Desarrollo de una técnica SLAM para ambientes dinámicos tridimensionales

Daniel Hernando Gómez Gómez

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica

Bogotá, Colombia

2015

Desarrollo de una técnica SLAM para ambientes dinámicos tridimensionales

Daniel Hernando Gómez Gómez

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Magister en Automatización Industrial

Director (a):

Ph.D Flavio Augusto Prieto Ortiz

Línea de Investigación:

Visión de Máquina

Grupo de Investigación:

GAUNAL

Universidad Nacional de Colombia

Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica

Bogotá, Colombia

2015

Dedicatoria:

A mi madre que me guía

A mi padre que me aconseja

A mis maestros quienes me ilustran

Y a la Universidad Nacional, mi segunda casa

Agradecimientos

Agradezco a la Universidad Nacional quien me ha formado como ingeniero y magister, de igual forma agradezco a la Vicedecanatura de Investigación y Extensión de la Facultad de Ingeniería quienes apoyaron económicamente esta propuesta, de igual forma un agradecimiento a la Technische Universität München por el apoyo técnico recibido. Quiero agradecer a mi familia y amigos por la paciencia, el apoyo incondicional y sus buenos deseos, por último quiero agradecer especialmente a mi director, profesor y mentor Flavio Prieto, por la paciencia, disposición y colaboración a lo largo de este proyecto.

Resumen

El problema SLAM nació a mediados de los años 80 como una extensión de los métodos de control que pretendían mejorar el control en el movimiento de los sistemas autónomos, rápidamente el área fue creció dadas las potenciales aplicaciones en el ámbito militar, recreativo, industrial, doméstico entre otras. El interés en el área surgió en la conferencia IEEE de robótica y automatización del 86, pero no es hasta que se introduce el algoritmo EKF que se genera un aumento considerable en las publicaciones. Sin embargo, las técnicas usadas requieren condiciones ajustadas en el ambiente, como la dimensión, la iluminación, la complejidad y el dinamismo, aun así el potencial de las aplicaciones que deben hacer uso de esta técnica demandan que se pueda reconstruir y localizar en el menor tiempo posible cualquier sistema autónomo bajo condiciones naturales. Dado el significativo aumento en la capacidad de computación de los últimos años y la introducción de sensores de rango tridimensionales de bajo costo, se pretende abordar la reconstrucción y localización con el uso mixto de sensores, características extendidas a la nueva información sensorial, clasificadores heurísticos bioinspirados y un sistema de almacenamiento jerárquico.

Palabras clave

SLAM, Odometría Visual, Navegación Autónoma, Mapeo, ROS, Procesamiento de Imágenes, OpenCV, PCL, SURF, KNN, Aprendizaje de Máquina.

Abstract

The SLAM problem was born in the mid-80s as an extension of the methods of control that sought to quickly improve the control on the movement of autonomous systems, the area was raised given the potential applications in the field of military, recreational, industrial, domestic, among others. Interest in the area arose in the IEEE Conference on Robotics and automation of 86, but it is not until EKF algorithm is created, that there is a considerable increase in publications. However, the used techniques required conditions set in the environment, dimension, lighting, complexity and dynamism, yet the potential of applications that should use this technique demand that it can be reconstructed and locate any autonomous system under natural conditions in the shortest possible time. Given the significant increase in the capacity of computing in recent years and the introduction of low-cost three-dimensional range sensors, is intended to reconstruct and locate with the mixed use of sensors, features extended to new sensory information, Bio-inspired heuristic classifiers and a hierarchical storage system.

Keywords

SLAM, Visual Odometry, Autonomous Navigation, Mapping, ROS, Image Process, OpenCV, PCL, SURF, KNN, Machine Learning.

Índice general

Índice de figuras

Pag.

Índice de cuadros

Pag.

Lista de Símbolos y abreviaturas

Símbolos con letras latinas

Símbolo	Termino	Unidad	Definición
C	Vector de puntos espaciales obtenidos por la medición de un sensor RGBD en el marco de referencia de la cámara.	1	$C = \left\{ \begin{array}{cccccc} K_0 & K_1 & K_2 & \cdots & K_n \end{array} \right\}$
c_x	Ubicación del centro de cámara sobre el eje X	px	
c_y	Ubicación del centro de cámara sobre el eje Y	px	
D_{xx}	Aproximación a la convolución L_{xx} con un kernel en vez de la función gaussiana.	px	
D_{xy}	Aproximación a la convolución L_{xy} con un kernel en vez de la función gaussiana.	px	
D_{yy}	Aproximación a la convolución L_{yy} con un kernel en vez de la función gaussiana.	px	
d	Distancia entre cada muestra μy su respectivo par ϕ	1	$d = \sum_{i=1}^{M} \lvert\mu - \phi\rvert$
E	Matriz esencial que define la relación entre coordenadas normalizadas.	1	$E = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix}$
E_i	Error relativo o absoluto de pose en un instante del tiempo	1	Ecuación 4.1Ecuación 4.3
F	Vector de M características numéricas para cada punto K en una medición C.	1	$F = \left\{ \begin{array}{cccc} F_1 & F_2 & \cdots & F_M \end{array} \right\}$
Fc	Matriz fundamental que define la relación entre coordenadas sobre el plano de la imagen.	1	$Fc = \begin{bmatrix} Fc_{11} & Fc_{12} & Fc_{13} \\ Fc_{21} & Fc_{22} & Fc_{23} \\ Fc_{31} & Fc_{32} & Fc_{33} \end{bmatrix}$

Símbolo	Termino	Unidad	Definición
f_x	Distancia focal sobre el eje X	mm	
f_y	Distancia focal sobre el eje Y	mm	
h_σ	Función gaussiana		$h_\sigma(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$
H	Matriz Hessiana		$\mathcal{H}(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix}$
I	Imagen como arreglo de pixeles en forma bidimensional	px	$I = \begin{Bmatrix} p_1 & p_2 & \cdots & p_{u*v} \end{Bmatrix}$
K	Punto específico en el espacio definido por una coordenada y un color en formato RGB.	m, rgb	$K = \begin{bmatrix} x & y & z & r & g & b \end{bmatrix}$
K_c	Matriz de calibración de la cámara RGBD	mm, px	$K_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$
L_{xx}	Convolución de la imagen con la segunda derivada de la función gaussiana en XX	px	$L_{xx}(m,\sigma) = I\,[m] * \frac{\partial^2 h_\sigma(x,y)}{\partial x^2}$
L_{xy}	Convolución de la imagen con la segunda derivada de la función gaussiana en XY	px	$L_{xy}(m,\sigma) = I\,[m] * \frac{\partial^2 h_\sigma(x,y)}{\partial x \partial y}$
L_{yy}	Convolución de la imagen con la segunda derivada de la función gaussiana en YY	px	$L_{yy}(m,\sigma) = I\,[m] * \frac{\partial^2 h_\sigma(x,y)}{\partial y^2}$
m_i	Vector de ubicación de la etiqueta i		
m	Ubicación de todas las etiquetas		$m = \{m_1, m_2, ..., m_k\}$
N	Número de puntos existentes en el mundo W.	1	
n	Número de puntos existentes en cada medición C.	1	
P	Coordenada normalizada que corresponde a un punto en una imagen traspasado al espacio euclidiano.	m	$P = \begin{bmatrix} x & y & z \end{bmatrix}$
p	Punto en una imagen	px	$p = \begin{bmatrix} u & v \end{bmatrix}$
Q	Trayectoria real en el espacio $SE(3)$	1	$Q = \begin{bmatrix} x & y & z & q_w & q_x & q_y & q_z \end{bmatrix}$
R	Matriz de rotación entre el marco de referencia de la cámara y el marco de referencia del mundo.	rad	$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$
S	Trayectoria estimada en el espacio $SE(3)$		$S = \begin{bmatrix} x & y & z & q_w & q_x & q_y & q_z \end{bmatrix}$
s	Usado como valor de escala		

Símbolo	Termino	Unidad	Definición
T	Número de iteraciones en el algoritmo de recocido simulado adaptativo.		
T	Matriz de transformación entre el marco de referencia de la cámara y el marco de referencia del mundo.	m, rad	$T = \begin{bmatrix} R & t \end{bmatrix}$
T_K	Actualización de temperatura en el algoritmo de recocido simulado adaptativo. .		$T_K = \frac{T}{k^{1/M}}$
t	Vector de traslación entre el marco de referencia de la cámara y el marco de referencia del mundo.	m	$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$
t	Hace referencia a la línea temporal de las mediciones C.		
$U_{0:k}$	Historial de las acciones de control.		$U_{0:k} = \{u_0, u_1, ..., u_k\}$
u_k	Vector de control aplicado en el instante k.		
v	Vector de características en cada sub-región de cada punto SURF		$v = \begin{Bmatrix} \sum dx & \sum \lvert dx \rvert & \sum dy & \sum \lvert dy \rvert \end{Bmatrix}$
W	Vector con todos los puntos existentes en el marco de referencia del mundo.	1	$W = \begin{Bmatrix} K_0 & K_1 & K_2 & \cdots & K_N \end{Bmatrix}$
$X_{0:k}$	Historial de las ubicaciones.		$X_{0:k} = \{x_0, x_1, ..., x_k\}$
x_k	Vector de estado que describe la ubicación en $SE(3)$ en el instante k.		
$Z_{0:k}$	Conjunto de todas las etiquetas observadas en el instante k.		$Z_{0:k} = \{z_1, z_2, ..., z_k\}$
z_k	Vector de ubicación de una etiqueta		

Símbolos con letras griegas

Símbolo	Termino	Unidad	Definición
Δ	Intervalo de tiempo	s	$\Delta = t_2 - t_1$
ε	Energía		
ϕ	Representa al material con N número de puntos		Ecuación 3.6
μ	Representa las muestras a clasificar		Ecuación 3.7
φ	Espacio de movimiento libre para cada partícula μ	1	$\varphi = \frac{\exp(0.693 * T) - 1}{2}$

Abreviaturas

Abreviatura	Término
ATE	Absolute trajectory error
CAD	Computer aid design
DVO	Dense visual odometry
FLANN	Fast library for aproximate nearest neighbors
GICP	Generalized iterative closest point
ICP	Iterative closest point
IMU	Inertial measurement unit
IR	Infrared
KNN	K-nearest neighbors
ORB	Oriented FAST and Rotated BRIEF
SANN	Simulated Annealing Nearest Neighbors
$SE3$	Hace referencia al espacio euclidiano tridimensional.
$SO3$	Hace referencia al espacio ortogonal tridimensional
SIFT	Scale-invariant feature transform
SURF	Speeded-Up Robust Features
RANSAC	Random sample consensus
RGBD	Hace referencia a una imagen en el espacio de color RGB(Red Green Blue) y una imagen de profundidad.
RMSE	Root mean square deviation
RPE	Relative pose error
UAV	Unmanned Aerial Vehicle
Voxel	Mínima unidad cubica que define un objeto tridimensional

Introducción

¿Dónde estoy? es una pregunta básica a la cual se ha intentado responder desde los inicios de la humanidad, principalmente con fines económicos, políticos, militares, filosóficos, de supervivencia y ahora robóticos. La necesidad de ubicar y conocer el entorno en el que opera una máquina determinada ha sido un tema activo de investigación desde hace 30 años, sin embargo aún quedan muchos temas que investigar antes de tener robots completamente autónomos y operando sin necesidad de una supervisión humana.

Resolver la pregunta inicial implica que la unidad robótica debe usar sensores para obtener información sobre el entorno, procesar esa información y reconocer su posición mientras actualiza el mapa del entorno, todo esto en tiempo real y de forma paralela al desarrollo de otras actividades. Es un proceso análogo a la forma en que los humanos usan sus sensores (de visión), para navegar reconociendo y/o aprendiendo del entorno.

Para resolver el problema se debe transformar la pregunta inicial, convirtiéndose en: *¿Cómo sabe una unidad robótica dónde se encuentra?*, la primera propuesta, común en la comunidad, es con el uso de un sensor GPS (global positioning system). Sin embargo, este no trabaja en áreas submarinas, debajo de tierra, dentro de edificios y aun no en marte, lo que deja un espacio de operabilidad correspondiente al espacio aéreo, a la superficie terrestre que esté dentro de construcciones, bosques o elementos que bloqueen la señal GPS, y a la superficie acuática, en estos espacios el departamento de defensa de los Estados Unidos de América garantiza una localización con un margen de error de 7.8 metros a un nivel de confianza del 95% [51], este error comprende efectos atmosféricos, obstrucción del espacio aéreo, calidad del receptor entre otros. Sin embargo, el error actual de los sensores GPS se estima actualmente en 3.286 metros horizontales y 6.301 metros verticales, en espacios abiertos, según el último reporte de la FAA [65], de cualquier forma estos valores de error son superiores a la dimensión de muchos objetos presentes en entornos humanos, lo que llevaría a que la unidad robótica, que use GPS como único sensor, corra el riesgo de colisionar con elementos cotidianos cuya dimensionalidad sea menor al margen de error. Por otro lado, el GPS no proporciona información acerca del entorno ni de los objetos que lo componen o que transitan a través de este, y depende fuertemente de un servicio de mapas. Una forma de afrontar este problema es usar un sensor mixto GPS + cámara, aunque este no es el enfoque de este documento se recomienda el trabajo de [62], en el cual se demuestra cómo reducir el margen de error a pocos centímetros navegando en un entorno

abierto cuyo mapa es previamente conocido.

Este documento planteará una estrategia para resolver la segunda pregunta formulada a través del uso de un sensor mixto Camara+IR (imagenes RGB-D), descriptores, base de datos y algunas técnicas de inteligencia artificial. Cada capítulo de este documento entrará en detalle del proceso, si bien es una propuesta que pretende resolver el problema de localización y mapeo simultáneo, no es la única. Se hace especial énfasis en propuestas cómo [36, 68, 34, 4], cuyos aportes al conocimiento han sido relevantes e importantes para la realización de este trabajo.

Motivación

Emprender la investigación y desarrollo para resolver la primera pregunta formulada, implica necesariamente responder *¿Para qué necesita una unidad robótica saber dónde se encuentra?,* considerando el impacto que ha tenido la introducción de la robótica en las labores humanas, se puede considerar como uno de los mayores logros de la ingeniería de finales del siglo XX e inicios del XXI, el desarrollo de las unidades robóticas les ha otorgado progresivamente habilidades de los seres vivos, movimiento, interacción, comunicación, percepción, procesamiento de datos entre otras habilidades, con el único fin de remover el elemento humano de las tareas repetitivas sin valor cognitivo agregado. El sentido en el cual se ha desarrollado el área del conocimiento sobre robótica y automatización indica que las unidades robóticas deben seguir adquiriendo habilidades con el objetivo de facilitar las tareas humanas, una de estas habilidades es la compleja navegación con la que cuentan un gran número de especies, producto de un arduo proceso de selección natural, esta habilidad permitiría automatizar el transporte urbano, el traslado de materiales y productos, la seguridad, algunas labores domésticas, entre otras tareas. Entonces, la necesidad de ubicación y conocimiento del entorno hace parte del desarrollo común que se le ha dado al campo de la robótica en los últimos años, otorgarle esa capacidad a las unidades robóticas conllevaría a aumentar su autonomía en pro de asignarles tareas que no precisen un mayor grado de abstracción en las cuales la vida humana pueda correr riesgo o en las que los seres humanos no pueden acceder por limitaciones físicas.

Aunque la investigación sobre *localización y mapeo simultaneo* en unidades robóticas se ha incrementado desde 1985, la principal motivación de este documento es afrontar el desarrollo de un sistema que pueda manejar entornos tridimensionales con características dinámicas, esta especificación nace de un análisis detallado del estado del arte que se extenderá en una próxima sección, independientemente de los resultados finales, también se pretende con el desarrollo de este documento motivar la investigación de esta área en el entorno local y nacional, a nivel académico e industrial.

Objetivos

El principal objetivo de este documento es **desarrollar una técnica de localización y mapeo simultaneo (SLAM) para ambientes dinámicos tridimensionales que use información sensórica 2D y 3D**, es importante señalar que no se pretende abarcar todos los problemas actuales en SLAM, puntualmente se trabajará en el problema de asociación de datos espaciales en el cual el desarrollo de SLAM ha tenido diversas dificultades, para lograr este objetivo principal se trabajarán los siguientes objetivos específicos:

1. La primera etapa sobre adquisición y tratamiento de señales contará con el uso de un sensor mixto Cámara+IR, con la información mixta de señales se pretende **extender el algoritmo de detección de características SURF a un espacio tridimensional y evaluar su desempeño en el problema SLAM**.

2. La segunda etapa compara la información adquirida con la que se encuentra en el entorno reconocido, esta comparación se considera actualmente un problema debido al tiempo usado y la precisión obtenida, con la información adicional del sensor mixto se pretende **adaptar la etapa de reconocimiento en SLAM a un problema de optimización bioinspirado y evaluar su desempeño contra otras técnicas de clasificación**. Considerando que las técnicas de optimización han demostrado en la comunidad académica ser eficientes con menor cantidad de recursos.

3. La organización de la información obtenida es otro factor cuyo impacto se analizará en este documento, cambiar el estilo de mapa a un sistema de información geográfica con relaciones, puede mejorar considerablemente las peticiones e inserciones al mapa, en este sentido se propone **integrar al algoritmo SLAM una descripción del entorno como un sistema jerárquico probabilístico que indique la posible existencia de ciertos objetos en el espacio de acuerdo a su dimensionalidad**.

4. Finalmente, el documento realizara un análisis de la efectividad del sistema SLAM con las modificaciones propuestas, puntualmente se **evaluara el desempeño del algoritmo SLAM-bioinspirado-jerárquico propuesto**.

Organización de la tesis

Este documento de tesis está organizado en 6 capítulos numerados, cada capítulo abarca un proceso específico de la metodología de trabajo empleada:

1. Introducción: En este capítulo se realiza un análisis inicial de la necesidad de investigar acerca de las técnicas SLAM, se plantean las motivaciones de la investigación y luego se presentan los objetivos que se desean lograr para

2. El primer capítulo inicia con una revisión histórica de los antecedentes sobre los temas que trabaja este documento, seguido a esto se presenta un análisis detallado del estado actual del arte en el cual se hará un análisis semántico sobre diversas bases de datos acerca de SLAM, este estado del arte estará sustentado en un análisis profundo sobre cada agrupación o área del conocimiento encontrada, señalando los logros alcanzados y retos propuestos. Finalmente este capítulo cierra con una descripción matemática de SLAM y de las propuestas más reconocidas en el mundo académico

3. El segundo capítulo inicia con una revisión histórica del desarrollo de las cámaras CMOS y de los sensores infrarrojos, luego presenta el sensor Kinect como un dispositivo integrado por ambas cámaras para obtener nubes de puntos con información de color. El capítulo presentará el método de calibración usado y los resultados obtenidos.

4. El tercer capítulo inicia con la revisión del algoritmo SLAM con el cual se inició el desarrollo de esta propuesta, luego en cada sub-capítulo se presenta el desarrollo de las modificaciones propuestas, su definición e implementación. Cada sub-capítulo tiene un desarrollo independiente dada la diferente naturaleza de las propuestas realizadas, por tal motivo no hay una fuerte conexión entre ellos, sin embargo cada sub-capitulo está relacionado directamente con un sub-capítulo del capítulo de evaluación, y con la evaluación del método general.

5. El cuarto capítulo realiza una evaluación de cada una de las propuestas presentadas en el capítulo 3, indicando los métodos usuales de comparación de resultados y datos comparativos para la mayoría de pruebas; cada evaluación esta seguida de un análisis de los resultados obtenidos, indicando las ventajas y desventajas, así como la viabilidad de integrar cada modificación a la técnica SLAM propuesta. El capítulo finaliza con una evaluación completa del algoritmo propuesto contra el algoritmo actual, los métodos se probarán con una base de datos estándar de escenas con diferentes composiciones.

6. El último capítulo concluye los resultados obtenidos en cada uno de los desarrollos realizados a partir de los resultados obtenidos, adicionalmente realiza un análisis global del método de trabajo y de posibles investigaciones a futuro.

Capítulo 1

Estado del Arte

Este capítulo inicia con una revisión en la tendencia investigativa que el área de localización y mapeo simultaneo (SLAM) ha tenido en la comunidad investigativa a través de las últimas 3 décadas, esta revisión incluye un análisis semántico de las investigaciones relacionadas al tema SLAM encontradas en diferentes bases de datos, la búsqueda se realizó a través del meta-buscador Scopus con palabras clave SLAM o Simultaneous Localization and Mapping, filtrando por temas de Engineering y Computer Science, esta configuración se muestra en la Figura 1.1.

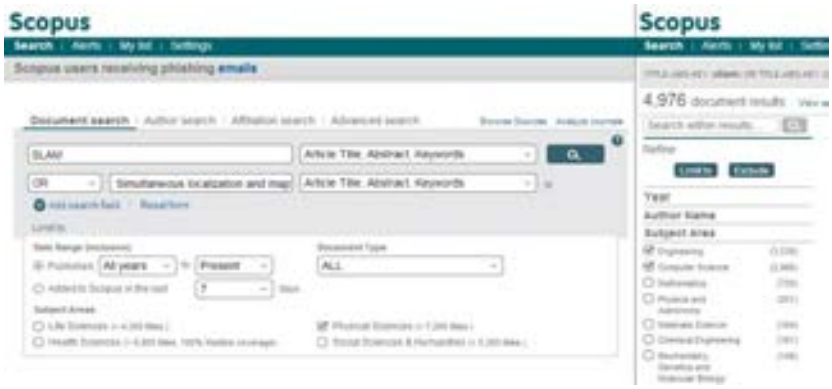

Figura 1.1: Parámetros de búsqueda en Scopus sobre SLAM. .

Con los resultados obtenidos se ofrecerá un análisis del estado del arte desde una perspectiva global, mencionando los artículos más relevantes en cada sub-área del desarrollo. La segunda sección de este capítulo realiza un estado del arte de las características junto con los métodos de extracción comúnmente usados, de los clasificadores comunes y de los sistemas de información geográfica estándar como una forma de introducir los temas que se tratarán en la propuesta de tesis, la tercera sección realiza una descripción de las técnicas SLAM usadas actualmente, su descripción matemática y resultados obtenidos por diversos autores. Finalmente, el capítulo termina con una definición precisa del problema SLAM, los resultados de algunas investigaciones populares y los problemas actualmente presentes.

1.1 Estado del arte

La investigación acerca de SLAM ha sido un tema de investigación desde 1975 y un tema activo desde 1985. Sin embargo, el 83.9% de las investigaciones sobre el tema se encuentran en el periodo 2000-2014, en la Figura 1.2 se puede observar cómo evoluciona el número de investigaciones activas a través de los años, también se resaltan 5 periodos investigativos:

- Periodo 1 (1975 - 1984): En la década previa a la formulación del problema SLAM las investigaciones, planeamiento de ruta [55] y planteamiento de conceptos de reconstrucción [44], las soluciones planteadas son aplicaciones euclidianas y análisis de la geometría de los cuerpos presentes en el entorno, de esta forma [44] plantea el entorno y ruta como una serie de nodos interconectados espacialmente y [55] usa los ángulos de los objetos en el mapa para hallar la dirección del movimiento, existen otras aproximaciones a los movimientos con incertidumbre como [56], en conjunto estas investigaciones permitieron el desarrollo del área SLAM en términos de movimientos con incertidumbre, reconstrucción y movimiento.

- Periodo 2 (1985 - 1994): El génesis del problema SLAM surgió en 1986 en la conferencia “IEEE Robotics and Automation - San Francisco” en la que se discutió algunos trabajos previos sobre localización y reconstrucción realizados por Peter Cheeseman, Jim Crowley, and Hugh Durrant-Whyte [35]; el resultado de esta discusión fue el reconocimiento del SLAM como uno de los problemas con mayores problemas conceptuales y computacionales en el área de la robótica [35]. En esta época el campo del mapeo estuvo fuertemente dividido entre aproximaciones métricas y topológicas. Dos de las representaciones métricas de esta época fueron realizadas por Alberto Elfes [22, 21], en las cuales se hace referencia a algoritmos probabilísticos, estos algoritmos basan el problema de mapeo en datos con ruido que proveen sensores, los cuales están en un robot con posición inicial conocida. Los algoritmos métricos suelen hacer uso de grillas de ocupación, como se describe en [22], los cuales son representaciones detalladas del ambiente donde se indica la presencia de obstáculos. Una representación topológica de esta época fue realizada por Benjamin Kuipiers [42], en la cual se hace referencia a una lista de sitios conectados por arcos, estos arcos contienen la información necesaria para navegar entre los nodos. Sin embargo, los métodos probabilísticos dominaron debido a la introducción de marcos probabilísticos robustos para resolver simultáneamente el mapeo y la localización [64, 57].

- Periodo 3 (1995 - 1999): Se introduce el concepto de navegación autónoma y se define completamente el problema SLAM en el simposio internacional ’Robotic research’ de 1995 [35], desde este año el número de publicaciones se mantiene constante hasta 1997, cuando se resuelve el problema de forma teórica desde un punto de vista pro-

babilístico por Michael Csorba [15]. En el simposio internacional 'Robotic research' de 1999 se sostuvo la primera sesión acerca de SLAM [35], en la cual se presentaron resultados de convergencia en las técnicas SLAM basadas en el filtro de Kalman [66], la convergencia del mapa a partir de sensores ultrasónicos con un número considerable de iteraciones.

- Periodo 4 (2000 - 2014): Con la introducción del filtro de Kalman extendido en 2000 se logra resolver el problema SLAM en 2001 [53] para ambientes estáticos, cerrados y con el uso de varias cámaras. Desde 2001 hasta 2010 el número de investigaciones tuvo una tendencia creciente donde se destacan múltiples intentos para mejorar los algoritmos, filtro y uso de diversos sensores, entre los que se destaca [20, 16]. Las soluciones monoculares como [17, 20] utilizan las características visuales con sus descriptores, para estimar la posición real de la cámara, el movimiento es modelado por el filtro de Kalman. SLAM se considera un problema teóricamente resuelto [18, 35], sin embargo a final de este periodo de tiempo las investigaciones se han enfocado en resolver diversos problemas prácticos de las técnicas SLAM (Observabilidad, convergencia, modelado de sensores, consistencia, eficiencia computacional) [18]. Cuando el entorno está cambiando en el tiempo, induce errores en la localización debido a que el mapa obtenido en un tiempo anterior no se puede emparejar con la información nueva, lo que conlleva a errores de convergencia o de consistencia. En [47] se usa la distribución de probabilidad Dirichlet processes para estimar la presencia o desaparición de cluster de características, posteriormente [26] usa la diferencia entre features para encontrar los objetos de un mapa que ya no están presentes y así hacer una corrección al mapa.

Figura 1.2: Número de investigaciones en SLAM por año.

Un análisis semántico realizado al periodo 4 (2000 - 2014) se muestra en la Figura 1.3, el análisis se realiza como método para agrupar las investigaciones cuyo contenido es textualmente similar, del análisis podemos concluir que existen 4 técnicas principales de SLAM (FastSLAM, Unscented Kalman filter SLAM, Structure from motion y Visual Odo-

metry) [18, 35, 48, 53, 10], las cuales usan un conjunto diverso de sensores (Monocular Camera, Range Sensor, RGB-D, IMU) y se en enfoca en 3 tipos de aplicaciones (Vehicle, Cooperative robots, Aerial-UAV). También podemos destacar de la Figura 1.3 la cercanía entre investigaciones, lo cual indica similitud entre los temas investigados, de esta forma el grupo de investigaciones con el uso de sensores RGB-D e IMU es el menos disperso, esto se explica debido a que los sensores IMU han sido ampliamente usados en robótica, con algoritmos estocásticos como el filtro de partículas o el filtro de Kalman. Además la aparición de las cámaras RGB-D de bajo costo permitieron realizar mayor número de investigaciones, extendiendo los resultados de los algoritmos monoculares. Se puede determinar que el campo de aplicación Arial-UAV es el menos denso e investigado, esto se debe a que la presencia de los UAV está aún en auge, esta área se caracteriza por usar en su totalidad cámaras monoculares y algoritmos de visual odometry.

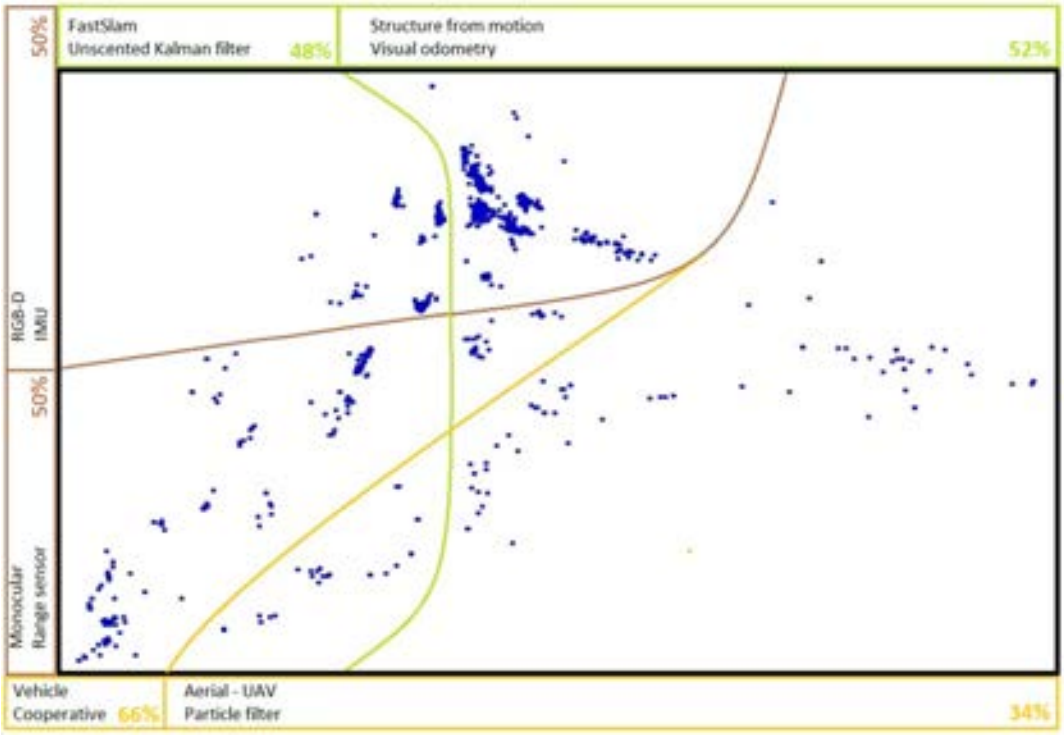

Figura 1.3: Análisis semántico del periodo 4.

1.2 Problema SLAM

1.2.1 Descripción del problema SLAM

El problema de localización y mapeo simultaneo (SLAM) consiste en darle la habilidad a la unidad robótica, ubicada inicialmente en una posición desconocida y en un entorno desconocido, de construir un mapa de su entorno incrementalmente y al mismo tiempo determinar su posición respecto a este mapa.

En orden de reconstruir un mapa, la unidad robótica debe interpretar la información de los sensores, estos sensores pueden ser cámaras, sensores de rango, sonares, infrarrojos etc. Sin embargo, se señala que todos los sensores poseen un margen de error debido a limitaciones físicas, ruido del ambiente o desgaste operativo. Las acciones de control en el desplazamiento no son una fuente confiable de datos para estimar la posición ni la orientación de la unidad robótica, debido a que la acción de control puede no ser ejecutada en su totalidad en el ambiente, dadas las condiciones físicas del entorno y de la unidad robótica.

1.2.2 Formulación probabilística

El error inherente en los sensores de las unidades robóticas, y el error inherente a las acciones de control, conllevan a formular el problema SLAM en un marco estadístico. Podemos definir:

- x_k: Vector de estado que describe la ubicación y orientación del vehículo en SE(2) o SE(3),
- $X_{0:k} = \{x_0, x_1, ..., x_k\}$: Historial de las ubicaciones y orientaciones del vehículo,
- u_k: Vector de control aplicado en el tiempo *k-1* al sistema de navegación del vehículo en SE(2) o SE(3).
- $U_{0:k} = \{u_0, u_1, ..., u_k\}$: Historial de las acciones de control,
- m_i: Vector de ubicación espacial de la etiqueta *i*,
- $m = \{m_1, m_2, ..., m_k\}$: Ubicación de todas las etiquetas.
- z_{ik}: Vector de ubicación de la *i* etiqueta encontrada en el instante *k,*
- $Z_{0:k} = \{z_1, z_2, ..., z_k\}$: Conjunto de todas las etiquetas observadas en el instante *k*.

De esta forma, SLAM es la probabilidad de obtener una nueva posición x_{t+1} y la ubicación actualizada de todas las marcas M, dado los vectores históricos $X_{1:t}$, $U_{1:t}$ y $Z_{1:t}$, como se muestra en la Figura 1.4. Es entonces que la distribución se define como:

$$P(x_{t+1}, M|Z_{0:k}, U_{0:k}, X_{1:t}). \tag{1.1}$$

La Ecuación 1.1 es calculada para cada tiempo t, sin embargo la operación de la probabilidad implica realizar operaciones con espacios de estado grandes, para esto la solución probabilística de SLAM asume que el mundo es estático y que los vectores se manejan como una cadena de Markov. Debido a las limitaciones físicas de los sensores, no se pueden

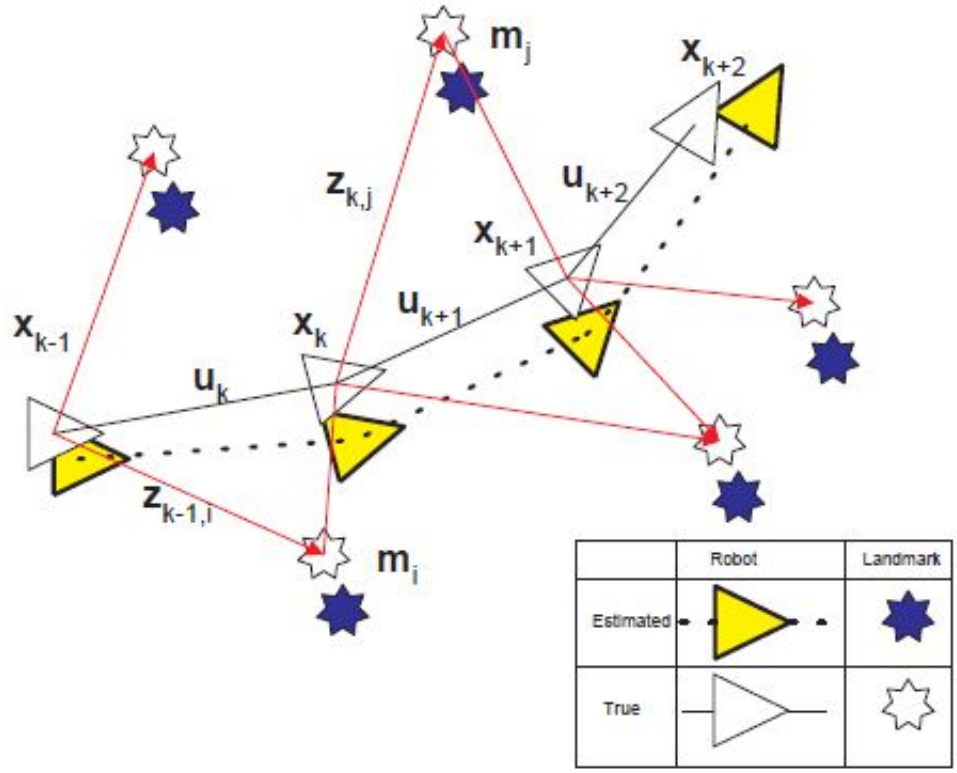

Figura 1.4: Problema SLAM [35].

obtener mediciones constantes, por lo que la formulación probabilística debe operar en 2 ciclos paralelos:

- Actualización temporal: Esta actualización constante en el tiempo, es la multiplicación de la distribución de la posición estimada con la distribución de última posición conocida, integrada por el movimiento realizado entre mediciones.

$$P(x_{t+1}, M|Z_{0:k-1}, U_{0:k}, X_{1:t}) = \int P(x_{t+1}|x_t, u_k) * P(x_t, M|Z_{0:k-1}, U_{0:k-1}, X_{1:t-1})dx_t. \quad (1.2)$$

- Actualización por medición: Cuando una medición es realizada, la actualización consiste en multiplicar la distribución de la medición según la posición con la distribución de la última actualización temporal y dividirlo por la distribución de la medición, según las acciones de control.

$$P(x_{t+1}, M|Z_{0:k}, U_{0:k}, X_{1:t}) = \frac{P(Z_{t+1}|X_{t+1}, M) * P(x_{t+1}, M|Z_{0:k-1}, U_{0:k}, X_{1:t})}{P(Z_{t+1}|Z_{0:k-1}, U_{0:k})} \quad (1.3)$$

La localización está entonces relacionada con la distribución $P(x_{t+1}|Z_{0:k}, U_{0:k}, M)$, asumiendo que el mapa es el conjunto de marcas M que son conocidas con alto grado de precisión.

1.3 Soluciones actuales

La solución común a las distribuciones de probabilidad mencionadas en las ecuaciones 1.1, 1.2 y 1.3, hacen uso de distribuciones gaussianas para el ruido. El filtro de Kalman extendido fue introducido para solucionar el problema SLAM [53], desde entonces EKF-SLAM se ha convertido en una solución diversificada en algoritmos extendidos como UKF. Otras distribuciones no gaussianas como el filtro de partículas Rao-Blackwellized definido en [48] se ha implementado, generando el algoritmo FastSLAM. Otras aproximaciones recientes al problema SLAM implementan imágenes RGB+D, consecutivas como lo plantea [10] en odometría visual.

1.3.1 EKF-SLAM

El filtro de Kalman se puede usar con sistemas lineales, sin embargo para sistemas no lineales se debe aplicar una expansión de taylor al modelo alrededor del último vector de estado. Considerando un sistema no lineal:

$$X_{t+1} = f(X_t, U_t) + \varepsilon_x. \tag{1.4}$$

$$Z_{t+1} = h(X_t) + \varepsilon_z. \tag{1.5}$$

Aplicando la expansión de taylor se obtiene:

$$X_{t+1} = f(X_0, U_t) + f'(X_0, U_t)\Delta X + \varepsilon_x. \tag{1.6}$$

$$Z_{t+1} = h(X_0) + h'(X_0)\Delta X + \varepsilon_z. \tag{1.7}$$

Considerando que $f(X_0, U_t)$ y $h(X_0)$ son constantes, entonces se definen la las constantes A, C del filtro de Kalman:

$$A = f'(X_0, U_t),$$

$$C = h'(X_0).$$

1.3.2 Keypoint-Alignement

Keypoint-Alignement es una técnica para encontrar la transformación $T = \begin{bmatrix} R & t \end{bmatrix} \in SE3$ necesaria para transformar una nube de puntos $C = \begin{Bmatrix} K_0 & K_1 & K_2 & \cdots & K_n \end{Bmatrix}$ desde el marco de referencia de la cámara al marco de referencia del mundo W, donde R es la rotación y t es la traslación. Esta transformación se halla a partir de puntos K comunes en

ambas nubes de puntos, la similitud se da por sus características $F = \left\{ \begin{array}{cccc} f_1 & f_2 & \cdots & f_M \end{array} \right\}$ obtenidas en el pre-procesamiento de cada nube de puntos C. Se puede asumir que el mundo W es un conjunto espacial discreto de N puntos K, cada uno con la posibilidad de tener características F como se muestra en la Ecuación 1.8. Cada nueva nube C podría encontrar una transformación del marco de referencia de la cámara hacia el marco de referencia del mundo, cuando comparta suficientes puntos con el mundo, en caso contrario es un nuevo espacio no referenciado en el mundo.

$$W = \begin{cases} K_0 \left(\begin{array}{cc} x_0 & y_0 \end{array} \right) = & \left\{ \begin{array}{cccc} f_1 & f_2 & \cdots & f_M \end{array} \right\} \\ K_1 \left(\begin{array}{cc} x_1 & y_1 \end{array} \right) = & \left\{ \begin{array}{c} - \end{array} \right\} \\ K_2 \left(\begin{array}{cc} x_2 & y_2 \end{array} \right) = & \left\{ \begin{array}{cccc} f_1 & f_2 & \cdots & f_M \end{array} \right\} \\ \vdots & \vdots \\ K_{N-20} \left(\begin{array}{cc} x_{N-20} & y_{N-20} \end{array} \right) = & \left\{ \begin{array}{c} - \end{array} \right\} \\ \vdots & \vdots \\ K_{N-1} \left(\begin{array}{cc} x_{N-1} & y_{N-1} \end{array} \right) = & \left\{ \begin{array}{c} - \end{array} \right\} \\ K_N \left(\begin{array}{cc} x_N & y_N \end{array} \right) = & \left\{ \begin{array}{cccc} f_1 & f_2 & \cdots & f_M \end{array} \right\} \end{cases} . \tag{1.8}$$

Cada uno de los puntos K existen debido a que han sido extraídos a través del procesamiento de la información sensorial de cada medición C del entorno, múltiples extractores han sido propuestos tales como SIFT, SURF, ORB, entre otros. Para cada C existe un número variable de puntos K como se muestra en la Ecuación 1.9, estos puntos tienen propiedades tridimensionales y de color cuando son adquiridos a través de una cámara RGBD. El valor de N siempre va a ser mayor o igual a n, considerando que los puntos se agregan consecutivamente evitando duplicados.

$$\begin{array}{rl} C_0 = & \left\{ \begin{array}{ccccc} K_0 & K_1 & K_2 & \cdots & K_{n+1} \end{array} \right\} \\ C_1 = & \left\{ \begin{array}{ccccc} K_0 & K_1 & K_2 & \cdots & K_{n-1} \end{array} \right\} \\ C_2 = & \left\{ \begin{array}{ccccc} K_0 & K_1 & K_2 & \cdots & K_{n+3} \end{array} \right\} \\ \vdots & \vdots \\ C_\infty = & \left\{ \begin{array}{ccccc} K_0 & K_1 & K_2 & \cdots & K_{n+9} \end{array} \right\} \end{array} . \tag{1.9}$$

Considerando que las mediciones C son consecutivas, existe la probabilidad de que varias mediciones C compartan puntos K, entonces en un par de mediciones C_1, C_2 se puede hallar un conjunto de puntos comunes a través de un proceso de comparación, el algoritmo comúnmente usado es vecinos más cercanos (KNN) a través de su implementación FLANN, en este algoritmo los datos de entrenamiento crean una estructura tipo árbol, y cada clasificación está dada por una serie de comparaciones entre los nodos del árbol. Las correspondencias se usan para encontrar la transformación necesaria para añadir C_2 al

mapa espacial W. Para cada correspondencia hallada entre 2 puntos p_1, p_2 se define desde la geometría epipolar la relación de la Ecuación 1.10, donde E es la matriz esencial y P_1, P_2 son las coordenadas normalizadas en cada imagen.

$$P_1^T E P_2 = 0. \tag{1.10}$$

Las coordenadas normalizadas se relacionan con las coordenadas de la imagen según la Ecuación 1.11, en esta ecuación el parámetro K_c es la matriz de calibración hallada en el Capítulo 2 .

$$P = K_c^{-1} p. \tag{1.11}$$

Usando el movimiento espacial entre los puntos normalizados (Ecuación 1.10) y la relación entre puntos normalizados y píxeles (Ecuación 1.11), se puede desarrollar una ecuación de movimiento entre conjuntos de píxeles (Ecuación 1.12), esta Ecuación define Fc, como la matriz fundamental y está relacionada con la matriz esencial como lo muestra la Ecuación 1.13.

$$p_1^T K_c^{-T} E K_c^{-1} p_2 = 0,$$

$$p_1^T \, Fc \, p_2 = 0. \tag{1.12}$$

$$Fc = K_c^{-T} E K_c^{-1}. \tag{1.13}$$

La Ecuación 1.12 puede escribirse en detalle considerando la matriz fundamental Fc como una matriz 3×3, y los puntos p_1, p_2 como vectores tridimensionales $\begin{bmatrix} u & v & 1 \end{bmatrix}$, de esta forma se obtiene la Ecuación 1.14 que corresponde a la relación entre píxeles a través de la matriz fundamental en un formato extendido.

$$\begin{bmatrix} u_0 & v_0 & 1 \end{bmatrix} \begin{bmatrix} Fc_{11} & Fc_{12} & Fc_{13} \\ Fc_{21} & Fc_{22} & Fc_{23} \\ Fc_{31} & Fc_{32} & Fc_{33} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0. \tag{1.14}$$

Resolviendo y reagrupando los términos de la Ecuación 1.14 se obtiene la Ecuación 1.15, este sistema se puede resolver si se tienen al menos 9 pares de correspondencias, la solución a esta ecuación consiste en hallar cada uno de los términos f que componen las matriz fundamental Fc. Conociendo Fc y con la calibración de cámara K se puede calcular la matriz esencial E usando la Ecuación 1.13.

$$\begin{bmatrix} u_0u_1 & u_0v_1 & u_0 & u_1v_0 & v_0v_1 & v_0 & u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0. \tag{1.15}$$

Para obtener la matriz de transformación T a partir de E se usa la factorización por SVD (Singular value descomposition), esta factorización transforma una matriz compleja en un sistema de 3 matrices U, L, V^T. Aplicando la factorización a la matriz esencial E se tiene 4 posibilidades de obtener las matrices de rotación y traslación:

1. $T_1 = \begin{bmatrix} R_1 & t \end{bmatrix} = \begin{bmatrix} UWV^T & U_z \end{bmatrix}$.
2. $T_2 = \begin{bmatrix} R_2 & -t \end{bmatrix} = \begin{bmatrix} UWV^T & -U_z \end{bmatrix}$.
3. $T_3 = \begin{bmatrix} R_3 & t \end{bmatrix} = \begin{bmatrix} UW^TV^T & U_z \end{bmatrix}$.
4. $T_4 = \begin{bmatrix} R_4 & -t \end{bmatrix} = \begin{bmatrix} UW^TV^T & -U_z \end{bmatrix}$.

Donde:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

Y

$$U_z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}.$$

Para decidir cuál valor de T es el correcto, se debe probar con un punto aleatorio P que sea visible en ambas imágenes, con coordenadas normalizadas $P = \begin{bmatrix} x & y & 1 \end{bmatrix}$ en cada imagen, para cada una de estas imágenes el cálculo de la coordenada Z debe tener un valor positivo, este valor se calcula de acuerdo con la Ecuación 1.16.

$$Z = \frac{(r_1 - x * r_3) * t}{(r_1 - x * r_3) * P^T}. \tag{1.16}$$

Debido a errores de calibración, ruido, o desempeño, la transformación calculada puede no ser precisa, para esto se hace una pequeña transformación considerando la disminución de la distancia entre puntos respecto C_2 y W, esta transformación considera únicamente

traslación y es en cada eje el valor promedio de la distancia de los puntos más cercanos en ambas nubes.

En resumen, el procedimiento de la técnica *keypoint alignment* se muestra en la Figura 1.5, desde la etapa de pre-procesado hasta la reconstrucción de la escena, existen consideraciones posteriores como *loop closure* que considera el análisis de todas las transformaciones previas para mejorar la consistencia del mapa.

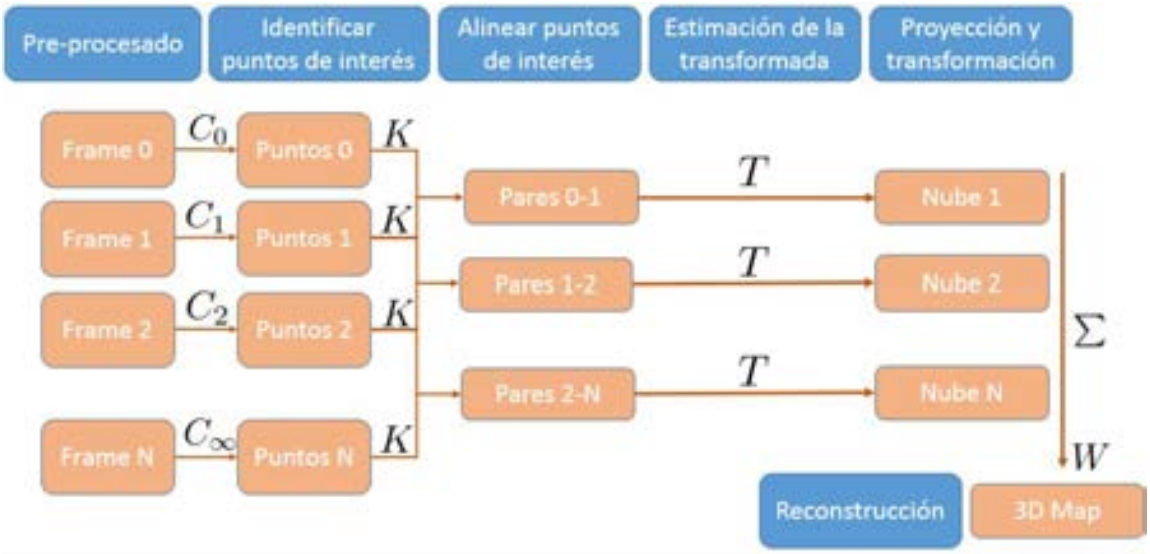

Figura 1.5: Proceso de alineamiento de keypoints

Este enfoque de alineamientos consecutivos en SLAM ha sido usado en [34, 36, 25, 33], en estas implementaciones las características son emparejadas con algoritmos métricos como *nearest neighbors*, y luego se refina su alineamiento a través de algoritmos iterativos como RANSAC o ICP. En [23] se presenta un desarrollo completo de esta técnica, a partir de las imágenes RGB se extraen características SURF, SIFT y ORB, las cuales se comparan con sus respectivos *frames* anteriores, al encontrar las coincidencias estas características se convierten en puntos tridimensionales. Con las características como un conjunto de puntos tridimensionales se halla la alineación en SE(3), la cual es refinada a través de RANSAC o GICP, en [23] se presenta también una optimización a través del framework g^2O [43]. Luego de que se ha calculado la matriz de transformación entre 2 *frames* consecutivos, la nube de datos es ingresada a un sistema de mapas, en [23] se hace uso de un mapa voxelizado [12], el esquema de la técnica SLAM usada por [23] se observa en la Figura 1.6.

Evaluaciones y comparaciones de las diversas implementaciones de esta técnica SLAM muestran que la técnica se desempeña de forma aceptable, en [23] se muestra que las características SIFTGPU y SURF son más precisas que las características ORB, con valores

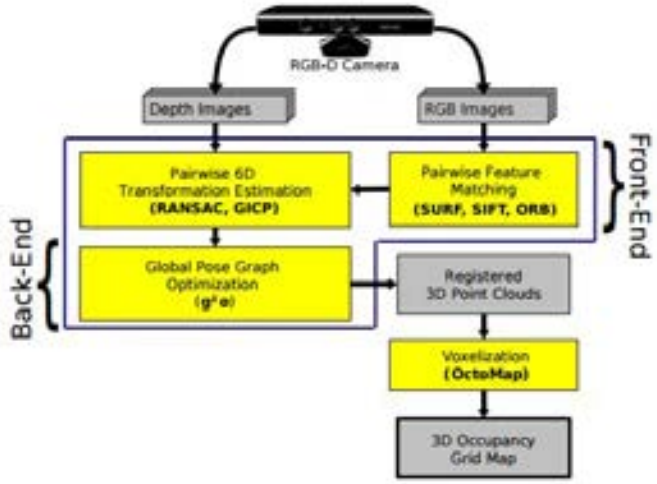

Figura 1.6: Técnica SLAM keypoint alignment realizado en [23].

de 2.1 cm y 4.1 cm en el RMSE de la trayectoria respectivamente, sin embargo la extracción de características toma el doble de tiempo para SURF/SIFTGPU que para ORB. Esta técnica ha sido usado por [32, 23] en escenas interiores, los resultados se pueden observar en la Figura 1.7.

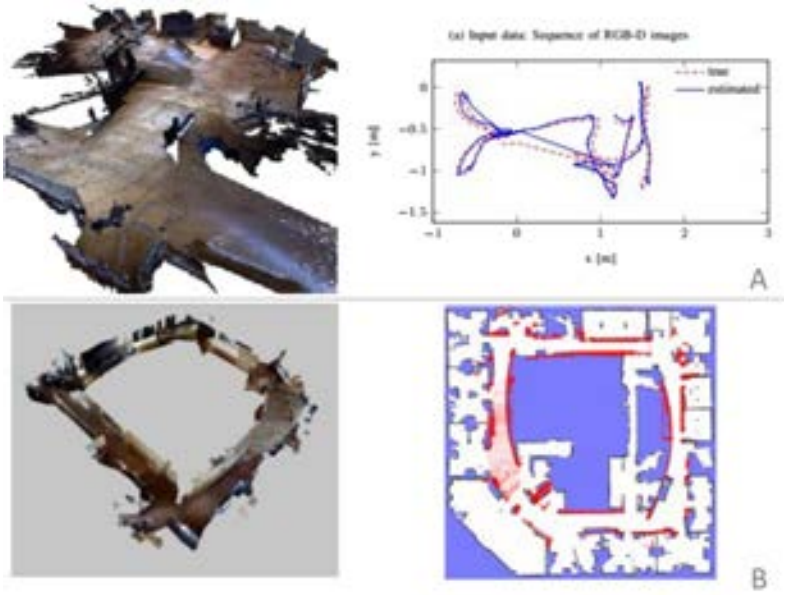

Figura 1.7: Mapas 3D reconstruidos con la tecnica *SLAM keypoint alignment* para ambientes. Imagen A: Reconstrucción de un laboratorio de la base de datos Freiburg [23], Imagen B: Reconstrucción del Laboratorio Intel. [32]

1.3.3 Odometría Visual

La odometría visual es una técnica que determina la transformación incremental $T = \begin{bmatrix} R & t \end{bmatrix} \in SE3$ en una serie de imágenes RGBD consecutivas. Esta alineación de image-

nes se conoce como *pairwise aligment,* la cual se basa en disminuir el error entre 2 imágenes cuando se aplica una transformación en el espacio, finalmente se optimiza la transformación a través de algoritmos iterativos tales como ICP o RANSAC.

Una transformación rígida T entre 2 frames consecutivos $I(t_0); I(t_1)$, induciría una transformación en $I(t_1)$, en el espacio euclidiano, T pertenece al espacio euclidiano y se define como una matriz de $4x4$ como se muestra en la Ecuación 1.17.

$$T = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}. \tag{1.17}$$

Dónde R y t son las matrices de rotación y traslación respectivamente, con estas matrices se define una función G que transforma un punto P en el espacio:

$$G(T, P) = RP + t. \tag{1.18}$$

Y una función π que transforma el punto P en el espacio al plano, esta función usa los parámetros de la matriz de calibración K_c:

$$\pi(G, K_c) = \begin{bmatrix} \frac{G[0]*f_x}{G[3]} - c_x & \frac{G[1]*f_y}{G[3]} - c_y \end{bmatrix}. \tag{1.19}$$

En el álgebra de lie SO(3), se tienen diferentes elementos que se definen:

$$S = \begin{pmatrix} w_x & w_y & w_z & v_x & v_y & v_z \end{pmatrix}^T. \tag{1.20}$$

$$\hat{S} = \begin{pmatrix} 0 & -w_z & w_y & v_x \\ w_z & 0 & -w_x & v_y \\ -w_y & w_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{1.21}$$

Donde $\hat{S}$ es un *twist*, la relación entre SO(3) y SE(3) está establecida por los mapas exponenciales y logarítmicos, de esta forma:

$$T = exp(\hat{S}). \tag{1.22}$$

$$\hat{S} = log(T). \tag{1.23}$$

Haciendo uso de las relaciones se puede definir una función de error como:

$$I(t_1) = \pi(G(T, P), K).$$

$$I(t_1) = \pi(G(exp(log(T) * S_t), P), K).$$

$$E(S_1) = I(t_0) - I(t_1). \tag{1.24}$$

La minimización de la Ecuación 1.24 produce la transformación entre ambos pares de imágenes, al realizarlo consecutivamente se obtiene una serie de transformaciones de cámara y el mapa del entorno. Esta técnica ha sido usado por [13] en escenas interiores, los resultados se pueden observar en la Figura 1.8.

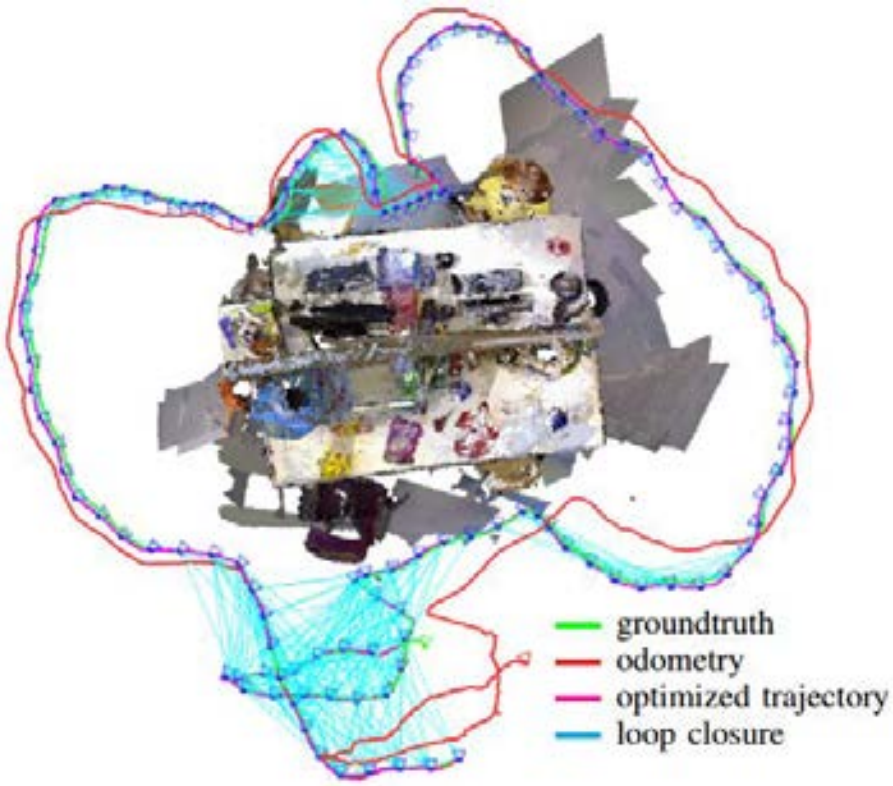

Figura 1.8: Mapa 3D reconstruido con la técnica *dense visual SLAM* para ambientes.[13]

Capítulo 2

Calibración sensorial

Las técnicas SLAM están correlacionadas primordialmente con el tipo de sensores que están disponibles, dada la naturaleza del sensor se puede obtener información de rango, de color, de calor u otros. Sin embargo, los parámetros de cada sensor varían de acuerdo al proceso de fabricación, por esto existe una diferencia importante entre los valores teóricos y los valores reales, y por tanto el modelo matemático de cada sensor se afecta al momento de usarlo con los modelos matemáticos de reconstrucción o localización. En este capítulo se iniciará con una revisión de los diferentes tipos de sensores usados en SLAM, se presenta a su vez el modelo matemático del Kinect y finaliza con un proceso de calibración de un Kinect real, estimando los parámetros para la cámara de color y la cámara infrarroja.

2.1 Sensores

Actualmente existe una amplia gama de oferta de sensores en el mercado, muchos de ellos han sido objeto de investigación desde la mitad del siglo XX. Con el fin de adquirir información útil del entorno se han usado comúnmente en SLAM sensores ultrasónicos, sensores de rango láser, sensores de rango IR, cámaras CMOS, sensores inerciales, GPS, odómetros, y giroscopios. Un breve recorrido por algunas soluciones SLAM muestran que Navlab11 [71] implementó entre 2002 y 2004 un sensor de rango láser y odómetros, Pioneer 2 [31] desarrolló en 2003 un sensor de rango láser 2D al igual que RW1 B21 Rhino, Robotic wheelchair [49] implementó en 2005 un sensor de rango láser 2D, el robot de [52] desarrolló en 2007 2 cámaras CMOS, el Mercedes Benz E class [69] en 2009 implementó un sensor de rango láser 2D, odómetros y 2 sensores de rango IR, diamler [70] en 2010 implementó 2 sensores de rango láser 2D, odómetros, 2 sensores de rango IR y una cámara CMOS. En el Capítulo 2 se evidencia que la tendencia de usar sensores de rango láser debido a la alta precisión y alcance que tienen, sin embargo los sensores de rango láser son costosos y de alto consumo energético. Las cámaras CMOS han tenido gran acogida por su bajo costo y facilidad de operación, desde 2010 se integró el sensor de rango IR con las cámaras

CMOS con el fin de obtener un sensor para detección de personas en espacios cerrados, a continuación se describirán brevemente el origen de ambos sensores concluyendo en el sensor Kinect junto con su modelo matemático.

2.2 Cámara

El concepto de cámara digital fue propuesto por Eugene F. Lally del Jet Propulsion Laboratory en 1961 para generar información de navegación espacial a través de un fotosensor en mosaico el cual registraba la ubicación de estrellas y planetas. La primera cámara digital fue construida por la empresa Kodak en 1975, esta usó sensores CCD desarrollados por Fairchild Semiconductor en 1973 y guardaba la información en cassette; en 1988 la empresa Fuji generó la primera cámara digital que almacenaba las imágenes en formatos JPEG y MPEG desarrollados en el mismo año, lo que permitió una integración completa con los computadores.

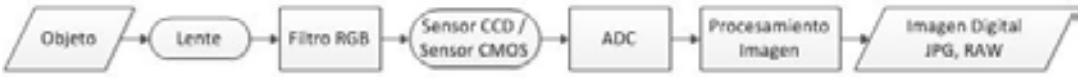

Figura 2.1: Funcionamiento de una cámara digital.

La estructura de funcionamiento de una cámara digital se observa en la Figura 2.1, el objeto es captado a través del lente de la cámara dando posibilidades a ajustes manuales a través del uso de los principios ópticos, la imagen capturada se filtra en los 3 canales del espacio de color RGB debido a que los sensores solo captan la intensidad de iluminación, este filtro puede ser:

- Fijo en matriz: Donde cada píxel tiene asociado un canal específico, el filtro más común es el mosaico de Bayer conformado por 50% filtros verdes, 25% filtros azules y 25% filtros rojos; esta distribución se debe a que el ojo humano es más sensible al color verde por lo que la imagen final es más real para el ojo humano.
- Filtro prisma dicroico: Usando un prisma se generan 3 proyecciones independientes que inciden en un sensor exclusivo para cada canal de color.

Luego de realizar el filtrado, la luz incide sobre los sensores CCD o CMOS, ambos se basan en el concepto del efecto fotoeléctrico descrito por Heinrich Hertz en 1887, y se componen de una matriz de fotositos/fotodiodos que emite un conjunto de señales eléctricas de acuerdo a la intensidad de la luz; los sensores CCD (Charge-Coupled Device) fueron inventados en los laboratorios Bell por Willard Boyle y George Smith en 1969, en estos dispositivos la corriente eléctrica generada por cada fotodiodo es enviada a un amplificador universal y luego enviada a un conversor análogo digital; los sensores CMOS difieren de los CCD en que por cada fotodiodo existe un amplificador y un conversor análogo digital, lo cual reduce

la cantidad de componentes externos y acelera el proceso de lectura, una comparación de ambos sensores se muestra en la Tabla 2.1.

Tabla 2.1: Comparación sensores CCD/CMOS [54].

Parámetros	CMOS	CCD
Calidad de la imagen	Excelente	Buena
Tecnología	Desarrollo	Establecida
Tamaño pixel	Pequeño	Grande
Procesamiento	Rápido	Estándar
Consumo potencia	Bajo	Alto
Ruido	Alto	Bajo

En la última etapa se guarda la imagen, en esta etapa existen múltiples formatos de almacenamiento tales como JPEG, TIFF o RAW; JPEG (Joint Photographic Experts Group) es un formato desarrollado por un grupo de expertos en digitalización en 1986, el objetivo es comprimir la imagen usando la transformada del coseno lo cual induce algunas perdidas en la calidad de la imagen no perceptibles al ojo humano, logrando una reducción del espacio en memoria, TIFF (Tagged Image File Format) es un formato desarrollado en 1993 por la compañía Aldus con el objetivo de crear imágenes de alta calidad para impresión, el fundamento del formato se encuentra en los encabezados los cuales contienen información útil de la imagen y por el uso de capas distintas para cada color.

Las cámaras digitales obtienen imágenes del ambiente dentro del rango permitido por el lente, sin embargo la imagen no proporciona información real sobre el ambiente que lo rodea, por tal motivo se usan múltiples algoritmos de procesamiento de imágenes con el fin de obtener información de la imagen; las cámaras actuales son de bajo costo y permiten la adquisición de imágenes en términos de milisegundos.

2.3 Sensor Infrarrojo (IR sensor)

Los sensores infrarrojos fueron inventados en 1941 por Robert J. Cashman y producidos en masa desde 1944 en la 'Northwestern University', se componen básicamente de un diodo emisor (Ga(NO3)3 y CsNO3) y un diodo receptor (HgCdTe). Las emisores son unidades que pueden emitir rayos en el rango de la frecuencia infrarroja (0,7 - 1000 μm), la cual es reflejada por los objetos según el material y geometría de su superficie. Los sensores infrarrojos tienen 2 campos activos:

- Activos: Los sensores IR activos tienen 2 clases: 'sensores de ranura' y 'sensores reflexivos', los primeros se caracterizan por tener un emisor junto con un receptor ubicados en la misma línea de acción con el fin de identificar las interrupciones al flujo continuo de fotones, en este campo se destacan aplicaciones industriales de

conteo de objetos, y ruedas perforadas de conteo de posición; la segunda clase de sensores utilizan un emisor y receptor en paralelo, al encontrar un objeto cercano con ciertas propiedades de reflectancia, los rayos infrarrojos regresarán indicando que se encuentra un objetos, [2] muestra que con un modelo correcto se puede estimar la distancia en términos de la corriente, aunque el alcance máximo es aproximado de 50mm y es propenso a errores de precisión.

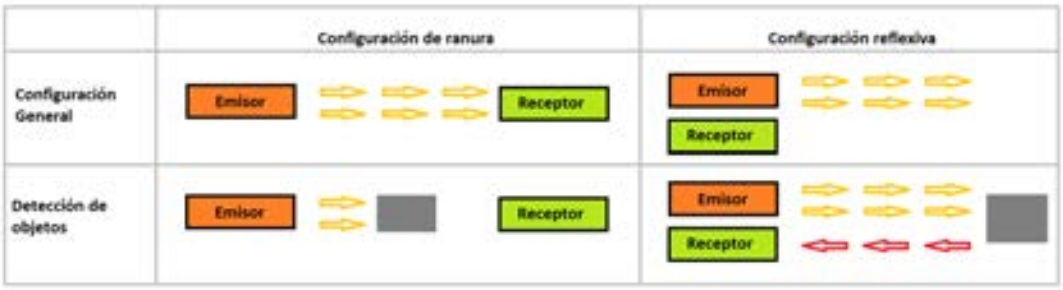

Figura 2.2: Configuración general de sensores IR pasivos.

- Pasivos: Los sensores IR pasivos son usualmente 'cámaras infrarrojas', esta clase de cámaras siguen el mismo tipo de funcionamiento que las cámaras digitales, como se muestra en la Figura 2.1, sin embargo los filtros se adecuan para dejar pasar solo la luz infrarroja que viene del ambiente la cual tiene una alta relación con la temperatura del objeto, por lo que también se usan como termómetro a distancia.

2.4 Kinect

El uso de sensores Activos para detectar objetos en el espacio ha aumentado considerablemente en los últimos años por la fabricación comercial de diferentes dispositivos como Kinect o Xion. El dispositivo Kinect fue lanzado al mercado el 4 de noviembre de 2010 por la empresa Microsoft y dada su popularidad se generaron versiones para desarrollo desde junio de 2011, esta popularidad se debe a su bajo costo, tamaño pequeño, bajo consumo, confiabilidad y velocidad en las medidas. Se usa comúnmente en aplicaciones robóticas de espacios cerrados, tales como reconstrucción y reconocimiento de objetos. Este dispositivo cuenta con un proyector IR, un receptor CMOS para IR, una cámara RGB y un micrófono como se muestra en la Figura 2.3.

La cámara infrarroja puede tomar imágenes con una resolución de 1280×1024 píxeles, con un campo de visión de 57 grados en el eje horizontal y de 45 grados en el eje vertical, 6.1 mm de distancia focal y un tamaño de píxel de 5.2 μm [63], junto con el emisor infrarrojo tienen la capacidad de hallar la profundidad de un objeto en el espacio con un alcance entre

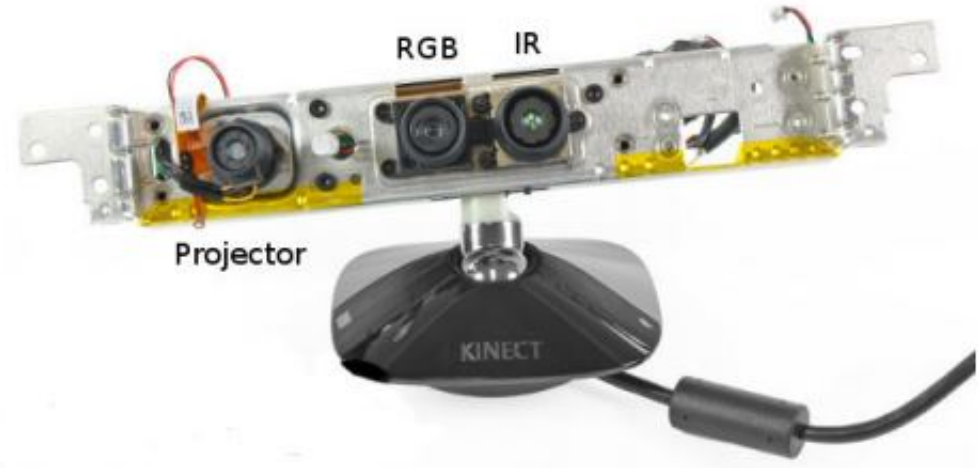

Figura 2.3: Estructura de un Kinect [63] .

0.685 m y 15 m. La cámara CMOS puede tomar imágenes en RGB con una resolución de 1280×1024 píxeles, con un campo de visión de 63 grados en el eje horizontal y de 50 grados en el eje vertical, 2.9 mm de distancia focal y un tamaño de píxel de 2.8μm [63].

En [63] se modela el kinect como un sistema multi-vista con 2 cámaras (Color e infrarroja) y un proyector infrarrojo, como se muestra en la Figura 2.4, para cada una de las cámaras se usa el modelo pinhole, este modelo geométrico proyecta diversa cantidad de puntos en el espacio $P = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$ en sus correspondientes pares de puntos en la imagen $p = \begin{bmatrix} u & v \end{bmatrix}^T$ a través de la matriz de cámara de color K_{RGB} y de la matriz de cámara infrarroja K_{IR} , la proyección implica también remover la distorsión radial de ambas cámaras a través de las funciones f_{RGB}, f_{IR}, finalmente se aplica una transformación T^{IR}_{RGB} desde el marco de referencia de la cámara IR a la cámara de color.

De esta forma, el modelo del kinect inicia como un punto en la cámara infrarroja p_{IR} que se transforma al plano de la imagen infrarroja a través de K_{IR} y se remueve su distorsión con f_{IR}, este punto se denomina P_{IR}

$$P_{IR} = f_{IR}^{-1}\left(K_{IR}^{-1} p_{IR}\right),$$

luego, el punto en el plano de la imagen infrarroja se traslada al plano de la imagen de color con la matriz de transformación T^{IR}_{RGB} , entonces para cada píxel de la imagen de color hay un píxel correspondiente con la información de profundidad, el modelo de proyección es entonces:

$$P = K_{RGB} f_{RGB}\left(\frac{1}{Z_{RGB}}\begin{bmatrix} I & 0 \end{bmatrix} p_{RGB}\right). \tag{2.1}$$

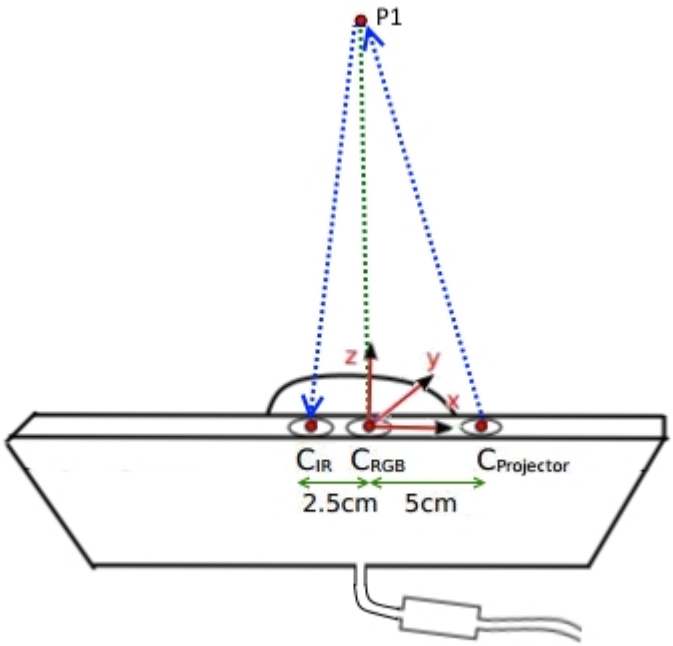

Figura 2.4: Modelo geométrico de un Kinect.

2.5 Calibración

La calibración de cámara es la primera etapa común en visión de máquina, el objetivo es obtener parámetros internos de la cámara, estos parámetros permiten mejorar la precisión de las operaciones que se apliquen o efectúen con estas imágenes. Para este proyecto la calibración de ambas se realizó con el modelo pinhole, que se muestra en la Figura 2.5, en el cual se asume que unos puntos en el espacio están conectados a unos puntos en el plano de la imagen, a través de líneas rectas que se intersectan en un punto llamado 'pinhole'.

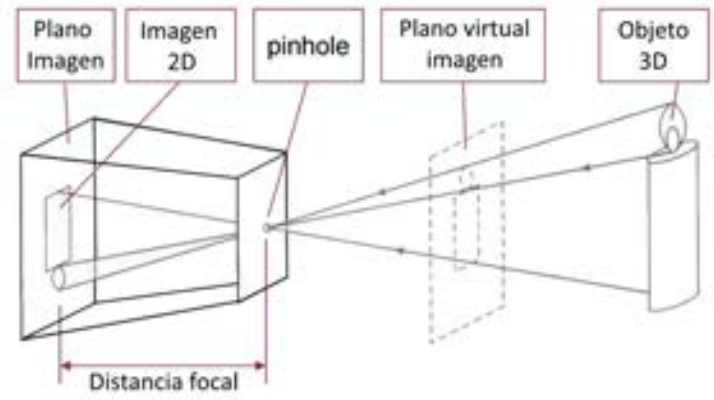

Figura 2.5: Modelo Pinhole.

Matemáticamente, la calibración consiste en hallar la matriz C, con dimensiones 3×4 y con parámetro $C(3,4) = 1$, lo que devela 11 parámetros a encontrar. Para encontrar los parámetros se necesitan conocer correspondencias entre los puntos 3D y los puntos en las

imágenes. Si un punto en el espacio $P(X, Y, Z)$ corresponde a uno en la imagen $p(u, v)$, entonces la transformación entre ambos es:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{u}{w} \\ \frac{v}{w} \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = C \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{2.2}$$

Donde el primer vector corresponde a los píxeles en la imagen obtenida, los cuales corresponden a un vector en el espacio tridimensional en el plano de la imagen, y a su vez son producto de una transformación del punto real en el espacio con ayuda de la matriz C. Se definen las filas de C como r_1, r_2, r_3:

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}. \tag{2.3}$$

Entonces la relación de la Ecuación 2.2 puede definir como:

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} P. \tag{2.4}$$

Lo cual produce 2 ecuaciones lineales:

$$u'(r_3 P) - r_1 P = 0, \tag{2.5}$$

$$v'(r_3 P) - r_2 P = 0. \tag{2.6}$$

De esta forma, una correspondencia en el plano y en el espacio genera 2 ecuaciones lineales, para resolver los 11 parámetros de C se necesitan como mínimo 6 ecuaciones, sin embargo, en la práctica, se usan más de 6 para mejorar la precisión de los valores.

El primer paso para calibrar es obtener imágenes a través de los sensores, entonces con ambas cámaras se capturaron diferentes escenas con un patrón en forma de ajedrez, el patrón de ajedrez consiste en una grilla de cuadrados de 8×11, en la Figura 2.7 se puede observar las 15 capturas realizadas tanto para la cámara RGB como para la cámara IR, con cada una de las imágenes en ambas cámaras se encuentra el patrón a través de la función *findChessboardCorners* de OpenCV. Con cada una de estas imágenes se hallaron las esquinas internas, la función implementada aplica filtros gaussianos con diferentes valores de sigma, luego aplica restas de imágenes y halla las líneas rectas de la imagen, las líneas cuyas separaciones espaciales sean similares al valor del lado del cuadrado se consideran como bordes del tablero [5], lo que corresponde a 70 puntos espaciales, en la Figura 2.6 se

muestra la detección de estos puntos para ambas cámaras.

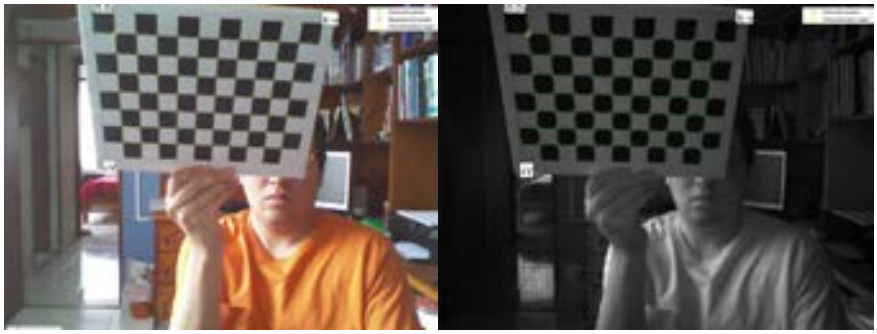

Figura 2.6: Ejemplo de detecciones en el patrón en forma de ajedrez.

Se establece la esquina superior izquierda del tablero como centro coordenado del mundo, de esta forma la componente Z de todos los puntos es 0 debido a que están en un plano, las componentes X, Y son conocidas puesto que la estructura y distribución física del tablero se conocen. De esta forma las ecuaciones 2.5 y 2.6 se pueden escribir de la siguiente manera:

$$u' \begin{bmatrix} c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = 0. \tag{2.7}$$

$$v' \begin{bmatrix} c_{31} & c_{32} & c_{33} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} c_{21} & c_{22} & c_{23} & c_{24} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = 0. \tag{2.8}$$

Al tener más correlaciones que las necesarias, se reducen los parámetros de C por optimización de mínimos cuadrados, escribiendo C en términos de $\phi, \varphi, \psi, T, \frac{f}{S_x}\alpha, O_x, O_y$ la optimización halla los parámetros aproximados de la matriz intrínseca K. Los parámetros internos encontrados para este proyecto se muestran en el Tabla 2.2, la calibración se realizó a través del software GML Camera Calibration Toolbox 0.72.

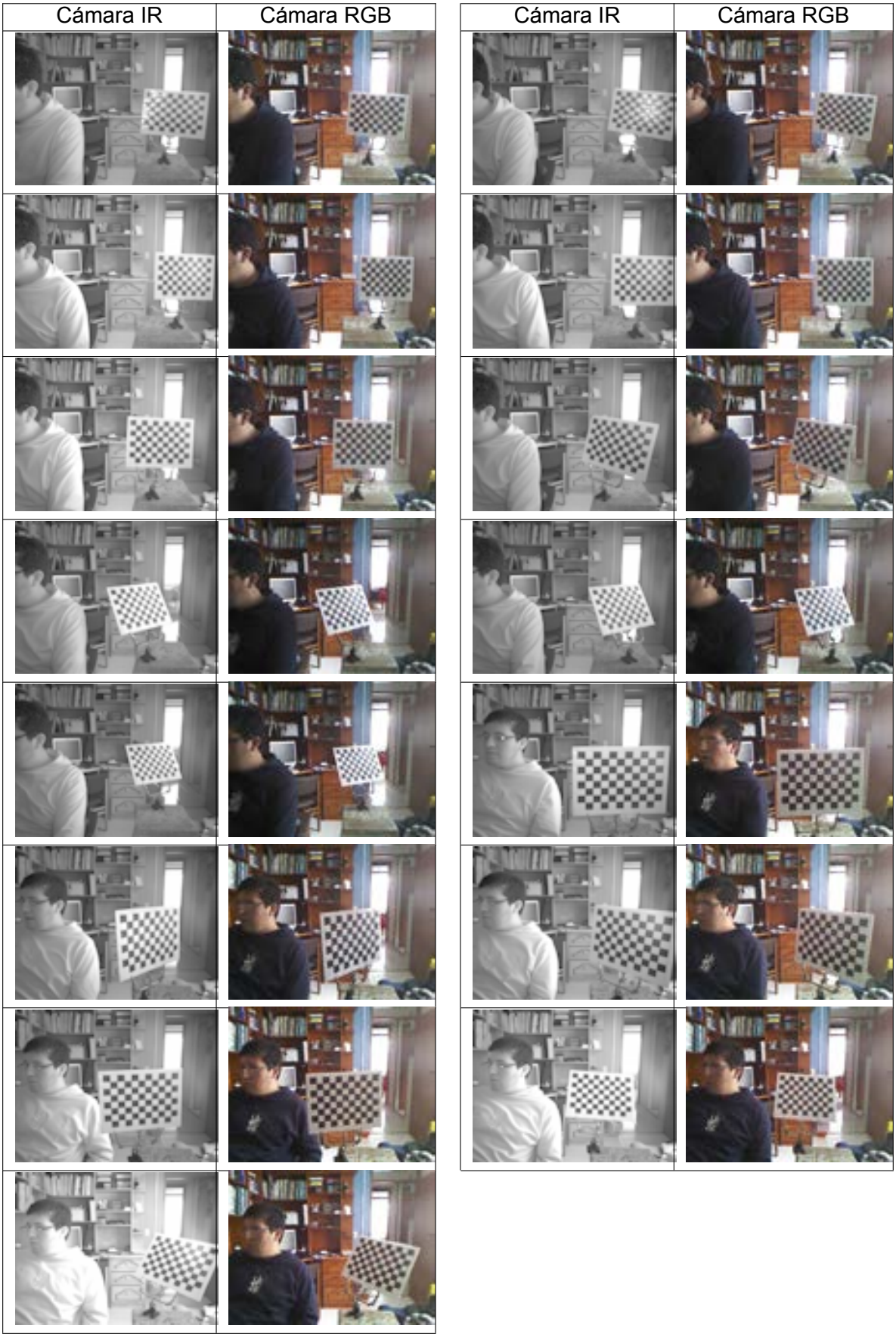

Figura 2.7: Capturas para la calibración.

Tabla 2.2: Parámetros internos de las cámaras IR y RGB.

IR		
Variable	Estándar	Valor
f_x	571.26	615.432059
f_y	571.26	577.039286
c_x	319.5	342.207119
c_y	239.5	242.174101
k_1	0.0	0.044879
k_2	0.0	0.078431
k_3	0.0	0.0
p_1	0.0	0.002916
p_2	0.0	0.002718

RGB		
Variable	Estándar	Valor
f_x	525.0	535.501896
f_y	525.0	537.504906
c_x	319.5	330.079632
c_y	239.5	248.201700
k_1	0.0	0.119773
k_2	0.0	-0.369037
k_3	0.0	0.0
p_1	0.0	0.002639
p_2	0.0	0.007776

$$T_{RGB}^{IR} = \begin{bmatrix} 1 & -0.0026 & -0.006 & -24.568 \\ 0.0026 & 1 & -0.0018 & 0.271 \\ 0.006 & 0.0017 & 1 & 0.801 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Capítulo 3

Técnica SLAM propuesta

Dada la trayectoria del problema SLAM en la comunidad investigativa, se han formulado múltiples soluciones, cada una orientada hacia el uso de un tipo específico de sensor o la generación de mapas con estructuras definidas. Este trabajo se orienta hacia las técnicas SLAM que hacen uso de sensores RGB-D y que generan mapas voxelizados, en esta área de desarrollo se usan actualmente 2 técnicas independientes, basados en el alineamiento de características visuales (Keypoint Alignement) [34, 36, 25, 33] y los que se basan en la foto-consistencia visual (Dense visual odometry) [23, 10, 24]. La propuesta que se desarrollará está basada en la técnica Keypoint-Alignement, que hace uso de un sensor RGB-D, el cual proporciona información sobre el ambiente en forma de un par de imágenes de color y profundidad, el movimiento de la cámara es estimado aplicando SLAM a través de los frames consecutivos, cuya precisión depende de la calibración previa del sensor. De esta forma el sensor RGB-D se puede usar de forma libre para reconstruir el ambiente y/o reconocer su posición, la información adquirida consecutivamente por la cámara RGB-D se puede ordenar de acuerdo al tiempo de adquisición por medio de *frames*, al ser consecutivos la probabilidad de que se encuentren correspondencias en 2 o más *frames* aumenta considerablemente, estas correspondencias permiten estimar la transformación de la cámara en el espacio. Sin embargo, las técnicas SLAM presentan algunos problemas respecto a la carga computacional necesaria, esta carga está representada principalmente en los algoritmos iterativos de refinamiento de pose y de emparejamiento de características, también existen problemas de desempeño en ambientes no estáticos, el emparejamiento de características no considera que los objetos a los que pertenecen las características pueden aparecer, moverse, o desaparecer en el ambiente. Para solucionar el problema de desempeño se propone un nuevo clasificador con base a un algoritmo heurístico, esto permite reducir la complejidad computacional con el fin de comparar las características en menor tiempo, de igual forma se proponen un nuevo tipo de características extendidas a partir de SURF, esta extensión tiene como objetivo mejorar la capacidad discriminativa de las características para reducir el error en la clasificación y con esto reducir la complejidad computacional necesaria para hallar las correspondencias entre *frames*. La invariabilidad a los ambientes

dinámicos es abordada desde el tipo de mapa, en esta propuesta se desarrolla un tipo de mapa que considera la probabilidad de ocupación de las características de un objeto, con esto la alineación de las características con el mapa creado evita los errores cuando objetos del mapa cambian de posición. Este capítulo continúa con una formalización matemática y técnica de la técnica SLAM elegida, esta formalización contendrá la definición de los problemas de técnica, que se enfrentan en este trabajo. El capítulo termina con una formulación matemática y técnica de las soluciones propuestas, y la forma en que se desarrollan en la solución.

3.1 Modificaciones propuestas

En el Capítulo 1 se presentó la técnica *Keypoint-Alignement*, la cual como base para el desarrollo de esta propuesta. La técnica usa la alineación de características para encontrar una transformación T entre los puntos de interés, con el fin de mejorar el desempeño de la técnica se propone modificar el tipo de características y el alineador de características.

El desempeño de los algoritmos de clasificación depende de las meta-características del conjunto de datos a clasificar [58, 38, 29], de esta forma la estructura interna de las características influye considerablemente en cual algoritmo de clasificación usar. Normalmente las características extraídas en aplicaciones SLAM como [23, 34] son SIFT [45], SURF [9], y ORB [59]. La extracción de características se hace siempre sobre la transformación a grises de la imagen de color, al omitir la imagen de profundidad se pierde información espacial, se propone entonces la introducción de información espacial al algoritmo de detección de características, con el objetivo de mejorar la capacidad discriminativa, logrando una mayor cantidad de clasificaciones correctas.

El emparejamiento de características es un problema identificado en diferentes evaluaciones como [23], en particular el tiempo de emparejamiento y la cantidad de *frames* a comparar son parámetros de difícil elección. El tiempo de emparejamiento depende básicamente de la complejidad computacional del algoritmo, una elección común a todos los algoritmos revisados muestra que se usa Nearest Neighbors (FLANN) [50], el cual tiene una complejidad de construcción $O(F * \log(F))$ y una complejidad de búsqueda $O(\log(F))$. Se propone el uso de un algoritmo heurístico [30] el cual podría reducir el tiempo de clasificación y con esto mejorar el desempeño de la técnica SLAM, la reducción del tiempo de clasificación también impactaría en el número de *frames* a comparar, dando mayor consistencia en los mapas generados.

Los entornos de desempeño robóticos son normalmente dinámicos, pero la mayoría de soluciones propuestas al problema SLAM [34, 36, 25, 33] asumen que el espacio es estático, lo que permite mapear el entorno como un conjunto sucesivo de mediciones con una transformación T. Sin embargo los entornos presentan cambios constantes [74, 72], objetos en movimiento, objetos que aparecen o desaparecen en el tiempo, cambios de estado o de

forma, entre muchas consideraciones que afectan el desempeño de las técnicas SLAM. Se propone que el sistema de mapas considere la posibilidad de que un píxel cambie, asumiendo que todos los objetos en el mundo son masivos y poseen la misma densidad, entonces la probabilidad de que un objeto cambie de estado está relacionado directamente con su volumen, considerando que la cantidad de energía necesaria para moverlo es directamente proporcional a su volumen y que la probabilidad del uso de energía es inversamente proporcional a su cantidad, de tal forma objetos muy pequeños no se consideran parte del mapa ni de las características del entorno dado que tienen una probabilidad alta de movimiento, y la consistencia del mapa mejoraría si se mantiene solo los objetos masivos.

3.1.1 Características SURF extendidas

Los vectores de características se hallan a través del procesamiento de la imagen de color en cada medición con extractores como SIFT, SURF y ORB, estos son comúnmente usados en técnicas SLAM debido a su efectividad, la imagen de color usualmente se convierte a grises para dar invariabilidad al color. Para esta propuesta se usará el extractor SURF (Speeded up robust features) [9], SURF es un algoritmo basado en histogramas de gradientes, fue desarrollado para acelerar el cálculo de características respecto a SIFT. La modificación propuesta para este trabajo consiste en calcular los puntos de interés en la imagen de profundidad y en la imagen de color, estos puntos se adicionan en un solo vector como se observa en la Figura 3.1.

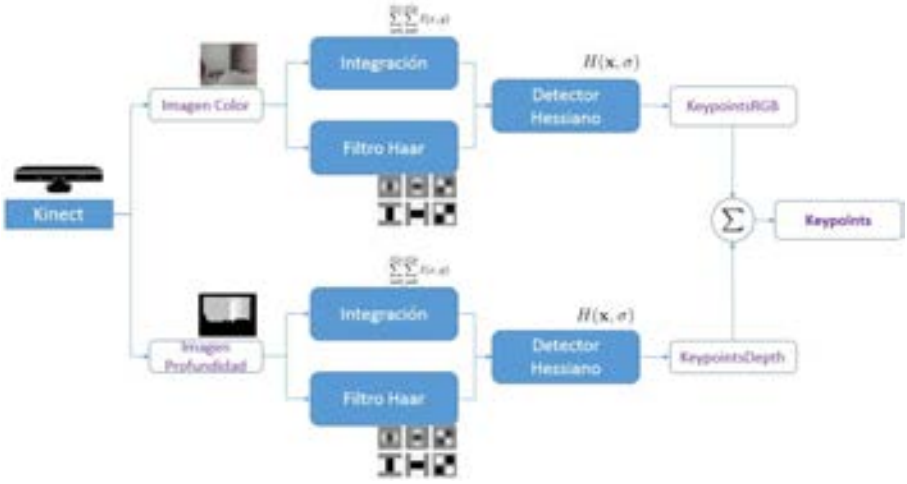

Figura 3.1: Proceso propuesto para la extracción de características extendidas, el Kinect realiza la captura de la imagen a color y la imagen de profundidad, cada una de ellas se les aplica una integración y una serie de filtros, luego se extraen los puntos de cada imagen con el detector hessiano y se mezclan para obtener los puntos característicos.

El cálculo de descriptores se realiza sobre la imagen de color con el vector previamente calculado, de esta forma se introducen puntos de interés de origen geométrico al vector de

características, el resultado final es un vector de características SURF con mayor cantidad de puntos, estas características se denominan *Características SURF extendidas*. A continuación se detalla el procedimiento de extracción de características, el cual está dividido en la identificación de puntos característicos y el cálculo de características, el primer paso es aplicado a las imágenes de color y profundidad mientras que el segundo paso es solamente aplicado a la imagen de color.

El primer paso para la extracción de características es identificar el conjunto de puntos en la imagen que son distintivos y puedan ser localizados bajo diferentes variaciones en el punto de vista o en la presencia de ruido. Para identificar estos puntos el método genera una imagen integral, la cual se define como la suma de todos los píxeles anteriores a cada coordenada x, y como se define en la Ecuación 3.1.

$$I(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(x, y). \tag{3.1}$$

La identificación de puntos se hace a través de un detector Hessiano, el determinante de la matriz Hessiana está correlacionada directamente con el cambio del área respecto a cada punto, esta matriz se define en la Ecuación 3.2, donde L_{xx} es la convolución de la imagen con la segunda derivada de la gaussiana, sin embargo esta convolución es computacionalmente costosa por lo que SURF aproxima esta convolución con kernels aplicados en la imagen integral de la Ecuación 3.1.

$$\mathcal{H}(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}. \tag{3.2}$$

Los kernels usados para aproximar las gaussianas se muestran en la Figura 3.2, donde las regiones grises tienen valores de 0, las regiones blancas valores de 1, y las regiones negras un valor de -2 para L_{xx}, L_{yy} y un valor de -1 para L_{xy}.

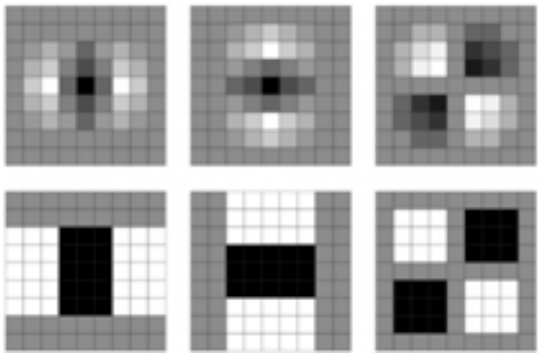

Figura 3.2: Aproximaciones a las derivadas gaussianas por filtros, L_{xx}, L_{yy}, L_{xy}.[9]

Al realizar la convolución de la imagen integral con cada kernel aproximado se obtiene: D_{xx} en vez de L_{xx}, D_{yy} en vez de L_{yy} y D_{xy} en vez de L_{xy}, el determinante aproximado de

la matriz hessiana es la Ecuación 3.3 con un parámetro w que se usa como constante en 0.9 [9], los puntos en los que el determinante sea mayor a 0 se consideran keypoints.

$$Det(\mathcal{H}) = D_{xx}D_{yy} - (w * D_{xy})^2. \tag{3.3}$$

A diferencia de SIFT, SURF examina la imagen a diferentes escalas y octavas del filtro con el fin de garantizar la invariabilidad a la escala, esta pirámide de imágenes se observa en la Figura 3.3.

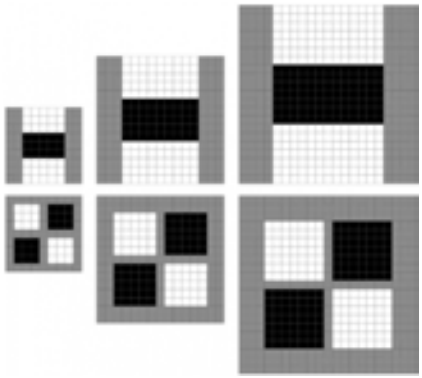

Figura 3.3: Incremento de escala del filtro a convolucionar con la imagen. [9]

El segundo paso para la extracción de características es el cálculo de los valores descriptivos, en cada punto obtenido se genera una región de interés, esta región es un cuadrado con lados de dimensión $20s$, donde s es la escala a la cual se detectó el punto, esta región se subdivide en 16 regiones como se muestra en la Figura 3.4.

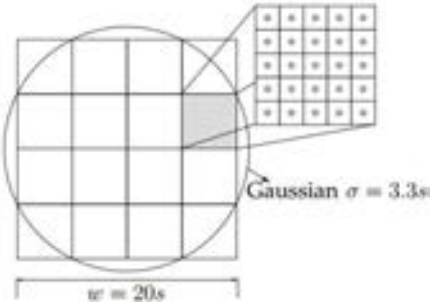

Figura 3.4: Subregiones para cada punto de interés [9].

En cada subregión se calcula las respuestas a los filtros de Haar, los cuales son unos kernel binarios que se muestran en la Figura 3.5, con la respuesta se realiza la sumatoria en cada dirección y una sumatoria absoluta en cada dirección como se muestra en la Ecuación 3.4, estos 4 valores son las características de cada subregión, lo que crea para cada punto de interés un vector de 64 valores.

$$v = \left\{ \sum dx \quad \sum |dx| \quad \sum dy \quad \sum |dy| \right\} \tag{3.4}$$

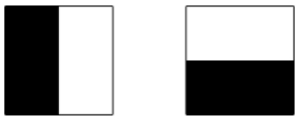

Figura 3.5: Filtros de Haar ∂x y ∂y [9].

El objetivo de la identificación de puntos, a través del determinante de la matriz Hessiana, es obtener coordenadas distinguibles bajo modificaciones de escala, rotación e iluminación, al usar los filtros de la Figura 3.2 como aproximaciones a las gaussianas, se identifican posiciones en la imagen donde el gradiente de la región cambie lo suficiente. En las imágenes estos puntos suelen estar relacionados con bordes o texturas como se puede observar en la Figura , en el desarrollo de este proyecto se propone usar la imagen de profundidad para extraer puntos de interés, la imagen de profundidad captura la geometría del entorno bajo ciertas condiciones de ruido, rango e iluminación exterior, sin embargo los bordes o los cambios abruptos de geometría espacial que pueden ser identificados en las imágenes de color bajo otras perspectivas son visibles en la imagen de profundidad.

3.1.2 Alineador de características heurístico

La alineación de características consiste en comparar dos agrupaciones de características y encontrar las correspondencias respectivas basadas en una función de costo. En procesamiento de imágenes normalmente se usa KNN, un algoritmo de aprendizaje supervisado propuesto en 1951, que se basa en que las instancias de una misma clase deben tener propiedades similares, por ende, la clase de cualquier instancia puede ser inferida a través de las clases de K instancias vecinas [41]. KNN es un método ampliamente usado, sin embargo tiene varias desventajas, como la necesidad de un gran espacio de almacenamiento en memoria; sensibilidad a la función de similitud escogida; sensibilidad al valor de K y tiempos altos de clasificación. Este tiempo de clasificación se determina por la complejidad del algoritmo, actualmente los algoritmos KNN usan una estructura de datos tipo kd-tree, la complejidad de construcción del kd-tree (proceso de entrenamiento) es $O(N \log N)$ y la clasificación de una instancia tiene complejidad promedio de $O(\log N)$ y en el peor de los casos de $O(N)$, siendo N el número de instancias en la etapa de clasificación.

En este trabajo se propone cambiar el algoritmo KNN por un algoritmo meta-heurístico inspirado en el algoritmo optimización Fast Simulated Annealing [37], el cual mejora el algoritmo meta-heurístico Simulated Annealing [61]. El objetivo es reducir el tiempo de clasificación sin afectar significativamente el porcentaje de clasificaciones correctas. El recocido (Annealing) es un tratamiento térmico de materiales que conjuga en el aumento de la temperatura en un sólido y su enfriamiento lento, eliminando así las tensiones internas entre los granos del material, recuperando la estructura molecular. De esta manera, las partículas se reacomodan en estados de más baja energía, hasta que se obtiene un sólido con partículas en mejor equilibrio térmico, conforme a una estructura de cristal [6].

El algoritmo de recocido simulado tiene como base el método de Monte Carlo, el sólido en cada iteración i tiene una energía ε_i representada como el valor de una función objetivo $f(i) = \varepsilon_i$, luego se aplica una perturbación al sistema que genera un estado alterno con función objetivo $f(j) = \varepsilon_j$, para aceptar o rechazar la perturbación el algoritmo de recocido simulado usa el criterio de metrópolis, entonces si $f(i)$ es mejor que $f(j)$ se acepta el cambio, en caso contrario si $f(i) - f(j) > 0$, la probabilidad de aceptar el cambio está dada por $\tan^{-1}\left(exp\left(-\frac{f(j)-f(i)}{c}\right)\right)$, donde c es un parámetro de control. Originalmente este procedimiento se repite T veces, siendo T una variable análoga a la temperatura inicial. En [46, 1] se han sugerido técnicas para estimar el valor de T, sin embargo, en esta propuesta se usará un valor fijo con el fin de garantizar una complejidad constante. El número de iteraciones y la tasa de enfriamiento son parámetros que influyen en el desempeño del algoritmo; en este trabajo se aplica la propuesta de [37] en la cual, la función de enfriamiento está dada por la Ecuación 3.5, lo que le permite al algoritmo adaptar la taza de acuerdo a la cantidad de características M de cada problema y, además se hace visible en la actualización de la temperatura a través de cada iteración, donde k es el número de iteraciones realizadas.

$$T_K = \frac{T}{k^{1/M}}. \tag{3.5}$$

Durante el entrenamiento se tienen N puntos en el mundo, cada uno con M características, este espacio se asumirá como el material ϕ. El material debe tener una organización que garantice la máxima dispersión de características. A partir de las muestras se halla la desviación estándar sobre cada una de las M características σ_M. El material ϕ se ordena de mayor a menor en la columna con mayor desviación estándar, en caso de valores iguales se ordena con base en la siguiente columna con mayor desviación estándar. El material ϕ se representa en la Ecuación 3.6, contiene M características y para cada una de ellas su valor σ correspondiente, se hace notar que el material de la Ecuación 3.6 se puede considerar como el espacio W de la Ecuación 1.8 considerando solo los puntos que tienen características.

$$\phi = \begin{cases} \phi_0 \begin{pmatrix} x_0 & y_0 \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \phi_2 \begin{pmatrix} x_2 & y_2 \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \phi_{10} \begin{pmatrix} x_{10} & y_{10} \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \vdots & \vdots \\ \phi_{N-25} \begin{pmatrix} x_{N-25} & y_{N-25} \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \vdots & \vdots \\ \phi_{N-5} \begin{pmatrix} x_{N-5} & y_{N-5} \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \phi_N \begin{pmatrix} x_N & y_N \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \sigma & \begin{Bmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_M \end{Bmatrix} \end{cases}. \tag{3.6}$$

En la clasificación se adquiriere una cantidad n de muestras y se aplica el mismo algoritmo de extracción de características, este nuevo grupo de muestras conforman el espacio μ, como se observa en la Ecuación 3.7.

$$\mu = \begin{cases} \mu_0 \begin{pmatrix} x_0 & y_0 \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \mu_7 \begin{pmatrix} x_7 & y_7 \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \vdots & \vdots \\ \mu_{n-10} \begin{pmatrix} x_{n-10} & y_{n-10} \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \\ \vdots & \vdots \\ \mu_n \begin{pmatrix} x_n & y_n \end{pmatrix} = & \begin{Bmatrix} f_1 & f_2 & \cdots & f_M \end{Bmatrix} \end{cases}. \tag{3.7}$$

Con el material ϕ y la lista de clasificación μ se asignan aleatoriamente cada muestra del espacio μ con una muestra en el espacio ϕ, como se muestra en la Figura 3.6, esta configuración inicial es usualmente usada en los algoritmos de optimización bio-inspirada con el fin de garantizar la diversidad de la solución.

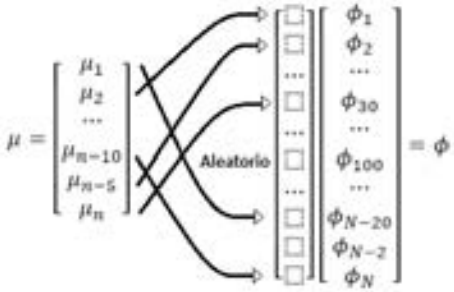

Figura 3.6: Distribución aleatoria de las muestras del espacio μ en el espacio ϕ .

El proceso de clasificación es iterativo, determinado por un valor T definido previamente, al inicio de cada iteración se calcula el valor de la distancia d_1 entre cada muestra μ y su respectivo par ϕ, esta distancia entre 2 vectores usa la métrica cityblock, la cual es la sumatoria de la diferencia absoluta de los vectores.

$$d = \sum_{i=1}^{M} |\mu - \phi|. \tag{3.8}$$

Cada punto en μ tiene un espacio propio para moverse en ϕ, dependiendo de la temperatura, el tamaño de este espacio es el número de muestras N cuando T es máxima y 1 cuando $T = 0$, este cambio se planteó empíricamente con una función exponencial mostrada en la Ecuación 3.9. Este espacio es su índice actual más o menos φ, en este rango se elige aleatoriamente un punto del espacio ϕ como propuesta de cambio para cada partícula.

$$\varphi = \frac{\exp\left(0.693 * T\right) - 1}{2}. \tag{3.9}$$

Cuando se eligen un punto como propuesta de cambio se calcula la distancia d_2 con el punto de muestra, el cálculo de esta distancias sigue el mismo procedimiento de la Ecuación3.8. Con las distancias calculadas a dos puntos distintas por punto de muestra, puede ocurrir que:

- Si $d_2 < d_1$ se acepta el punto propuesto como nueva pareja del punto de muestra.
- Si $d_1 \geq d_2$ se elige un número aleatorio entre 1 y N, el cambio es aceptado si este número es menor a $\frac{1}{10}(1 - \frac{T}{T_{max}})$.

El clasificador propuesto, para la etapa de entrenamiento considera todas las características del mundo como partículas de un material metálico y las organiza de acuerdo a la variabilidad de sus valores. Para la etapa de clasificación, considera todas las características en las observaciones como partículas independientes que entran aleatoriamente al material organizado previamente. Al reducir la temperatura, las partículas se mueven a un estado de menor energía, la energía está representada por la distancia entre sus características y las de una partícula aleatoria vecina, existe también la posibilidad de cambiar a un estado de mayor energía siempre y cuando una función de probabilidad acepte este cambio con el fin de garantizar la diversidad de la solución. Cuando el ciclo termina, la partícula queda emparejada con la partícula de la última iteración.

3.1.3 Mapas jerárquicos y probabilísticos

Las soluciones propuestas actualmente al problema SLAM, tales como [34, 36, 25, 33], generan una reconstrucción del entorno a partir de la alineación consecutiva de mediciones, esta alineación está fuertemente basada en la suposición de que la mayoría de los puntos clave en una medición se encuentran en las mediciones anteriores, esto implica que el ambiente debe ser estático en todo momento. Sin embargo los ambientes reales son dinámicos [74, 72], este carácter dinámico implica que los objetos están en constante movimiento, aparecen o desaparecen, o cambian sus propiedades visuales a través del tiempo.

Los mapas para sistemas SLAM suelen ser nubes de puntos voxelizadas [74, 28], estas nubes de puntos suelen basarse en estructuras tipo árbol como se plantea en [72]. En [74] se genera un algoritmo robusto para eliminar puntos dinámicos en las mediciones, basándose en los datos de múltiples cámaras en el entorno pueden clasificar estos puntos como estáticos o dinámicos, de acuerdo con la consistencia de la triangulación. En [28] se propone la creación de un modelo del entorno en CAD siguiendo la metodología de [16], donde las características sean borradas si no han sido observadas en más del 50% de las mediciones donde deberían ser visibles.

El mapa es una representación del mundo, el símbolo de cada mapa es entonces W_i donde i es un número consecutivo, los mapas se crean cuando el sistema de navegación no sea capaz de encontrar una transformación adecuada entre ellos y existe una cantidad

significativa de mediciones que puedan definir cada uno de los espacios. El mapa propuesto se basa en la estructura de un árbol de 8 hijos, cada uno para una dirección en el espacio tridimensional, este espacio tiene el nombre de *voxel*, en cada voxel se representa una posición espacial x, y, z, un color r, g, b un valor de incertidumbre b y un vector de características F. Un espacio no descubierto del mapa tiene la máxima incertidumbre $b = 1$, con la cual se garantiza que cualquier medición en este espacio va a ser aceptada, al ingresar el valor de un píxel en el mapa la medición debe proveer el área del grupo al cual pertenezca el píxel en la imagen de profundidad, esta área se obtiene a través de la segmentación de la imagen de profundidad como se observa en la Figura 3.7, el área en esta imagen es entonces una medida proporcional al volumen del objeto en el espacio.

Figura 3.7: Segmentación en imagen de profundidad.

Con la medida del área del grupo del píxel A_p, se halla la proporción respecto a la máxima área posible en la imagen A_{max}, esta proporción es la medida de incertidumbre dada por la Ecuación 3.10, la cual asume que la probabilidad de cambio está directamente relacionada con el tamaño del objeto en el entorno, asumiendo una densidad uniforme en el espacio este tamaño está relacionado con la cantidad de energía necesaria para realizar el movimiento de los objetos.

$$b = \frac{A_p}{A_{max}}. \tag{3.10}$$

El sistema de mapa propuesto considera 3 funcionalidades:

1. Inserción: La operación de inserción recibe el conjunto de puntos C y una transformada T desde el marco de referencia de la cámara hacia el marco de referencia del mundo W_i. La inserción se realiza por cada punto P, considerando los errores de medición, proyección y alineación, un punto P_m no va a estar en la posición exacta que debe

reemplazar, en vez de esto el mapa selecciona la partícula P_W más cercana dentro de un espacio de error ε como se observa en la Figura 3.8, el punto de la medición P_m reemplaza al punto del mapa si su valor de incertidumbre N_M es mayor al que tiene el mapa N_W. El reemplazo consiste en eliminar en la estructura del mapa el punto previo y adicionar un punto con su valor de posición espacial, color, incertidumbre y características.

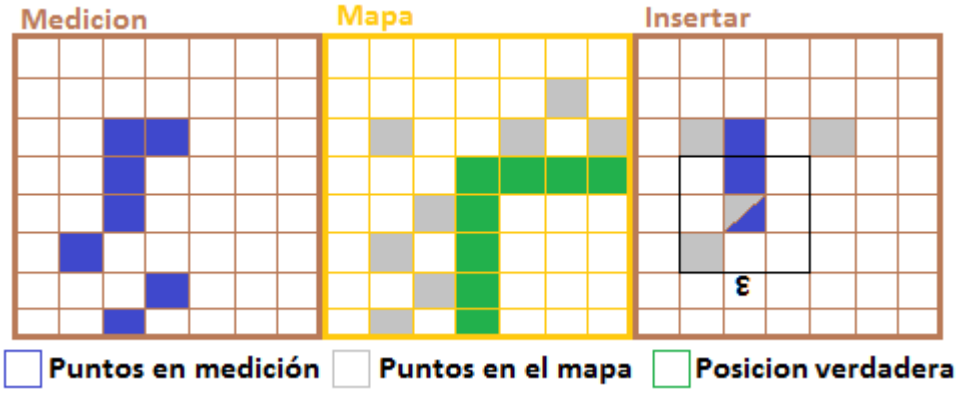

Figura 3.8: Procedimiento de inserción en el mapa.

2. Obtención: La operación de obtención recibe los límites en el marco de referencia del mapa de los cuales se desea obtener un conjunto de puntos C con sus respectivas características F.

3. Refinar: La operación de refinar no recibe ni devuelve información al sistema, su función es la de recorrer la estructura del mapa verificando que cada vóxel este en la posición correcta de la estructura según su valor espacial. Al tiempo que recorre el mapa W se buscan correlaciones entre los puntos para corregir desplazamientos o para hallar *loop closures*, como se muestra en la Figura 3.9.

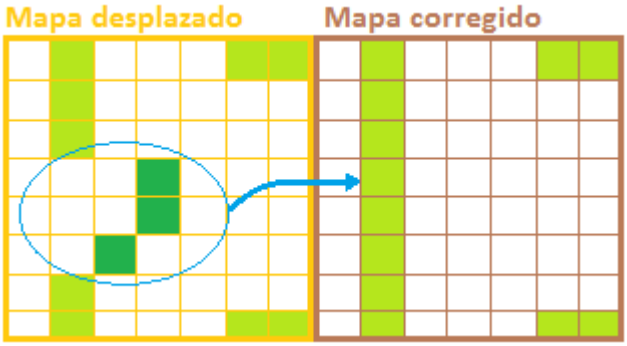

Figura 3.9: Procedimiento de refinar en el mapa

3.2 Odometría por características extendidas

Con las modificaciones propuestas en la Sección 3.1, se adaptó la técnica *Keypoint Alignement* para evaluar la mejora en cuanto al rendimiento y calidad del mapa generado.

La técnica modificada se denomina Odometría por características extendidas y recibe un par de imágenes RGBD, la imagen de color I_{rgb} se convierte a grises en la etapa de preproceso, luego se calculan los puntos de interés sobre la imagen de color I_{rgb} y sobre la imagen de profundidad I_{depth}, como resultado se obtienen un vector de puntos espaciales para cada imagen. Este proceso se muestra en la Ecuación 3.11 y en la Ecuación 3.12, donde Ψ representa la función de extracción de puntos de interés, esta función realiza la convolución entre la imagen y filtros de las gaussianas aproximadas de la Figura 3.2, calcula la matriz Hessiana y devuelve un vector de los puntos cuyo determinante Hessiano sea mayor a 0.

$$C_{rgb} = \Psi(I_{rgb}) \tag{3.11}$$

$$C_{depth} = \Psi(I_{depth}) \tag{3.12}$$

El vector de puntos C_{rgb} corresponde a los puntos en el espacio donde el cambio de textura tiene capacidad discriminante, mientras que el vector de puntos C_{depth} corresponde a los puntos en el espacio donde el cambio de geometría tiene capacidad discriminante. Asumiendo que los cambios en la geometría también corresponden a cambios en la textura, se calcula el vector C como la suma de los vectores calculados, como se muestra en la Ecuación 3.13.

$$C = C_{rgb} + C_{depth} \tag{3.13}$$

Con el vector de puntos de interés C, se realiza el cálculo de las características F sobre la imagen I_{rgb}. Este proceso se muestra en la Ecuación 3.14, donde Γ representa la función de extracción de características. Esta función genera una subregión sobre cada punto en la lista C, sobre la cual se realiza la convolución con los filtros de Haar de la Figura 3.5; al medir el resultado según la Ecuación 3.4 se obtiene la respuesta de la función Γ, la cual es un arreglo de M características por cada punto en C. Este arreglo se denomina μ para el proceso de alineamiento con el material ϕ.

$$\mu = \Gamma(C, I_{rgb}) \tag{3.14}$$

Adicionalmente, con la imagen I_{depth} se realiza un proceso de segmentación con el fin de identificar las zonas de diferente dimensión como se muestra en la Figura 3.7. Si un punto de interés C cae sobre una zona identificada, el punto adquiere el valor b correspondiente.

Si el material ϕ se encuentra vacío, entonces μ se transforma en ϕ y se espera otra iteración. En caso contrario, se realiza la distribución aleatoria de la muestra μ en el material ϕ, como se muestra en la Figura 3.6. Para cada par (μ, ϕ) se calcula la distancia como se propone en la Ecuación 3.8, y luego se propone un posible cambio aleatorio dentro del espacio φ de cada partícula en C, la probabilidad de elección de los puntos depende de su

valor b, con el fin de descartar puntos sobre regiones de menor dimensión. Si la distancia a ese punto aleatoriamente elegido es menor, se acepta el cambio, de lo contrario se estima la probabilidad de cambio de acuerdo a la temperatura T_k y se acepta aleatoriamente. El proceso se repite de acuerdo a una variable T que garantiza que el tiempo de alineación se mantenga estable. Al finalizar el proceso se devuelve la lista de los puntos en ϕ que corresponden a los puntos en μ. Con más de 8 correspondencias se puede resolver la Ecuación 1.12 para obtener la matriz fundamental F_c, con la cual se calcula la matriz esencial E con la ayuda de K_c según lo describe la Ecuación 1.13.

Finalmente la estimación de la rotación y traslación se realiza por descomposición de SVD, como se describe en la sub-sección 1.3.2. La rotación y traslación componen la matriz de transformación T, la cual se aplica a la nube proyectada a través de la imagen de color I_{rgb}, la imagen de profundidad I_{depth}, y la matriz de calibración de cámara K_c. La transformación también se aplica a la lista de puntos C para trasladarlos espacialmente. Finalmente la nube proyectada se inserta al mundo W y los puntos de interés se insertan al vector m.

3.3 Implementación

La implementación de la técnica SLAM de odometría por características extendidas se realiza a través de las librerías Boost, Eigen, FLANN, Octomap, Octovis, OpenCV, OpenNI2, y PCL sobre la plataforma ROS Indigo. En la Figura 3.10 se muestran las dependencias usadas para desarrollar y probar la técnica de odometría por características extendidas, el algoritmo de la técnica propuesta hace uso de las librerías OpenCV, PCL y Octomap. OpenCV [11] es una librería de código abierto para visión por computador y el aprendizaje de máquina, esta librería se usa para el cálculo de las características SURF y para el manejo de las imágenes RGBD. PCL [60] es una librería de código abierto para el procesamiento de información 2D y 3D, esta librería se usa para el manejo de nubes de puntos y la construcción de mapas tridimensionales. Octomap [12] es una librería de código abierto para el procesamiento de nubes de puntos, esta librería se usa para el almacenamiento de los mapas generados por la técnica propuesta.

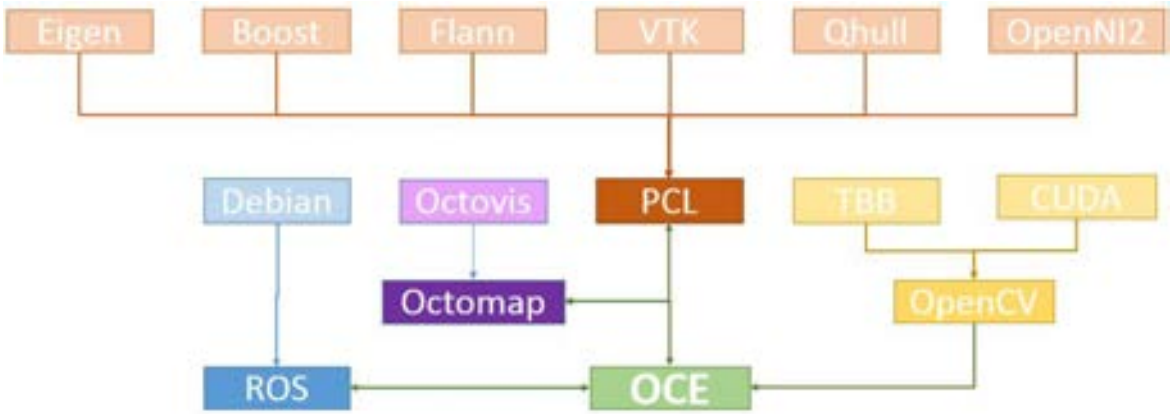

Figura 3.10: Dependencias necesarias para la técnica de Odometría por características extendidas

El sistema operativo robótico ROS es usado en esta propuesta con el fin de conectar los módulos de adquisición de información con los módulos de procesamiento y visualización; en la adquisición de las imágenes RGBD se desarrolló un nodo ROS denominado rgbd2bag, cuyo propósito es leer las imágenes RGBD del disco duro y publicarlas con cierta frecuencia dentro del sistema. Los módulos de procesamiento contienen los algoritmos SLAM de odometría por características extendidas y dense visual odometry, ambos reciben las imágenes RBGD publicadas y publican la transformación entre la publicación actual y la anterior. La visualización de los resultados se hace a través de Rviz y de Octovis, ambas son herramientas de uso libre que permiten observar las nubes de puntos. El uso de ROS en esta propuesta se debe a la necesidad de separar las etapas de adquisición, procesamiento y visualización, con el fin de poder evaluar el comportamiento de la técnica propuesta, y a su vez de aprovechar mejor los recursos del sistema.

3.4 Conclusiones

En este capítulo se presentó una propuesta de una técnica SLAM que se denomina odometría por características extendidas, la cual está basada en la técnica de odometría por alineación de características (Keypoint-Alignement). Se presentaron 3 modificaciones a la técnica base, la primera de ellas consistió en modificar la lista de puntos característicos, extrayendo puntos de interés adicionales de la imagen de profundidad, bajo el argumento de que los cambios pronunciados en las geometrías del espacio son indicadores de zonas de interés, los puntos estimados en la imagen de profundidad se añaden a la lista de puntos de interés de la imagen de color, y son usados para extraer las características SURF de la imagen. La segunda modificación consistió en cambiar el alineador de características, adaptando el algoritmo heurístico de recocido simulado al algoritmo de vecinos más cercanos, con el fin de reducir el tiempo de alineación y mejorar el desempeño de la técnica

SLAM propuesta. La última modificación propuesta consistió en crear un mapa jerárquico probabilístico a partir de la alineación de las nubes de puntos obtenidas por el kinect y transformadas por la técnica SLAM, este mapa permitirá la interacción con otras librerías al mismo tiempo que una representación de los objetos que podrían modificar su posición en el tiempo. Cada una de las propuestas realizadas se orienta al cumplimiento de los objetivos específicos presentados en esta propuesta, se concluye entonces que la técnica propuesta presenta cambios respecto de la técnica original con el fin de resolver el problema de desempeño y de consistencia.

Capítulo 4

Evaluación

En el Capítulo 3 se presentó la técnica SLAM de odometría por características extendidas, esta técnica está basada en diferentes modificaciones a la técnica *Keypoint Alignement*, dado que las modificaciones realizadas implican algoritmos de diferente naturaleza, la evaluación de la técnica propuesta se realiza midiendo el desempeño de cada una de las modificaciones y el desempeño de la técnica completa respecto a una técnica SLAM actual: *Dense Visual Odometry* [13, 14, 27].

Los métodos de evaluación varían de acuerdo a cada modificación propuesta, debido a que cada una de ellas corresponde a un espacio investigativo diferente; de esta forma la evaluación de las características SURF extendidas corresponde a la comparación de la capacidad discriminativa de las mismas respecto a las características SURF originales sobre conjuntos cruzados de objetos y escenas. La evaluación de características también se realiza a través de la comparación de meta-características, las cuales fueron extraídas sobre los mismos conjuntos de datos y permiten identificar el comportamiento de cada extractor. El alineador de características heurístico se evalúa con respecto al clasificador KNN, dentro de los parámetros a evaluar está el porcentaje de aciertos, el tiempo de entrenamiento y el tiempo de clasificación, cada uno de estos parámetros es medido para cada clasificador sobre los 17 conjuntos de datos seleccionados de la base de datos UCI machine learning [7], a cada conjunto de datos se le calcula sus meta-características, las cuales son las propiedades que definen la capacidad discriminativa, y se hallan con el fin de determinar en qué casos específicos es mejor usar el clasificador KNN o el clasificador heurístico propuesto.

La evaluación de la técnica SLAM propuesta se realiza a través de la medición del error relativo entre poses (RPE) y el error absoluto de trayectoria (ATE), estos errores se calculan para las trayectorias determinadas por la técnica SLAM propuesta y la técnica SLAM actual. El error se calcula para cada secuencia de la base de datos, adicionalmente se mide la tasa de error al combinar secuencias del mismo escenario, las cuales difieren por diversos objetos presentes en toda la secuencia. Este capítulo continúa con la definición de los métodos de evaluación para las características extendidas, el clasificador heurístico y el mapa probabilístico, analizando los resultados de cada prueba en cada sección, finaliza con una

evaluación de toda la técnica SLAM propuesta usando las definiciones de error RPE y ATE, y mostrando los escenarios reconstruidos por la técnica SLAM propuesta.

4.1 Características SURF extendidas

Las características se evalúan en 2 formas: en comparación con otras características cuando se usa un clasificador predeterminado, o evaluando las metacaracterísticas de las instancias a clasificar. Los primeros 2 métodos de prueba tienen como objetivo determinar de forma práctica el cambio en la capacidad discriminativa de los descriptores hallados por cada uno de los métodos de extracción de características. Dado que el objetivo de esta propuesta se basa en ambientes dinámicos, es necesario entender el comportamiento de las nuevas características sobre escenarios, y sobre los objetos que lo vuelven dinámico. El clasificador seleccionado, Vecinos más cercanos, permite evaluar este cambio al realizar variaciones de K, de esta forma una fuerte separación entre los valores de los descriptores generaría un porcentaje constante de aciertos al aumentar K en un espacio N-dimensional. El tercer método tiene como objetivo evaluar las características SURF extendidas a través de diferentes índices, tales como el porcentaje de acierto, la tasa discriminativa de Fisher, el número de instancias que recaen sobre los límites de la clase, el porcentaje de intersección de clases, etc. Estos índices son calculados sobre las características extraídas a conjuntos de imágenes previamente establecidos, para esta propuesta se usó la base de datos de la universidad de washington [40], la cual consta de 299 objetos comunes y 14 escenarios comunes; por cada uno de los objetos existen en promedio 700 imágenes RGBD, mientras que por cada uno de los escenarios existen en promedio 900 imágenes. Los índices se calculan para objetos y para escenarios por separado, aunque todas las imágenes corresponden a objetos comunes, la geometría y textura de cada uno es diferente. Las características SURF extendidas definidas en el Capítulo 3 se compararán respecto a las características SURF bajo los siguientes tres métodos:

1. Variación en el porcentaje de clasificaciones correctas con el clasificador KNN sobre diferentes objetos comunes elegidos al azar.

2. Variación en el porcentaje de clasificaciones correctas con el clasificador KNN sobre diferentes escenarios comunes elegidos al azar.

3. Evaluación de las meta-características obtenidas con cada extractor sobre objetos y escenarios elegidos al azar.

La primera prueba evalúa el porcentaje de acierto del clasificador KNN cuando se usan características SURF extendidas con objetos comunes, el parámetro de referencia será entonces el porcentaje de acierto del clasificador KNN cuando se usan características SURF.

La prueba consiste en seleccionar un número n_o de imágenes de objetos en la base de datos, estas imágenes contienen solamente al objeto sobre un fondo neutro en baja resolución, una muestra de las imágenes disponibles se muestra en la Figura 4.1.

Figura 4.1: Muestra de 13 de 299 objetos en la base de datos [40].

En esta prueba se seleccionaron 50 imágenes al azar por objeto, extrayendo de cada una las características SURF extendidas y SURF, con las cuales se entrena el clasificador KNN, luego se seleccionan al azar el 20% de las imágenes de los objetos entrenados, con las cuales se realiza la clasificación, comparando el resultado con la clase original de cada imagen se halla el porcentaje de acierto. El procedimiento de entrenamiento y validación se realizó con valores de K desde 2 hasta 14, estos valores se eligen tomando en cuenta que el valor óptimo de K se encuentra entre el rango $K = \{5, 11\}$ [8], se asume un margen de ± 3, con el fin de analizar el comportamiento de cada uno de los descriptores, con lo cual el rango de evaluación es $K = \{2, 14\}$. La prueba se realiza con 2, 5 y 10 objetos dado que el objetivo es determinar el cambio de la capacidad discriminativa de las características en cada extractor propuesto, la variación del número de objetos permite identificar la separabilidad de las clases cuando existen mayor cantidad de características y de clases.

Si las características no son suficientemente discriminativas, el K-vecino más cercano tiene más probabilidad de pertenecer a otra clase, por tanto se busca que al aumentar el valor de K el porcentaje de clasificaciones correctas sea mayor que la referencia y se mantenga constante; con esto se compara la capacidad discriminativa de los extractores a través de la separabilidad de clases. El resultado de la primera prueba se puede observar en la Figura 4.2, en la cual se observa las líneas de tendencia del acierto de clase para cada clasificador al variar el número de K, el eje vertical secundario indica el número de objetos utilizados para la respectiva tendencia, finalmente el color azul identifica el resultado de las características SURF y el resultado naranja identifica el resultado de las características SURF extendidas.

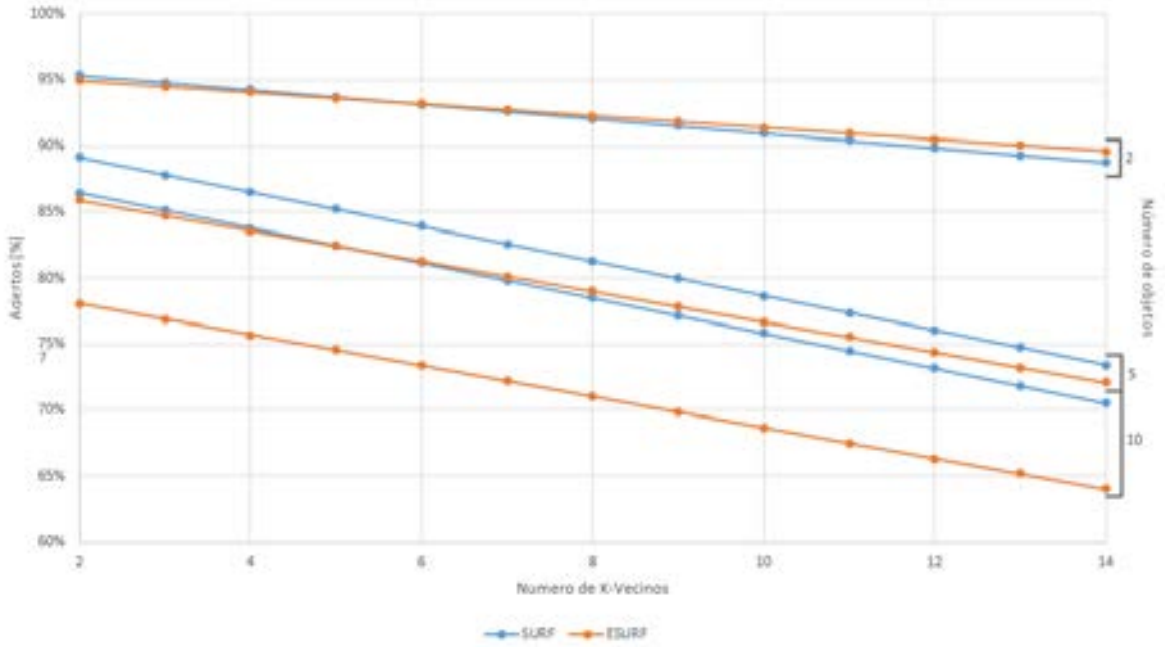

Figura 4.2: Comparación del promedio de acierto al variar el número de K-Vecinos más cercanos con características SURF y SURF extendidas.

Como se observa en la Figura 4.2, las características SURF extendidas no mejoran el porcentaje de acierto de objetos. En el caso de 2 objetos el porcentaje de acierto para ambos tipos de características es prácticamente similar, no obstante para los casos de 5 y 10 objetos el porcentaje de acierto para SURF es considerablemente mayor que las características propuestas en 3% y 9%, respectivamente. Sin embargo, se señala que la tendencia lineal indica que las características SURF extendidas poseen una pendiente menor que SURF, lo cual sugiere una menor superposición entre clases. La Figura 4.2 demuestra que el extractor de características SURF tiene mayor capacidad discriminativa sobre grupos de objetos. Esto se explica por la geometría de los objetos, los cuales no presentan cambios abruptos en su superficie ni poseen grandes dimensiones, por consiguiente los puntos de interés calculados con la imagen de profundidad no contienen información útil que pueda

mejorar la capacidad discriminativa de las características SURF extendidas. En la evaluación de la tasa de acierto con grupo de 2 objetos las características propuestas superan a las características SURF en 1%, para valores de K mayores a 6; sin embargo esto no representa mejora alguna dado que no se puede generalizar para ambientes comunes donde la presencia de estos objetos es mayor a 2.

La segunda prueba evalúa el porcentaje de acierto del clasificador KNN cuando se usan características SURF extendidas con escenarios comunes, el parámetro de referencia es entonces el porcentaje de acierto del clasificador KNN cuando se usan características SURF, la prueba consiste en seleccionar un número n_s de imágenes de escenarios de la base de datos, estas imágenes muestran escenarios en alta resolución, una muestra de las imágenes disponibles se muestra en la Figura 4.3.

Figura 4.3: Muestra de 4 de 14 escenarios en la base de datos [40].

En esta prueba se seleccionaron 10 imágenes al azar por escenario, extrayendo de cada una las características SURF extendidas o SURF, con las cuales se entrena el clasificador KNN, luego se seleccionan 20 imágenes al azar entre todas las imágenes de los escenarios entrenados, con las cuales se realiza la clasificación, los parámetros de esta prueba son los mismos de la primera prueba para objetos, en cuanto a elección del rango de K, o la elección de un número variable de número de escenarios a entrenar y validar. El resultado

de la segunda prueba se puede observar en la Figura 4.4, en la cual se observa las líneas de tendencia del acierto de clase para cada clasificador, al variar el número de K, el eje vertical secundario indica el número de escenarios utilizados para la respectiva tendencia. Finalmente, el color azul identifica el resultado de las características SURF y el resultado naranja identifica el resultado de las características SURF extendidas.

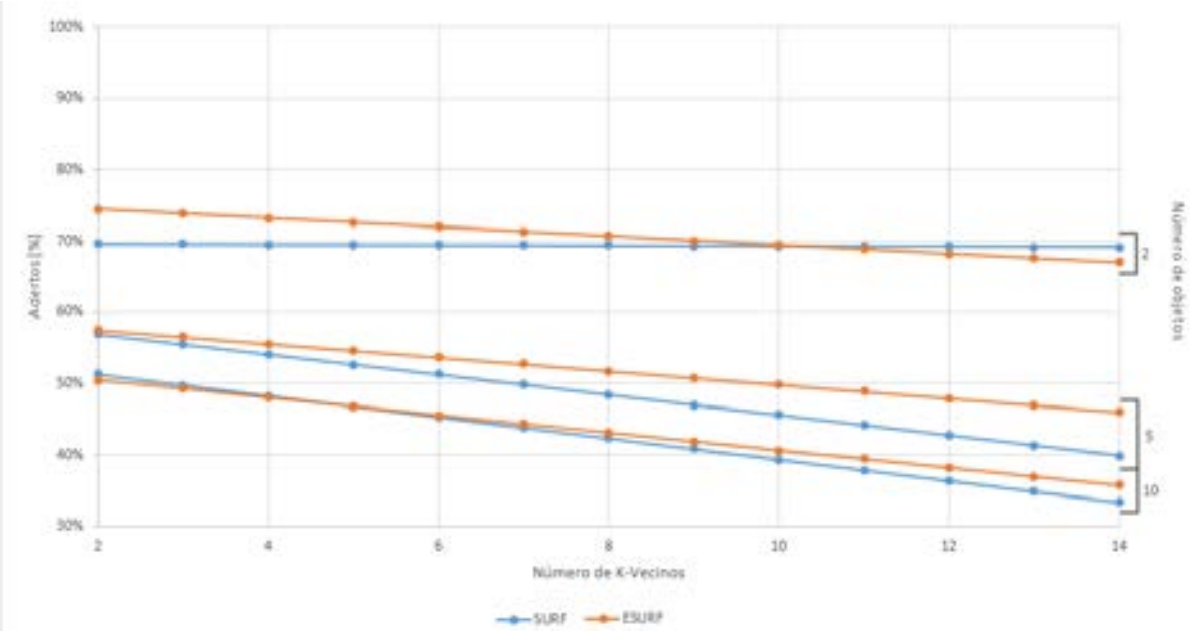

Figura 4.4: Comparación del promedio de acierto al variar el número de K-Vecinos más cercanos con características SURF y SURF extendidas.

Como se observa en la Figura 4.4, las características propuestas mejoran el porcentaje de acierto, en el caso de 2 escenarios ambas tendencias del porcentaje de acierto presentan un comportamiento similar, con una ventaja máxima de 3% para las características SURF extendidas en valores de K menores a 10, y una ventaja máxima del 1.5% para SURF en valores de K mayores a 10. Sin embargo, en los casos de 5 y 10 escenarios el porcentaje de acierto de las características SURF extendidas tiene una ventaja máxima de 4% y 1% respectivamente, esta ventaja es constante sobre gran parte del rango K. Las tendencias para características SURF extendidas para los casos de 5 y 10 escenarios presentan una menor pendiente que para las características SURF, lo cual sugiere una menor superposición entre clases. La Figura 4.4 indica que el extractor SURF extendido tiene mayor capacidad discriminativa sobre grupos de escenarios para valores de K mayores a 5. Esto se explica por las imágenes de profundidad detalladas que generalmente tienen los escenarios, este detalle se debe a la dimensión de los espacios comunes y la diversa cantidad de objetos que lo componen. La mayor cantidad de detalles aumenta la cantidad de puntos de interés en la imagen de profundidad, los cuales aportan información útil que mejora la capacidad discriminativa de las características. El valor de K necesario para mejorar la tasa de clasificación sugiere que las características SURF extendidas están más dispersas que las

características SURF.

Tanto en la Figura 4.2 como en la Figura 4.4 se observa que las tendencias del porcentaje de acierto son inversamente proporcionales al número de vecinos, esto señala que las características de las clases se intersectan en el espacio de tal forma que al aumentar el número de vecinos se tiene mayor probabilidad de encontrar características de otras clases. Las características SURF extendidas demuestran menor pendiente tanto en objetos como en escenarios, lo que significa que reduce las regiones de intersección de características.

La tercera prueba consiste en extraer meta-características teóricas a las características SURF extendidas de 10 y 10 escenarios, los parámetros de referencia son las meta-características teóricas extraídas a las características SURF de 10 objetos y 10 escenarios. Las meta-características calculadas son la máxima tasa discriminativa de Fisher (F1), la superposición de los cuadros delimitantes por clase (F2), la máxima eficiencia discriminativa de un atributo (F3), la eficiencia discriminativa colectiva (F4), la fracción de los puntos en los límites de clase (N1), el radio de la distancia promedio entre vecinos (N2), la tasa de error del vecino más cercano excluido (N3). Cada uno de estos parámetros es calculado con la librería Dcol [3] sobre un archivo KEEL con las características de los 10 objetos o escenarios. El resultado de la tercera prueba para objetos se puede observar en el Tabla 4.1, y el resultado de la tercera prueba para escenarios se puede observar en el Tabla 4.2.

Tabla 4.1: Comparación de las meta-características teóricas de las características SURF extendidas y SURF, halladas sobre una muestra de 10 objetos aleatorios.

Meta-Característica	SURF extendido	SURF
F1	0.362189	0.263666
F2	0.0000000267171	0.000000132727
F3	0.337488	0.308348
F4	4.81579	3.82794
N1	0.576177	0.540034
N2	0.846984	0.860878
N3	0.369344	0.332198

Tabla 4.2: Comparación de las meta-características teóricas de las características SURF extendidas y SURF, halladas sobre una muestra de 10 escenarios aleatorios.

Meta-Característica	SURF extendido	SURF
F1	0.0122883	0.0149936
F2	0.000254966	0.00426996
F3	0.00786443	0.0112998
F4	0.110652	0.146291
N1	0.802732	0.782918
N2	1.00833	0.998792
N3	0.53214	0.565718

Comparando las meta-características de la Tabla 4.1 contra las meta-características de la Tabla 4.2, se observa que la tasa discriminativa de Fisher para objetos es mayor en las características SURF extendidas, mientras que para escenarios es mayor en SURF, sin embargo la diferencia en el valor de la tasa discriminativa en escenarios es menor que la diferencia en el valor de la tasa discriminativa en objetos. La superposición de clases es menor en las características propuestas tanto para objetos como escenarios, lo cual permite señalar que las características SURF extendidas son más discriminativas que SURF. La reducción en la superposición de clases coincide con la Figura 4.2 y con la Figura 4.4, esta reducción en la superposición implica que al aumentar el valor de K existe mayor probabilidad de encontrar una característica de la misma clase, lo cual aumenta el porcentaje de aciertos y reduce la pendiente de la línea de tendencia. La mayor eficiencia por característica individual en objetos se encuentra en las características extendidas SURF. Mientras que en escenarios se encuentra en las características SURF, de la misma forma la mayor eficiencia por característica colectiva para objetos la tienen las características propuestas y para escenarios la tienen las características SURF. La fracción de puntos sobre los límites de clase es mayor en las características SURF extendidas, adicionalmente la tasa de puntos fuera de los límites de clase para objetos es menor en las características propuestas, mientras que para escenarios tienen menor tasa las características SURF. Las evaluaciones de las características SURF extendidas señalan que este tipo de características tienen mayor capacidad discriminativa en escenarios que en objetos, tanto por la menor superposición de clases como por la menor cantidad de puntos fuera de los límites de la clase y la mayor concentración sobre los límites. Estas características propuestas pueden reducir el error de pose en la técnica SLAM propuesta, dado que al tener mayor capacidad discriminativa se reduce la posibilidad de alinear características con correspondencia errónea.

4.2 Alineador de características heurístico

La evaluación de los métodos de clasificación es un tema activo de investigación [58, 38, 29], debido a que el desempeño de cualquier clasificador depende de la complejidad interna del conjunto de datos que se desean clasificar, normalmente el parámetro de evaluación usado es el porcentaje de clasificaciones correctas o porcentaje de acierto. Otros parámetros de comparación son el tiempo de entrenamiento y clasificación usados en comparaciones como el proyecto StatLog [19] o el análisis sobre la base de datos UCI machine learning [7] realizado por [41]. Considerando las referencias en métodos de evaluación para clasificadores, en esta propuesta se midió el tiempo de entrenamiento, el tiempo de clasificación, el porcentaje de clasificaciones correctas y la desviación en una validación cruzada de 10 conjuntos, aplicado sobre 17 conjuntos de datos seleccionados de la base de datos UCI machine learning [7]. En el Tabla 4.3 se muestran los 17 conjuntos seleccionados y sus propiedades calculadas en [30].

Tabla 4.3: Propiedades y meta-características del conjunto de datos usados para la prueba comparativa de clasificadores [30]

	Instancias	Atributos	Clases	Entropía datos	Entropía de clases	F1	F2	N1	N2	N3
Iris	150	4	3	4,483	1,585	16,041	0,0054	0.1	0.212	0.047
Wine	178	13	3	6.249	1.567	2.673	6.13e-5	0.118	0.575	0.051
Sonar	208	60	2	7.469	0.997	0.466	1.05e-6	0.288	0.741	0.125
Glass	214	9	5	5.749	2.176	1.576	6.013	0.486	0.682	0.299
Heart-Statlog	270	13	2	2.793	0.991	0.76	0.196	0.367	0.672	0.244
Haberman	306	3	2	4.019	0.834	0.185	0.718	0.539	0.754	0.353
Balance	625	4	3	2.322	1.318	0.204	3	0.317	0.677	0.208
Breast-w	699	9	2	2.255	0.929	3.477	0.248	0.067	0.34	0.046
Diabetes	768	8	2	5.718	0.933	0.576	0.252	0.438	0.84	0.294
Vehicle	846	18	4	5.242	1.999	0.35	0.169	0.452	0.801	0.303
Airfoil	1503	5	10	3.956	2.886	0.259	20.474	0.769	1.592	0.49
Segment	2310	19	7	7.773	2.807	15.614	1.65e-4	0.077	0.181	0.026
Wine Quality	4898	11	10	5.139	1.862	0.281	0.458	0.563	0.67	0.329
Waveform	5000	40	3	9.110	3.083	0.614	0.007	0.32	0.889	0.226
ccpp	9568	4	10	10.777	1.585	7.904	6.791	0.541	0.776	0.346
Shuttle	14500	9	7	4.192	0.945	2.596	1.88e-5	0.005	0.018	0.001
Letter	20000	16	26	3.092	4.700	1.554	2.162	0.094	0.451	0.037

La medición se realiza con el clasificador propuesto SANN y los parámetro de referencia son las mediciones realizadas con el clasificador KNN con valor de $K = 1$. En el Tabla 4.4 se observa la evaluación en porcentaje de aciertos y la desviación estándar correspondiente, los parámetros de tiempo de clasificación y entrenamiento se observan en la Tabla 4.5.

Tabla 4.4: Comparación de porcentajes de clasificación correcta entre el clasificador heurístico y vecino más cercanos.

	KNN		SANN	
	μ	σ	μ	σ
Iris	95.933	0.21	86.993	3.07
Wine	76.243	1.07	74.670	4.110
Sonar	82.271	0.96	74.351	3.05
Glass	73.246	0.70	63.976	2.74
Heart-Statlog	58.481	1.55	61.111	2.88
Haberman	67.484	1.30	67.668	2.04
Balance	79.059	0.43	64.950	1.31
Breast-w	95.264	0.29	90.659	1.06
Diabetes	64.882	0.54	68.300	1.28
Vehicle	68.194	0.44	64.936	1.11
Airfoil	64.882	0.41	21.634	1.19
Segment	21.458	0.20	90.082	0.73
Wine Quality	21.458	0.35	49.747	0.81
Waveform	78.002	0.17	68.104	0.48
ccpp	63.428	0.16	58.138	0.45
Shuttle	99.815	0.02	98.714	0.11
Letter	95.995	0.07	83.502	0.21

Tabla 4.5: Comparación del tiempo de entrenamiento y el tiempo de clasificación entre el clasificador heurístico y vecinos más cercanos.

	KNN			SANN		
	T_e	T_c	$Total$	T_e	T_c	$Total$
Iris	0.0047	0.0010	0.0057	0.0001	0.0021	0.0022
Wine	0.0043	0.0011	0.0054	0.0001	0.0030	0.0032
Sonar	0.0044	0.0014	0.0058	0.0002	0.0025	0.0027
Glass	0.0049	0.0011	0.0060	0.0001	0.0023	0.0024
Heart-Statlog	0.0041	0.0011	0.0052	0.0001	0.0022	0.0023
Haberman	0.0043	0.0010	0.0053	0.0001	0.0020	0.0021
Balance	0.0054	0.0013	0.0067	0.0002	0.0026	0.0028
Breast-w	0.0054	0.0014	0.0068	0.0003	0.0027	0.0031
Diabetes	0.0049	0.0011	0.0060	0.0002	0.0026	0.0028
Vehicle	0.0040	0.0020	0.0060	0.0003	0.0030	0.0032
Airfoil	0.0070	0.0017	0.0087	0.0005	0.0044	0.0049
Segment	0.0048	0.0091	0.0139	0.0009	0.0097	0.0107
Wine Quality	0.0050	0.0290	0.0340	0.0015	0.0239	0.0254
Waveform	0.0052	0.0388	0.0441	0.0019	0.0407	0.0425
ccpp	0.0171	0.0041	0.0211	0.0022	0.0630	0.0652
Shuttle	0.0270	0.0240	0.0510	0.0058	0.2665	0.2723
Letter	0.0142	0.5354	0.5495	0.0151	0.5179	0.5330

Se observa que el porcentaje de acierto para ambos clasificadores está relacionado inversamente con las meta-características N1, N2 y N3, la Figuras 4.5, 4.6, 4.7, presentan respectivamente las gráficas comparativas de las meta-características N1, N2 y N3 contra el porcentaje de acierto, los puntos rojos reflejan los conjuntos donde el clasificador SANN tuvo mejor desempeño y los puntos azules reflejan los conjuntos donde el clasificador KNN tuvo mejor desempeño. Se observa que para valores altos de N1, N2 y N3 el algoritmo de clasificación heurístico presenta mejor desempeño que el clasificador KNN.

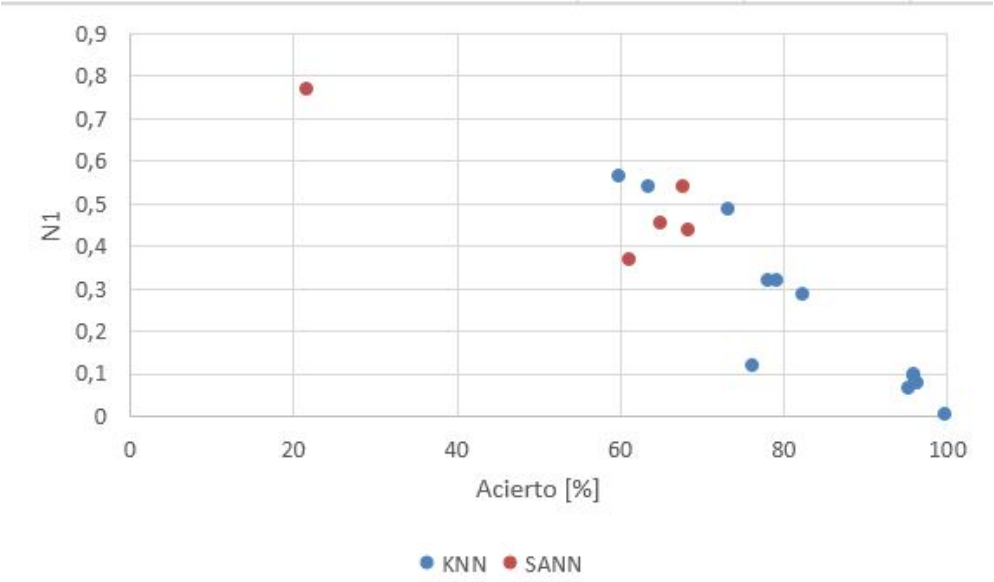

Figura 4.5: Valor de N1 contra el porcentaje de acierto en ambos clasificadores.

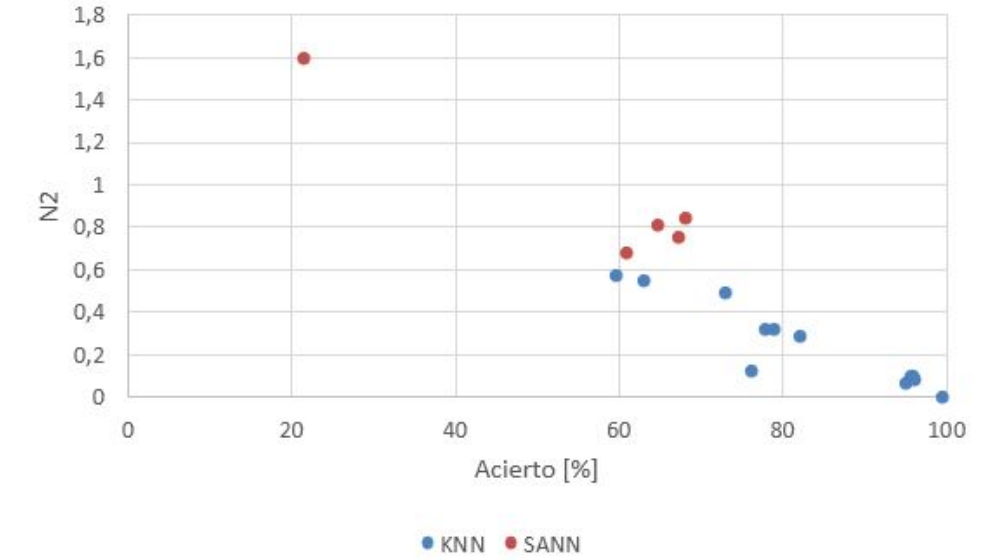

Figura 4.6: Valor de N2 contra el porcentaje de acierto en ambos clasificadores.

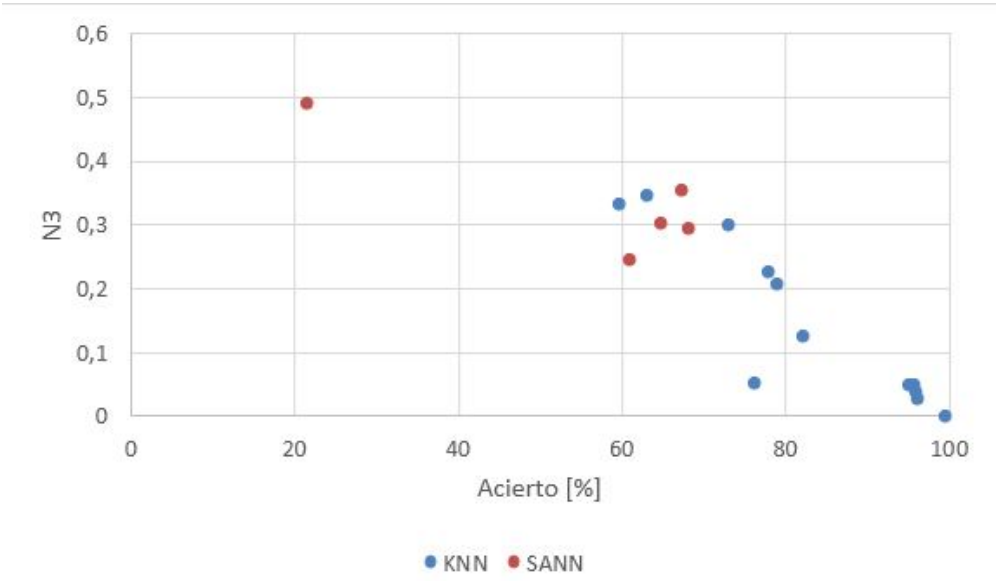

Figura 4.7: Valor de N2 contra el porcentaje de acierto en ambos clasificadores.

El tiempo de entrenamiento del alineador de características heurístico es menor que el tiempo de entrenamiento de KNN para conjuntos menores a 20.000 muestras, según la Tabla 4.5. Este comportamiento se debe a la forma de entrenamiento, el cual al organizar las muestras en una lista de acuerdo a sus complejidades, siendo el peor de los casos un procedimiento de complejidad $O\left(N^2\right)$, mientras que el algoritmo estándar se entrena con una complejidad de $O\left(N \log\left(N\right)\right)$. El tiempo de clasificación es mayor en el alineador de características heurístico que el tiempo de clasificación de KNN, la única excepción ocurre en la base de datos *'Wine Quality'*, sin embargo ninguna de las características calculadas en la Tabla 4.3 para cada base de datos tiene una correlación directa con este fenómeno. En el tiempo de entrenamiento, el alineador de características propuesto aumenta a una mayor tasa que KNN. Sin embargo, el tiempo de clasificación en el alineador de características tiende a un valor constante mientras que KNN aumenta. Ambas tendencias concurren en aproximadamente 20.000 muestras, de igual forma la Tabla 4.4 muestra que el porcentaje de clasificaciones correctas es mayor para bases de datos menores a las 2.500 muestras con el alineador de características heurístico propuesto. Analizando las meta-características de las bases de datos utilizadas se observa en las Figura 4.5, Figura 4.6 y Figura 4.7 una correlación inversa entre las características N1, N2, N3 respecto al porcentaje de acierto, por lo cual se puede determinar que el alineador de características propuesto tiene mejor desempeño cuando los valores de N1, N2, y N3 son superiores a 0.5, 0.6 y 0.25, respectivamente, y el número de muestras es menor a 2500. Finalmente se señala que la complejidad teórica de clasificación en el clasificador heurístico (SANN) es $O\left(1\right)$ en el rango de 0 a 2500 muestras, mientras que en el clasificador KNN es $O\left(\log\left(N\right)\right)$.

4.3 Mapa probabilístico

La implementación del mapa probabilístico se hace a través de la librería OctoMap [12], la cual permite realizar las operaciones de insertar, obtener y refinar a través de las funciones implementadas *insertRay/insertScan*, *computeRay* y *Prune*, adicionalmente la librería implementa la probabilidad de ocupación de cada celda, lo cual cumple con los requisitos planteados en el Capítulo 3. La estructura base de la librería es un árbol de 8 hijos, denominado OctTree, el cual se puede ajustar para obtener diferentes niveles de detalle, como se observa en la Figura 4.8.

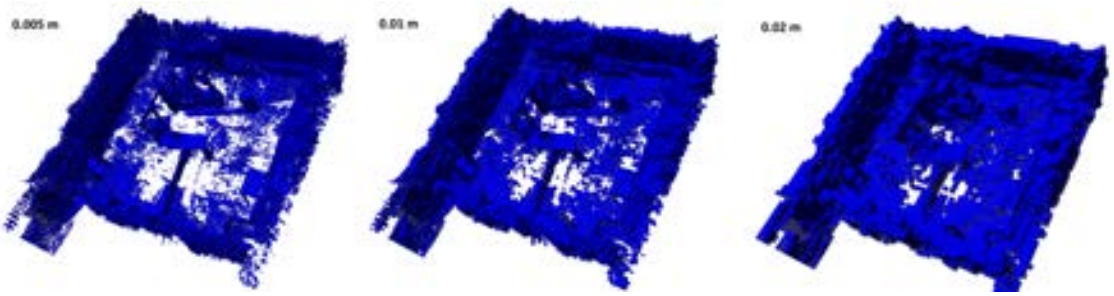

Figura 4.8: Mapa jerárquico variando el nivel de resolución de 0.005 m a 0.2 m por píxel.

Los mapas almacenados por OctoMap son sensibles a la configuración de detalle, debido a que cada nivel adicional de detalle eleva exponencialmente la cantidad de memoria requerida para almacenar el mapa. Con base a esto se determina que el nivel máximo de detalle será de $0.005m$ como se observa en la Figura 4.8.

El valor de probabilidad es asignado por cada píxel de acuerdo al área de la segmentación de la imagen RGBD, en la Figura 4.9 se muestra el ejemplo de diversos escenarios con su respectivo valor de probabilidad, las imágenes de la derecha corresponden a la probabilidad de cada píxel de permanecer en el tiempo de acuerdo a la Ecuación 3.10, la escala inicia en 0 (negro) hasta 255 (blanco). Se puede observar que donde se encuentran los objetos más pequeños, la probabilidad es baja, mientras que en superficies como las mesas o pisos la probabilidad tiene un valor intermedio, finalmente los fondos tienen un valor de probabilidad alto. Con esta probabilidad basada en la segmentación se garantiza que las características calculadas sobre superficies grandes tendrán mayor importancia que las características que se calculan sobre pequeñas superficies.

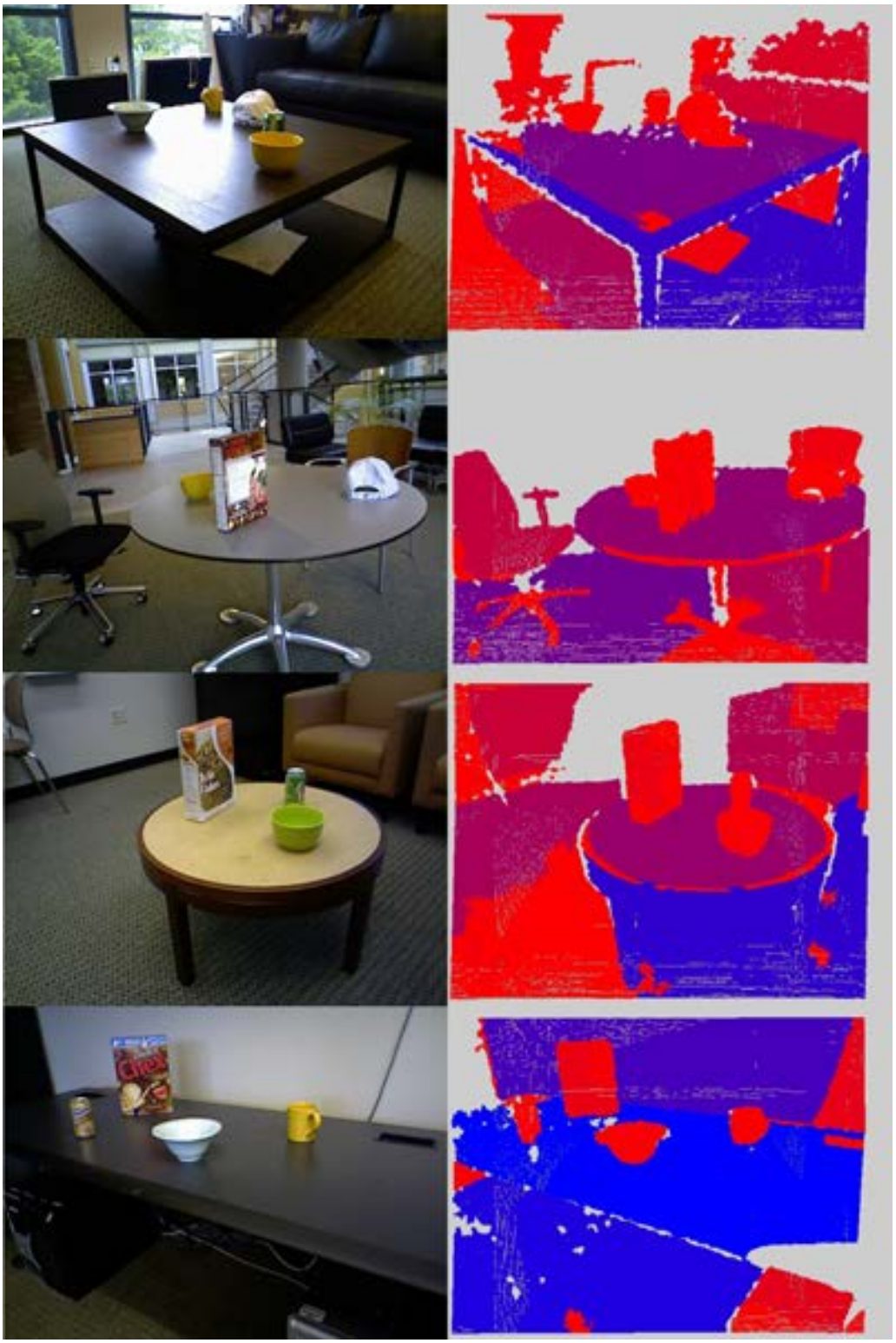

Figura 4.9: Valor de probabilidad en segmentos de diferentes escenarios de la base de datos [40].

4.4 Técnica de odometría por características extendidas

Las técnicas SLAM tienen como objetivo encontrar la trayectoria realizada por la cámara a partir de las mediciones disponibles, con base en esta trayectoria se obtiene la posición final y las transformaciones necesarias para construir el mapa. En este sentido la variable principal en las técnicas SLAM es el vector de estado espacial que cambia a través del tiempo, este vector se puede comparar contra la trayectoria real (*Ground Truth*), o contra otra técnica dada la dificultad de obtener la trayectoria real [39]. Existen 2 métodos para evaluar la calidad de la trayectoria obtenida por un método particular: el error relativo de pose (RPE) y el error absoluto de trayectoria (ATE) [39]. El RPE mide la precisión local de la trayectoria sobre un intervalo fijo de tiempo Δ, el error se define como el desplazamiento entre 2 intervalos de tiempo como se muestra en la Ecuación 4.1, el error de una trayectoria con m intervalos se muestra en la Ecuación 4.2, donde se halla la raíz cuadrada de la varianza del error, considerando el error solo en su componente traslacional.

$$E_i = \left(Q_i^{-1} Q_{i+\Delta}\right)^{-1} \left(S_i^{-1} S_{i+\Delta}\right) \tag{4.1}$$

$$RMSE\left(E_{1:m}, \Delta\right) = \left(\frac{1}{m} \sum_{i=1}^{m} \left| trans\left(E_i\right) \right|^2\right) \tag{4.2}$$

donde $S_i \in SE(3)$ hace referencia a la posición estimada en un instante del tiempo y $Q_i \in SE(3)$ hace referencia a la posición real en un instante del tiempo, E_i es el error relativo de pose entre la posición estimada y la posición real en un instante del tiempo.

El ATE evalúa la distancia absoluta entre las trayectorias, el método halla una transformación rígida T entre las trayectorias, lo que corresponde a la solución de mínimos cuadrados, y luego para cada instante halla el error usando la Ecuación 4.3, el error de una trayectoria con n instancias se muestra en la Ecuación 4.4, donde se halla la raíz cuadrada de la varianza del error, considerando el error solo en su componente traslacional.

$$E_i = Q_i^{-1} T S_i \tag{4.3}$$

$$RMSE\left(E_{1:n}, \Delta\right) = \left(\frac{1}{m} \sum_{i=1}^{n} \left| trans\left(E_i\right) \right|^2\right) \tag{4.4}$$

En resumen, la técnica SLAM de *odometría por características extendidas* recibe imágenes RGBD de escenarios comunes, a cada una de estas imágenes RGBD se les extrae las características SURF extendidas, estas características se ingresan en el alineador heurístico para determinar correspondencias, con las correspondencias calculadas se estima el cambio de pose en términos de una matriz de transformación T, finalmente se realiza la proyección de las imágenes RGBD a una nube de puntos, la cual se inserta en el mapa,

logrando una actualización del espacio. La evaluación de la técnica SLAM propuesta se realizará a través de la comparación del error entre la técnica propuesta respecto al valor verdadero y el resultado de otra técnica en diferentes escenarios de una base de datos con objetos dinámicos. Esta prueba tiene como objetivo medir el error relativo de pose (RPE) y el error absoluto de trayectoria (ATE), de la técnica SLAM propuesta y de una técnica SLAM actual denominada *Dense Visual Odometry* [13, 14, 27], sobre 4 escenarios de la base de datos de la universidad de washington [40]. Para cada uno de estos escenarios se tienen secuencias de imágenes RGBD con variaciones en los objetos que componen la escena, los escenarios y su contenido son:

- Escenario 1: Espacio común tipo sala, el cual se rodea de muebles, cajas y estantería, el centro de la escena se compone de una mesa y diferentes objetos ubicado en posiciones aleatorias, se tienen 4 secuencias con diferentes composiciones de objetos sobre la mesa, en las Figuras 4.10, 4.11, 4.12 y 4.13 se muestran las reconstrucciones tridimensionales del escenario 1 para cada secuencia, además se identifican los objetos y su distribución espacial.

- Escenario 2: Espacio común tipo cafetería, el cual se rodea de muebles y mesas, el centro de la escena se compone de una mesa y diferentes objetos ubicados en posiciones aleatorias, se tienen 4 secuencias con diferentes composiciones de objetos sobre la mesa, en las Figuras 4.14, 4.15, 4.16 y 4.17 se muestran las reconstrucciones tridimensionales del escenario 2 para cada secuencia, además se identifican los objetos y su distribución espacial.

- Escenario 3: Espacio común tipo salón, el cual se rodea de muebles y estantería, el centro de la escena se compone de una mesa y diferentes objetos ubicados en posiciones aleatorias, se tienen 4 secuencias con diferentes composiciones de objetos sobre la mesa, en las Figuras 4.18, 4.19, 4.20 y 4.21 se muestran las reconstrucciones tridimensionales del escenario 3 para cada secuencia, además se identifican los objetos y su distribución espacial.

- Escenario 4: Espacio común tipo estudio, el cual se rodea de pared y estantería, el centro de la escena se compone de una mesa y diferentes objetos ubicados en posiciones aleatorias, se tienen 2 secuencias con diferentes composiciones de objetos sobre la mesa, en las Figuras 4.22 y 4.23 se muestran las reconstrucciones tridimensionales del escenario 4 para cada secuencia, además se identifican los objetos y su distribución espacial.

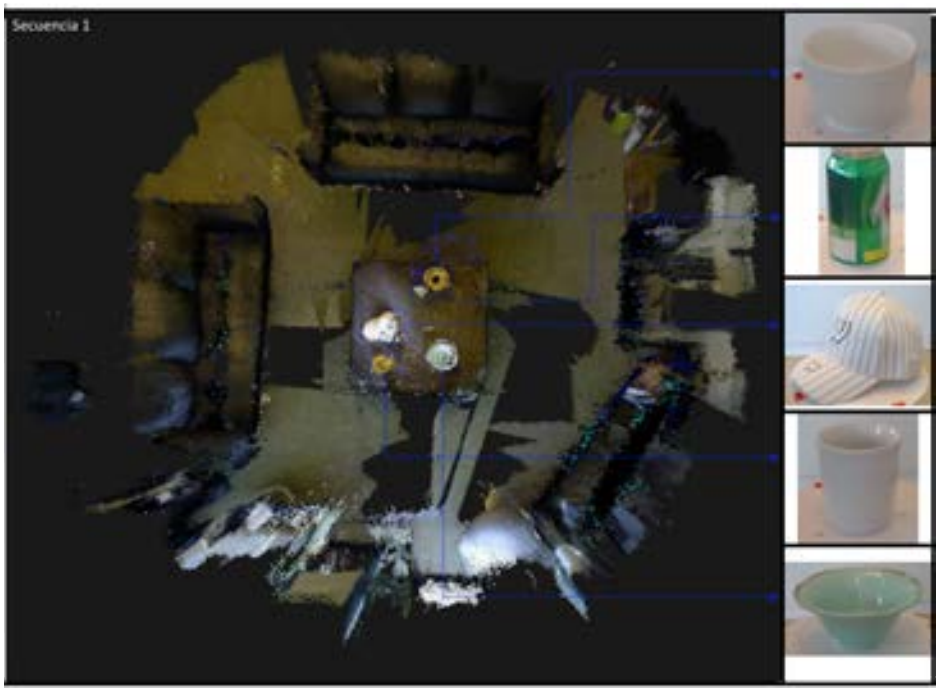

Figura 4.10: Reconstrucción de la secuencia 1 del escenario 1 a partir de la información de movimiento verdadero.

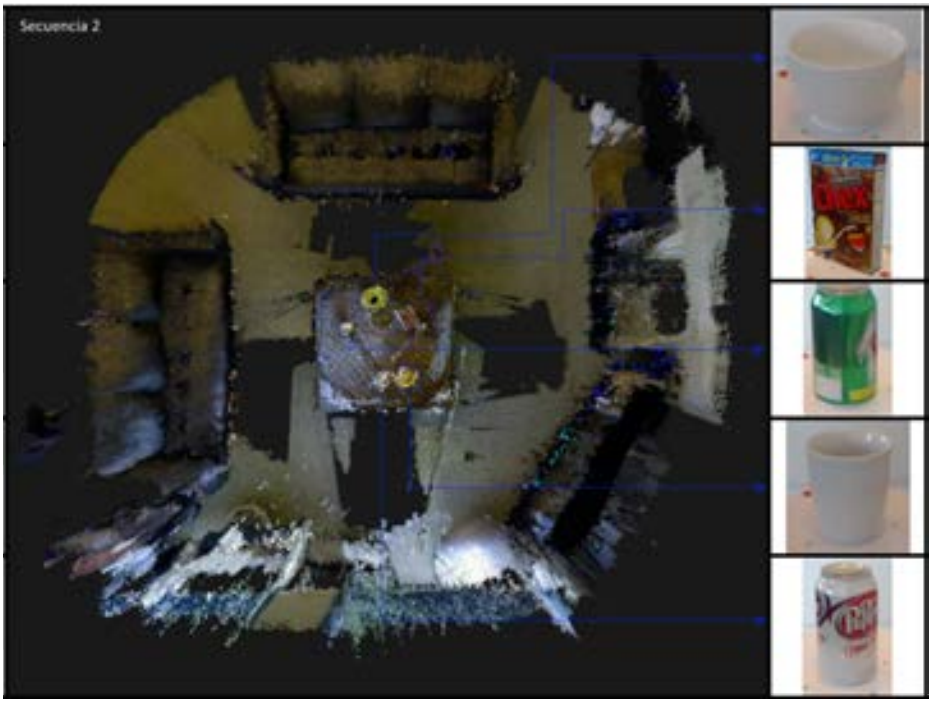

Figura 4.11: Reconstrucción de la secuencia 2 del escenario 1 a partir de la información de movimiento verdadero.

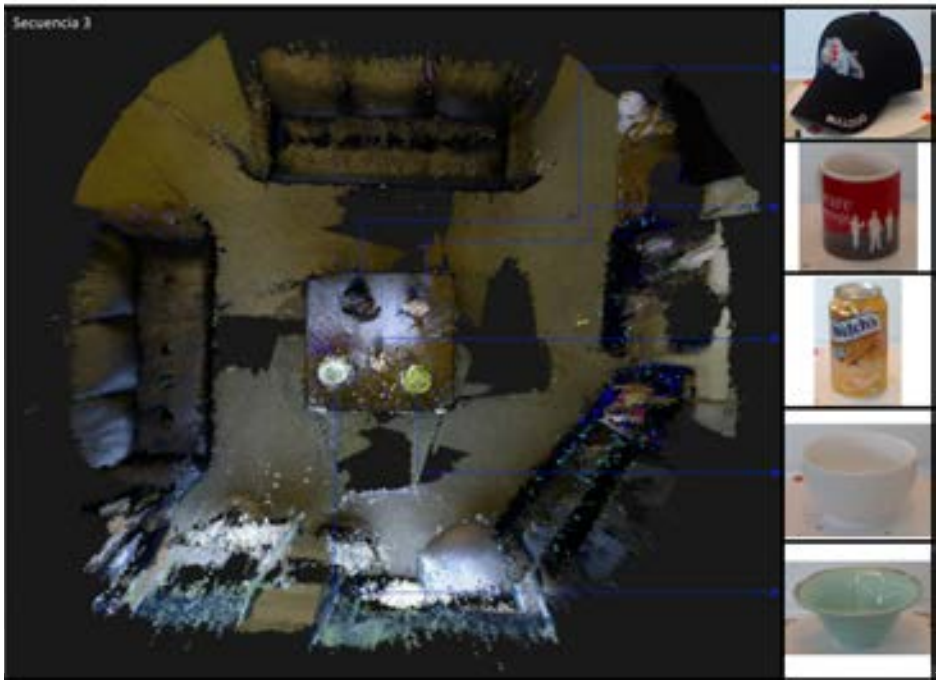

Figura 4.12: Reconstrucción de la secuencia 3 del escenario 1 a partir de la información de movimiento verdadero.

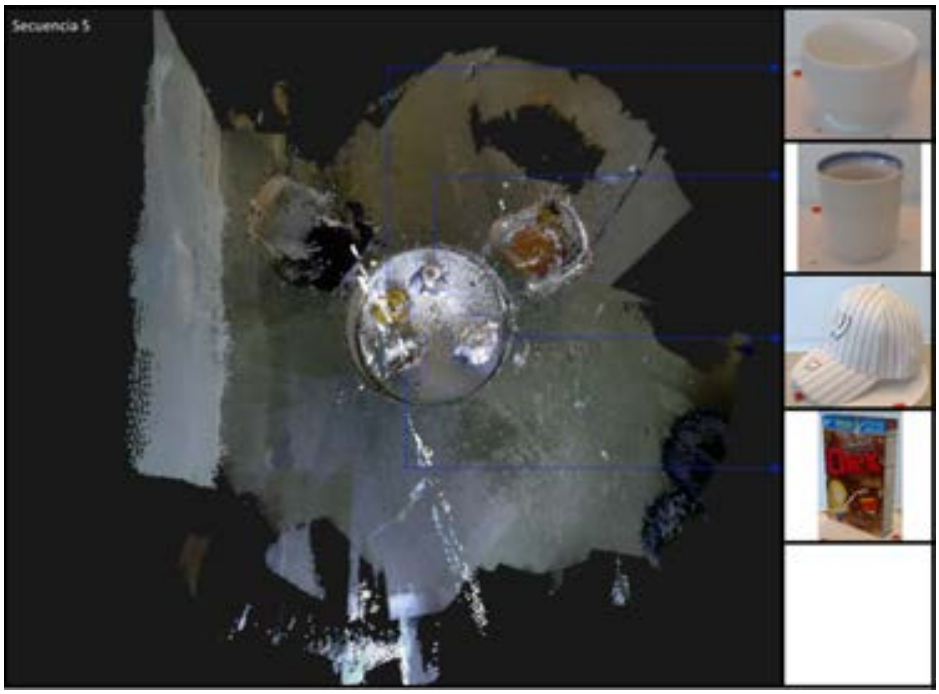

Figura 4.13: Reconstrucción de la secuencia 4 del escenario 1 a partir de la información de movimiento verdadero.

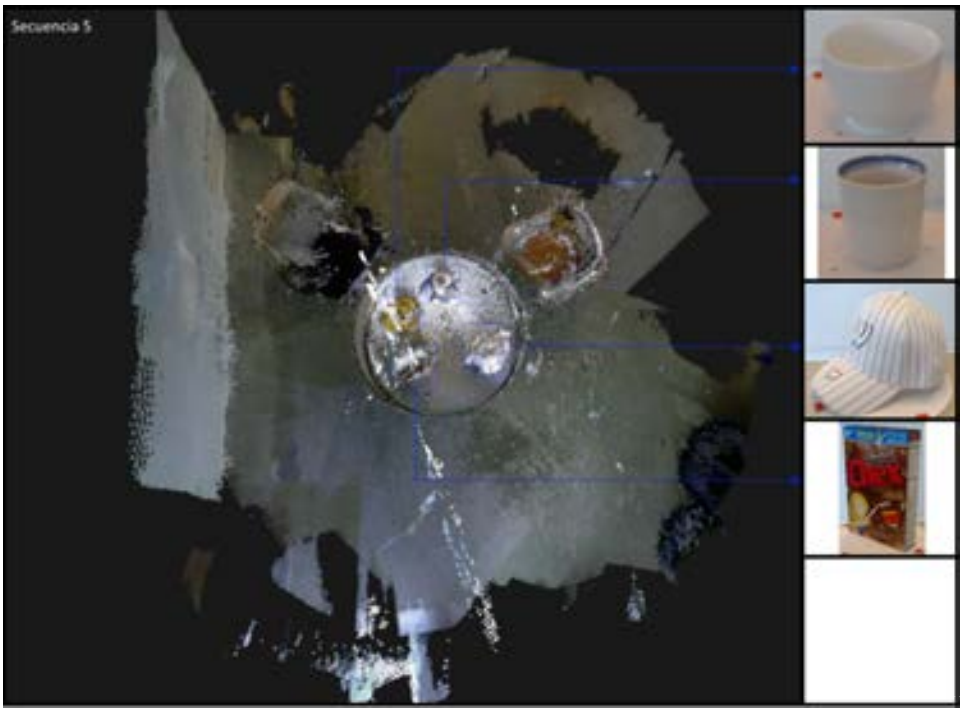

Figura 4.14: Reconstrucción de la secuencia 5 del escenario 2 a partir de la información de movimiento verdadero.

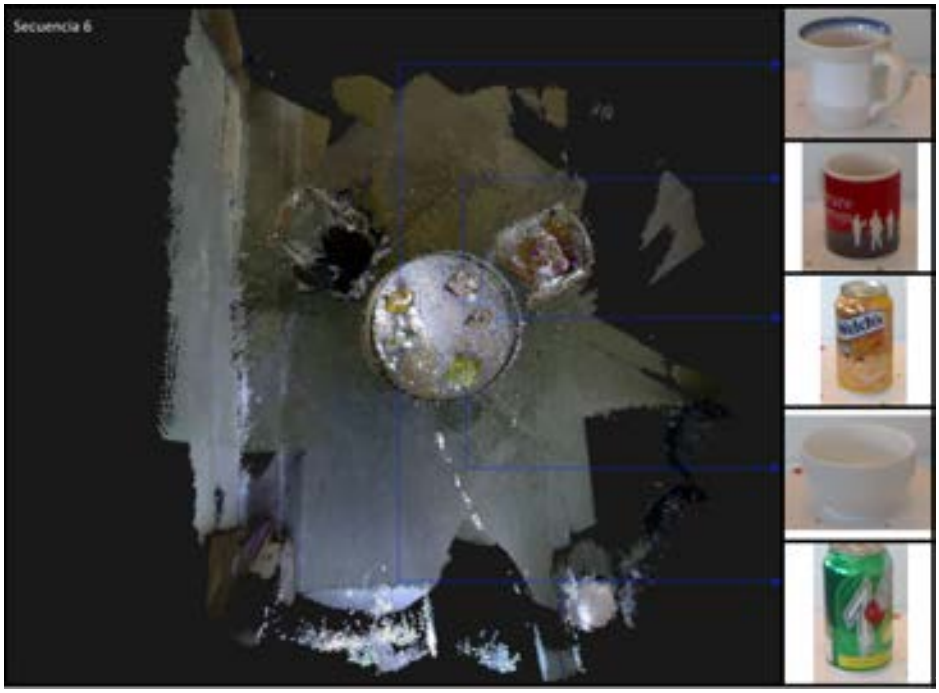

Figura 4.15: Reconstrucción de la secuencia 6 del escenario 2 a partir de la información de movimiento verdadero.

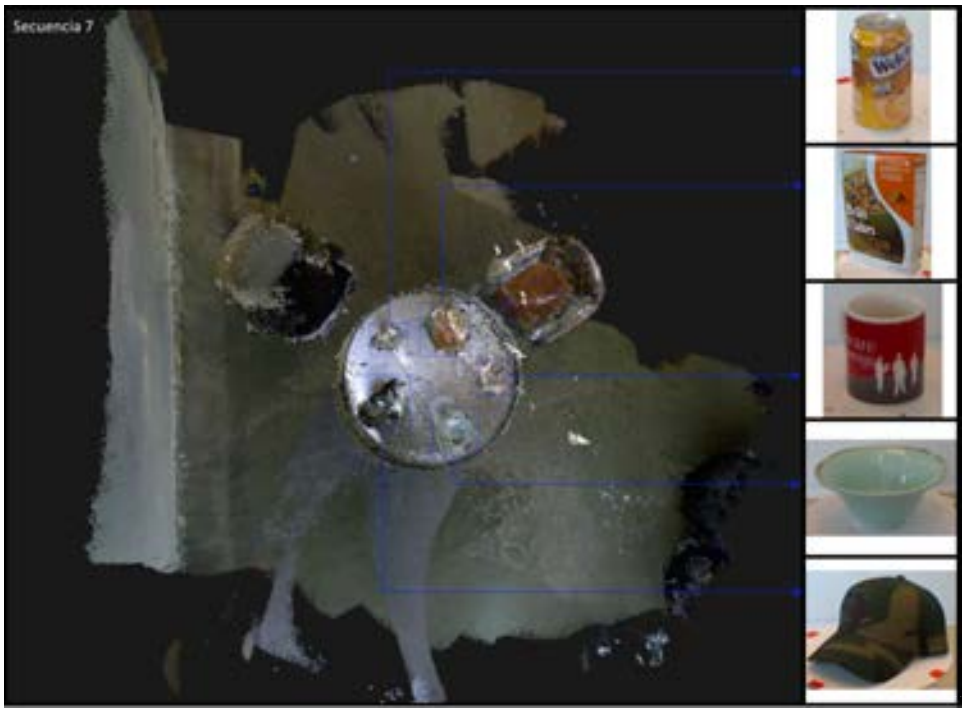

Figura 4.16: Reconstrucción de la secuencia 7 del escenario 2 a partir de la información de movimiento verdadero.

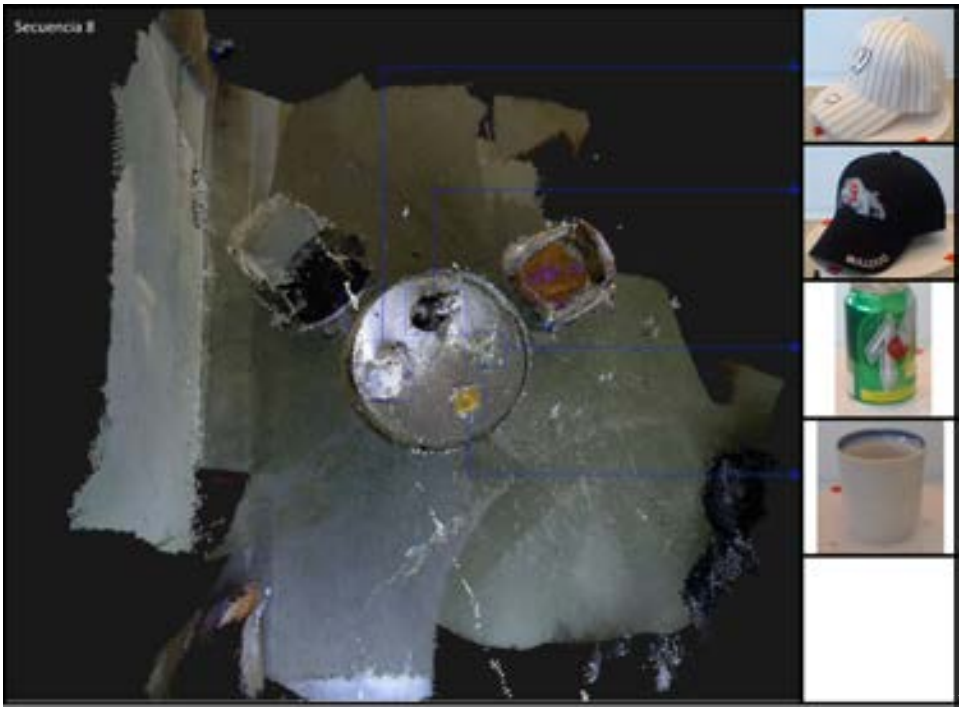

Figura 4.17: Reconstrucción de la secuencia 8 del escenario 0 a partir de la información de movimiento verdadero.

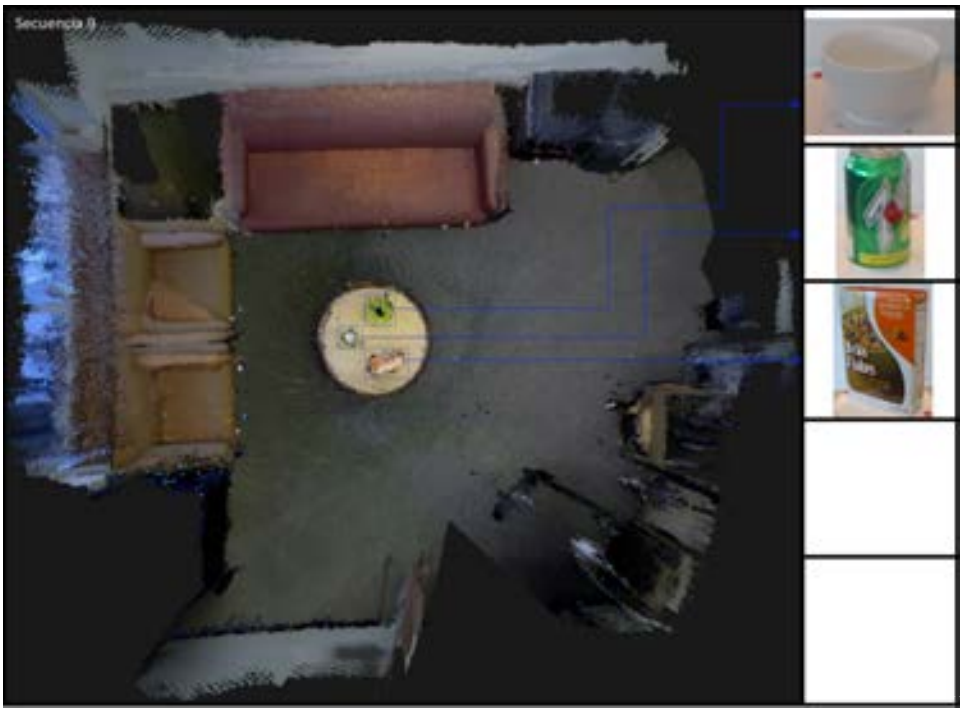

Figura 4.18: Reconstrucción de la secuencia 9 del escenario 3 a partir de la información de movimiento verdadero.

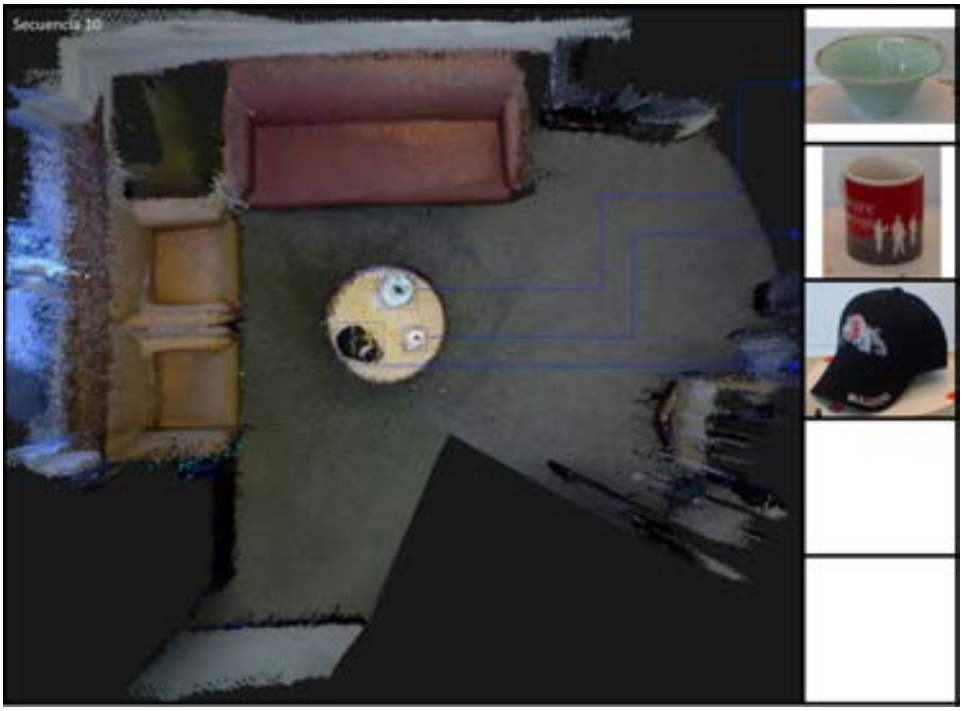

Figura 4.19: Reconstrucción de la secuencia 10 del escenario 3 a partir de la información de movimiento verdadero.

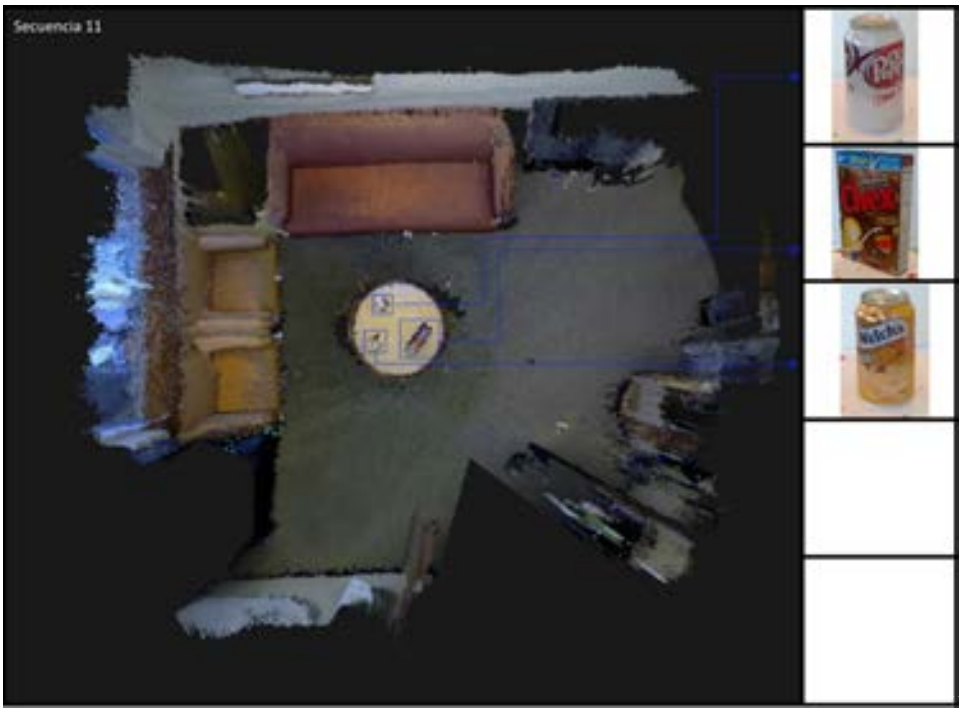

Figura 4.20: Reconstrucción de la secuencia 11 del escenario 3 a partir de la información de movimiento verdadero.

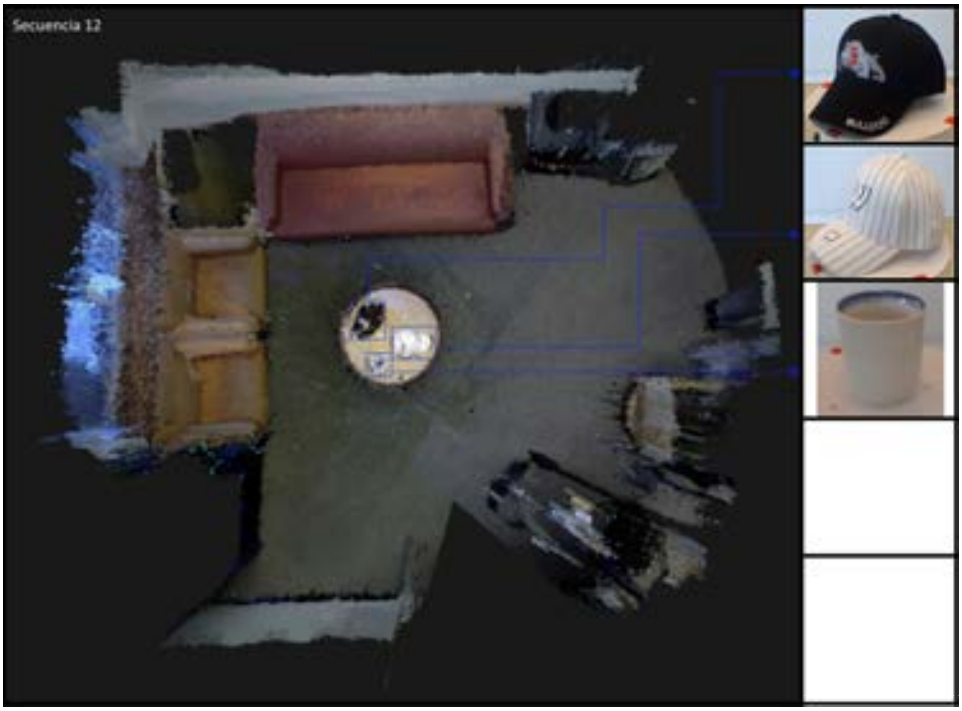

Figura 4.21: Reconstrucción de la secuencia 12 del escenario 3 a partir de la información de movimiento verdadero.

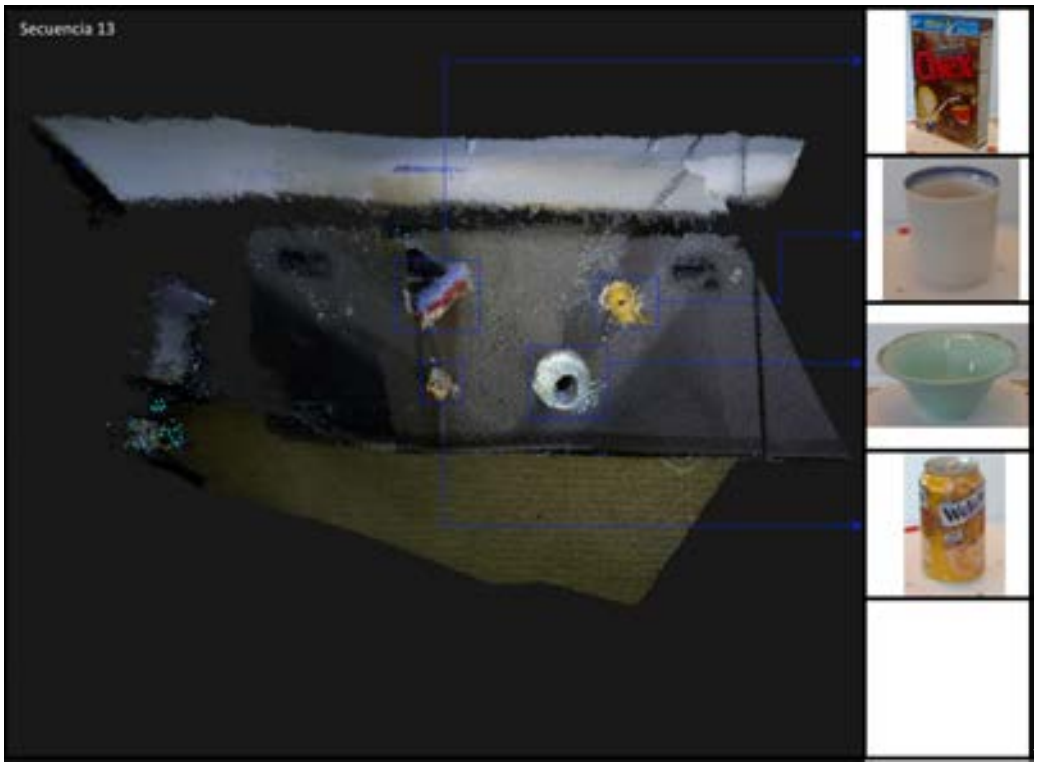

Figura 4.22: Reconstrucción de la secuencia 13 del escenario 3 a partir de la información de movimiento verdadero.

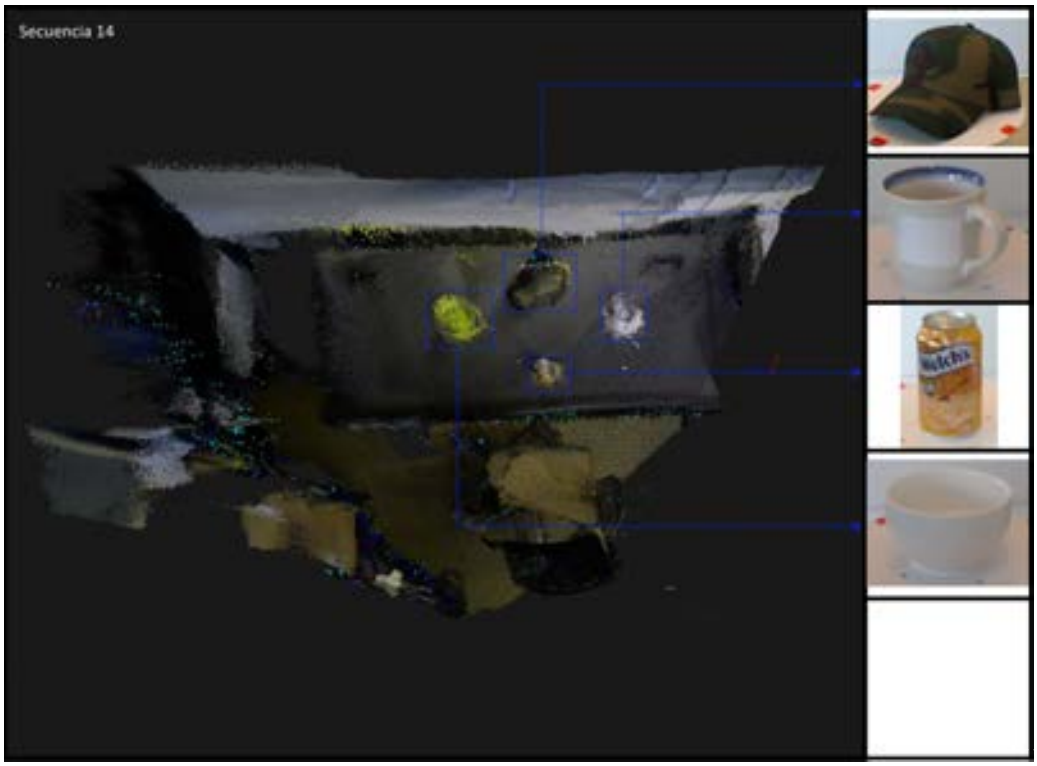

Figura 4.23: Reconstrucción de la secuencia 14 del escenario 4 a partir de la información de movimiento verdadero.

Los escenarios presentan elementos fijos como las paredes, ventanas, mesas y muebles, al igual que elementos dinámicos como cajas, recipientes, tasas entre otros, cada escenario se compone de secuencias con diferentes objetos dinámicos visibles todo el tiempo. La técnica SLAM produce como resultado una secuencia de poses tridimensionales en un archivo plano, cada pose indica la posición de la cámara para cada secuencia de imágenes RGBD. En el Apéndice A se puede observar la gráfica comparativa de las poses obtenidas con la técnica SLAM propuesta, respecto al resultado de otra técnica SLAM y la pose verdadera dada por la base de datos para cada escenario y secuencia. Con las poses calculadas se puede generar la reconstrucción de los escenarios, el Escenario 1 se muestra en la Figura 4.24, el Escenario 2 se muestra en la Figura 4.25, el Escenario 3 se muestra en la Figura 4.26 y el Escenario 4 se muestra en la Figura 4.27. El cálculo del error se realiza con las herramientas de evaluación de error RPE y ATE [39], cada una admite la lista de poses calculadas y las verdaderas; para cada escenario se calcula el error ATE, comparativamente se muestra en el Tabla 4.6. De igual forma se calcula el error RPE, el cual se muestra comparativamente en el Tabla 4.7.

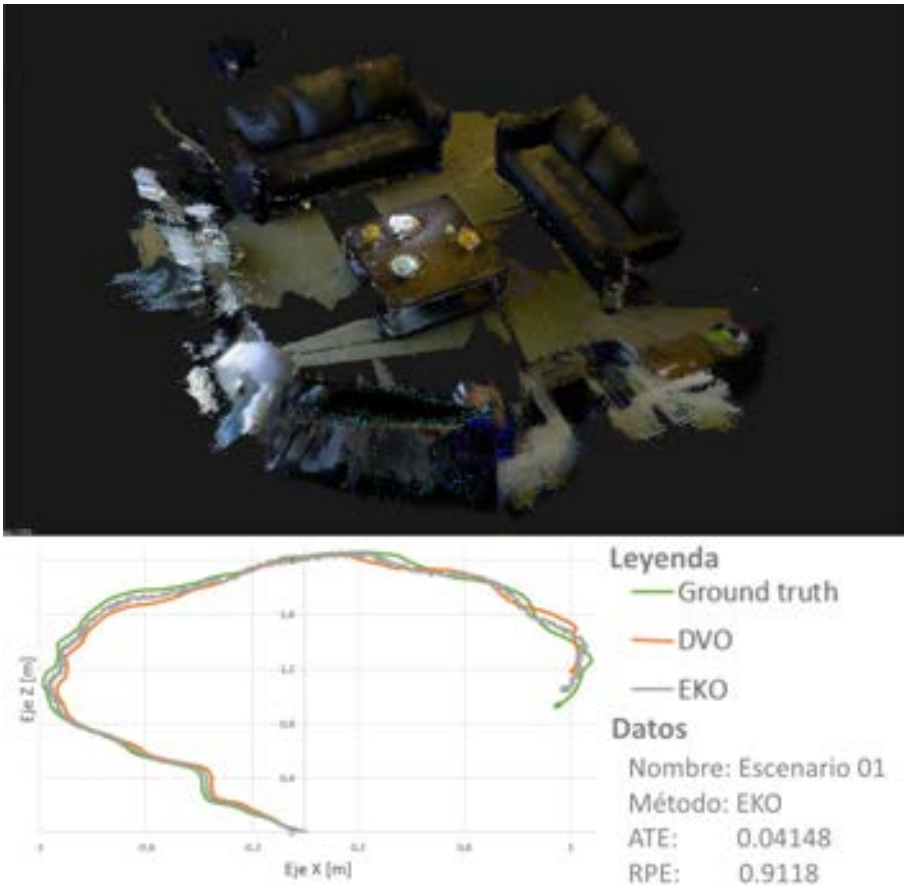

Figura 4.24: Reconstrucción del Escenario 1 con los resultados de la técnica SLAM propuesta.

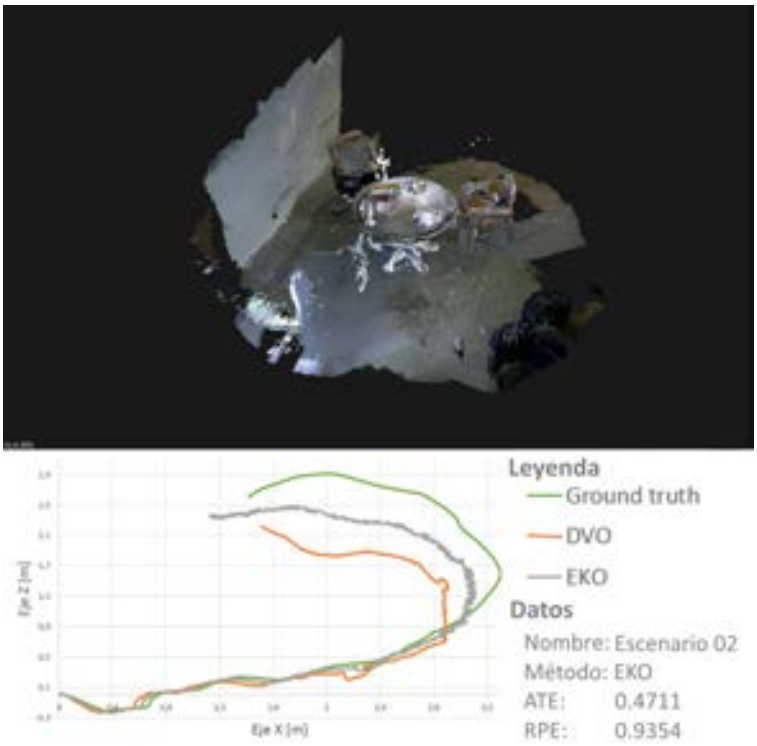

Figura 4.25: Reconstrucción del Escenario 2 con los resultados de la técnica SLAM propuesta.

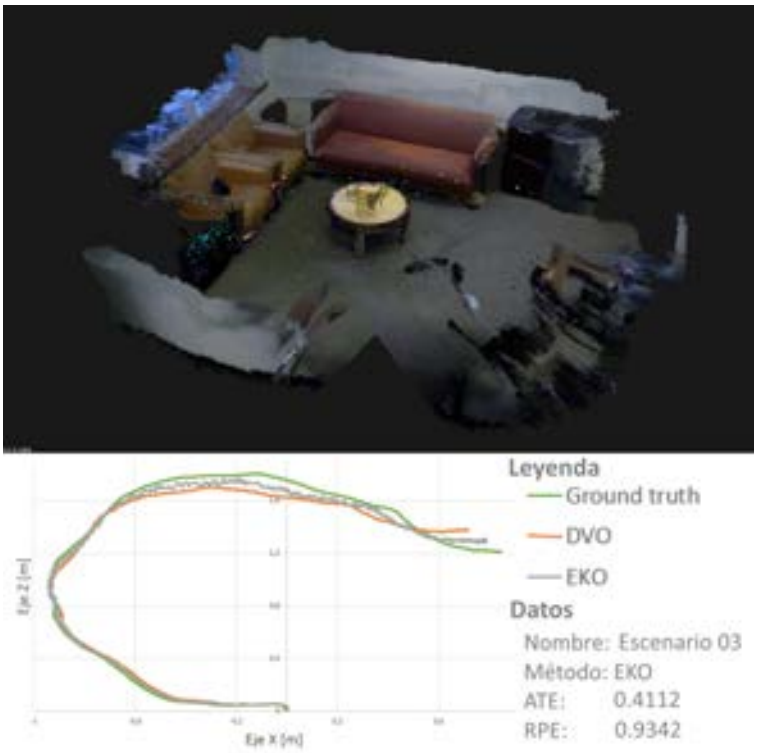

Figura 4.26: Reconstrucción del Escenario 3 con los resultados de la técnica SLAM propuesta.

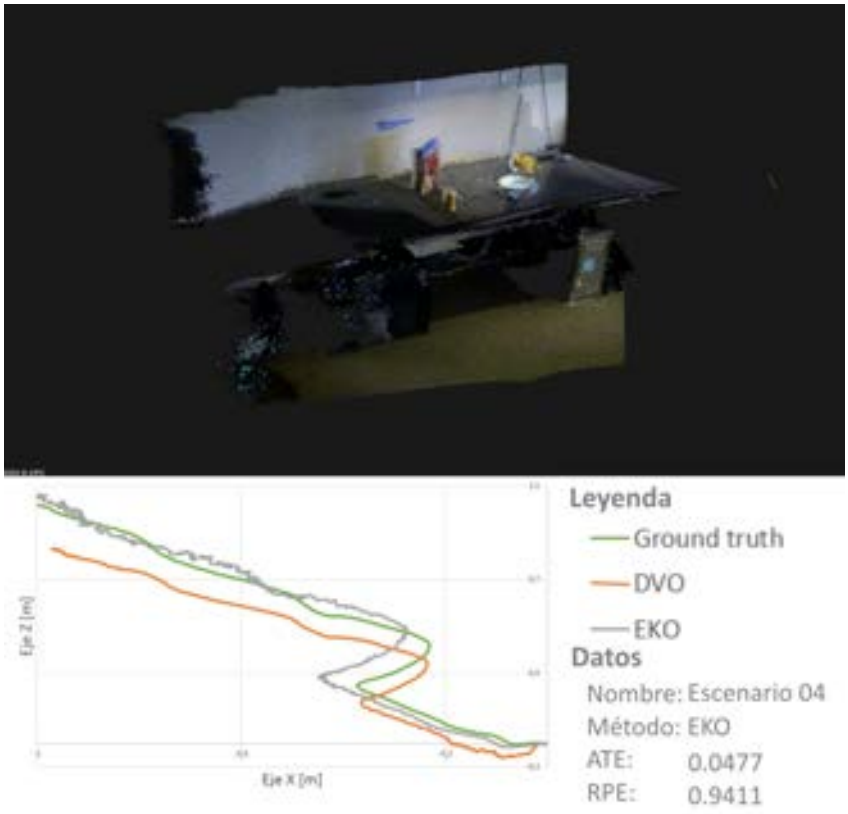

Figura 4.27: Reconstrucción del Escenario 1 con los resultados de la técnica SLAM propuesta.

Tabla 4.6: Error ATE comparativo para la técnica SLAM propuesta y la técnica de referencia.

Escenario	Secuencia	Técnica Propuesta	Técnica Referencia
1	1	0.041484	0.083926
1	2	0.049274	0.091655
1	3	0.050082	0.08795
1	4	0.043369	0.085722
2	1	0.047119	0.088135
2	2	0.048393	0.092582
2	3	0.049307	0.084097
2	4	0.046872	0.092397
3	1	0.041216	0.084637
3	2	0.047582	0.084694
3	3	0.040795	0.084282
3	4	0.043406	0.086774
4	1	0.047726	0.090637
4	2	0.04884	0.086898

Tabla 4.7: Error RPE comparativo para la técnica SLAM propuesta y la técnica de referencia.

Escenario	Secuencia	Técnica Propuesta	Técnica Referencia
1	1	0.941843	0.837364
1	2	0.937723	0.841875
1	3	0.938964	0.845629
1	4	0.935071	0.838202
2	1	0.935436	0.838892
2	2	0.938059	0.842721
2	3	0.940648	0.840462
2	4	0.935453	0.836725
3	1	0.9342	0.842269
3	2	0.933938	0.840878
3	3	0.934948	0.843299
3	4	0.935968	0.840329
4	1	0.941114	0.838029
4	2	0.941752	0.8411

Los resultados obtenidos en error ATE muestran que la técnica propuesta mejora el error absoluto de pose, lo cual concuerda con las gráficas de posición absoluta del Apéndice A, la estimación de la pose es mejor en la técnica propuesta que en la técnica de referencia en valores absolutos; los resultados obtenidos en error RPE muestran que la técnica propuesta empeora debido al ruido inducido por el alineador heurístico, el cual al tener un componente aleatorio induce un ruido en la estimación de la pose, por lo que los movimientos relativos tienen mayor error con la técnica propuesta. El cálculo del tiempo promedio de procesamiento en cada técnica SLAM se muestra en la Tabla 4.8, se observa que la técnica propuesta requiere mayor tiempo que la técnica de referencia, esto se debe principalmente a la etapa de cálculo de características y a la etapa de segmentación. El cálculo de características incide negativamente en el tiempo procesamiento debido a que procesa el doble de pixeles en cada iteración, mientras que el cálculo de la probabilidad de las características implica mayor procesamiento dado que se compara cada píxel con sus vecinos con el fin de conectar regiones.

Tabla 4.8: Tiempo comparativo para la técnica SLAM propuesta y la técnica de referencia.

Escenario	Secuencia	Técnica Propuesta	Técnica Referencia
1	1	2.168663	0.987759
1	2	3.858398	0.99405
1	3	3.170981	0.966787
1	4	2.358619	0.95001
2	1	1.798244	0.966787
2	2	3.173523	0.96469
2	3	3.173079	0.962593
2	4	1.79405	0.926941
3	1	3.454425	0.968884
3	2	2.810384	0.947913
3	3	1.143718	0.93533
3	4	3.206412	0.958398
4	1	3.083343	0.970981
4	2	3.050230	0.991953

4.5 Conclusiones

La técnica SLAM planteada en el Capítulo 3 se compone de varias propuestas que se han evaluado individualmente a lo largo de este Capítulo, a continuación se presentan las conclusiones de dichas evaluaciones, además de determinar la viabilidad de su uso en la técnica final.

Las evaluaciones realizadas a las características SURF extendidas permiten afirmar que tienen mayor capacidad discriminativa debido a que reducen el área de superposición entre las clases como se observa con el indicador F2 en la tercera evaluación presentada en la sección 4.1, sin embargo aumenta el número de puntos sobre el límite de la clase como se observa con el indicador N1, por lo cual es fundamental la elección de un clasificador que no dependa únicamente de los centros de las clases. De igual forma con las primeras 2 evaluaciones, es importante resaltar que las características propuestas aumentan la capacidad discriminativa siempre a medida que la imagen de profundidad tenga mayor complejidad. Se concluye que las características extendidas pueden ser usadas en la técnica SLAM propuesta, dado que se calcularían sobre las imágenes de las escenas consecutivas, y dada la ventaja mostrada en ambientes complejos se puede disminuir el error al mejorar la capacidad discriminativa de las características.

Se concluye que el alineador de características heurístico propuesto es dependiente de las meta-características del problema a resolver, ganando y perdiendo en diferentes funciones de costo, siguiendo el teorema No Free Lunch (NFL) [73, 67]. En la técnica SLAM propuesta se propone que el alineador use las características extendidas SURF, las cuales fueron propuestas en el Capítulo 3, de acuerdo a las evaluaciones en este Capítulo 4, las

características propuestas cumplen el requisito en valores N1, N2 y N3 tanto para objetos como para escenarios, según la Tabla 4.1 y la Tabla 4.2. Las características SURF también cumplen el parámetro de número de muestras, dado que el número mínimo de puntos tomados para la alineación es 8, sin embargo se tienen valores usuales entre 100 y 500 con los que se hace una alineación estadística. La reducción del tiempo de entrenamiento representa un ahorro significativo para la actualización constante del mapa con cada medición, motivo por el cual esta propuesta puede aumentar la velocidad de la técnica SLAM presentado. Se concluye que el alineador propuesto puede mejorar la velocidad de la alineación, y que puede ser usado junto con las características SURF extendidas.

El código fuente del mapa probabilístico no fue implementado en el desarrollo de esta propuesta, sino que se utilizó una librería que cumplía con todos los requerimientos establecidos, de esta forma se realizó la integración del código fuente de la propuesta con la librería, dando como resultado la posibilidad de generar mapas en forma de árbol jerárquico como se observa en la Figura 4.8, cada nivel de la jerarquía agrega un nivel adicional de detalle que puede variar de acuerdo a las necesidades. Al ser una librería externa el mapa no se realizaron evaluaciones, debido a que ya existen investigaciones que realizan diferentes pruebas sobre estas librerías, sin embargo el resultado final y la capacidad de generar los mapas en este tipo de archivo se pueden considerar una extensión o mejora de la técnica SLAM.

La técnica SLAM propuesta reduce el error absoluto de pose, sin embargo la introducción de un alineador de características heurístico introduce un ruido que afecta el error relativo de pose. El desempeño en tiempo del algoritmo es mayor que la técnica de referencia, esto se debe a que el cálculo de características extendidas y el cálculo de áreas segmentadas son computacionalmente más complejas, y la reducción del tiempo del alineador heurístico no alcanza a compensar el tiempo requerido por las otras modificaciones propuestas.

Capítulo 5

Conclusiones

En este trabajo se ha realizado la propuesta de una técnica SLAM para ambientes dinámicos tridimensionales, el trabajo presentó un estado del arte que permitió identificar los problemas actuales con los algoritmos de reconstrucción y localización basados en imágenes RGBD. Desde el estado del arte se concluye que los métodos SLAM generan un valor agregado muy alto a las cámaras RGBD debido a que permiten obtener la reconstrucción del escenario en ambientes no estándares , esto sin duda puede ayudar a impulsar las aplicaciones de realidad virtual e Internet de las cosas, sin embargo es la alta demanda de procesamiento la que genera mayores problemas para llevar esta tecnología al usuario final, debido a que la mayoría de equipos de procesamiento no serían capaces de ejecutar algoritmos SLAM durante tiempos prolongados. Adicionalmente se concluye que la investigación sobre métodos SLAM ha impulsado grandes desarrollos en términos de librerías abiertas y disponibles, con el fin de procesar datos tridimensionales de maneras más fáciles.

Entre los problemas identificados se encontró que las técnicas SLAM actuales tienen problemas de asociación de datos en ambientes donde los objetos son dinámicos, con base a esto se realizó la propuesta de una técnica SLAM que resolviera el problema de los ambientes dinámicos considerando que estos pueden afectar la integridad del mapa. Se realizaron varias propuestas de modificación que incluían el cambio del tipo de descriptores, un nuevo tipo de clasificador y la integración de un mapa jerárquico, cada una de estas propuestas se trabajaron por separado permitiendo su evaluación de acuerdo de diversos parámetros establecidos por la comunidad investigativa. Con los resultados de evaluación individual se pudo establecer la pertinencia de su implementación en la técnica SLAM propuesta, de esta forma se concluyó que las características extendidas mejoraban la capacidad discriminativa entre escenarios mas no entre los objetos, el alineador de características heurístico mejora la tasa de aciertos cuando las características usadas tienen ciertos metavalores, sin embargo su uso induce un ruido en las poses estimadas dado el componente aleatorio.

La técnica propuesta implico que las características extendidas se usaran con alineador heurístico propuesto, debido a que al analizar el desempeño de las meta-características se

encontró que mejoran el porcentaje de acierto, este trabajo conjunto permitió reducir el error absoluto. Con base a los resultados obtenidos por parte del alineador heurístico, se generó la publicación [30] en la revista IEEE Latin America Transactions, con el objetivo de mostrar este método de clasificación a la comunidad investigativa.

El mapa usado con la técnica fue incorporado por una librería externa, sin embargo su uso permite el almacenamiento de la información más compacta. El cálculo de la probabilidad de ocupancia de acuerdo a las dimensiones de los objetos permite priorizar las características de objetos de mayor volumen, lo que incide directamente en la reducción del error. El tipo de mapa usado permite almacenar estos valores de probabilidad con el fin de determinar los objetos con mayor probabilidad de movimiento. La técnica SLAM propuesta se denomina *Odometria por características extendidas*, es un método que mejora la estimación de la pose en términos de error absoluto, sin embargo el error inducido por el alineador heurístico hace que el mapa reconstruido sea difuso o con poco detalle a pequeñas escalas, en el Capítulo 4 se muestra para cada escenario las reconstrucciones realizadas por el algoritmo. El método propuesto demanda mayor cantidad de tiempo para procesar las características extendidas y el cálculo de probabilidad desde la segmentación. En este trabajo se realizó la propuesta de modificar el algoritmo de alineador de características con el fin de reducir el tiempo de clasificación, la reducción de tiempo es en promedio del 50% lo que corresponde a $3ms$, sin embargo el aumento de tiempo por la segmentación corresponde en promedio a $1.5s$, y el cálculo de características aumenta en $0.50s$ respecto a las características SURF.

La técnica *Odometría por características extendidas* fue presentada en este trabajo, los resultados obtenidos demuestran que la reducción del error es considerable respecto a otra técnica actual como lo es *Dense Visual Odometry*, sin embargo es importante que se reduzca el ruido inducido por el alineador, como trabajos futuros se propone agregar una etapa adicional de refinamiento de pose basándose en la etapa de *bundle adjustment* de algunos métodos investigados, esta etapa permitiría contrarrestar el ruido inducido por el alineador heurístico. Se plantea la posibilidad de cambiar el método de segmentación por un método de identificación de objetos basado en HOG/SVM o entrenar un clasificador en cascada HAAR, lo cual permitiría encontrar más rápido los objetos presentes en el mapa y evitar el uso de un segmentador espacial. Este cambio propuesto podría reducir considerablemente el tiempo de procesamiento total de la técnica, y podría reducir el error absoluto de pose. A futuro se propone continuar la investigación sobre el efecto de la información geometría sobre extractores de características, incluyendo diferentes extractores actuales como ORB, FAST, SIFT. Finalmente, se propone como trabajo futuro la creación de un repositorio con todos los métodos SLAM propuestos, funcionales, y con estadísticas de desempeño, debido a que la falta de estos datos retrasa considerablemente el desarrollo de mejoras u otras propuestas.

Apéndice A

Resultado comparativo de la pose estimada por diferentes algoritmos SLAM respecto al movimiento real en los escenarios de la base de datos

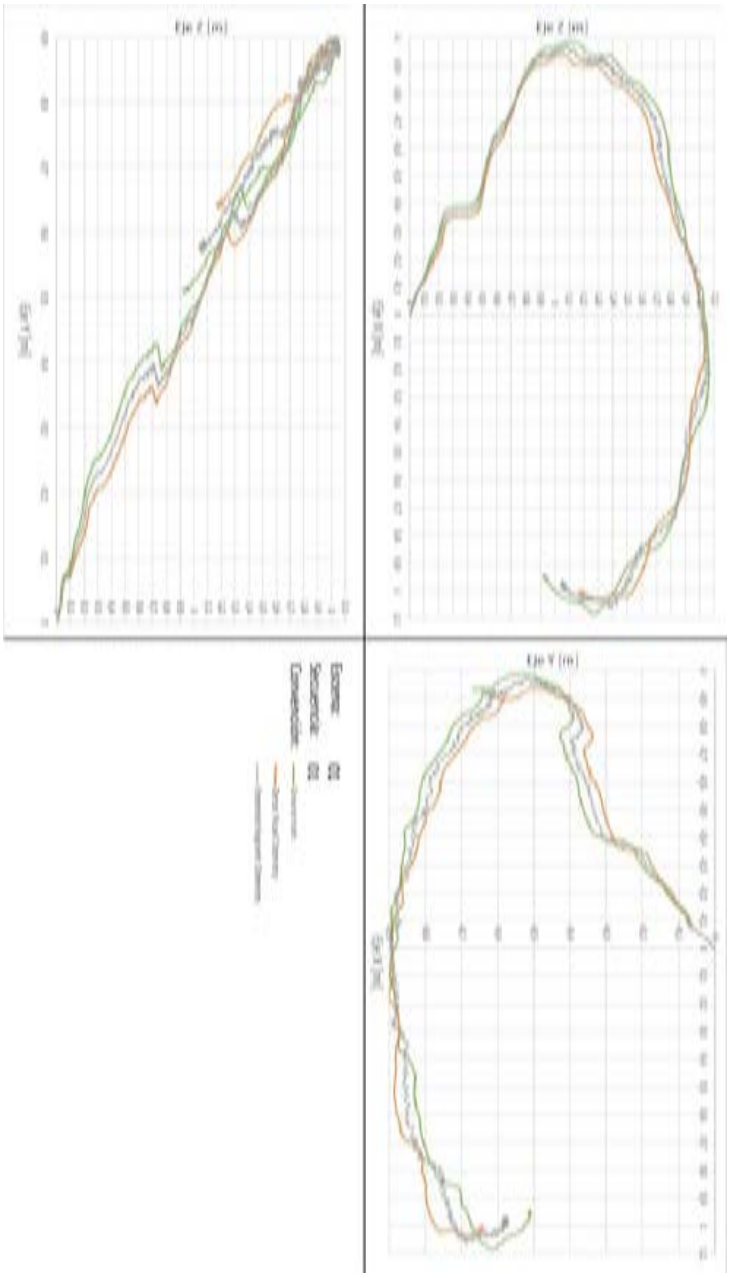

Figura A.1: Reconstrucción de la secuencia 1 del escenario 1 a partir de la información de movimiento verdadero

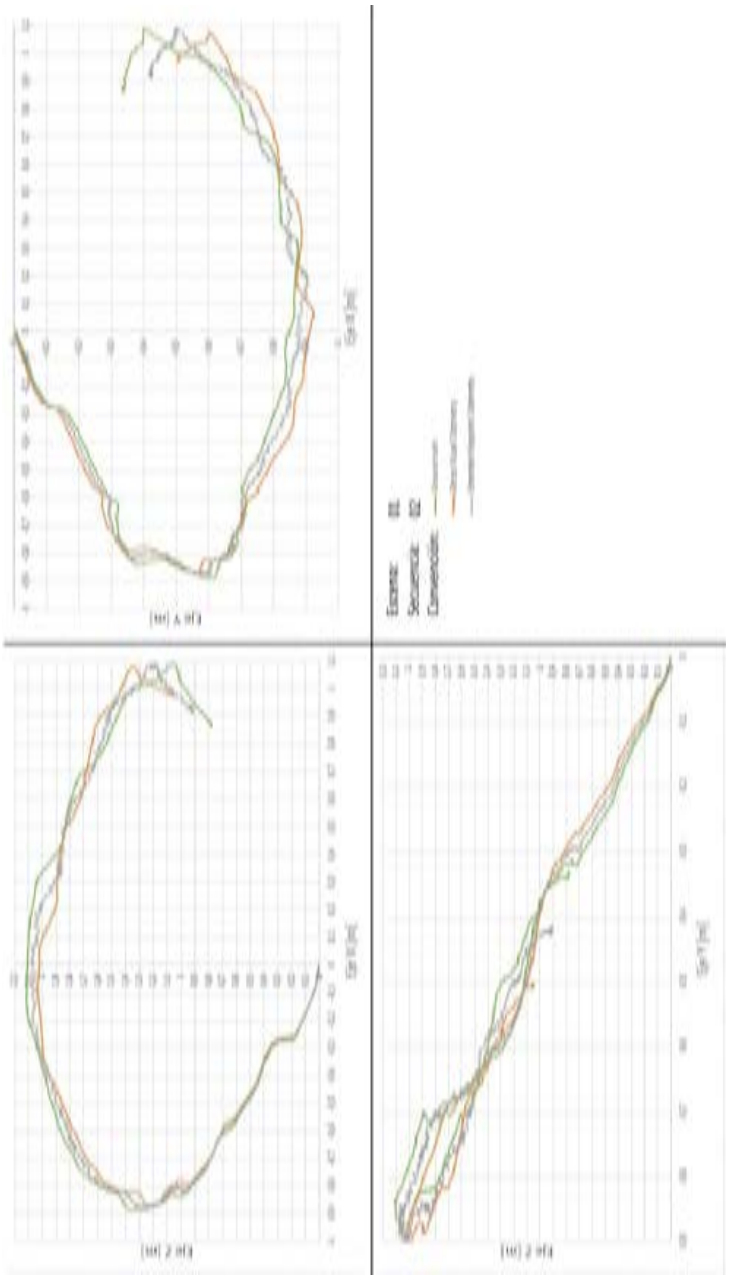

Figura A.2: Reconstrucción de la secuencia 2 del escenario 1 a partir de la información de movimiento verdadero

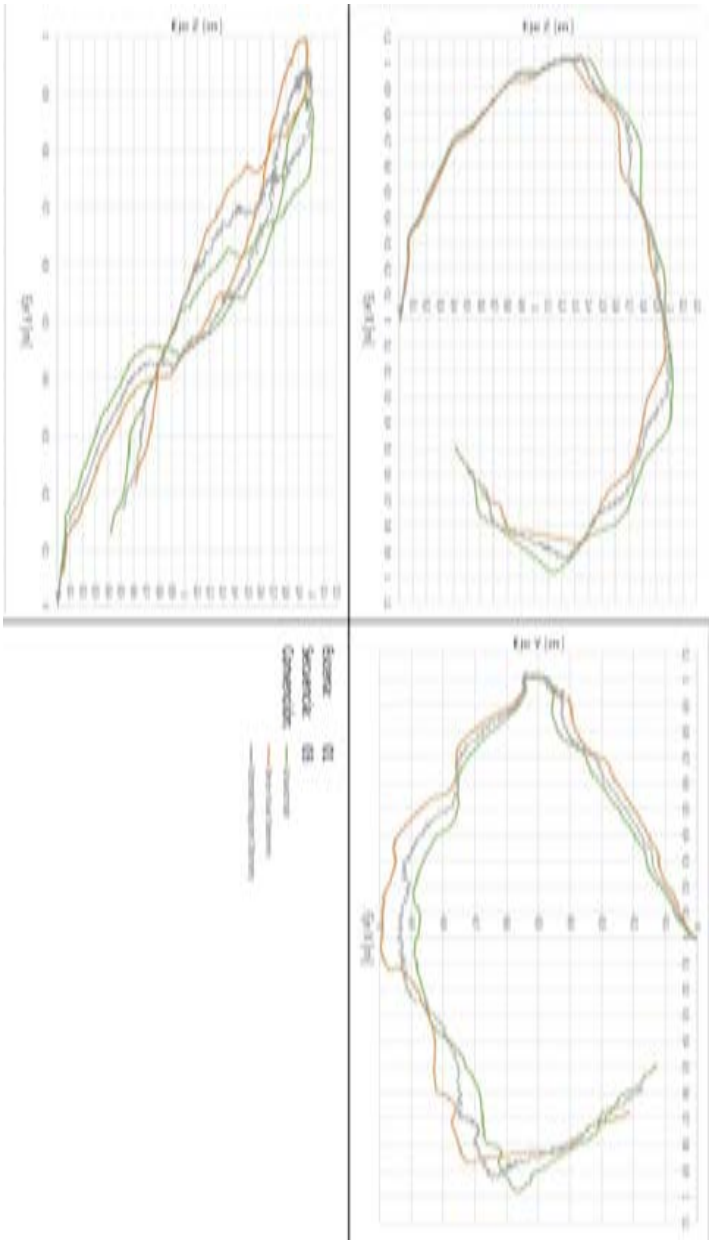

Figura A.3: Reconstrucción de la secuencia 3 del escenario 1 a partir de la información de movimiento verdadero

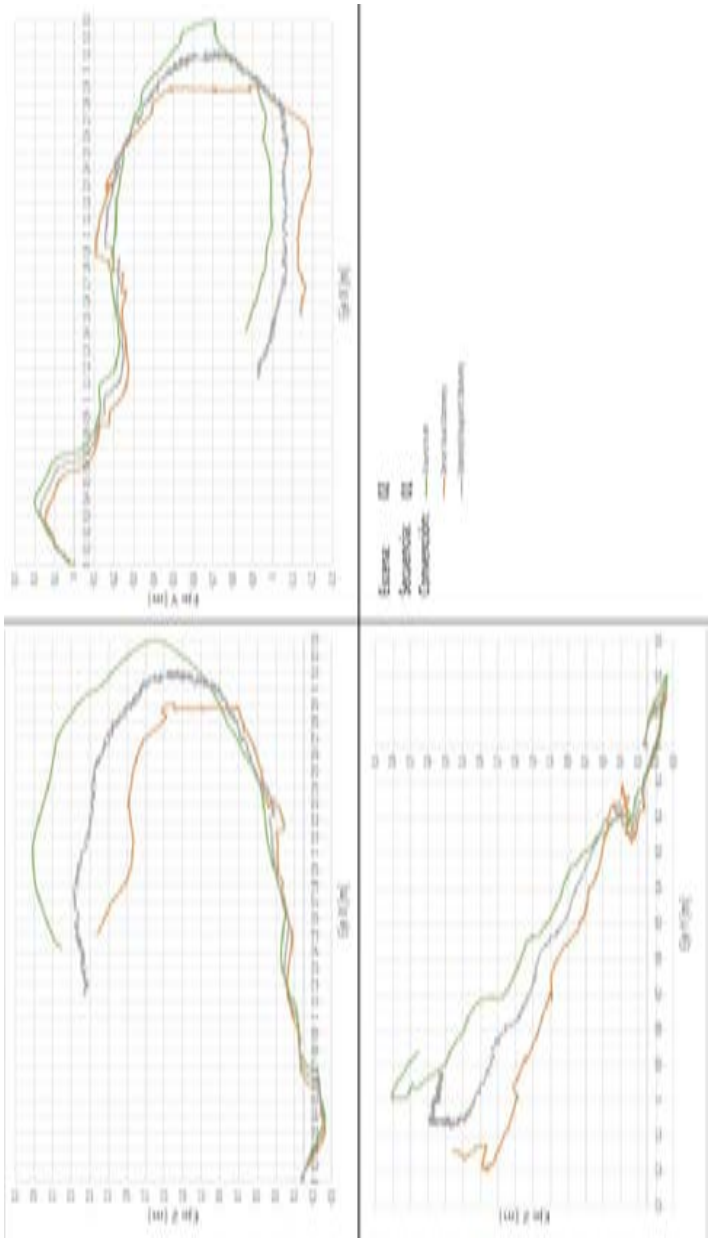

Figura A.4: Reconstrucción de la secuencia 4 del escenario 1 a partir de la información de movimiento verdadero

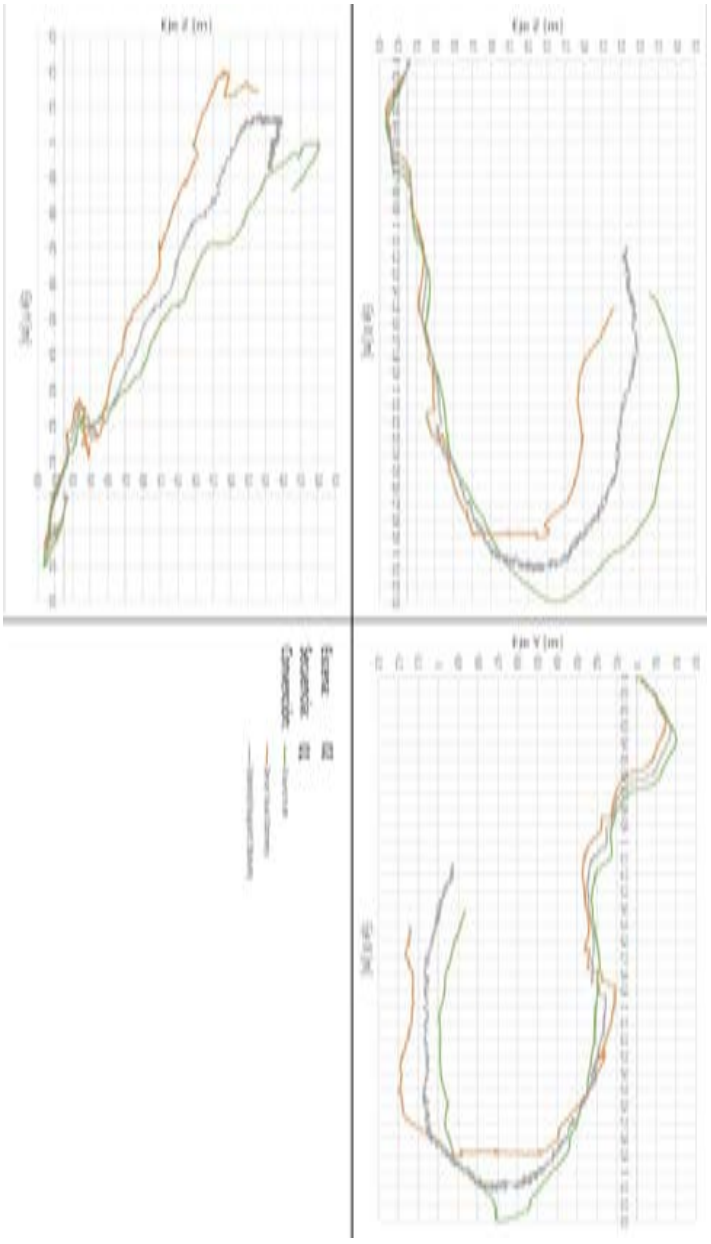

Figura A.5: Reconstrucción de la secuencia 5 del escenario 2 a partir de la información de movimiento verdadero

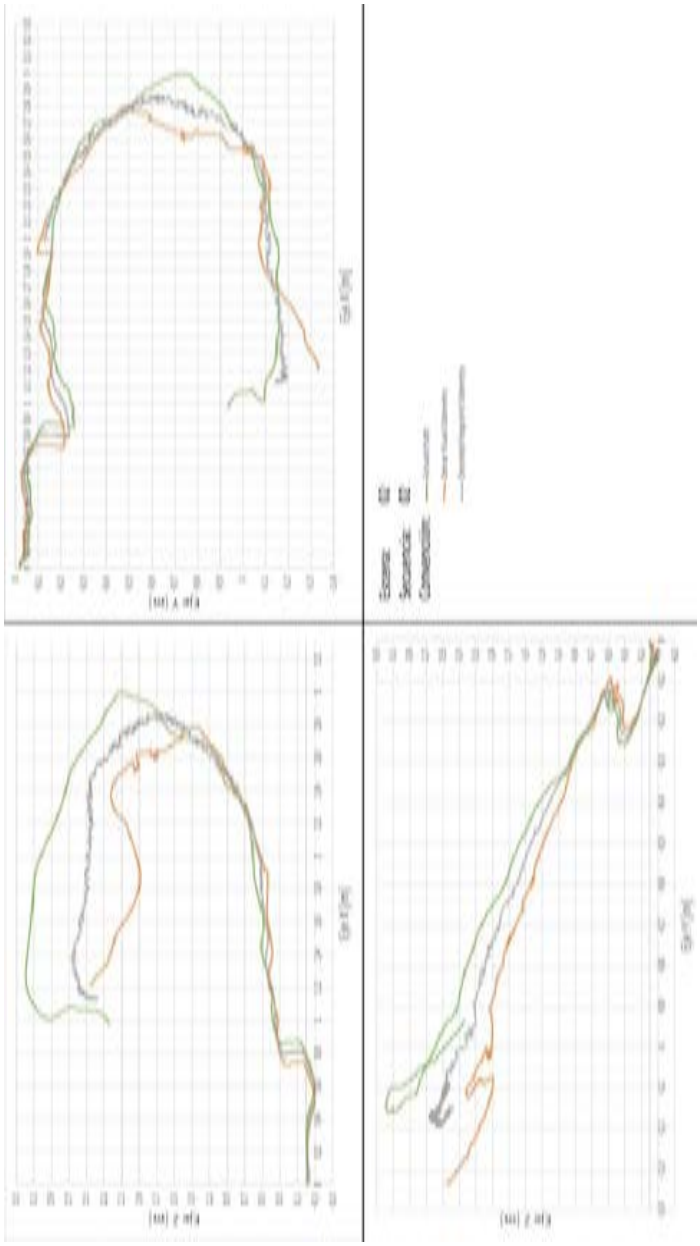

Figura A.6: Reconstrucción de la secuencia 6 del escenario 2 a partir de la información de movimiento verdadero

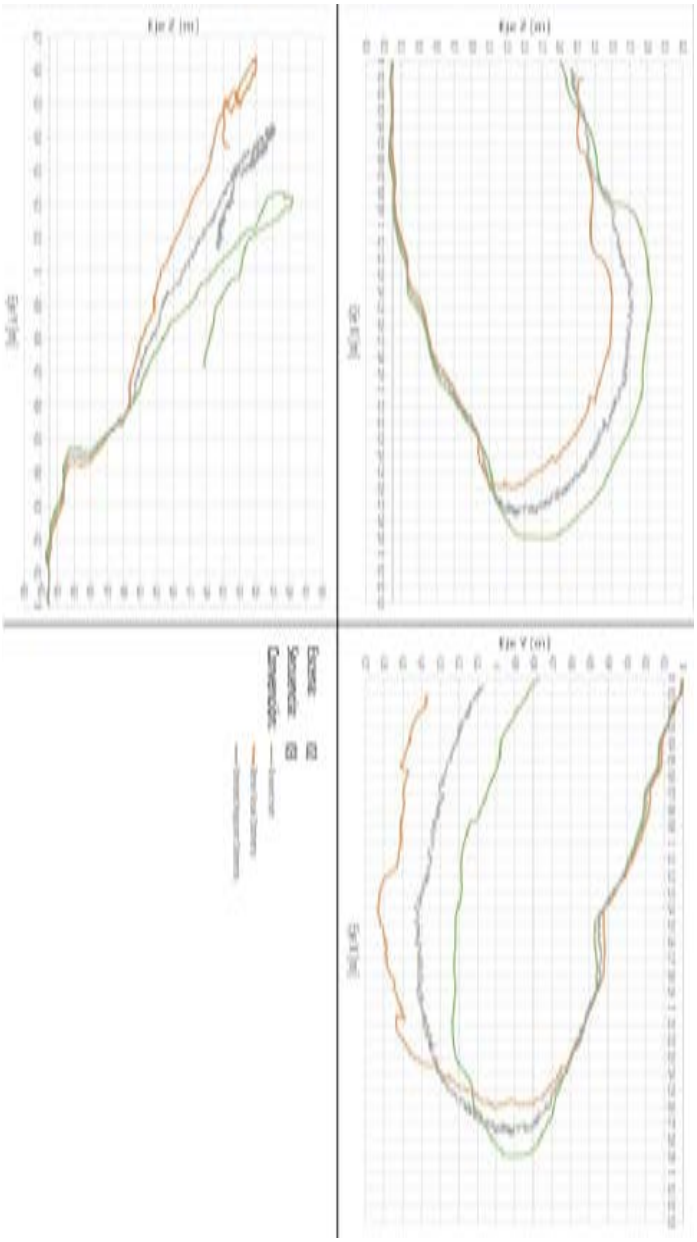

Figura A.7: Reconstrucción de la secuencia 7 del escenario 2 a partir de la información de movimiento verdadero

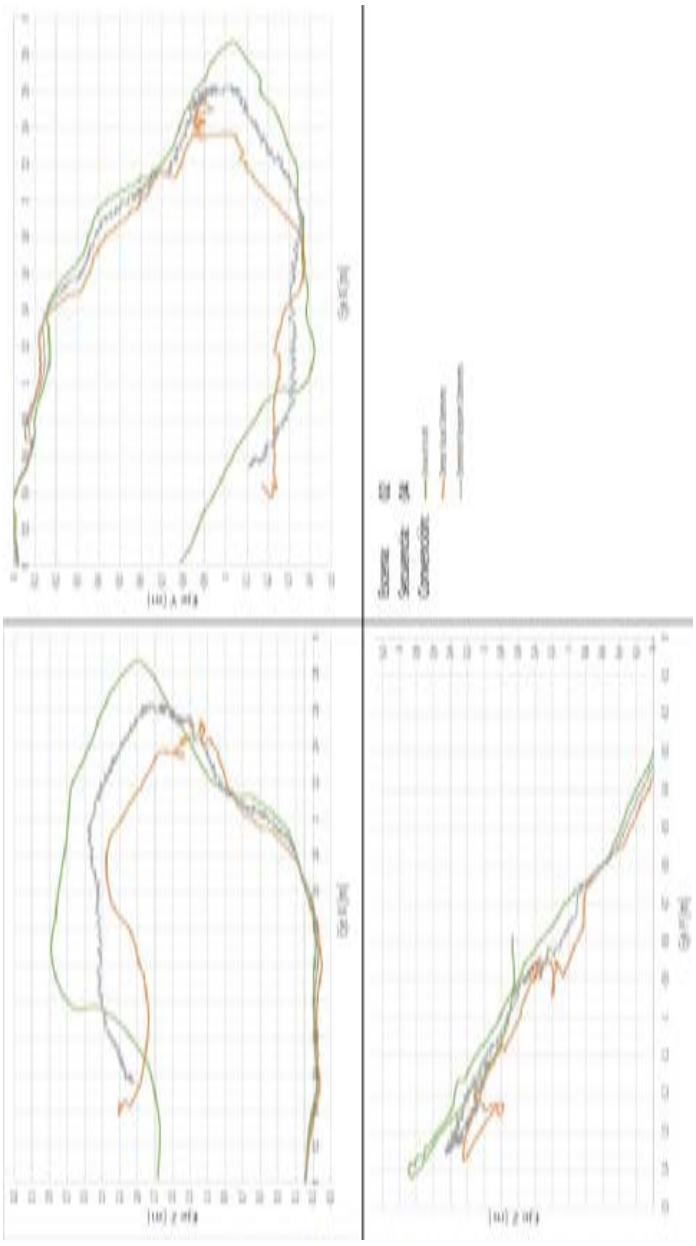

Figura A.8: Reconstrucción de la secuencia 8 del escenario 0 a partir de la información de movimiento verdadero

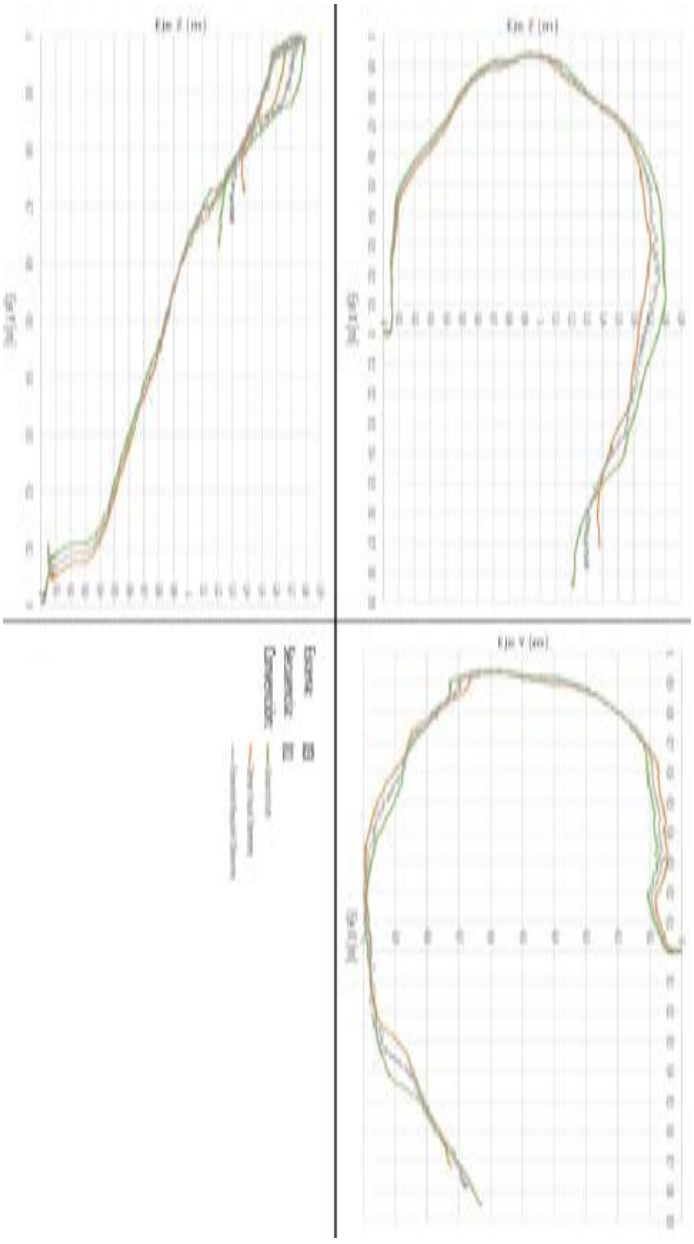

Figura A.9: Reconstrucción de la secuencia 9 del escenario 3 a partir de la información de movimiento verdadero

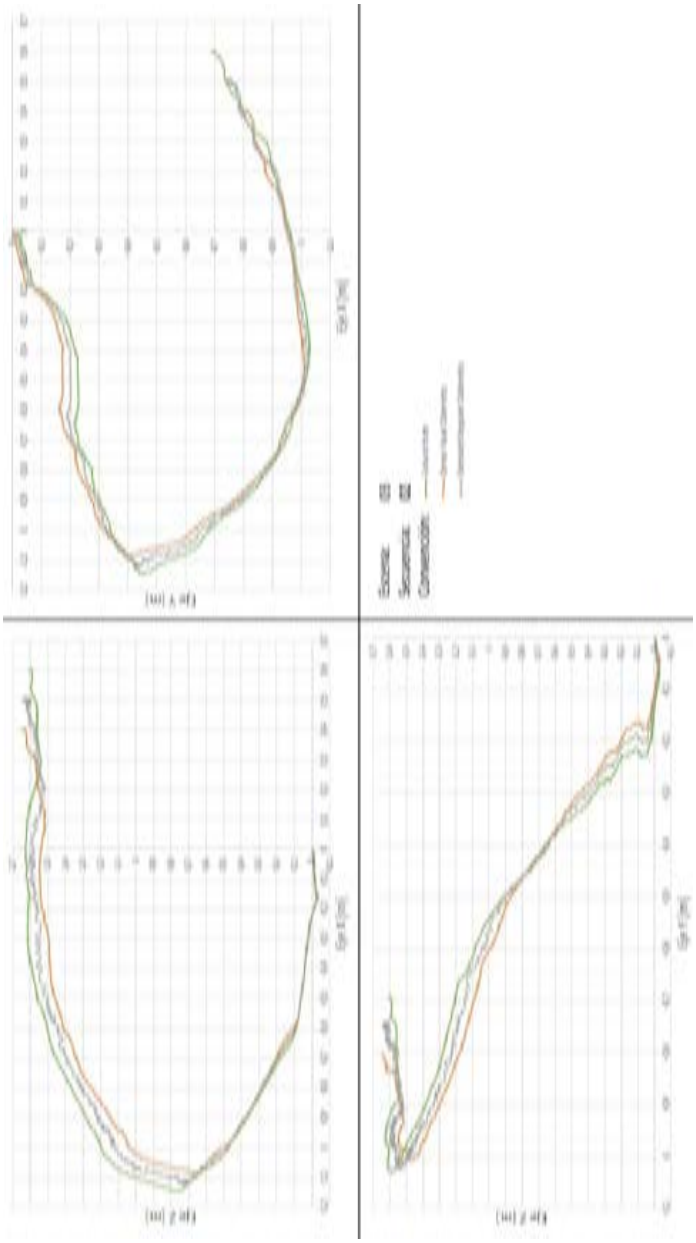

Figura A.10: Reconstrucción de la secuencia 10 del escenario 3 a partir de la información de movimiento verdadero

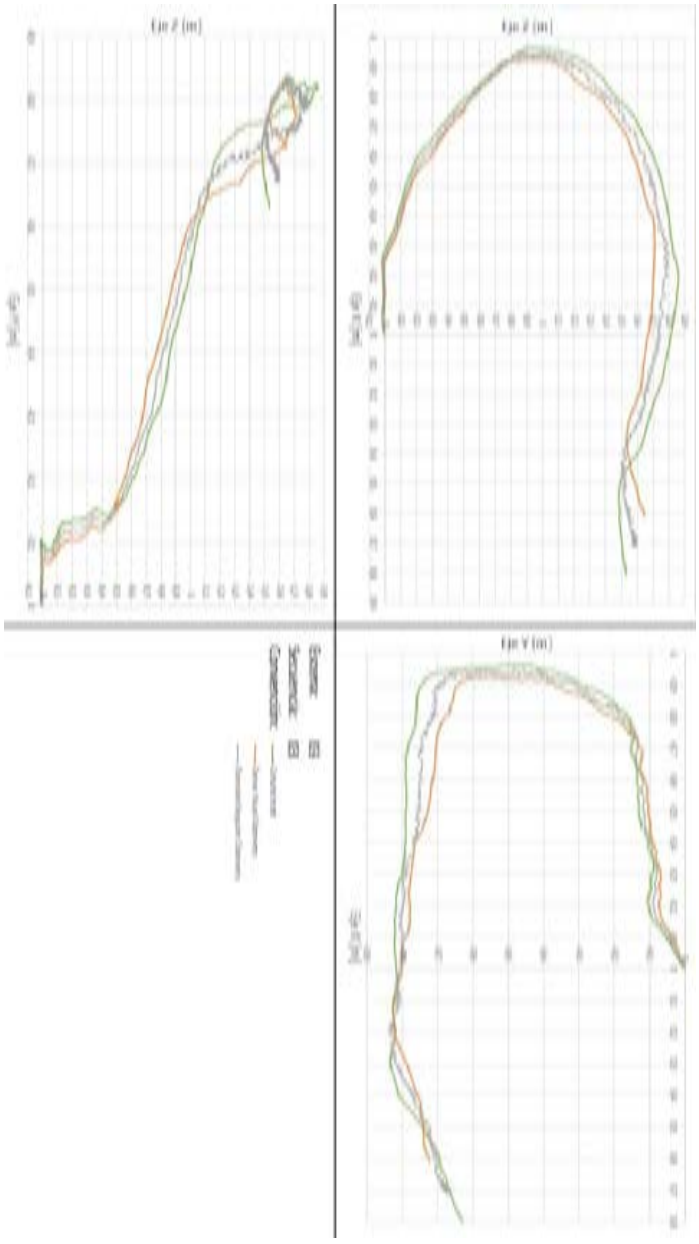

Figura A.11: Reconstrucción de la secuencia 11 del escenario 3 a partir de la información de movimiento verdadero

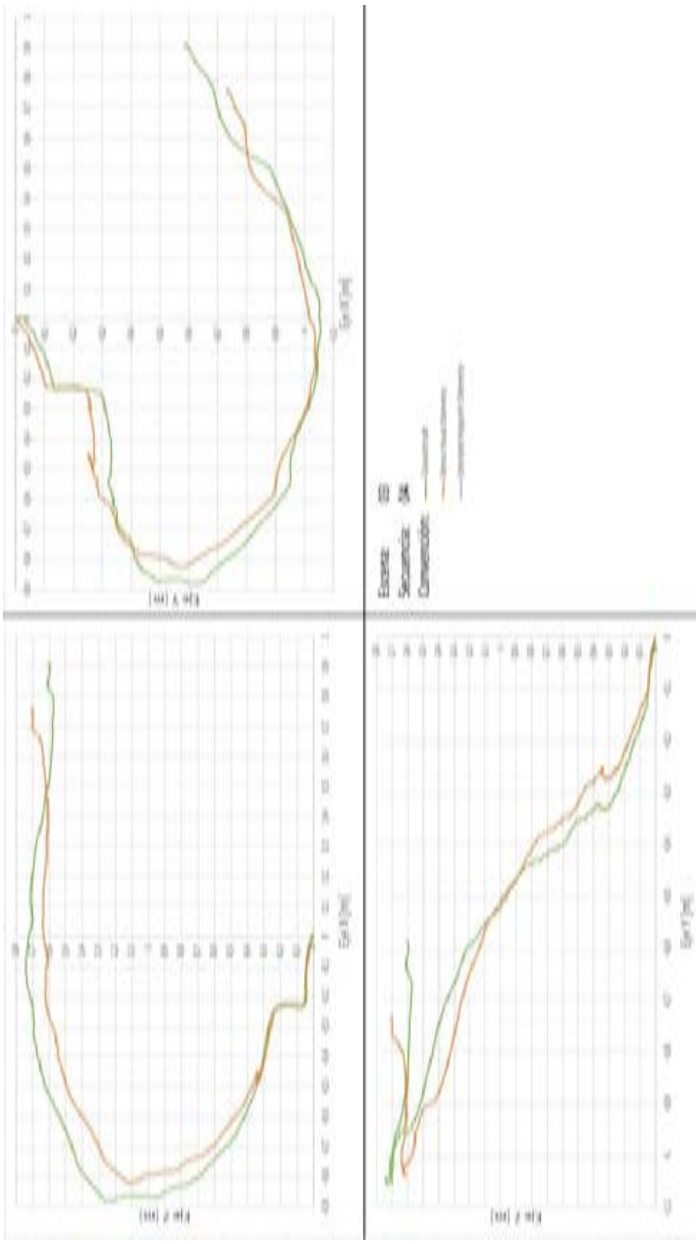

Figura A.12: Reconstrucción de la secuencia 12 del escenario 3 a partir de la información de movimiento verdadero

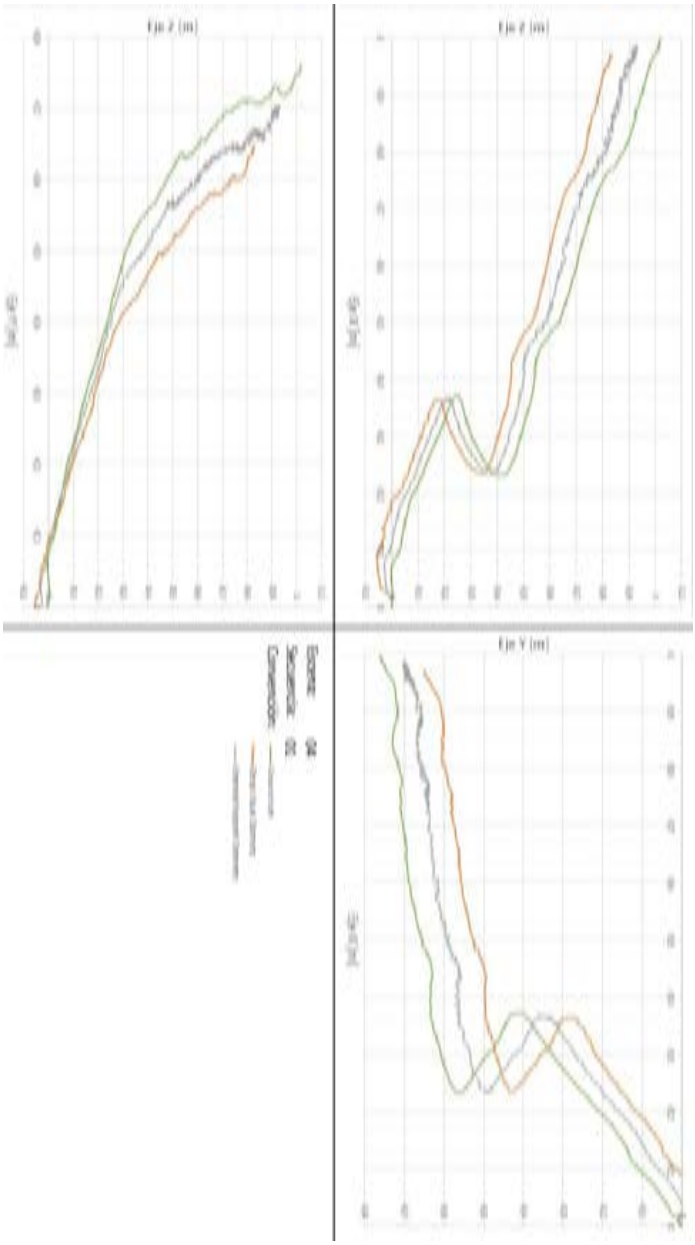

Figura A.13: Reconstrucción de la secuencia 13 del escenario 3 a partir de la información de movimiento verdadero

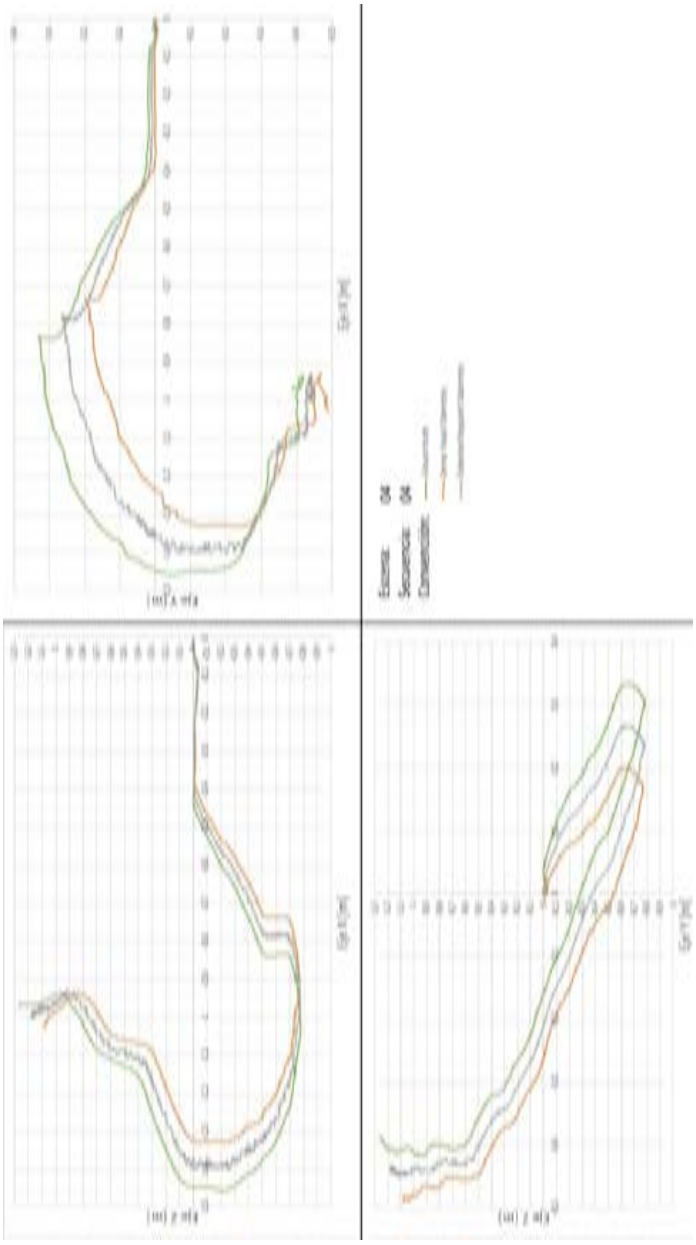

Figura A.14: Reconstrucción de la secuencia 14 del escenario 4 a partir de la información de movimiento verdadero

Bibliografía

[1] T. Lupo A. Certa and G. Passannanti. An efficient proposal for the application of simulated annealing algorithms. *1st International Conference on Engineering and Applied Sciences Optimization*, pages 910–927, 2014.

[2] Fabrizio Abrate, Basilio Bona, and Marina Indri. Monte carlo localization of mini-rovers with low-cost ir sensors. *2007 IEEE/ASME international conference on Advanced intelligent mechatronics*, 2007.

[3] Nuria Macia Albert Orriols-Puig and Tim Kam Ho. Data complexity library in c++. *dcol.sourceforge.net*, 2010.

[4] Rodrigo Francisco Munguia Alcala. Bearing-only slam methods. *Universitat politecnica de catalunya*, 2009.

[5] Stefano Arca, Elena Casiraghi, and Gabriele Lombardi. Corner localization in chessboards for camera calibration. *Universita degli Studi di Milano*, 2005.

[6] D. Askeland. Ciencia e ingenieria de los materiales. *CENGAGE*, 2005.

[7] K. Bache and M. Lichman. Uci machine learning repository. *University of California, Irvine, School of Information and Computer Sciences*, 2013.

[8] Gustavo Batista and Diego Furtado Silva. How k-nearest neighbor parameters affect its performance. *Simposio Argentino de Inteligencia Artificial*, 2009.

[9] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer Vision Image Understanding*, pages 346–359, 2008.

[10] Miguel Algaba Borrego. Desarrollo e implementacion de un metodo de generacion de mapas 3d usando el sensor kinect. *Universidad de Malaga*, 2012.

[11] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[12] Kai Wurm; Armin Hornung; Maren Bennewitz; Cyrill Stachniss; Wolfram Burgard. Octomap: A probabilistic flexible and compact 3d map representation for robotic systems. *Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[13] Jurgen Sturm Christian Kerl and Daniel Cremers. Dense visual slam for rgb-d cameras. *Proceedings of the conference on Robotics and Automation*, 2013.

[14] Jurgen Sturm Christian Kerl and Daniel Cremers. Robust odometry estimation for rgb-d cameras. *Proceedings of the conference on Robotics and Automation*, 2013.

[15] Michael Csorba. Simultaneous localisation and map building. *Oxford Universuty*, 1996.

[16] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *PAMI*, 29:1052–1067, 2007.

[17] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. *Proceedings of the IEEE International Conference on Computer Vision*, 2003.

[18] Gamini Dissanayake, Shoudong Huang, Zhan Wang, and Ravindra Ranasinghe. A review of recent developments in simultaneous localization and mapping. *International Conference on Industrial and Information Systems*, 2011.

[19] David Spiegelhalter; Charles Taylor Donald Michie. Machine learning, neural and statistical classification. *Overseas Press*, 1994.

[20] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. *proceedings 11th IEEE conference on computer vision*, pages 1–8, 2007.

[21] Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 1987.

[22] Alberto Elfes. Occupancy grids: A probabilistic framework for robot perception and navigation. *Department of Electrical and Computer Engineering, Carnegie Mellon University*, 1989.

[23] Felix Endres, Jurgen Hess, Nikolas Engelhard, Jurgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. *International Conference on Robotics and Automation*, 2012.

[24] Jakob Engel, Thomas Schops, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. *Proceedings of 13th European Conference in Computer Vision*, 2014.

[25] Nikolas Engelhard, Felix Endres, Jurgen Hess, Jurgen Sturm, and Wolfram Burgard. Real-time 3d visual slam with a hand-held rgb-d camera. *Proceedings of the RGB-D workshop on 3D perception in robotics at the European robotics forum*, 2011.

[26] Ross Finman, Thomas Whelan, Michael Kaess, and John J. Leonard. Toward lifelong object segmentation from change detection in dense rgb-d maps. *European Conference on Mobile Robots*, 2013.

[27] Jurgen Sturm Frank Steinbrucker and Daniel Cremers. Real-time visual odometry from dense rgb-d images. *Workshop on Live Dense Reconstruction with Moving Cameras at the International Conference on Computer Vision*, 2011.

[28] H. Wuest G. Bleser and D. Stricker. Online camera pose estimation in partially known and dynamic scenes. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 56–65, 2006.

[29] Pappa Gisele, Ochoa Gabriela, Hyde Matthew, Freitas Alex, Woodward Jhon, and Jerry Swan. Contrasting meta-learning and hyper-heuristic research: the role of evolutionary algorithms. *Genetic Programming and Evolvable Machines*, 15:3–35, 2014.

[30] Daniel Gomez, Flavio Prieto, and Maria Guzman. Nearest neighbors by adaptive simulated annealing. *IEEE Latin America Transactions*, 2015.

[31] Dirk Hahnel, Dirk Schulz, and Wolfram Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17:2003, 2003.

[32] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Using depth cameras for dense 3d modeling of indoor environments. *International Symposium on Experimental Robotics*, 2010.

[33] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. *Proceedings of the IEEE Intl. Conf. on Robotics and Automation*, 2011.

[34] Virgile Hogman. Building a 3d map from rgb-d sensors. *Royal Institute of Technology*, 2011.

[35] Tim Bailey Hugh Durrant-Whyte. Simultaneous localisation and mapping (slam):part i the essential algorithms. *Robotics & Automation Magazine, IEEE*, 13:99–110, 2006.

[36] Alexandru Eugen Ichim. Rgb-d handheld mapping and modeling. *Ecole Polytechnique Federale de Lausanne*, 2013.

[37] L. Ingber. Very fast simulated re-annealing. *Mathematical and Computer Modelling*, pages 967–973, 1989.

[38] Sanchez Jose, Mollineda Ramon, and Sotoca Jose. An analysis of how training data complexity affects the nearest neighbor classifiers. *Pattern Analysis & Applications*, 10:189–201, 2007.

[39] Felix Endres Wolfram Burgard Jurgen Sturm, Nikolas Engelhard and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. *In International Conference on Intelligent Robot Systems (IROS)*, 2012.

[40] Xiaofeng Ren Kevin Lai, Liefeng Bo and Dieter Fox. Detection-based object labeling in 3d scenes. *International Conference on Robotics and Automation*, 2012.

[41] S. Kotsiantis, I. D. Zaharakis, and P. Pintelas. Machine learning: a review of classification and combining techniques. *Springer Science*, pages 2564–2571, 2007.

[42] Benjamin Kuipiers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 1991.

[43] Rainer Kummerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g2o: A general framework for graph optimization. *Proceedings of the RGB-D workshop on 3D perception in robotics at the European robotics forum*, 2010.

[44] Jean Paul Laumond. Model structuring and concept recognition : two aspects of learning for a mobile robot. *Proceedings IJCAI-83*, pages 839–841, 1983.

[45] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal in Computer Vision*, pages 91–110, 2004.

[46] T. Hiroyasu M. Miki and T. Jitta. Adaptive simulated annealing for maximum temperature. *Trans. Information Processing*, pages 2787–2795, 2003.

[47] Julian Mason, Bhaskara Marthi, and Ronald Parr. Object disappearance for object discovery. *International Conference on Intelligent Robots and Systems*, 2012.

[48] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Factored solution to the simultaneous localization and mapping problem. *National Conference on Artificial Intelligence AAAI*, 2002.

[49] Luis Montesano. Detection and tracking of moving objects from a mobile platform. application to navigation and multi-robot localization, 2005.

[50] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

[51] Department of defense of United States of America. Global positioning system standard positioning service performance standard. 4, 2008.

[52] Joan Sola Ortega. Towards visual localization, mapping and moving objects tracking by a mobile robot: a geometric and probabilistic approach, 2007.

[53] Newman P., Clark S., and Durrant-Whyte H.F. A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on robotics*, 17:229 – 241, 2001.

[54] Atmaram Palakodety. Cmos active pixel sensors for digital cameras. *University of north Texas*, 2007.

[55] Tomas Lozano Perez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, pages 108–120, 1983.

[56] Brooks R. Symbolic error analysis and robot planning. *Journal of Robotics Researc*, 1984.

[57] Peter Cheeseman Randall Smith. On the representation of spatial uncertainty. *The International Journal of Robotics Research*, 1987.

[58] Matthias Reif, Faisal Shafait, Markus Goldstein, Thomas Breuel, and Andreas Dengel. Automatic classisifier selection for non-experts. *Pattern Analysis and Applications*, 17:83–96, 2014.

[59] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Proceedings of the 2011 international conference on computer vision. *Computer Vision Image Understanding*, pages 2564–2571, 2011.

[60] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[61] D. Gelatt S. Kirkpatrick and M. Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, 1983.

[62] Yun Shi, Shunping Ji, Zhongchao Shi, Yulin Duan, and Ryosuke Shibasaki. Gps-supported visual slam with a rigorous sensor model for a panoramic camera in outdoor environments. *Sensors*, pages 119–136, 2012.

[63] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3d with kinect. *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on Computer Vision*, 2011.

[64] Randall Smith, Mattew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 1990.

[65] NSTB/WAAS TE Team. Global positioning system (gps) standard positioning service (sps) performance analysis report. 85, 2014.

[66] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine learning*, 1998.

[67] Tor Lattimore Tom Everitt and Marcus Hutter. Free lunch for optimisation under the universal distribution. *IEEE Congress on Evolutionary Computation*, 2014.

[68] Andres Alejandro Diaz Toro. Localizacion y construccion simultanea de mapas en tiempo real a traves de entornos a gran escala empleando una camara monocular. *Universidad del valle*, 2012.

[69] Trung-Dung VU. Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects, 2009.

[70] Trung-Dung Vu, Julien Burlet, and Olivier Aycard. Grid-based localization and local mapping with moving object detection and tracking. *Information Fusion*, 12:58–69, 2011.

[71] Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and racking of moving objects. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA*, pages 842–849, 2002.

[72] Zilong Dong Guofeng Zhang Wei Tan, Haomin Liu and Hujun Bao. Robust monocular slam in dynamic environments. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 209 – 218, 2013.

[73] David Wolpert and William Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1997.

[74] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE Transactions on pattern analysis and machine intelligence*, pages 354–366, 2013.