UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA - PROCESAMIENTO DIGITAL DE SEÑALES

ESTIMACIÓN DE LA TRAYECTORIA DE UN ROBOT MÓVIL USANDO ODOMETRÍA VISUAL CON CÁMARA RGB-D

TESIS
PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
BAYRON ALEJANDRO GARZÓN CIFUENTES

TUTOR PRINCIPAL
Dr. JESÚS SAVAGE CARMONA
FACULTAD DE INGENIERÍA

CIUDAD UNIVERSITARIA, CD. MX, NOVIEMBRE, 2018

JURADO ASIGNADO:

Presidente:	Dr. Miguel Moctezuma Flores
Secretario:	M.I. Larry Hipólito Escobar Salguero
Vocal:	Dr. Jesús Savage Carmona
1er. Suplente:	Dr. Carlos Rivera Rivera
2o. Suplente:	Dr. Boris Escalante Ramírez

La tesis se realizó en el posgrado de Ingeniería, UNAM.

TUTOR DE TESIS:

Dr. JESÚS SAVAGE CARMONA

.

A mi amada familia por su amor y apoyo incondicional durante toda mi vida.
Madre, padre y hermana.

A ti Juliana, la mujer más excelsa que se ha cruzado en mi camino, con quien deseo compartir el resto de los días que me quedan por vivir. Mi amor.

A ti hija, que recién has llegado y ya es evidente que eres el amor de mi vida.
Te amaré cada día.

A la Facultad de Ingeniería y a la Universidad Nacional Autónoma de México, por la formación que me han dado.

A México, que me ha brindado un nuevo hogar.

Es gracias a ustedes que es posible el presente trabajo.
Gracias.

Reconocimientos

Al Dr. Jesús Savage Carmona por la confianza brindada para trabajar en el laboratorio de BioRobótica. Por sus conocimientos y enseñanzas que me fueron útiles durante la realización de este trabajo.

A los profesores del Programa de Maestría y Doctorado en Ingeniería del campo de Procesamiento Digital de Señales por brindar los fundamentos y encaminarme a la investigación.

A todos mis compañeros con quienes trabajé y quienes aportaron valiosos consejos y comentarios .

Finalmente quiero agradecer al CONACyT por el apoyo económico que me otorgó durante mis estudios de maestría y en el desarrollo de este trabajo de tesis.

A DGAPA UNAM por el apoyo recibido a través del proyecto PAPIIT IG100818 "Aplicación de modelos probabilísticos en el estudio de procesos cognitivos y el diseño de algoritmos para robots de servicio".

Acrónimos

SLAM	Simultaneous Localization And Mapping
VO	Visual Odometry
SFM	Structure From Motion
PNP	Perspective-n-Points
EPNP	Efficient Perspective-n-Points
SURF	Speeded-Up Robust Features
SIFT	Scale-Invariant Feature Transform
ORB	Oriented FAST and Rotated BRIEF
FAST	Features from Accelerated Segment Test
EKF	Extended Kalman Filter
LMS	Least-Mean-Square
NLMS	Normalized Least-Mean-Square
LMS-SER	Least-Mean-Square Sequential-Regression
RPE	Relative Pose Error
RMSE	Root-Main-Squares Error

Resumen

La odometría visual, es el proceso mediante el cual a partir de imágenes obtenidas desde una o multiples cámaras en un agente (robot, vehículo o humano, entre otros) se estima el ego-movimiento de éste. Las ventajas ampliamanete demostradas ante los modelos tradicionales de odometría clásica con encoders ha generado un fuerte interés entre múltiples centros y grupos de investigación, en universidades y en el sector empresarial gracias a que funcionan efectivamente en interiores como en exteriores y a su naturaleza, la cual es básicamente analizar el movimiento de una escena, permitiendo obtener trayectorias más acertadas en entornos complejos en términos de iluminación, superficies adversas y complicadas, donde un sistema típico por encoder tiene el problema de que su error de estimación crecería y generaría rápidamente una trayectoria degradada de la real.

Debido a las ventajas presentadas en los sistemas de visión, se desarrolló un sistema de odometría visual disperso para solucionar un problema de cuerpo rígido, con corrección por filtros adaptativos, específicamente filtros con sus coeficientes encontrados por tres algoritmos, el Least-Mean-Square (LMS), Normalized Least-Mean-Square (NLMS) y Least-Mean-Square Sequential-Regression (LMS-SER). El sistema propuesto consta de dos etapas fundamentales: una etapa de extracción de características de imagen, y una correspondencia de dichas características temporalmente, es decir en frames posteriores. A la primera etapa se le conoce como front-end. La segunda etapa es la estimación de la matriz de transformación que permite asociar las características encontradas en distintos tiempos y distintas coordenadas relativas, por lo que a partir de una concatenación de matrices de transformación es posible encontrar la trayectoria seguida por el sistema de visión.

Los resultados de este trabajo permiten concluir que es posible implementar un sistema de odometría visual disperso con robustez ante problemas de correspondencias en las características, usando técnicas de filtrado de bajo costo computacional y, con fundamento en los gradientes estocásticos y solución a la ecuación de Wiener por el algoritmo de LMS con algunas variaciones de éste.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Desde muchos años atrás, el ser humano viene haciendo uso de los recursos que tiene a disposición para moldear el entorno en su beneficio, desarrollando herramientas que le permitan ejecutar tareas de una manera más efectiva, rápida y automática, con sistemas capaces de llevar a cabo procesos de alta complejidad gracias al vertiginoso avance de la tecnología. A partir del deseo de mejorar la productividad, delegar tareas peligrosas y repetitivas para el ser humano, surgieron técnicas de diseño y construcción de robots y/o aparatos que realizan operaciones o trabajos, generalmente en instalaciones industriales, buscando crear sistemas que cumplan con el objetivo de suplantar al hombre en labores donde la integridad de éste tiene un riesgo latente, su presencia imposible o indeseable. Entre los múltiples beneficios de estos sistemas está el de potenciar la ejecución de tareas a realizar, liberando al hombre de funciones rutinarias, desagradables y complejas, lo que posibilitaría una nueva sociedad fundamentada en el desarrollo de otras actividades como arte, ciencia y tecnología. Robótica es un término acuñado por Isaac Asimov en sus obras de ciencia ficción, es la ciencia que se encarga de percibir y manipular el mundo físico a través de dispositivos electro-mecánicos. Esta ciencia de carácter interdisciplinario que involucra diferentes áreas como electrónica, control, mecánica, cibernética, computación entre otras, [8], esta inmersa en muchos campos de aplicación.

En las últimas décadas la construcción de robots ha sido un campo de gran desarrollo, teniendo como área la robótica; que se enfoca en el diseño, desarrollo e implementación de robots, los cuales son sistemas electromecánicos, cuyo objetivo más general es el de reemplazar al ser humano en la ejecución de tareas, tanto en lo que se refiere a la actividad física como en la toma de decisiones.

Los Robots tienen múltiples categorías en las que se pueden clasificar. Por su entorno de influencia como pueden ser robots industriales, de servicio, rescate entre otros o por su estructura física, grado de autonomía o nivel de movimiento; caso en el cual se pueden subdividir en robots de base móvil y de base fija. En los robots móviles podemos encontrar otras clases como son los robots que tienen desplazamiento por ruedas, orugas o patas.

Para el caso del desarrollo de robots de servicio, autónomos y móviles, se tienen aún muchos problemas por solucionar en busca de tener un robot propiamente autónomo, donde la autonomía se refiere a la capacidad de trabajar para cumplir su objetivo (s) o tarea (s), sin asistencia humana y adaptándose a los cambios del entorno, navegar y actuar sobre sí mismo y su ambiente con base en su diseño operativo.

De las múltiples tareas necesarias para lograr cierto grado de autonomía se encuentra la planeación de movimientos que se puede separar en cuatro problemas: primero la navegación, tarea que busca encontrar movimientos libres de colisiones que permitan ir de un punto X a un punto X'. Segundo la cobertura de la información necesaria del ambiente por medio de la correcta disposición de los sensores. En el tercer caso, la autolocalización del robot en una descripción del ambiente como un mapa en el que se quiere obtener las configuraciones de las poses en éste, teniendo como consideración que la pose de un robot es la orientación y posición relativa de éste a un sistema de coordenadas. Finalmente el cuarto problema a solucionar es el mapeo, que consiste de la exploración y sensado de un ambiente desconocido con el objetivo de obtener una representación de dicho ambiente que sea útil para algunas de las otras tareas del robot, este problema es conocido como SLAM (Simultaneous Localization And Mapping) [9].

El problema de la autolocalización puede ser visto en tres diferentes niveles:

- Localización global: Implica que un robot tenga la capacidad de ubicarse dentro de un ambiente previamente conocido (sin conocimiento sobre su posible ubicación).
- Rastreo de posición: Consiste en calcular la posición en la que se encuentra el robot en función de su(s) posición(es) previa(s), los comandos de control, su odometría y las mediciones (observaciones) realizadas durante su recorrido.
- Secuestro del robot: El robot es sustraído de su ambiente de trabajo y ubicado en una nueva posición sin haber realizado lectura alguna durante el

desplazamiento, por lo que su tarea será localizarse de nuevo.

En el amplio espectro de la robótica se tiene un área de gran crecimiento en los últimos años, la robótica de servicio, campo con tareas aun más complicadas a ejecutar en comparación con el área industrial la cual tiene tareas más repetitivas y definidas, donde los robots industriales son usualmente descritos por modelos matemáticos que trabajan con las dinámicas de estos y en algunos casos las dinámicas del entorno con el objetivo de lograr las tareas para las que se diseño, generalmente estas tareas son problemas de optimización matemática o procedimientos heurísticos, por el contrario tenemos que un robot de servicio autónomo está diseñado para realizar tareas de servicio sin intervención de las personas durante prolongados periodos de tiempo, con tareas menos repetitivas y ambientes menos controlados. Para lograr los objetivos el robot debe ser capaz de desempeñarse en entornos dinámicos y complejos, lo que nos lleva a tareas que los humanos consideramos triviales y de simple ejecución, contrario a la complejidad que representa para la ejecución de estas tareas para un robot autónomo y móvil, por lo que la complejidad del problema dificulta describir de manera adecuada las tareas del robot con modelos dinámicos, y se trabaja con la planeación de movimientos que desempeña un rol importante en el funcionamiento de los robots móviles actuales, permitiendo tener una respuesta a ciertas dinámicas tanto del robot como del entorno, logrando cierto grado de autonomía.

En la planeación de movimientos se tienen cuatro tareas principalmente; navegación, cobertura, localización y mapeo, estas dos últimas tareas han sido foco de interés para muchos grupos de investigación en los últimos años. La localización consiste en determinar la configuración de un robot dado un mapa y un conjunto de lecturas de los sensores. Por otro lado el mapeo, parte del desconocimiento del entorno, por lo que se realiza una exploración y sensado del ambiente con el fin de construir una representación como un mapa que sea útil para la ejecución de las demás tareas, este problema puede ser resuelto si simultáneamente se estima el estado del robot y se construye un modelo del entorno (mapa), dicha solución es el SLAM.

La comunidad que investiga y desarrolla algoritmos de SLAM ha venido progresando los últimos treinta años, obteniendo resultados de alta calidad, con aplicaciones en exteriores y admitiendo cierto grado de dinámicas, sin embargo dependiendo de las características del problema de SLAM como el robot y su tipo de movimiento, sensores disponibles y recursos computacionales, entorno estático o dinámico, planar o 3D, tipo de características, entre otros requerimientos de la solución como el acierto, la latencia y máximo tamaño de área de mapeo, estos

últimos son aspectos que permiten determinar el grado de madurez del problema del SLAM [10].

De acuerdo a la combinación de robot, entorno y rendimiento, se tiene un camino amplio de investigación por realizar; es el caso del enfoque del SLAM visual, el cual haciendo uso de técnicas de visión por computadora se extrae la información percibida del entorno por sistemas de visión como cámaras, en el que, generalmente esta información se obtiene de un conjunto de imágenes que son procesadas con métodos que permiten la construcción de descriptores de escenas que permitan filtrar en cierta medida información irrelevante, para el caso del SLAM a partir de esta información conocer la pose de plataformas a través de un análisis de cambios entre serie de imágenes y la construcción de un modelo consistente del ambiente.

Para este trabajo de investigación se plantea la implementación de un sistema de odometría visual que solucione el problema de cuerpo rígido a partir de técnicas de visión por computadora encontrando características que permitan describir puntos de interés y relacionando estos puntos o características en las imágenes RGB con sus respectivos valores espaciales en una nube de puntos obtenida con un sistema de coordenadas sobre el Kinect de Microsoft, el cual hace uso de la profundidad obtenida por luz estructurada para la formación de una escena tridimensional. Con la relación $2D$ a $3D$ de las características en distintos frames de la escena encontrar las transformaciones de la plataforma para realizar un posterior seguimiento y estimación de trayectorias con técnicas de filtrado. Los puntos característicos extraídos son usados de referencia con el fin de realizar una estimación de la pose de un robot, y de manera continua ir estimando la trayectoria y corrigiendo esta estimación.

Los resultados obtenidos muestran que es posible tener una reducción del error de pose relativa (RPE, Relative Pose Error), comparando el sistema sin corrección por filtrado, el cual tiene un RMSE=0.0541 metros con la trayectoria real del conjunto de datos trabajado, a un error de pose relativa con RMSE=0.0272 metros para el sistema con una corrección con el algoritmo NLMS, comparado nuevamente con la trayectoria real.

1.1. Hipótesis

Primordialmente se debe utilizar un dispositivo que permita adquirir información visual de un entorno de forma efectiva, práctica y confiable, considerando como principales inconvenientes encontrar adecuadamente los parámetros

intrínsecos del dispositivo, como son: las distorsiones de los lentes, las distancias focales y los cambios propios del uso continuo de los dispositivos que requieren calibración para una correcta relación entre las medidas del mundo y las que el dispositivo permite obtener.

Los sistemas de visión deben ser calibrados, de modo que su uso para configuraciones particulares permitan tener certeza en las estimaciones que con éste se realicen. Otro punto fuerte a enfrentar para el desarrollo de un sistema de odometría visual, es la adquisición de información relevante de una escena, aislando información dinámica, poco estable a cambios de iluminación y de baja frecuencia, ya que puntos característicos con baja certeza de ubicación y poca recurrencia no permiten un cálculo adecuado de transformaciones de rotación y traslación, que son los datos a encontrar en un sistema de odometría visual. Por lo que es de suma importancia identificar valores atípicos que introducen error en la información visual.

En términos de acierto, convergencia y costos computacionales, algunos métodos empleados en odometría para la identificación de puntos característicos en imágenes, es por puntos esparcidos distinguibles por medio de descriptores, que permiten hacer correspondencias, por lo que juegan un papel crucial para solucionar el problema de la estimación de pose del robot.

En términos generales se debe tener un nivel de abstracción (Front-End) de datos del sensor, que extraiga características relevantes y que por ende permita asociación de estas en correspondientes características de subsecuentes medidas, es un preprocesamiento dependiente del sensor. El otro nivel (Back-End) debe producir inferencias a partir de la abstracción de datos producidos por el primer nivel. Actualmente en la formulación de odometría visual se tiene un problema de estimación de máximo a posteriori (MAP), y es en el Back-end donde se realiza esta estimación de MAP que puede ser refinado por un proceso de optimización no lineal.

Considerando los módulos necesarios para el sistema de odometría visual dispersa y la naturaleza degenarativa del problema al ir acumulando los errores propios del sensor, así como los algoritmos de calibración y estimación de la rotación y traslación, conforme transcurre el tiempo de funcionamiento, es pertinente decir que un algoritmo de correción de trayectoria por gradiente estocástico adaptable permite reducir el grado de error de la trayectoria estimada por el sistema de odometría visual. Para validar esta hipótesis planteada se implementarón los módulos necesarios para un sistema de odometría visual y sobre estos se realizarón las comparaciones del sistema sin corrección de trayectoria contra el sistema con

corrección de trayectoria.

1.2. Motivación

La visión es el sentido que permite a algunas especies percibir y extraer información apropiada de su entorno físico, tener conocimiento de su propio ser en el mundo e interactuar con éste, lo que lo convierte en el sentido de exterocepción sobre el que los seres humanos más confían para la toma de decisiones. En un escenario donde un robot necesita desplazarse, es necesario conocer que información se puede inferir del entorno para que el robot logre navegar y tener habilidades de interacción, por lo que la necesidad de saber que información es útil, sobre que límites trabajar para no tener redundancia de conocimiento, diferenciar, reconocer y etiquetar datos, son problemas que aun no están resueltos, pero que están siendo fuertemente investigados desde la visión computacional.

Un robot móvil de servicio cuya principal tarea es la navegación, que además conlleva dos problemas característicos como lo es el conocimiento de su entorno y la localización, requiere un sensor que entregue información suficiente y adecuada para cumplir sus objetivos con razonable acierto. Algunos sensores tradicionales utilizados en la navegación como el sonar, infrarrojo, basados en láser y sensores de inercia, tienen limitaciones físicas y de costos, además adquieren menos información sobre el ambiente de lo que una cámara puede extraer, por lo que se ha hecho de las cámaras sistemas de sensado por visión ampliamente usados para abaratar costos, tener sistemas más robustos y flexibles.

Un robot autónomo móvil tiene la habilidad de ganar información de su entorno por medio de sistemas de sensores, trabajar por un extendido periodo de tiempo sin intervención humana, adaptarse a cambios de su entorno, adquirir nuevas capacidades que le permitan completar sus tareas, entre múltiples habilidades en las que se vienen trabajando en el campo científico. Considerando la clasificación de robot móvil se tiene que una de las principales capacidades que debe tener el robot es la navegación adecuada de su área de trabajo, que permitan al robot ir a destinos deseados, evitar obstáculos tanto estáticos como dinámicos, por lo que resulta imperativo percibir los escenarios con suficiente robustez y acierto. Para el caso de un robot basado en visión, una importante tarea para lograr en cierta medida estos aspectos son la extracción de la información pertinente que brinde estabilidad, rendimiento y robustez ante cambios físicos como oclusiones parciales, cambios de punto de vista, iluminación, escala y deformación, entre algunos retos difíciles y abiertos en el campo de la visión computacional. Un aspecto más particular pero sumamente importante para la navegación es la estimación de

la pose del robot, en donde la odometría visual ha ganado atención los últimos años, logrando estimar egomovimiento de un robot a partir de un análisis de los cambios de vista de una cámara.

Desde el 2011, en el Laboratorio de Biorobótica, ubicado en la Facultad de Ingeniería de la UNAM, se viene desarrollando el proyecto Justina (1.1), a cargo del Dr. Jesus Savage Carmona, el cual tiene como objetivo desarrollar un robot de servicio que ayude a las tareas del hogar. Entre las múltiples tareas que Justina ejecuta se encuentra la de navegación, por lo que es de interés contribuir en el proyecto realizando un sistema robusto que permita a Justina avanzar en su desarrollo tanto en la capacidad de navegar en su entorno de manera más acertada y con una mayor inferencia de su ambiente, como progresar en habilidades de interacción con el entorno desde la visión computacional.

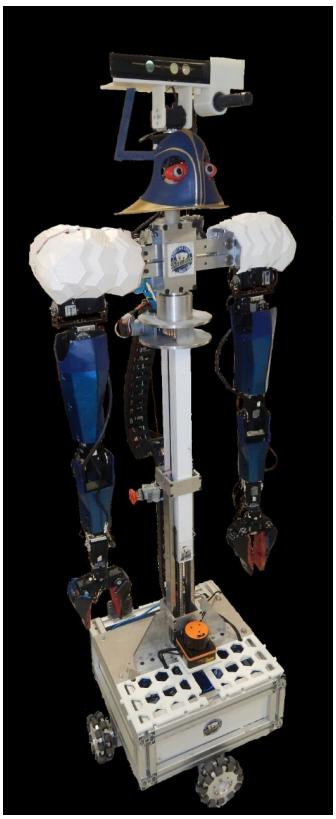


Figura 1.1. *Robot Justina del laboratorio de Biorobotica UNAM.*

1.3. Objetivos

1.3.0.1. Objetivo general

Diseñar e implementar un sistema de odometría visual para un robot móvil de servicio basado en información visual, utilizando técnicas de estimación y filtrado con construcción conjunta de un entorno disperso de puntos.

1.3.0.2. Objetivos específicos

- Extracción de puntos característicos para la descripción del entorno en el cual el robot opera.
- Evaluación de técnicas de correspondencia de puntos.
- Filtrado de valores atípicos para eliminar errores de coincidencia de características.
- Implementación de algoritmos de seguimiento y predicción de posición de características, así como la localización del robot.
- Construcción de una trayectoria consistente espacialmente.

1.4. Estructura de la tesis

Los capítulos posteriores a esta introducción, se desarrollan de la siguiente forma:

En el capítulo 2 se aborda el estado del arte referente a este trabajo, analizando la anatomía típica de un sistema SLAM y de odometría basado en visión para la estimación del estado de un robot y la construcción de un mapa a partir de extracción de características. El capítulo 3 abarca el fundamento teórico necesario para la realización de cada una de las etapas del sistema desarrollado, involucrando temas selectos de probabilidad, visión computacional y sistemas adaptables. Posteriormente se tiene en el capítulo 4 una descripción de cada uno de los módulos fundamentales del sistema implementado, el cual tiene una estructura de dos componentes esenciales que abarcan todos los módulos, front-end y back-end; en el front-end tenemos un nivel de abstracción de datos del sensor que extrae características relevantes y asociación de éstas en características de subsecuentes medidas, es un preprocesamiento dependiente del sensor. El back-end

produce inferencias de posición a partir de la abstracción de datos producidos por el front-end 1.2.

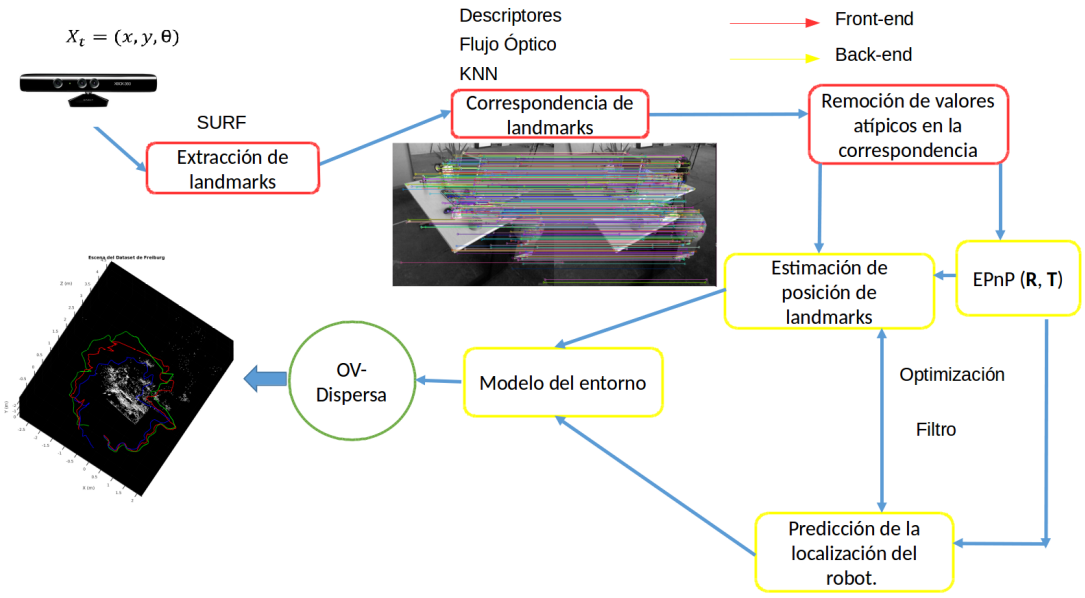


Figura 1.2. *Front-End y Back-End clásico de odometría visual.*

En el capítulo 5 se describe en términos de acierto, convergencia y costos computacionales los algoritmos implementados, y el desempeño final en conjunto para solucionar el problema de la estimación de pose del robot. Finalmente en el capítulo 6 se tienen las conclusiones y trabajos futuros, respectivamente.

Capítulo 2

Estado del arte

El desarrollo de sistemas de navegación visual en robots autónomos a través de ambientes desconocidos se ha convertido en un tema importante en la robótica debido a las múltiples ventajas que ofrecen los sensores basados en visión, los cuales han crecido en popularidad los últimos años, por ser una alternativa que reduce costos, permiten alto grado de inteligencia, flexibilidad y robustez, contrario a lo que sucede con los sistemas convencionales como son los sistemas basados en sensores ultrasónicos, de infra-rojo, o sensores láser, que tienen altos costos y limitantes físicas. Sistemas de visión como las cámaras RGB ofrecen alta resolución, amplio rango de sensado, nivel de acierto elevado, información robusta y beneficios inherentes al hardware como bajos costos, dimensiones y peso, permitiendo que los sistemas sean versátiles y generando,a su vez, una gran atención de investigadores, lo que propicia un rápido avance en el desarrollo de sistemas funcionales en tiempo real de odometría y SLAM visual.

En el caso del problema del SLAM visual, se busca estimar las poses del robot y una estructura de entorno desconocida dadas unas observaciones, dicho problema puede ser visto desde el enfoque de la robótica probabilística, donde la estimación de variables que no son directamente observables, pero que se pueden inferir a partir de los datos de sensores y que además están corruptas por ruido, define el núcleo central de la estimación de estados por métodos probabilísticos. La estimación de estados desde datos provenientes de un sensor, es un problema que puede verse de acuerdo a [11] como la estimación de la distribución de probabilidad conjunta (2.1):

$$p(x_{0:T}, m/z_{1:T}, u_{1:T}) \tag{2.1}$$

donde $x(0:T)$ son todas las poses del robot, m es el conjunto de todas las características, z son todas las observaciones de características y u son todos los controles de entrada. Por regla de Bayes y asunción de Markov es posible separar la distribución de probabilidad conjunta en la distribución condicional de probabilidad de la estimación de la pose del robot y en la distribución condicional de probabilidad de la configuración del mapa:

$$p(x_{0:T}, m/z_{1:T}, u_{1:T}) = p(x_{0:T}/z_{1:T}, u_{1:T}) \cdot (m/x_{0:T}, z_{1:T}) \tag{2.2}$$

Con el objetivo de conocer la distribución a posteriori para obtener la pose actual del robot se marginalizan las poses previas por integración, por lo que se tiene la ecuación 2.3

$$p(x_t/z_{1:t}, u_{1:t}) = \int_{x_0} ... \int_{x_{t-1}} p(x_{0:T}/z_{1:T}, u_{1:T}) dx_{t-1} ... dx_0 \tag{2.3}$$

De la anterior ecuación se llega a el conocido SLAM en linea:

$$p(x_t, m/z_{1:t}, u_{1:t}) = p(x_t/z_{1:t}, u_{1:t}) \cdot (m/x_{0:t}, z_{1:t}) \tag{2.4}$$

Para la solución del problema de estimación de la ecuación 2.4 se incorporan dos modelos, uno de movimiento y otro de observación. En el modelo de observación se tiene la distribución de probabilidad de la estimación de la pose del robot, usando poses previas conocidas y el control de entrada del robot.

$$p(x_t/x_{t-1}, u_t) \tag{2.5}$$

El modelo de observación es la función de distribución estimada de la medida z_t conociendo la pose estimada del robot x_t y la posición de las características

dentro del mapa, m.

$$p(z_t/x_t, m) \tag{2.6}$$

Finalmente el modelo gráfico que representa el enfoque anterior es la figura (2.1), en la cual se observa la secuencia de poses del robot, medidas del sensor, entradas de control, mapa del entorno y la relación entre éstas.

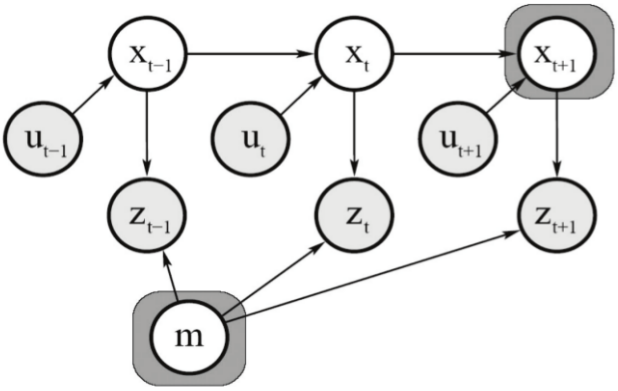


Figura 2.1. *Modelo de red Bayesiana del problema de SLAM* [1]

Para el caso de la navegación robótica basada en visión, técnica que guía a un robot móvil a lo largo de un camino a destinos deseados en un entorno evitando obstáculos estáticos y dinámicos, principalmente haciendo uso de sistemas encargados de obtener información del espectro electromagnético, son en su gran mayoría sistemas que manejan el rango del espectro conocido como el visible, donde a partir de la extracción de información desde una secuencia de imágenes del entorno con un sensor pasivo como una cámara RGB, es posible lograr sistemas de navegación robustos a cambios en la estructura de la escena e iluminación. Los algoritmos e implementaciones de estos sistemas tienen como fundamento teórico y de desarrollo la visión computacional, donde la tecnología VLSI (Very Large Scale Integration) ha permitido una significativa reducción de costos en los sistemas electrónicos, teniendo la visión computacional un vertiginoso desarrollo al aumentar la capacidad de los procesadores a un costo reducido.

Un sistema tradicional de visión de un robot autónomo móvil se compone de cinco principales subsistemas [12]:

- Mapa: El robot requiere de un modelo de conocimiento que represente el entorno con el fin de realizar tareas que requieran interactuar con su ambiente.
- Adquisición de datos: El sistema obtiene imágenes desde una cámara.
- Extracción de características: Este subsistema extrae características significantes a partir de las imágenes adquiridas. Las características pueden ser vértices, texturas y color, entre otras.
- Reconocimiento de características: El sistema busca posibles características que correspondan con características pre-almacenados o previamente observados bajo algún criterio establecido.
- Auto-localización: Calcula la posición actual del robot con base en una función de características detectadas y sus posiciones previas.

Por otro lado el problema de la navegación en robots móviles puede ser dividido en cuatro subproblemas [13]: Percepción del mundo, planificación, generación y seguimiento de trayectoria.

La percepción del mundo para cualquier robot autónomo móvil cuya principal tarea es la navegación comprende dos características principales: localización del robot y mapeo del entorno. Para solucionar el problema de la localización es menester conocer un mapa desde el cual el robot puede conocer su localización relativa y, para que el robot pueda extender el conocimiento de su mapa, debe conocer su actual localización relativa. Desde el campo de la visión por computadora esto se conoce como SLAM Visual, donde se tiene una secuencia de imagenes obtenidas desde una cámara en movimiento lo que permite reconstruir una escena (modelo del entorno) y estimar la pose de la cámara con respecto a un sistema de coordenadas que es definido sobre el entorno que se se va construyendo de manera incremenental.

El problema de SLAM Visual más investigado estima la pose de la cámara a partir de puntos característicos que deben ser identificables a lo largo del tiempo ante cambios de vista y condiciones de entorno variables. Estos puntos conocidos como características representan la información de la escena y en algunos casos conforman el mapa. Este método consiste en la extracción y correspondencia de puntos a través de frames donde la estimación conjunta del modelo del entorno y las poses de la cámara son un problema de optimización global, donde se busca

minimizar la distancia entre puntos predichos y observaciones.

Partiendo de una secuencia de imágenes se extrae y se hace correspondencia de puntos característicos para obtener información que permita estimar la pose de alguna plataforma analizando los cambios del entorno visto desde las imágenes, por lo que es posible determinar la posición de una cámara a partir de información acumulada de cuadros consecutivos, permitiendo realizar una localización relativa y además ir generando un mapa del entorno. Esta clase de SLAM se conoce como SLAM Visual disperso (Sparse).

2.1. SLAM visual

Los enfoques más comunes encontrados en la literatura para solucionar el problema del SLAM son los que utilizan características, basados en grillas o los topológicos. De acuerdo a la complejidad computacional, pensándolo en términos de asociación de datos y actualización de medidas, el enfoque basado en características es un enfoque muy ventajoso y ampliamente usado, donde el núcleo central de un SLAM de esta categoría es la extracción de características repetibles y robustas que permitan hacer correspondencias temporales, es decir de frame k a frame $k + t$, donde t habla de la elección de un frame posterior de acuerdo a algún tipo de decisión para considerar un nuevo frame pertinente para la extracción de información relevante y que permita contener en un margen de error la incertidumbre de las medidas, a esta selección de frames se le conoce en ingles como $keyframes$.

En [14] fue introducida la primera popuesta de SLAM donde se implementó un filtro de Kalman extendido (EKF). Este enfoque probabilístico limita el impacto de errores en las medidas adquiridas por los sensores, lo que incide directamente sobre el acierto en la creación del mapa. Desde esta primera implementación de SLAM con Kalman una gran cantidad de algoritmos nuevos ha sido implementada realizando mejoras en la reducción de error tanto de la localización como de la generación del modelo del entorno. Tiempo después del primer SLAM, en 2002 aparece uno de los algoritmos más reconocidos después de EKF-SLAM, el FastSLAM. Este algoritmo integra un filtro de partículas y un filtro extendido de kalman, lo que se convierte en un aumento en el acierto y en el costo computacional. Años después aparece la version siguiente llamada FastSLAM 2.0. A pesar de ser ampliamente estudiados, todos estos algoritmos tienen sus restricciones, el caso de Kalman extendido tiene la limitante de trabajar con una distribución unimodal, para el caso del filtro de partículas, a pesar de permitir trabajar dis-

tribuciones de probabilidad multimodal, tiene un costo computacional elevado a medida que el tamaño del entorno crece. Sin embargo los resultados obtenidos para estos algoritmos son bastante confiables.

Una parte importante del enfoque disperso en SLAM, es la extracción y correspondencia adecuada de las características, en [15] realizan un análisis de cual es el extractor de características más adecuado para usar en odometría visual monocular entre SIFT, SURF, ORB y A-kaze, realizando comparaciones de repetibilidad, robustez a cambios de escala y rotación. Los resultados de estos autores permiten discernir la relación costo beneficio, dictando el algoritmo SURF como el más adecuado, siendo similar al SIFT que es el extractor que encuentra la mayor cantidad de características, pero con la ventaja de ser más liviano computacionalmente, esto pesándolo en la oportunidad de una implementación en tiempo real. Posterior a la extracción de las características se encuentra la correspondencia o asociación de datos, la cual con un solo error de correspondencia puede llevar a que la solución de SLAM diverja. En [2] realizan una implementación de un modelo probabilístico de remoción de outliers con modelo RANSAC donde trabajan con vectores asociados a cada característica, de acuerdo a estos modelos y la matriz de covarianza del error se rechazan los posibles valores atípicos, dejando las correspondencias "*correctas*" como se puede observar en la figura 2.2.

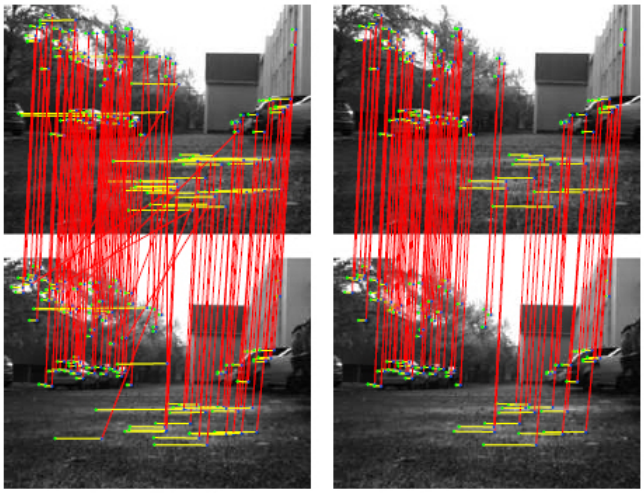


Figura 2.2. *Escenas para comparación de correspondencia sin y con remoción de valores atípicos, donde las imágenes superiores son la misma escena consideradas en un tiempo* $k-1$ *y las imágenes inferiores son la misma escena en un tiempo* k*. Imagen izquierda superior e izquierda inferior: correspondencias en rojo y su disparidad en amarillo, donde se observa correspondencias erróneas, debido a la ausencia de la eliminación de valores atípicos. Imagen derecha superior y derecha inferior: correspondencias con sistema de eliminación de valores atípicos.* [2]

La construcción adecuada del modelo del entorno va ligada con la correcta estimación de la pose, que a su vez depende de la correcta estimación del mapa, algunos algoritmos de SLAM trabajan refinando el mapa por medio de minimización de errores de distancia, calculando funciones de verosimilitud que encuentran la mejor ubicación de las características, o alineando las nubes de puntos por métodos iterativos, generalmente métodos de alto costo computacional.

En breves palabras, SLAM visual busca construir una representación del entorno a partir de la estimación del movimiento propio y el cierre de ciclos, este último es el hecho de reconocer correctamente que un vehículo o plataforma ha retornado a una locación previamente visitada. Sin este módulo de reconocimiento de lugares el sistema se reduce a un problema de odometría visual. En [16] y [17] se tienen los primeros sistemas en tiempo real, donde la idea de las técnicas usadas de estructura y movimiento a partir de un conjunto de características representativas en la escena se realizan un seguimiento de ellas a través de frames sucesivos y así se estima la ubicación en el espacio de coordenadas del mundo (3D) y el movimiento de la cámara en este sistema de referencia.

2.2. Odometría visual

Encontrar las poses relativas de una cámara y la estructura tridimensional de una escena desde un conjunto de frames es conocido en el campo de la visión computacional como estructura del movimiento (SfM: Structure from Motion) [18]. Por este método es posible obtener representaciones en tres dimensiones de escenas proyectadas en imágenes de dos dimensiones desde diferentes puntos de vista como la figura 2.3, en donde por lo general las poses de la cámara estimada y la estructura $3D$ inferida son refinadas con algún algoritmo de optimización. La odometría visual es un caso particular de SfM en donde se busca estimar el movimiento de la cámara en el mundo a partir de frames secuenciales y en tiempo real. El término de odometría visual fue acuñado por *Nister* en [17] y es debido a su similitud con la odometría por encoders en ruedas que estiman el movimiento por el número de giros en el tiempo. A partir del trabajo realizado por *Nister* surgieron sistemas de navegación visual aprovechando la ventaja ante terrenos lisos, morfológicamente accidentados y desiguales, en los cuales giros de las ruedas sin desplazamiento real inducirían errores de gran magnitud en la odometría clásica con encoders.

Inferir la posición y orientación de una plataforma a partir de una cámara

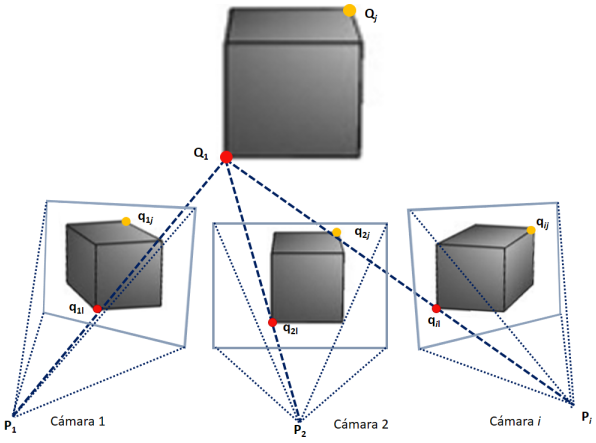


Figura 2.3. *Poses de una cámara y estructura 3D de una escena (SfM).*

fue una idea propuesta por [19], teniendo como objetivo estimar la posición con 6 grados de libertad para rovers planetarios en la exploración de Marte por parte de la NASA usando configuración estereoscópica a partir de un análisis de imágenes encontrando características esparcidas, más exactamente detectando esquinas, realizando correspondencias y eliminando características atípicas, para finalmente con las características aptas, calcular el movimiento a partir de la transformación de cuerpo rígido. Donde la transformación de cuerpo rígido esta definida por $g : R^3 \leftarrow R^3$ si y solo si cumple las siguientes propiedades:

- La distancia entre los puntos es constante:

$$||g(p) - g(q)|| = ||p - q|| \quad \forall\, p, q \in R^3 \tag{2.7}$$

- El producto cruz es preservado:

$$g(v \times w) = g(v) \times g(w) \quad \forall\, v, w \in R^3 \tag{2.8}$$

- El producto punto es preservado:

$$v_1^T v_2 = g(v_1)^T g(v_2) \quad \forall\, v_1, v_2 \in R^3 \tag{2.9}$$

Estas propiedades permiten inferir que la transformación g mantiene la ortogonalidad entre vectores por lo que se intuye que las transformaciones de cuerpo rígidos toman un sistema ortogonal y lo mapean a otro sistema ortogonal. Por lo tanto para realizar un seguimiento del movimiento por medio de cálculo de transformación de un cuerpo rígido se ubica un sistema de coordenadas cartesiano fijo

en cualquier punto del espacio o mundo donde se encuentra un cuerpo rígido, que a su vez tiene su propio sistema cartesiano (cámara), y una vez definidos los sistemas de coordenadas, se determina la configuración de los cuerpos rígidos al estudiar el movimiento relativo entre el sistema de coordenadas de un cuerpo y un sistema de coordenadas fijo y así obtener una estimación de la pose entre sistemas coordenados (R, t) (2.4).

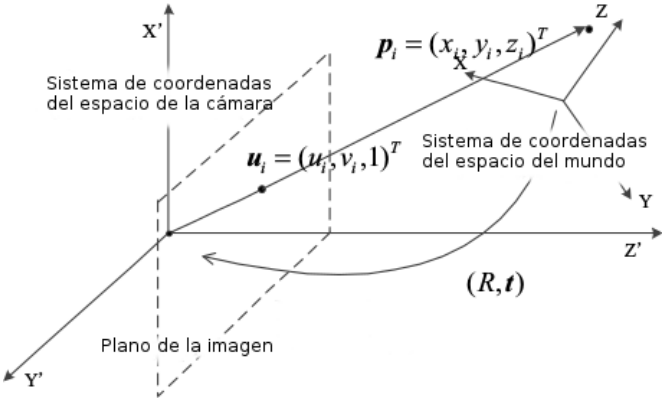


Figura 2.4. *Sistemas de coordenadas en un problema de estimación de pose con transformación de cuerpo rígido.* [3]

Dados unos puntos en el mundo y sus proyecciones sobre el plano de una imagen en una cámara, es posible estimar la posición de la cámara aplicando la transformación de cuerpo rígido con la parametrización de la rotación R y traslación t con respecto a un sistema de coordenadas. R y t son parámetros que se pueden estimar a partir de unas relaciones entre puntos en el espacio del mundo y puntos de proyección en el plano de la imagen, estas relaciones son obtenidas desde el modelo de cámara que asume el sistema de proyección "*pinhole*". Para la estimación de movimiento en sistemas de odometría visual, se calcula el movimiento entre la imagen actual y una imagen previa, y por concatenación de todas las poses con respecto a un sistema de referencia absoluto se obtiene la trayectoria completa (2.5). Se tiene entonces que la estimación de la pose de una cámara por características esparcidas a través de frames sucesivos involucra la estimación de la matriz R de rotación y el vector t de traslación a través de la detección y correspondencia de características que se encuentren en áreas comunes entre frames consecutivos, básicamente el principio fundamental de la odometría visual.

Obtener la pose desde tres correspondencias es la mínima cantidad de información necesaria y es conocido como problema de perspectiva de 3 puntos (perspective-3-point-problem, P3P). Teniendo una extension de más correspon-

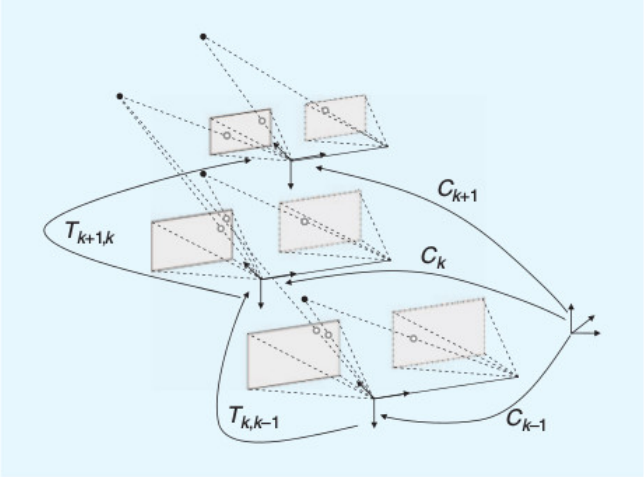


Figura 2.5. *Poses relativas entre sistemas coordenados de cámaras, y concatenación entre poses para obtener una pose absoluta respecto a un sistema absoluto global.* [4]

dencias se llega al problema de perspectiva desde n puntos (perspective-n-point problem, PnP) [20] [21] [22]. Este problema puede ser solucionado por métodos iterativos o no iterativos. En términos generales los métodos iterativos son más acertados y más lentos que los métodos no iterativos.

La solución más directa es el algoritmo de los 8 puntos [23] que soluciona un conjunto de ecuaciones lineales a partir de la correspondencia de 8 pares de puntos, si se tienen más de 8 puntos correspondientes se pasa a un proceso de optimización de una función de costo.

Múltiples trabajos han sido implementados en sistemas en tiempo real con bajo retraso, para propósitos de navegación, donde la gran mayoría de investigaciones pertenecen a la clase de algoritmos de estéreo visión desde los cuales se puede obtener la posición relativa de las características directamente por triangulación.

Investigaciones posteriores a las de [19] fueron desarrolladas en las cuales se diferenciaba la selección de puntos característicos definiendo matrices de covarianza del error de los puntos de la imagen y usando RANSAC para la eliminación de valores atípicos en la estimación de movimiento por mínimos cuadrados [24] [25] [26]. Estos trabajos tienen en común que el movimiento relativo es calculado desde puntos característicos $3D-3D$ de una imagen a otra, hasta el enfoque propuesto por [17] el cual calculó el movimiento de $3D-2D$ con mejores resultados en la estimación del movimiento. Recientemente se han desarrollado métodos [27]

[28] que buscan calcular geometría y movimiento directamente desde las imágenes omitiendo la búsqueda y correspondencia de puntos característicos.

2.3. Extracción y descripción de características

En la visión computacional se tienen dos campos separados para el procesamiento y abstracción de la información de las imágenes, los basados en apariencia que trabajan directamente sobre las intensidades de los pixeles y/o gradientes [29], lo que consecuentemente acarrea un alto costo computacional, comparado con el otro método que se basa en técnicas que abstraen información de las imágenes por identificación de regiones con información distinguible, que permite obtener características representativas.

Para el método de características esparcidas representativas se encuentran dos enfoques para encontrar los puntos característicos y sus correspondencias. El primer enfoque consiste en realizar seguimiento a las características encontradas a través de los frames realizando búsquedas locales en la imagen como correlación. El segundo enfoque se basa en la obtención de una posición central y algún vector descriptor que represente la región para realizar una correspondencia basada en alguna medida métrica entre vectores descriptores.

El proceso de detección de características consiste en buscar puntos clave (keypoints) en la imagen, también llamados puntos de interés o puntos característicos locales. Un punto de interés es un patrón en la imagen que difiere con respecto a sus vecinos inmediatos en términos de intensidad, color y textura, estos puntos tienen una alta probabilidad de ser encontrados nuevamente en imágenes siguientes. Varios algoritmos conocidos como algoritmos detectores realizan la tarea de determinar dichos puntos, usualmente se basan en buscar esquinas o manchas en la imagen.

Las propiedades que se desean en un detector de puntos característicos son:

- Precisión en la localización: los puntos detectados deben ser localizados precisamente en la imagen como en la escala. La precisión es especialmente importante en el paso de reconstrucción $3D$.
- Cantidad de puntos: el número ideal de puntos encontrados depende de la aplicación. Para el caso de la odometría visual es importante una gran

cantidad de puntos para tener una buena precisión durante la reconstrucción $3D$ y la estimación en la orientación de la cámara.

- Invarianza: las características encontradas deben ser invariantes a cambios de iluminación del ambiente, de escala y de perspectiva.
- Eficiencia computacional: es deseable que el tiempo de ejecución para la detección de características sea eficiente. En robótica, la mayoría de las aplicaciones se deben ejecutar en tiempo real. Sin embargo, el tiempo de ejecución está relacionado fuertemente con la invarianza deseada, entre mayor sea el nivel de invarianza, mayor es el tiempo de ejecución.
- Robustez: los puntos detectados deben ser robustos en términos de ruido, efectos de la discretización, procesos de compresión, invarianza a cambios fotométricos (iluminación) y cambios geométricos (rotación, escala y distorsión en la perspectiva).

Los algoritmos para identificación de características en imágenes por zonas constan de dos etapas, la primera etapa se encarga de encontrar las características claves, etapa en la cual generalmente se aplican operadores locales en una o múltiples escalas, con variaciones de tipo rotacional entre otras, que permitan obtener regiones con información estructural de la escena, para posteriormente en el centro de estas regiones ubicar los llamados puntos claves. La segunda etapa se encarga de extraer vectores que contengan características que permitan codificar la información de la región de ubicación del punto clave o característica permitiendo obtener descriptores que identifiquen y posibiliten encontrarlos nuevamente ante cambios de la escena como traslación, rotación, iluminación entre otros.

Tres algoritmos usados comúnmente en la solución del problema de estimación de pose son el SIFT, SURF y ORB [30] [5] [31] [32], este ultimo detector de un impacto dramático en la reducción del costo computacional, al tener como descriptor un vector binario. En [15] se encuentra que para un sistema de odometría visual el algoritmo de SURF muestra el mejor acierto. En [7] muestran una comparación de rendimiento y propiedades de diferentes detectores 2.1.

2.3.1. SURF

El SURF menos costoso computacionalmente es un derivativo del detector SIFT, por eficiencia computacional y acierto es un detector basado en el determinante del hessiano de la matriz [5], donde dado un punto $\mathbf{x} = (x, y)$ en una

Tabla 2.1. *Comparación entre detectores de puntos característicos [7].*

	Detector esquinas	Detector zonas	Invarianza rotación	Invarianza escala	Invarianza afín	Repetibilidad	Acierto localización	Robustez	Eficiencia
FAST	x		x	x		**	**	**	****
Shi-Tomasi	x		x			***	***	**	**
Harris	x		x			***	***	**	**
SIFT		x	x	x	x	***	**	***	*
SURF		x	x	x	x	***	**	**	**
CENSURE		x	x	x	x	***	**	***	***
ORB	x		x	x		*	**	**	****
A-KAZE		x	x	x	*	*	**	**	***

imagen I se define la matrix hessiana $H(\mathbf{x}, \sigma)$ siendo sigma una escala en la que el punto $\mathbf{x}$ se encuentra. Se define $H(\mathbf{x}, \sigma)$ como:

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{yx}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \tag{2.10}$$

Donde $L_{xx}(\mathbf{x}, \sigma)$ es una convolución de la derivada de la gaussiana de segundo orden $\frac{\partial^2}{\partial\, x^2} g(sigma)$ con el punto $\mathbf{x}$ en la imagen I, de manera similar con los demás valores de la matriz, se obtiene:

$$H(x, y, \sigma) = \begin{bmatrix} \frac{\partial^2}{\partial\, x^2} g(sigma) I(x, y) & \frac{\partial}{\partial\, x}\frac{\partial}{\partial\, y} g(sigma) I(x, y) \\ \frac{\partial}{\partial\, x}\frac{\partial}{\partial\, y} g(sigma) I(x, y) & \frac{\partial^2}{\partial\, y^2} g(sigma) I(x, y) \end{bmatrix} \tag{2.11}$$

Los autores de [5] mencionan lo óptimo de los gaussianos para el análisis del espacio en escala, pero debido a la necesidad de discretizar y recortar se producen problemas de aliasing tan pronto se lleva la imagen I a un proceso de sub-muestreo, por lo que optan realizar aproximaciones del modelo por gaussianos usando cajas de filtros 2.6, esta aproximación permite por filtros, una rápida evaluación por imágenes integrales, minimizando el numero de operaciones por realizar convoluciones con cajas de filtros independiente del tamaño.

Para la extracción del vector descriptor parten de una región cuadrada centrada y orientada en el punto de interés encontrado previamente. Para cada región se subdivide y se calcula la respuestas ante una *Haarwavelet* horizontal y vertical.

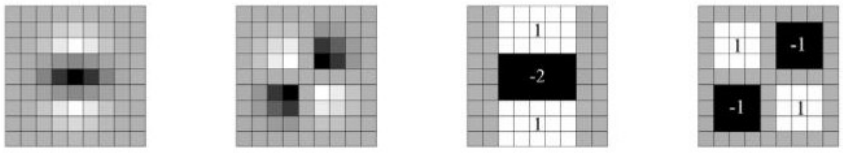


Figura 2.6. *Gaussianas de segundo orden y aproximaciones por filtros [5].*

En el proceso final de obtención de la información ante las *wavelets* se obtiene un vector descriptor de 64 elementos.

Las ventajas de este descriptor ante rotación, escalamiento, iluminación y bajo costo computacional han sido ampliamente evaluadas; acierto, repetibilidad y eficiencia, en diversas tareas como: detección de objetos, reconocimiento de objetos y reconstrucción de modelos $3D$, entre otras aplicaciones en visión, lo convierten en un detector adecuado para una implementación en tiempo real para estimación de trayectorias en robots autónomos móviles.

Capítulo 3

Marco teórico

En este capítulo se hará una revisión de la notación y fundamento matemático para el modelo de obtención de una imagen y de la estimación y método de obtención de información $3D$ de una escena desde una configuración de cámara RGB-D. Se proveerá la transformación de cuerpo rígido y el fundamento del algoritmo de obtención de la matriz de transformación y el vector de traslación basado en la minimización de una función de costo de error de retroproyección con características $3D$ a $2D$ con una mejora iterativa a la estimación de la pose. Finalmente se esbozara la teoría del algoritmo de LMS (Least-Mean-Square) y LMS-SER (LMS con Regresión secuencial) usados para corrección de la trayectoria.

3.1. Modelo de perspectiva de imagen

El proceso físico de obtención de una imagen del mundo real ocurre a partir de la interacción de la luz con la escena, donde ocurren fenómenos físicos como reflexión, refracción, dispersión y difracción, que dependiendo de las propiedades de los materiales de los elementos en la escena va a variar la contribución de la luz que entre en la cámara.

Una representación sencilla pero funcional, consiste en una cámara sin lentes con una apertura muy pequeña por donde la luz de la escena pasa y proyecta la imagen de manera invertida. El modelo matemático de este proceso se ha desarrollado para el campo de la visión computacional, donde se hace una consideración de imagen virtual ubicada en frente del "*agujero*"(*pinhole*), el plano que contie-

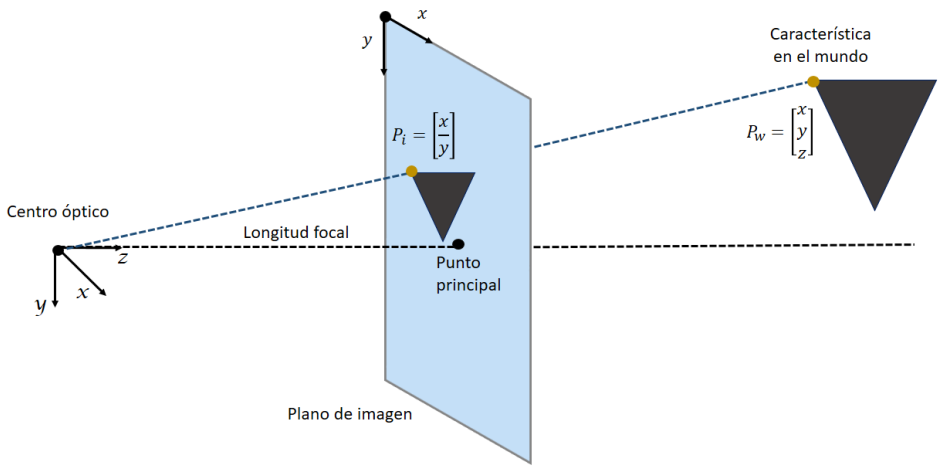


Figura 3.1. *Terminología del modelo de cámara pinhole.*

ne la imagen virtual es llamado plano de imagen o plano de proyección, el cual es desplazado del centro óptico (*pinhole*) una distancia w en el eje óptico, y la distancia entre el centro óptico y el punto de intersección del plano de proyección a lo largo del eje óptico es conocido como distancia focal 3.1.

La posición $\mathbf{P_i} = [x, y]^T$ en $2D$ correspondiente al punto $\mathbf{P_w} = [u, v, w]^T$ en $3D$ se puede encontrar a partir de la conexión de un rayo desde el centro óptico hasta el punto $\mathbf{P_w}$, siendo la posición de $\mathbf{P_x}$ la intersección de este rayo con el plano de proyección, este proceso es conocido como proyección de perspectiva.

Una vez se tienen los puntos $3D$ proyectados con el modelo *pinhole* se debe transformar el resultado obtenido en coordenadas de acuerdo al plano de la imagen, y en coordenadas del sensor, que vienen siendo *pixeles*.

Por defectos de fabricación el centro óptico no está exactamente en el centro del chip de la cámara, este tiene un desplazamiento por lo que se suelen introducir dos parámetros; C_x y C_y, que permiten considerar estos errores en el modelo. Para el caso de la forma individual de cada pixel que es más rectangular que cuadrada, se manejan dos distancias focales, una para el eje vertical f_y y otra para el horizontal f_x, donde cada una de estas longitudes focales es igual al producto de la distancia focal física del lente f por el tamaño de un elemento individual de la imagen s_x y s_y, para f_x para f_y respectivamente, por lo tanto el modelo resultante que nos permite obtener una posición en el plano de la imagen de un punto del

espacio del mundo cuyas coordenadas son (X, Y, Z) tiene las siguientes ecuaciones:

$$\mathbf{x_p} = f_x \cdot \frac{X}{Z} + C_x \tag{3.1}$$

$$\mathbf{y_p} = f_y \cdot \frac{Y}{Z} + C_y \tag{3.2}$$

Los parámetros intrínsecos f_x, f_y, C_x y C_y junto con los coeficientes de distorsiones radiales debidas a los lentes pueden ser obtenidos desde un procedimiento estándar de calibración, estas aberraciones y problemas inherentes a los lentes son removidos antes de proceder al procesamiento de las imágenes por el modelo de pinhole, lo que hace el modelo funcional y ampliamente usado.

3.2. Cámaras RGB-D

Las cámaras RGB-D son dispositivos digitales que proveen información de color y profundidad para cada pixel en la imagen, esta información de profundidad se obtiene desde distintas técnicas de estéreo visión.

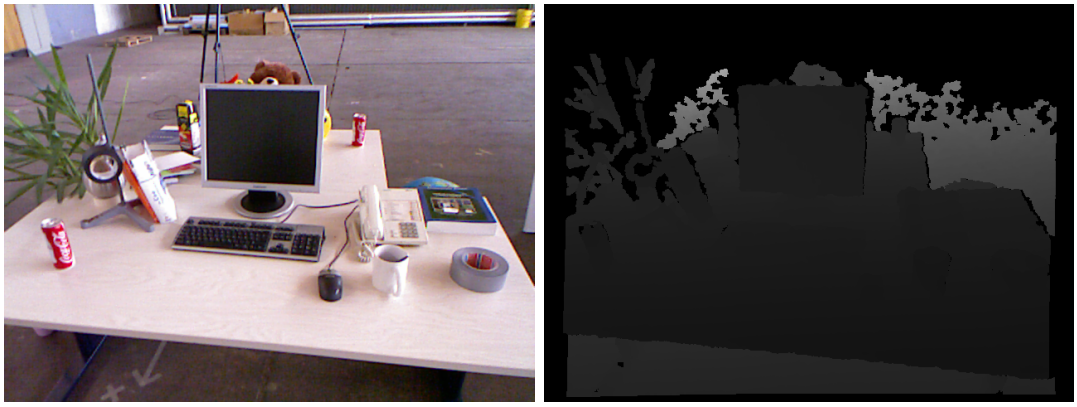
Una cámara RGB-D se basa en tecnología de proyección desarrollada por Prime-Sense como los casos del Kinect de Microsoft en la figura 3.2 y el ASUS Xtion PRO LIVE. Estos dispositivos han sido ampliamente usados en odometría visual ya que son baratos, con bajo consumo de potencia y poseen la ventaja añadida de realizar el cálculo de la profundidad directamente en el dispositivo.

Estos dispositivos generalmente tienen dos cámaras, una RGB y la otra infrarroja, además de un proyector que emite un patrón infrarojo que es capturado por la cámara infrarroja, de esta manera conociendo el patrón de emisión y el patrón obtenido se realiza la estimación de la profundidad utilizando la técnica de proyección de luz estructurada, que detecta cambios que se generan en el patrón emitido debido a las superficies de los objetos en los que incide. Analizando este cambio, se puede obtener una imagen de la profundidad de los objetos al plano de la cámara. Además de los componentes que permiten construir las imágenes RGB-D, los cuales son una cámara de color, un emisor infrarrojo y un sensor infrarrojo, el sensor kinect tiene incorporado un arreglo de 4 micrófonos y un motor que le permite girar con un ángulo de cabeceo en un rango de 54 grados.

Figura 3.2. *Sensor Kinect de Microsoft.*

El sensor Kinect fue desarrollado por la empresa Microsoft y puesto a disposición del público en noviembre del 2010, como dispositivo enfocado principalmente a la interacción natural en videojuegos, con el objetivo de estimar poses y movimientos del cuerpo de los usuarios para su video-consola.

Una vez obtenida del sistema emisor-receptor infrarrojo la información de profundidad, se relaciona con la información obtenida por la cámara RGB para finalmente obtener imágenes de 4 canales, tres con la información de color (Red, Green, Blue) y uno con información de profundidad (Depth) 3.3. Con la información de profundidad y los parámetros de la cámara, se puede calcular la posición tridimensional de cada pixel de la imagen con el origen referenciado al centro del sensor de la cámara, lo que nos permite tener una nube de puntos que describe de manera discreta el entorno registrado como se ve en la figura 3.4.



(a) ***Imagen de color (RGB).*** (b) ***Imagen de profundidad (D).***

Figura 3.3. *Imágenes obtenidas del Kinect [6].*

La imagen de profundidad que genera el sensor Kinect comúnmente presenta datos sin información, es decir, pixeles en donde el valor de profunidad es 0, los cuales representan error en la lectura. Existen diversas fuentes para este tipo de

Figura 3.4. *Nube de puntos obtenida del Kinect [6].*

errores. A continuación se describen algunas de las más importantes [33]:

- Condiciones de luz: en escenas con luz muy intensa, se presentan errores en la deteccion del patrón de puntos proyectado, lo que da origen a regiones sin información o errores en la medición.
- Distancia a los objetos: cuando los objetos se encuentran fuera del rango de medición del sensor, no se obtendrá el valor de profundiad de estos. En caso de que la distancia se encuentre fuera del rango, pero se tenga lectura, esta será muy ruidosa.
- Distancia entre el sensor y el emisor IR: dado que ambos, emisor y receptor se encuentran físicamente separados por una distancia considerable en el dispositivo, algunas partes de la escena pueden aparecer ocluidas o con sombras. Esto se debe a que, dependiendo de la escena, un objeto puede estar siendo proyectado por el emisor IR, pero no detectado por el sensor IR, lo que se traduce en un problema con regiones sin información.
- Propiedades y orientación de las superficies de los objetos: las propiedades reflectantes de las superficies de los objetos y su ángulo respecto al plano de la imagen pueden generar errores grandes en la medición e inclusive generar regiones sin información. Esto pasa comúnmente con materiales como metales, cerámicas o plásticos transparentes.

Aún con las limitaciones propias del Kinect éste ha sido ampliamente usado por un amplio sector de la comunidad científica, al tener bajo costo comparado

con otros sensores que proveen información similar, como pueden ser los escáneres láser. Debido a la rápida adopción por investigadores en el área de la robótica, se ha vuelto común ver robots que usan este tipo de sensor para desempeñar diversas tareas como la localización y mapeo simultáneo [34].

3.3. Problema de estimación de pose

La estimación de la pose de una cámara o un agente robótico móvil con sistema de visión óptico desde un enfoque de puntos característicos se fundamenta en la estimación de la matriz de transformación de cuerpo rígido compuesta por una matriz de rotación R y un vector de traslación t. El movimiento de un cuerpo rígido busca describir la posición y orientación relativa de un marco de referencia que va ligado al cuerpo rígido con respecto a un marco de referencia global (2.4).

Partiendo de un sistema de referencia absoluto y estático desde el cual se mide un punto estático, que en el caso de odometría visual es llamado característica (en ingles, landmark), se puede estimar la transformación que relacione este punto con otro sistema de referencia relativo con la siguiente expresión:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{3.3}$$

Si el agente móvil o la cámara relacionada al sistema de referencia definido por (x_r, y_r, z_r) cambia de posición respecto al sistema de referencia absoluto descrito por (x_w, y_w, z_w) en un intervalo de tiempo discreto definido por $[(k-1)T_s, kT_s]$ donde $k-1$ y k son el número de muestra en distinto tiempo, y T_s es el periodo de muestreo de dichas muestras, de estas definiciones tenemos que:

$$\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \tag{3.4}$$

Por tanto se puede inferir que la matriz $T_{k,k-1}$ puede ser estimada a partir del conocimiento de las coordenadas $3D$ en el espacio y $2D$ en la imagen de las características en los instantes $k-1$ y k asociadas tanto al sistema de referencia

absoluto como al relativo. En [7] se definen distintas metodologías para la estimación de la matriz de cuerpo rígido $T_{k,k-1}$; los métodos de puntos en el espacio $3D$ a $3D$ con una alta incertidumbre y propenso al error de estimación por problemas relacionados a los errores de estimación de las coordenadas de los puntos en el espacio. $2D$ a $2D$ y el más acertado $3D$ a $2D$.

3.3.1. Algoritmo PnP con ajuste iterativo para problema de estimación de pose

Una mejora iterativa al algoritmo EPNP relacionada al conocimiento de la profundidad de las caraterísticas, para estimación de pose es propuesta en [3], donde parten de la formulación del problema definiendo la relación entre coordenadas de puntos vistos desde un sistema de coordenadas de la cámara y las coordenadas de los mismos en relación a las coordenadas del mundo 3.5.

$$\mathbf{q}_i = R\mathbf{p}_i + \mathbf{t}, R \in SO(3), \mathbf{t} \in \mathbf{R}^3 \tag{3.5}$$

En donde $\mathbf{q}_i = (x_i^q, y_i^q, z_i^q)^T$ conjunto de coordenadas de puntos nocolineales expresados en coordenadas respecto de cámara. $\mathbf{p}_i = (x_i, y_i, z_i)^T$ sus correspondientes puntos expresados en el espacio de coordenadas del mundo con $i = 1, ..., n,\ n \geq 3$.

Por definición de coordenadas homogéneas se tiene $\mathbf{u}_i = (u_i, v_i, 1)^T$ para la proyección sobre el plano de la imagen de $\mathbf{p}_i$, a lo que se tiene:

$$w_i\mathbf{u}_i = K\mathbf{q}_i \tag{3.6}$$

Siendo K la matriz de parámetros intrínsecos de la cámara y w_i un valor escalar que denota la profundidad del punto característico en coordenadas de la cámara. Tomando de 3.5 se puede reformular 3.6 como:

$$w_i\mathbf{u}_i = K(R\mathbf{p}_i + \mathbf{t}),\ i = 1, ..., n \tag{3.7}$$

Una posterior normalización de lo puntos en el plano de la imagen se obtiene con:

$$\mathbf{m}_i = K^{-}1\mathbf{u}_i \tag{3.8}$$

A lo que combinando 3.8 con 3.7 tenemos:

$$w_i\mathbf{m}_i = (R\mathbf{p}_i + \mathbf{t}), i = 1, ..., n \tag{3.9}$$

De forma matricial:

$$\begin{bmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{bmatrix} \begin{bmatrix} \mathbf{m}_1^T \\ \vdots \\ \mathbf{m}_n^T \end{bmatrix} R + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \mathbf{T} = \begin{bmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_n^T \end{bmatrix} \tag{3.10}$$

Con $T = -\mathbf{t}^T R$ y $W = diag(w_1, \cdots, w_n)$ la matriz con los valores de profundidad de las características y $\mathbf{P}$ las coordenadas de las características en el espacio del mundo. Finalmente se tiene la ecuación 3.11 que asumiendo W conocida se puede obtener R y $\mathbf{T}$ por el método tradicional de descomposición en valores singulares (Single Value Decompositio, SVD) minimizando 3.12.

$$\mathbf{A}R + \mathbf{1}\mathbf{T} = \mathbf{P} \tag{3.11}$$

$$min_{R,\mathbf{T}} ||\mathbf{A}R + \mathbf{1}\mathbf{T} - \mathbf{P}||_F^2, R^T R = \mathbf{I} \tag{3.12}$$

Con R y $\mathbf{T}$ obtenidas por la solución de mínimos cuadrados los autores de [3] proponen el refinamiento de los valores por medio de iteraciones con una nueva matriz W que puede ser calculada con los parámetros de la pose estimada en el paso anterior hasta una convergencia razonable. A continuación los pasos del

algoritmo:

- Paso 1: Con un primer $W = W^{(0)}$ y asumiendo la kth iteración de W como $W^{(k)}$ donde $W^{(0)}$ puede ser una matriz identidad o cualquier estimación de parámetros de profundidad.
- Paso 2: Calcular $R^{(k)}$ y $\mathbf{T}^{(k)}$ con SVD.
- Paso 3: Calcular $\mathbf{t}^{(k)}$ desde $\mathbf{T}^{(k)} = (-\mathbf{t}^{(k)})^T R$.
- Paso 4: Calcular $W^(k+1) = diag(\mathbf{M}R^{(k)}(\mathbf{P}^T - (\mathbf{T}^{(k)})^T\mathbf{1}^T))diag(\mathbf{M}\mathbf{M}^T)^-1$.
- Paso 5: Finalizar las iteraciones si se tiene una convergencia razonable, en otro caso iterar desde el paso hasta el paso 4.
- Paso 6: Obtener la pose R y $\mathbf{t}$.

3.4. Sistemas adaptativos

Los sistemas adaptativos tienen una estructura cuyas propiedades pueden ser alteradas o ajustadas de acuerdo a algún criterio deseado, estos ajustes permiten que el rendimiento de estos sistemas mejore a través del funcionamiento con el entorno donde desempeñan la tarea para la cual son diseñados.

Las propiedades más generales de los sistemas adaptativos es su varianza en el tiempo y su desempeño auto-ajustable. Lo que da ventajas en instancias en las cuales un rango de condiciones de entradas, o los valores estadisticos no puedan ser conocidos. En circunstancias así, un sistema adaptable buscará el óptimo usando busquedas que minimicen alguna métrica dando rendimientos superiores comparado con sistemas fijos.

Una categoría de los sistemas adaptativos es que sean lineales o no lineales, en el caso que nos compete, los sistemas lineales adaptables son muy usados por su gran desempeño y tratabilidad matematica, lo que los hace generalmente fáciles de diseñar e implementar. La estructura no recursiva de una combinación lineal es en esencia un filtro digital y variante en el tiempo, variando su aplicabildad de acuerdo a su arquitectura y límites de desempeño.

Uno de los esquemas más simples para ajuste de pesos en una combinación lineal es el algoritmo LMS (Least-Mean-Square), el cual ha sido ampliamente

aplicado a diferentes tipos de sistemas adpatativos, por su simplicidad, costo computacional y eficiencia, además no requiere estimaciones de gradiente previas a su funcionamiento o repetición de datos.

3.4.1. Algoritmo LMS

El algoritmo LMS desarrollado por Widrow and Hoff en 1959 como parte de su investigación es un algoritmo simple y efectivo que nace para el diseño de filtros adaptativos transversales. Es un algoritmo de gradiente estocástico lo que lo distingue del algoritmo del método de máxima pendiente que usa un gradiente determinista en un cálculo recursivo de los filtros Wiener para entradas estocásticas. Una característica muy importante y a la vez atractiva del algoritmo LMS es su simplicidad. No requiere medición de funciones de correlación, tampoco necesita de inversión de matrices. La operación adaptativa del algoritmo es totalmente descrita por la siguiente ecuacion recursiva:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{u}(n)[d(n) - \mathbf{w}^H(n)\mathbf{u}(n)]^* \tag{3.13}$$

Donde $\mathbf{u}(n)$ es el vector de entrada, $d(n)$ es la respuesta deseada, y μ es el parametro del tamaño de paso, el asterisco denota complejo conjugado y la H denota transposición hermitiana. El error de la señal es $[d(n) - \mathbf{w}^H(n)\mathbf{u}(n)]$, por lo que el parametro estimado es $y(n) = \mathbf{w}^H(n)\mathbf{u}(n)$.

La ecuación 3.13 se puede derivar de:

$$e(n) = d(n) - \mathbf{w}^H(n)\mathbf{u}(n) \tag{3.14}$$

Y estimando el gradiente de $\varepsilon = E[e(n)^2]$ por consideraciones de estacionariedad en sentido amplio, se toma un estimado del valor esperado igual a $\varepsilon n = e(n)^2$. Entonces en cada iteración del proceso adaptativo tenemos un gradiente estimado como sigue:

$$\hat{\nabla}(n) = \begin{bmatrix} \frac{\partial\ e(n)^2}{\partial\ w_0} \\ \vdots \\ \frac{\partial\ e(n)^2}{\partial\ w_L} \end{bmatrix} = 2e(n) \begin{bmatrix} \frac{\partial\ e(n)}{\partial\ w_0} \\ \vdots \\ \frac{\partial\ e(n)}{\partial\ w_L} \end{bmatrix} = -2e(n)\mathbf{u}(n) \tag{3.15}$$

Con esta estimación de gradiente se puede especificar un tipo de paso de gradiente descendente que nos llevaría a 3.13.

Entre los algoritmos que estan relacionados con el algoritmo LMS se encuentran los algoritmos de aproximaciones estócasticas que realizan busquedas lineales para optimizar sistemas con relaciones entre variables controlables y variables resultado. Otro algoritmo relacionado con el algoritmo LMS es el de gradiente de lattice adaptativo (GAL), la principal diferencia entre ellos es la estructura en que se basan, ya que el LMS se basa en una estructura trasnversal y GAL se basa en una estructura lattice.

3.4.2. Algoritmo NLMS

El algoritmo LMS Normalizado (NLMS) tiene por objetivo independizar la convergencia de la potencia de la señal de entrada, es por ello, más robusto que el algoritmo LMS. En el algoritmo LMS, la corrección aplicada al vector de pesos $w(n)$ es proporcional al vector de entrada $u(n)$. Por tanto, si $u(n)$ es elevado al cuadrado, el algoritmo LMS experimenta un problema de amplificación de ruido del gradiente, con la normalización del parámetro de convergencia, este problema se reduce, de igual manera que se evita un aumento desmedido de la corrección del vector $w(n)$ cuando la entrada disminuye drásticamente. El algoritmo queda así:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu\mathbf{u}(n)[d(n) - \mathbf{w}^H(n)\mathbf{u}(n)]^*}{\|\mathbf{u}(n)\|} \tag{3.16}$$

3.4.3. Algoritmo LMS con algoritmo de regresión secuencial

Las distintas variaciones del LMS tienen por objetivo realizar la estimación de la matriz $\mathbf{R^{-1}}$ de la ecuación normal, por ejemplo el algoritmo de regresión

secuencial (SER) se aproxima más al ideal de estimar de manera directa la matriz.

El algoritmo de regresión secuencial cálcula un estimado de $\mathbf{R^{-1}}$ que mejora con cada iteración, de esta manera se acerca a la solución directa de la matriz, aumentando la exactitud de la estimación y velocidad de convergencia a la solución del filtro.

Se realizan consideraciones relacionadas a la memoria del filtro, de manera que se define $\mathbf{Q_k}$ para proporcionar una memoria de corto plazo en la estimación de $\mathbf{R}$.

$$\mathbf{Q_k} = \sum_{l=0}^{k} \alpha^{k-l} \mathbf{X_l} \mathbf{X_l^T} \tag{3.17}$$

Teniendo el estimado $\hat{\mathbf{R}}_\mathbf{k}$ y la siguiente definición:

$$\hat{\mathbf{R}}_\mathbf{k} \mathbf{W_k} = \hat{\mathbf{P}}_\mathbf{k} \tag{3.18}$$

con $\hat{\mathbf{R}}_\mathbf{k}$ y $\hat{\mathbf{P}}_\mathbf{k}$ definidos como:

$$\hat{\mathbf{R}}_\mathbf{k} = \frac{1-\alpha}{1-\alpha^{k+1}} \mathbf{Q_k} \tag{3.19}$$

$$\hat{\mathbf{P}}_\mathbf{k} = \frac{1-\alpha}{1-\alpha^{k+1}} \sum_{l=0}^{k} \alpha^{k-l} d_l \mathbf{X_l} \tag{3.20}$$

Usando 3.19 y 3.20 se cancela el factor y se obtiene:

$$\hat{\mathbf{Q}}_\mathbf{k} \hat{\mathbf{W}}_\mathbf{k} = \sum_{l=0}^{k} \alpha^{k-l} d_l \mathbf{X_l} \tag{3.21}$$

de 3.18, 3.21 y, asumiendo que $\mathbf{W_{k+1}}$ puede ser estimada en terminos de $\hat{\mathbf{R}}_\mathbf{k}$ y $\hat{\mathbf{P}}_\mathbf{k}$ tenemos:

$$\hat{\mathbf{Q}}_{\mathbf{k}}\hat{\mathbf{W}}_{\mathbf{k+1}} = \alpha \sum_{l=0}^{k-1} \alpha^{(k-1)-l} d_l \mathbf{X_l} + d_k \mathbf{X_k} \tag{3.22}$$

de 3.17:

$$\hat{\mathbf{Q}}_{\mathbf{k}} = \alpha \hat{\mathbf{Q}}_{\mathbf{k-1}} + \mathbf{X_k}\mathbf{X_k^T} \tag{3.23}$$

Ahora sustituyendo 3.23 en 3.22 y reescribiendo para la definición de la señal deseada d en el instante k:

$$\hat{\mathbf{Q}}_{\mathbf{k}}\hat{\mathbf{W}}_{\mathbf{k+1}} = \hat{\mathbf{Q}}_{\mathbf{k}}\hat{\mathbf{W}}_{\mathbf{k}} + \epsilon_k \mathbf{X_k} \tag{3.24}$$

Para obtener la operación adaptativa del algoritmo se multiplica 3.24 por Q_k^{-1}, quedando así:

$$\hat{\mathbf{W}}_{\mathbf{k+1}} = \hat{\mathbf{W}}_{\mathbf{k}} + \hat{\mathbf{Q}}_{\mathbf{k}}^{-\mathbf{1}} \epsilon_k \mathbf{X_k} \tag{3.25}$$

Desde 3.19, se define $\mathbf{Q_k^{-1}}$:

$$\hat{\mathbf{Q}}_{\mathbf{k}}^{-\mathbf{1}} = \frac{1-\alpha}{1-\alpha^{k+1}} \mathbf{R_k^{-1}} \tag{3.26}$$

Considerando el caso de estado estable para un k lo suficientemente grande como para permitir ignorar α^{k+1} se puede hacer una aproximación a un ideal caso del algoritmo LMS de Newton, ademas por consideraciones de no estacionariedad se llega a:

$$\hat{\mathbf{W}}_{\mathbf{k+1}} = \hat{\mathbf{W}}_{\mathbf{k}} + \frac{2\mu\lambda_{av}(1-\alpha^{k+1})}{1-\alpha} \hat{\mathbf{Q}}_{\mathbf{k}}^{-\mathbf{1}} \epsilon_k \mathbf{X_k} \tag{3.27}$$

finalmente por premultiplicar $\mathbf{Q_k^{-1}}$ y posmultiplicar por $\mathbf{Q_{k-1}^{-1}}$ en 3.23 se puede llegar a un procedimiento para calcular iterativamente $\mathbf{Q_k^{-1}}$. A continuación los pasos del algoritmo para calcular los parámetros adaptables:

- $\alpha \approx 2^{\frac{-1}{longitud\ de\ estacionariedad}}$
- $\mathbf{Q_0^{-1}} = (Constante) * \mathbf{I}$
- $\mathbf{W_0}$ = Pesos iniciales
- $\mathbf{W_1} = \mathbf{W_0} + 2\mu\lambda_{av}\mathbf{Q_0^{-1}}\epsilon_0\mathbf{X_0}$

Para $k \geq 1$:

- $\mathbf{S} = \mathbf{Q_{k-1}^{-1}}\mathbf{X_k}$
- $\gamma = \alpha + \mathbf{X_k^T S}$
- $\mathbf{Q_k^{-1}} = \frac{1}{\alpha}(\mathbf{Q_{k-1}^{-1}} - \frac{1}{\gamma}\mathbf{S S^T})$
- $\mathbf{W_{k+1}} = \mathbf{W_k} + \frac{2\mu\lambda_{av}(1-\alpha^{k+1})}{1-\alpha}\mathbf{Q_k^{-1}}\epsilon_k\mathbf{X_k}$
- $0 < \mu < \frac{1}{\lambda_{av}}$, o $\mu\lambda_{av} \ll 1$

Con los pasos anteriores es posible encontrar los parámetros de un filtro transversal adaptable que realice la predicción de una señal con la que se alimente. Todas las variaciones del algoritmo LMS son implementadas en acople con los demás módulos que conforman el sistema de odometría visual desarrollado para la estimación de la trayectoria del robot.

Capítulo 4

Sistema propuesto

El énfoque principal del sistema planteado para la estimación de la trayectoria de una plataforma móvil con odometría visual se centra en la detección de características entre imágenes sucesivas, para una correspondencia temporal entre puntos característicos persistentes temporal y espacialmente, que permiten la estimación de movimiento de la plataforma a partir de una concatenación de transformaciones sucesivas obteniendo una trayectoria de movimiento. Esta trayectoria es corregida por filtros de bajo costo computacional fundamentados en la solución de la ecuación normal de Wiener permitiendo por medio de estimadores de gradientes estocásticos solucionar implicitamente la ecuación normal.

El sistema se compone de los siguientes módulos:

- Detección y extracción de características.
- Correspondencia de características y filtrado de valores atípicos.
- Cálculo del problema de transformación de cuerpo rígido.
- Concatenación de estimación de poses y corrección de trayectoria por filtrado adaptable.

La figura 4.1 representa la forma en que los módulos se relacionan para entregar la trayectoria de desplazamiento estimada.

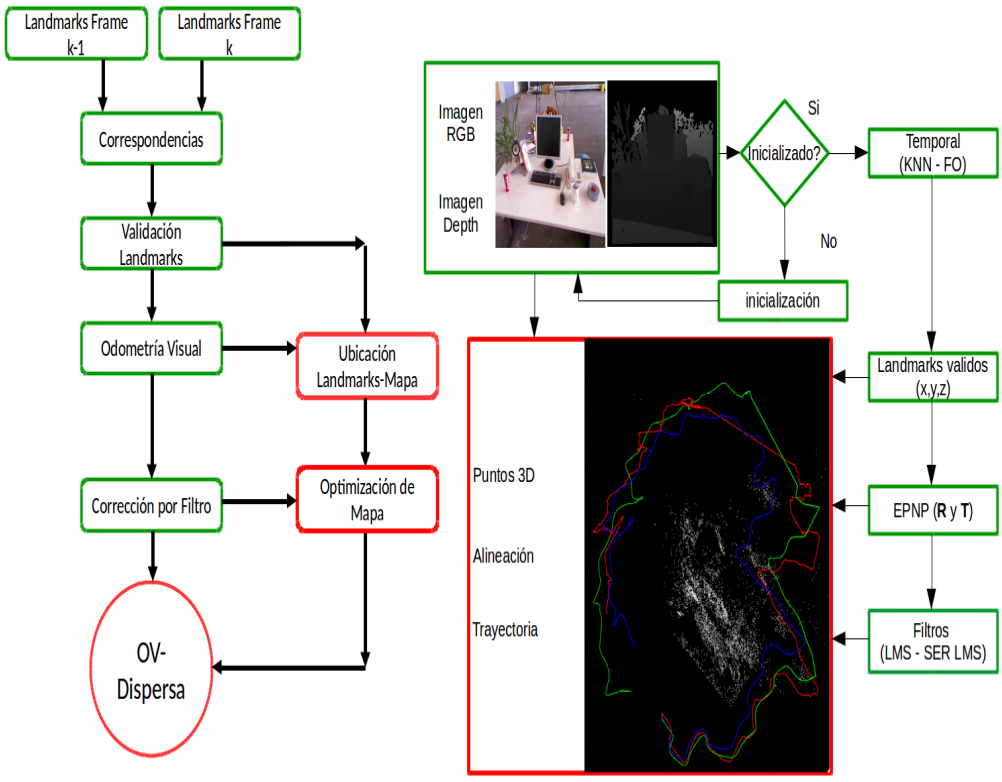


Figura 4.1. *Sistema de odometría visual.*

4.1. Detección y extracción de características

En un principio, cuando se obtiene una nueva imagen se buscan sus puntos de interés; después se calculan sus descriptores para finalmente emparejar esos puntos con los puntos del frame inmediatamente anterior estimado. Este proceso requiere de un detector que sea eficiente, que proporcione una gran cantidad de puntos y que tenga buena precisión en la localización. También es necesario que el descriptor a utilizar sea bastante robusto y al mismo tiempo eficiente. Por tales motivos se decidió por SURF como detector y descriptor.

Se escogió SURF porque es el detector con la mejor relación costo beneficio y que además proporciona las características suficientes para las estimaciones posteriores de la matriz de transformación. En 4.2 se observa la robustez del algoritmo al extraer características de la mesa, la cual con la resolución de la camara y la

distancia tiene una relativa baja textura, por otro lado encuentra características con una alta cercanía en objetos con muchos bordes como el teclado, que al momento de la correspondencia pueden tener una alta similitud entre características vecinas y corresponder equivocadamente las posiciones temporales entre frames de dichas características e introducir errores en las estimaciones de la rotación y traslación del sistema.

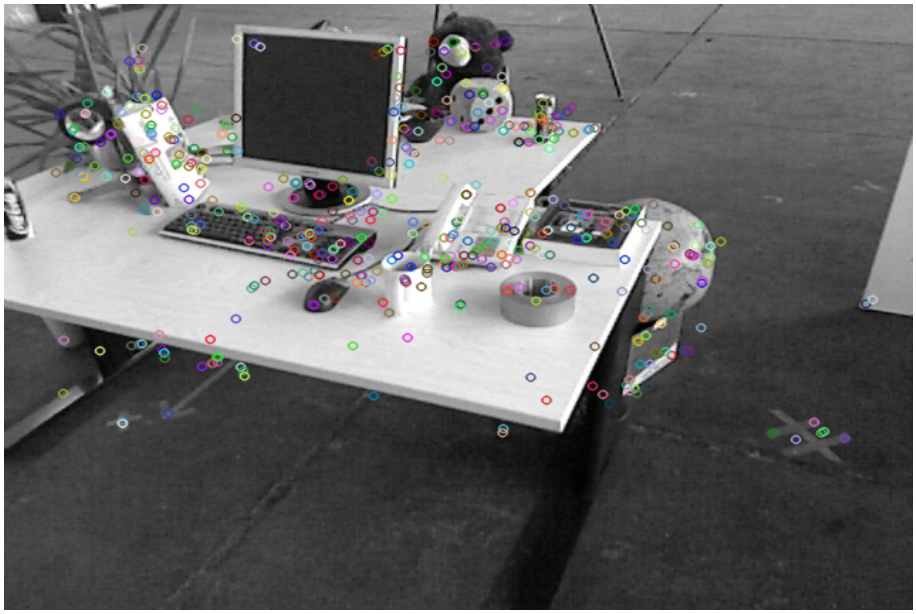


Figura 4.2. *Características de SURF en un frame del conjunto de datos seleccionado.*

4.2. Correspondencia de características y filtrado de valores atípicos

El emparejamiento de puntos entre la imagen I_{k-1} y la imagen I_k se lleva a cabo usando un algoritmo de K vecinos más cercanos (K nearest neighbors). Este algoritmo encuentra una cantidad de vecinos probablemente asociados al descriptor de iteración que se esté procesando con un margen de error óptimo en un tiempo menor al realizable por fuerza bruta, además de tener un grado de robustez ante características muy cercanas eligiendo el vecino más probable a corresponder o supuestas características nuevas de alta similitud con desplazamientos significativos en la imagen, las cuales introducen el mayor error en la estimación de la pose relativa entre frames 4.3.

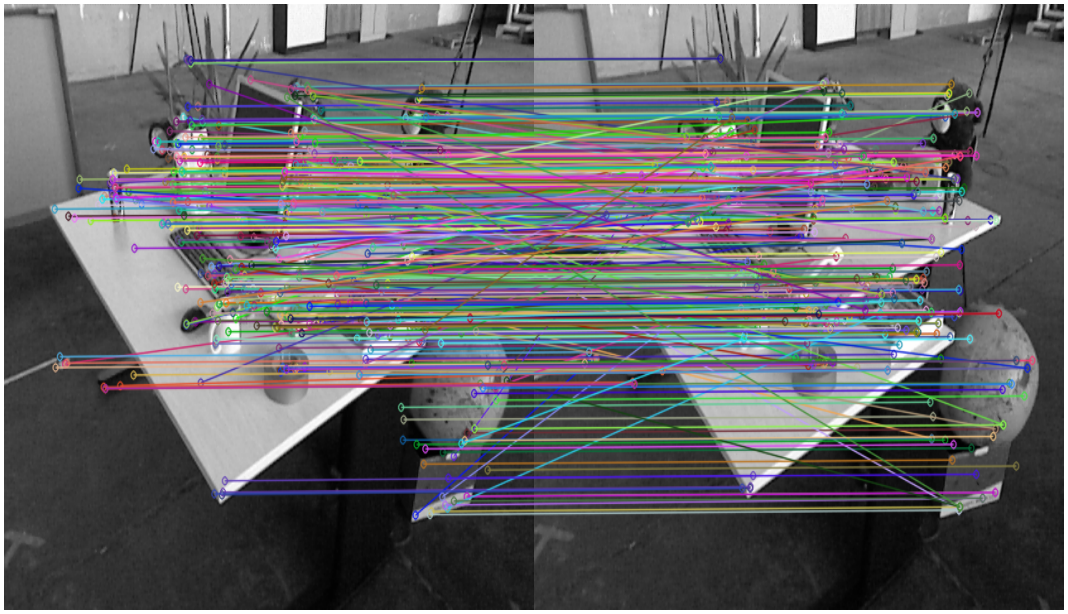


Figura 4.3. *Correspondencia de características con valores atípicos entre características correspondidas.*

Para incrementar la robustez en la asociación y evitar errores de correspondencia de características altamente similares se ajusta el umbral de distancia a una mínima distancia de comparación de los K vecinos más cercanos, lo que lleva a un rechazo de falsas correspondencias de características más elevado a costa de una reducción en el número de características correctamente correspondidas. A pesar de elegir este umbral lo más pequeño posible, existe una probabilidad considerable de encontrar asociaciones erróneas, por lo que se acopla un predictor de posición de trayectoria por cercanía espacial en el espacio de la imagen, lo que permite rechazar características que según el sistema de emparejamiento se han desplazado considerablemente 4.4. Cabe notar que solo se realizan emparejamientos entre dos frames para tomar como válida una característica, lo que difiere con la literatura para esta clase de sistemas, los cuales por lo general encuentran la relación que existe entre los puntos encontrados en las imágenes de por lo menos tres frames, es decir, se obtiene la trayectoria de los puntos entre estos tres frames vecinos y se considera como una característica válida para estimación aquellas que perduren en estos frames.

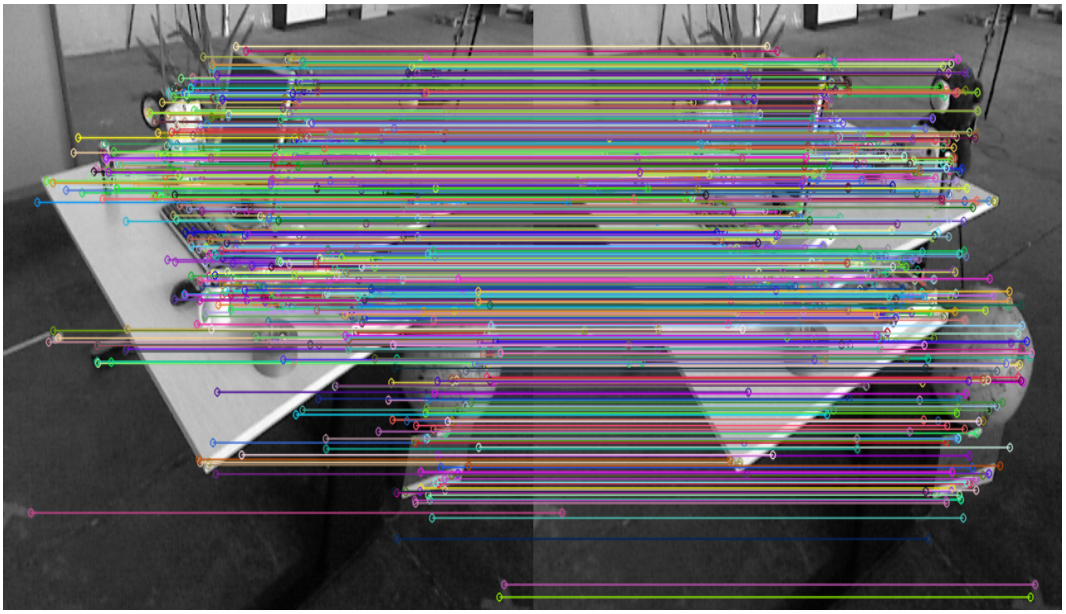


Figura 4.4. *Correspondencia de características con filtrado de valores atípicos.*

4.3. Cálculo del problema de transformación de cuerpo rígido

La estimación de la transformación que realiza el mapeo de un sistema coordenado a otro se determina usando las caracterisiticas encontradas en frames sucesivos, si y solo si este número de características es mayor a un umbral establecido para limitar el error de predicción por mínimo de características, en caso contrario se deja estático el último frame y se realiza búsqueda de características en los siguientes frames con mayores aciertos de emparejamientos.

Una vez se tienen las características necesarias se realiza con Perspectiva desde n Puntos (Pnp) la estimación de rotación y traslación del sistema, en específico se usa una versión iterativa de EPnP aprovechando el conocimiento que se tiene de la profundidad por el sensor. Para usar el algoritmo y solucionar el sistema que optimicce los parámetros de transformación, se requieren al menos 4 puntos en el espacio del mundo y sus correspondientes en el espacio de la imagen.

En el inicio del sistema de odometría visual dispersa se establece el sistema de coordenadas absolutas al primer frame y es desde este que se relacionan todas las transformaciones para realizar la trayectoria y al cual las nubes de puntos se mapean con el objetivo de tener un entorno disperso de puntos referenciados a un solo sistema cartesiano.

4.4. Concatenación de poses y corrección de trayectoria por filtrado

Una vez se tienen las estimaciones relativas entre frames sucesivos temporalmente y la relación de las poses de cámara en el instante $k-1$ y k por transformación de cuerpo rígido dada por 4.2, se obtiene C_k, que es la pose absoluta en el instante k y, teniendo por simplicidad las coordenadas del mundo igual al primer frame del agente $k=0$, se calculan por concatenación todas las transformaciones que permiten obtener la trayectoria completa 4.1.

$$C_k = C_{k-1} T_{k,k-1} \tag{4.1}$$

$$\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ \mathbf{0}_{1*3} & 1 \end{bmatrix} \tag{4.2}$$

Entre estimación y estimación de T_k, se realiza una concatenación que es refinada por filtros de grandiente estocástico, lo que permite limitar rotaciones y traslaciones abruptas y reducir los errores en las transformaciones debidos al sensor, la extracción y correspondencia de las características y demás.

Capítulo 5

Pruebas y resultados

En este capítulo se presentan los resultados del sistema de generación de trayectoria por odometría visual, para un robot móvil, cuyos módulos principales fueron descritos brevemente en el capítulo previo.

Para obtener medidas cuantitativas y confiables del comportamiento del sistema propuesto se eligió un video del repositorio de validación del grupo de visión por computadora de la Universidad Técnica de Munich [6]. Este conjunto de datos se ha utilizado para la validación de sistemas de navegación de robots móviles, para el caso que nos compete un robot de servicio por lo que fue oportuno contar con vídeos de espacios en interiores, además de incluir los parámetros de calibración del sensor utilizado. Los datos usados tienen disponibilidad pública y tiene por titulo: "$rgbd\_dataset\_freiburg2\_desk$".

El video fue realizado en una oficina artificial, sin objetos dinámicos y cualitativamente se puede decir que con una luz ambiental de poco impacto para el sensor de IR. El recorrido se realizó alrededor de una mesa que tiene objetos $'típicos'$ de oficina, con algunas excepciones. Este conjunto de datos contiene 20000 cuadros de imágenes RGB y de profundidad.

Figura 5.1. *Imagen de frames RGB del conjunto de datos usado [6].*

En la figura 5.2 se aprecian variaciones en los ángulos principales (Pitch, Yaw y Roll), igualmente es posible percibir la riqueza de texturas, aunque dichas texturas tienden a estar concentradas todas muy próximas al plano horizontal de la mesa, lo cual es un problema en las estimación de las transformaciones, afectando la estimación angular del Pitch y el Roll.

5.1. Resultados por módulos individuales del sistema global

5.1.1. Detección, extracción y correspondencía de características

El algoritmo de SURF comprende la detección de puntos de interés y un descriptor. Para la obtención de las características con el algoritmo SURF se consideró un umbral para el hessiano, donde matemáticamente el hessiano de una matriz describe las segundas derivadas de una función que representan las curvaturas, lo que permite buscar los máximos. SURF usa este fundamento de la matriz hessiana con sus valores propios y el umbral que se establezca como limite mínimo. Para detectar los puntos salientes en una imagen se debe obtener del determinante de la matriz de un punto de interés y para ser considerado una característica debe sobrepasar el umbral definido para el detector de esquinas hessiano. Con lo anterior, de acuerdo al umbral que se establezca se puede tener un gran número de puntos de interés, no todos robustos, que es el caso de un

umbral bajo, o teóricamente con un alto umbral se obtendran menos puntos característicos, pero más destacados y robustos.

Otro umbral que se preestableció fue el de la correspondencia de características, donde la mínima distancia entre vecinos más probables es aquella que va dada por los dos vecinos más cercanos en dimensiones del vector descriptor del punto característico en cuestión, multiplicando el segundo más probable por un umbral de cercanía, evitando elegir un descriptor de alta similitud al correspondiente vector real de la característica a corresponder.

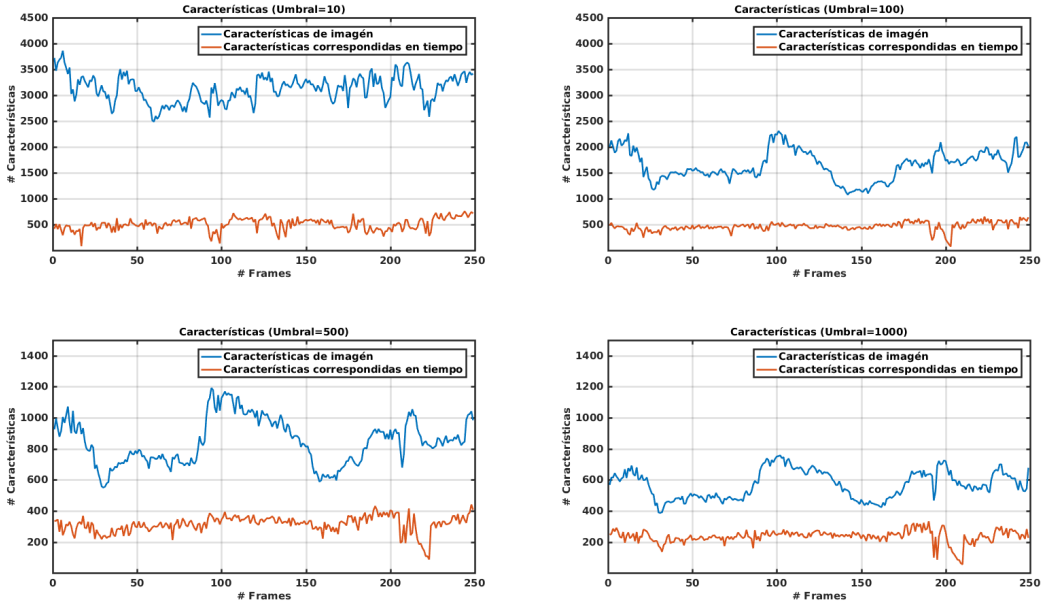


Figura 5.2. *Comportamiento del módulo de características empleando diferentes umbrales para el hessiano.*

La figura 5.2 muestra la disminución de características obtenidas por imagen, donde efectivamente a un menor umbral para el hessiano se obtienen mayor cantidad de características, que al momento de ser correspondidas, solo un porcentaje inferior al total de las características se corresponden adecuadamente, contrario a lo que sucede con valores de umbral altos, donde la brecha entre características encontradas y correspondidas es menor.

Tabla 5.1. *Comportamiento del módulo de detección, extracción y correspondencia de características entre frames sucesivos.*

Umbral	Num. de características	Num. de correspondencias	Tiempo de ejecución del módulo (ms)
10	3128	506	750.9
100	1650	465	466.2
500	847	314	305.0
1000	570	235	247.6

En la tabla 5.1 se tienen los resultados promedios obtenidos para 10 segundos de imágenes del dataset. Se observa un decremento de características encontradas a medida que aumenta el umbral, es de notar el decremento de características correspondidas, a una tasa inferior a las del incremento del umbral y el decremento de características. Por lo tanto se verifica que para un umbral definido pertinentemente se obtienen caracterísitcas suficientes y robustas para ser correspondidas temporalmente entre distintos frames con desplazamiento.

5.1.2. Estimación de la matriz de rotación y vector de traslación con EPNP

Desde los puntos de interés y sus respectivas correspondencias entre frames temporales tanto en $3D$ como en $2D$, se estimó la transformación entre frames sucecivos y la concatenación de poses para la trayectoria de movimiento del sistema.

Con el fin de probar el funcionamiento adecuado del módulo de estimación de las trasnformaciones espaciales del sistema se hicieron pruebas en el conjunto de datos en cinco sectores de este, los cuales fueron procesados con el módulo de extracción, correspondencia de características y la solución al problema de PNP.

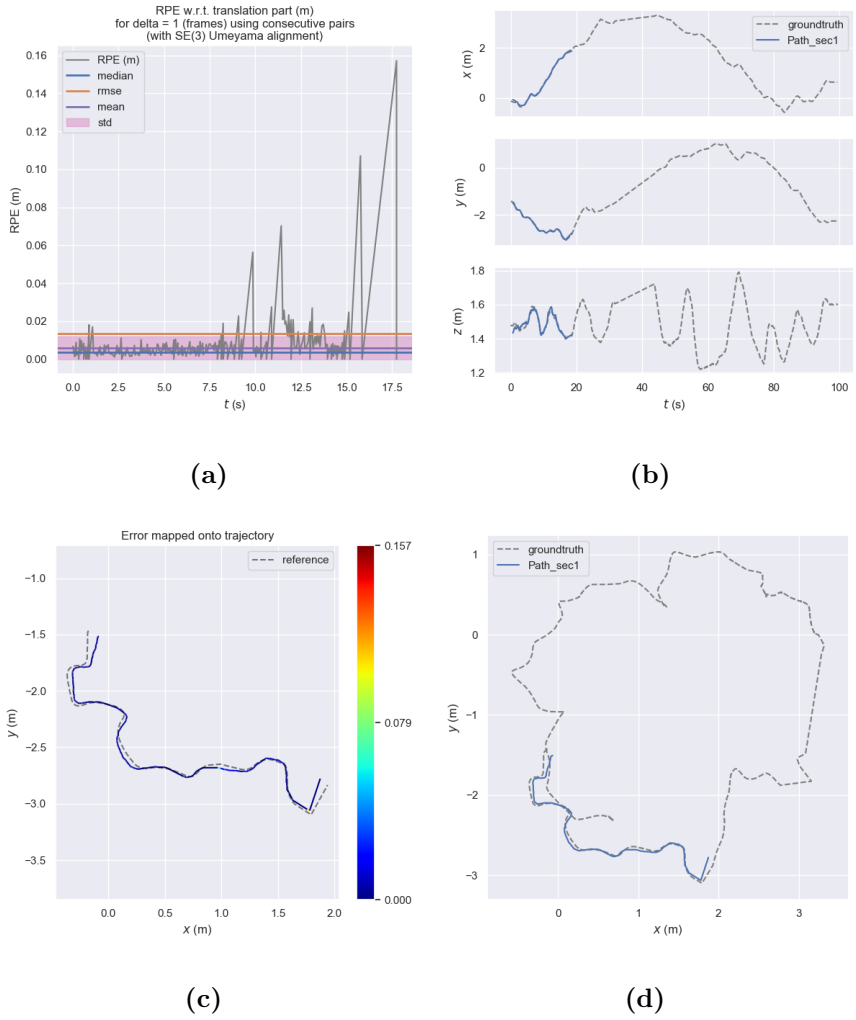


Figura 5.3. *Error de pose relativa, trayectoria por coordenada y trayectoria global. en (a) y (b), Error de posición relativa (RPE) y perfiles de trayectoria por coordenada respectivamente. En (c) vista aérea de la trayectoria del sector 1 y el sector respectivo de la trayectoria real. En (d) vista aérea de la trayectoria del sector 1 junto con la trayectoria real completa.*

En la figura 5.3 se tiene el comportamiento del sistema con el módulo de características y el módulo de PNP para la sector 1 que tiene una longitud de 4.205 metros y 17.9 segundos de duración. La imagen superior izquierda muestra el comportamiento del error de pose relativa, en donde se tiene un máximo de $0.1571m$ de error, con un valor $RMSE = 0.013m$, $media = 0.005961m$, $mediana = 0.003784m$ y $desviación\ estandar = 0.01193m$, todos los valores en un rango de error bastante acertado para ser el sistema sin correción de estimaciones ni ajustes globales de trayectoria, igualmente cabe notar desde la imagen de

vista aérea un problema en la estimación de las rotaciones, donde se aprecian las distorsiones en la alineación sobre la trayecyoria real de longitud igual a 20.339 metros y duración de 99.3 segundos.

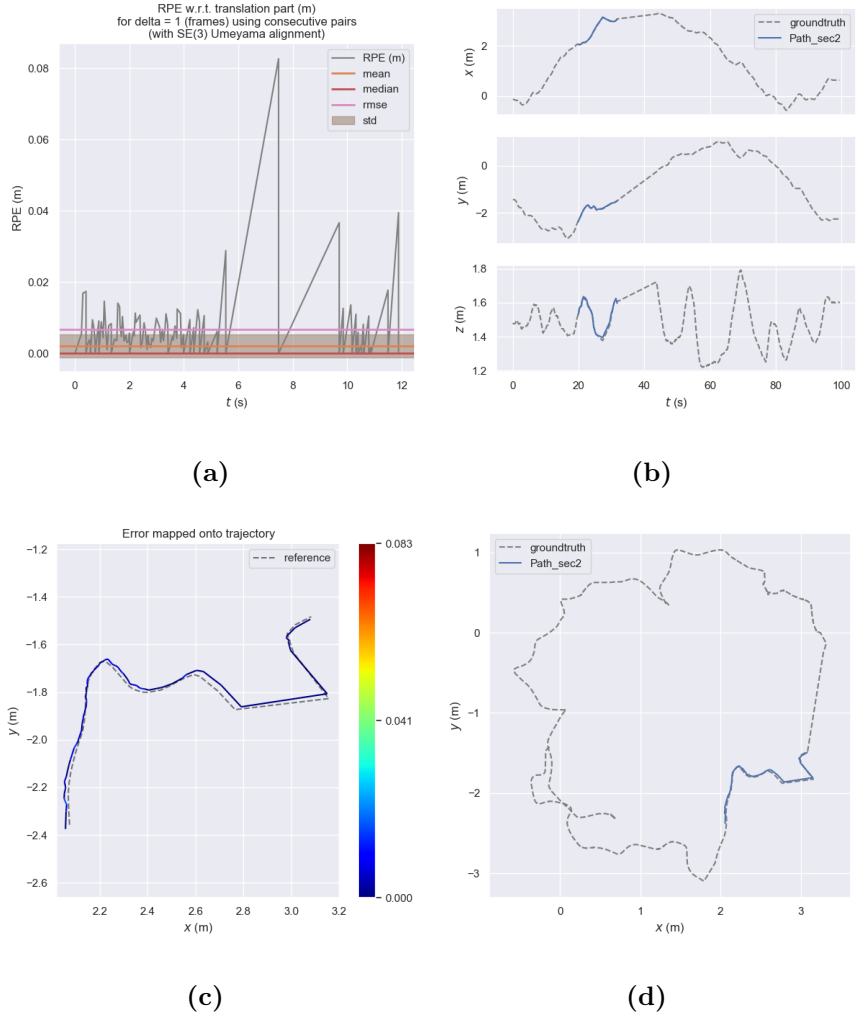


Figura 5.4. *Error de pose relativa, trayectoria por coordenada y trayectoria global. en (a) y (b), Error de posición relativa (RPE) y perfiles de trayectoria por coordenada respectivamente. En (c) vista aérea de la trayectoria del sector 2 y el sector respectivo de la trayectoria real. En (d) vista aérea de la trayectoria del sector 2 junto con la trayectoria real completa.*

Para la sector 2 se tiene la trayectoria con longitud de 2.418 metros y 11.877 segundos con medidas de error en 5.4 inferiores en comparación a la sector 1, con un máximo de 0.082645 metros de error, un valor de $RMSE = 0.006731$ metros, $media = 0.002163$ metros, $mediana = 0.000000$ metros y *desviación*

estandar = 0.006373 metros. Los perfiles de trayectoria por coordenada muestran una correcta estimación de odometría para este sector. Analizando la vista aérea de la imagen en (c) se observa un nivel de altura menos variable.

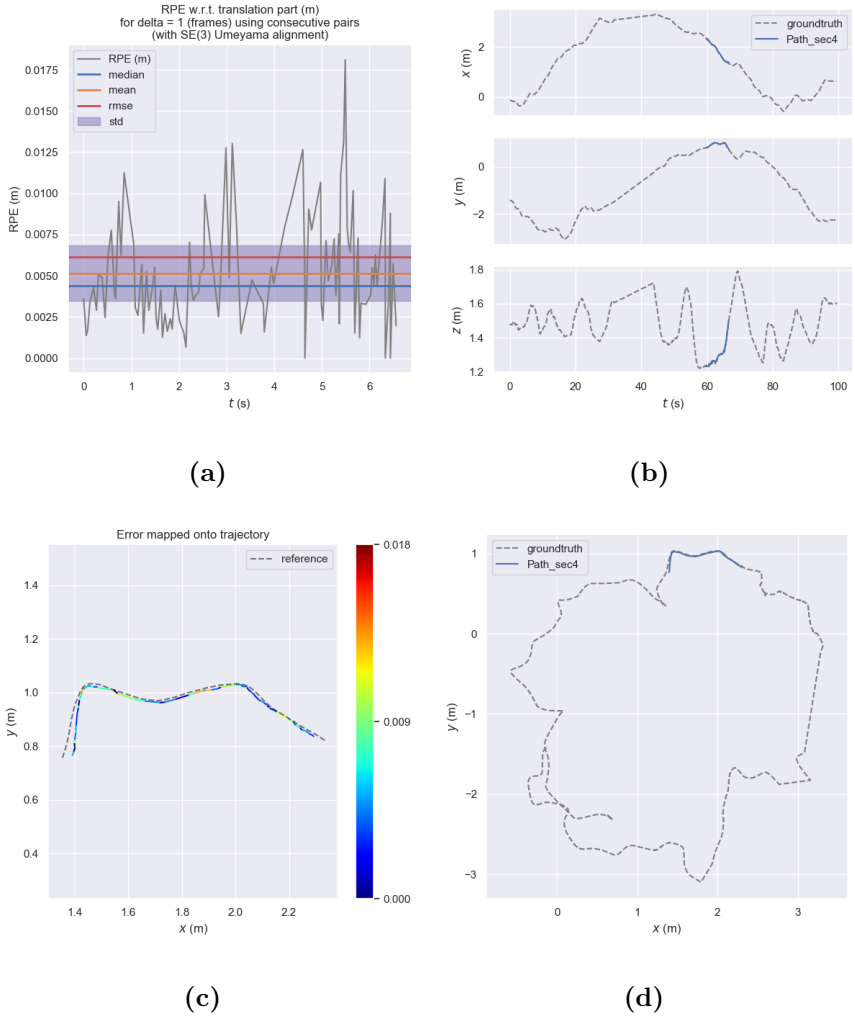


Figura 5.5. *Error de pose relativa, trayectoria por coordenada y trayectoria global. en (a) y (b), Error de posición relativa (RPE) y perfiles de trayectoria por coordenada respectivamente. En (c) vista aérea de la trayectoria del sector 3 y el sector respectivo de la trayectoria real. En (d) vista aérea de la trayectoria del sector 3 junto con la trayectoria real completa.*

Para el sector 3 se observa una trayectoria más suave en sus rotaciones, notandose una disminución y estabilidad en las medidas de error en 5.5, para este sector la longitud de la trayectoria es 1.344 metros y una duración de 6.713 segundos, teniendo así los siguientes valores en el error: máximo de 0.018107 metros

de error, RMSE= 0.006152 metros, media= 0.005155 metros, mediana= 0.004406 metros y *desviación estandar* = 0.003357 metros. Los perfiles de trayectoria por coordenada muestran una correcta estimación de odometría para este sector e igualmente que e sector 2 la vista aérea deja ver un nivel de altura poco variable, lo que influye en la disminución del error.

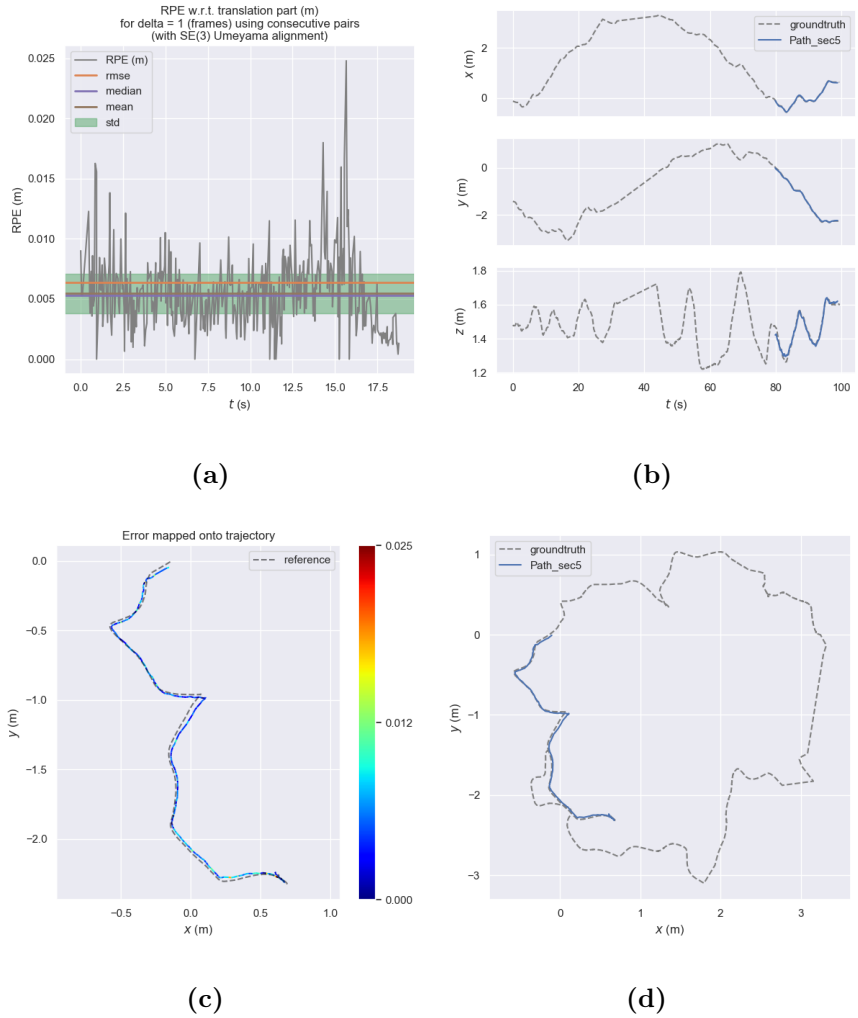


Figura 5.6. *Error de pose relativa, trayectoria por coordenada y trayectoria global. en (a) y (b), Error de posición relativa (RPE) y perfiles de trayectoria por coordenada respectivamente. En (c) vista aérea de la trayectoria del sector 4 y el sector respectivo de la trayectoria real. En (d) vista aérea de la trayectoria del sector 4 junto con la trayectoria real completa.*

Finalmente en la figura 5.6 que representa el sector 4 se tienen de nuevo valores aceptables en las estimaciones. A pesar de contener una rotación fuertemente

marcada, el sistema logró una buena aproximación al valor real. Para este último sector se tiene una longitud de 4.046 metros y duración de 18.954 segundos, con las siguientes medidas de error: un máximo de 0.024787 metros, $RMSE = 0.006348$ metros, $media = 0.005448$ metros, $mediana = 0.005292$ metros y $desviación\ estandar = 0.003258$ metros.

Tabla 5.2. *Módulo de estimación de rotación y traslación de puntos* $3D - 2D$ *para distintos sectores de la trayectoria total.*

Sectores	**Num. de características**	**tiempo (ms)**	**RMSE**	**Trayectoria (m)**
1	264	304.1	0.013339	4.205
2	342	334.5	0.006731	2.418
3	133	353.7	0.006152	1.344
4	194	237.8	0.006348	4.046

La tabla 5.2 refleja el comportamiento del algoritmo EPNP para las características obtenidas con SURF y correspondidas con KNN en los distintos sectores en los que se analizó el conjunto de datos. Los resutados obtenidos demuestran el funcionamiento correcto de la implementación para la solución del proplema de cuerpo rígido con características $2D$ a $3D$ propuesto y da prueba de que las características extraídas son robustas temporalmente al permitir concatenar posiciones en longitudes de uno a cuatro metros, con tiempos de recorrido de 10 a 18 segundos aproximadamente.

5.1.3. Algoritmo LMS, NLMS y LMS-SER

Una etapa posterior a la estimación de la transformación de rotación y traslación del sistema es la corrección por filtros. Se probaron los algoritmos LMS, NLMS y LMS-SER para la actualización de los pesos de filtros transversales. Las primeras pruebas que se realizaron fueron sobre una señal simulada con distintas relaciones de señal-ruido con el objetivo de explorar el número de coeficientes del filtro y parámetros propios de los algoritmos como el paso de convergencia.

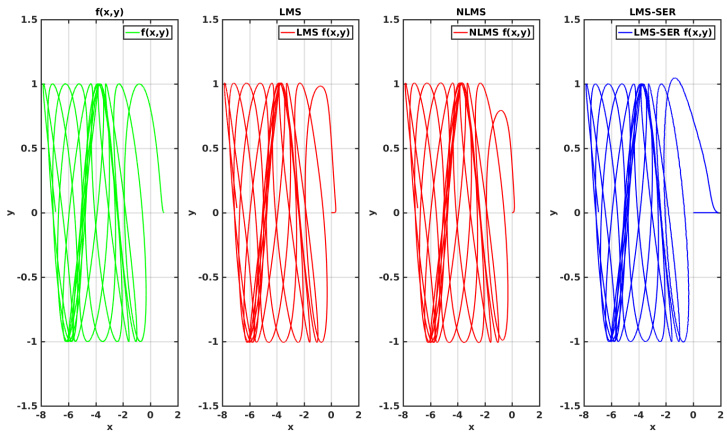


Figura 5.7. *Señal simulada de trayectoria 2D sin ruido. De izquierda a derecha, trayectoria simulada, trayectoria con predicción de NLMS y trayectoria con predicción de LMS-SER respectivamente.*

En la figura 5.7 se observa de izquierda a derecha la respuesta de predicción ante una entrada como la simulada, la señal artificial inicia desde $(x, y) = (1, 0)$ y por inicialización de los pesos en los predictores se visualiza un punto de partida distinto al real ("*simulada*"), hasta un tiempo después que los filtros convergen y se estabilizan a la señal.

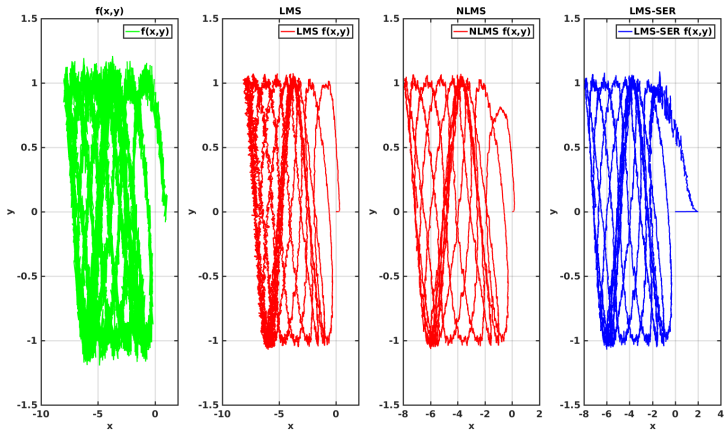


Figura 5.8. *Señal simulada de trayectoria 2D con ruido (SNR=0.7). De izquierda a derecha, trayectoria simulada, trayectoria con predicción de LMS y trayectoria con predicción de LMS-SER respectivamente.*

Para la figura 5.8 con una relación señal a ruido de 0.2 db los predictores infie-

ren una trayectoria más limpa que la señal simulada, cualitativamente se puede decir que la capacidad de los tres predictores ante el ruido es más eficiente y similar al momento de filtrar y predecir la señal de fondo "*real*" que el filtro.

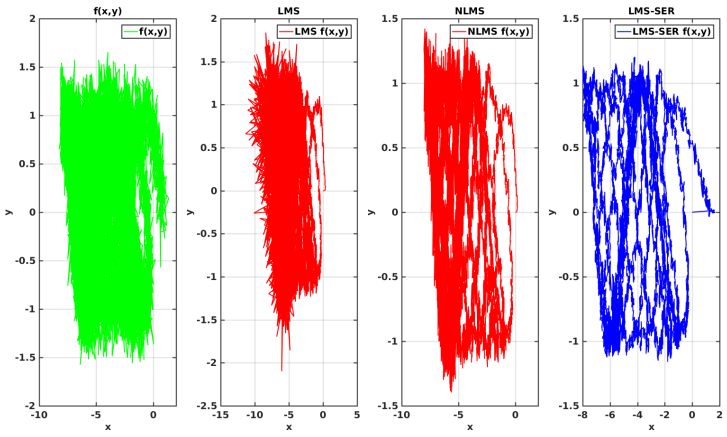
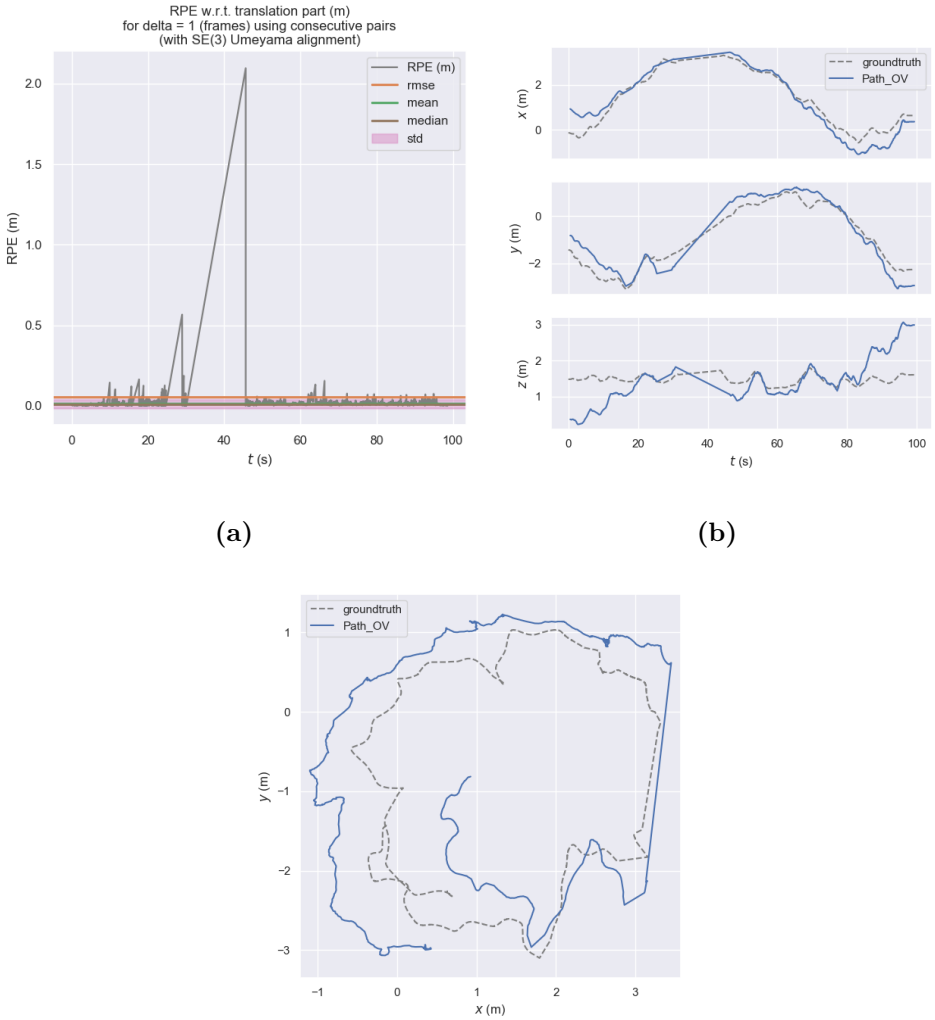


Figura 5.9. *Señal simulada de trayectoria 2D con ruido (SNR=0.5). De izquierda a derecha, trayectoria simulada, trayectoria con predicción de NLMS y trayectoria con predicción de LMS-SER respectivamente.*

Finalmente ante un aumento en la relación señal a ruido se tiene en la figura 5.9 una divergencia para el filtro con algoritmo LMS, y un aumento en la amplitud del NLMS, que terminara en inestabilidad, lo que lo llevara a diverger. Por otro lado el filtro con algoritmo LMS-SER es estable, además de efectivo ante las predicciones de la señal "*real*" inmersa en el ruido.

5.2. Resultados globales del sistema

Posterior a la descripción particular de cada módulo de los que se compone el sistema de odometría visual propuesto, se tiene un análisis en conjunto de los resultados obtenidos con los módulos acoplados. Para todos los experimentos se empleó el umbral del hessiano del algoritmo SURF con valor de 1000 por cuestiones de robustez en las características y resultados obtenidos en el módulo de extracción y correspondencias de características resultando en tiempos de computo óptimos y errores de estimación dentro de límites aceptables.

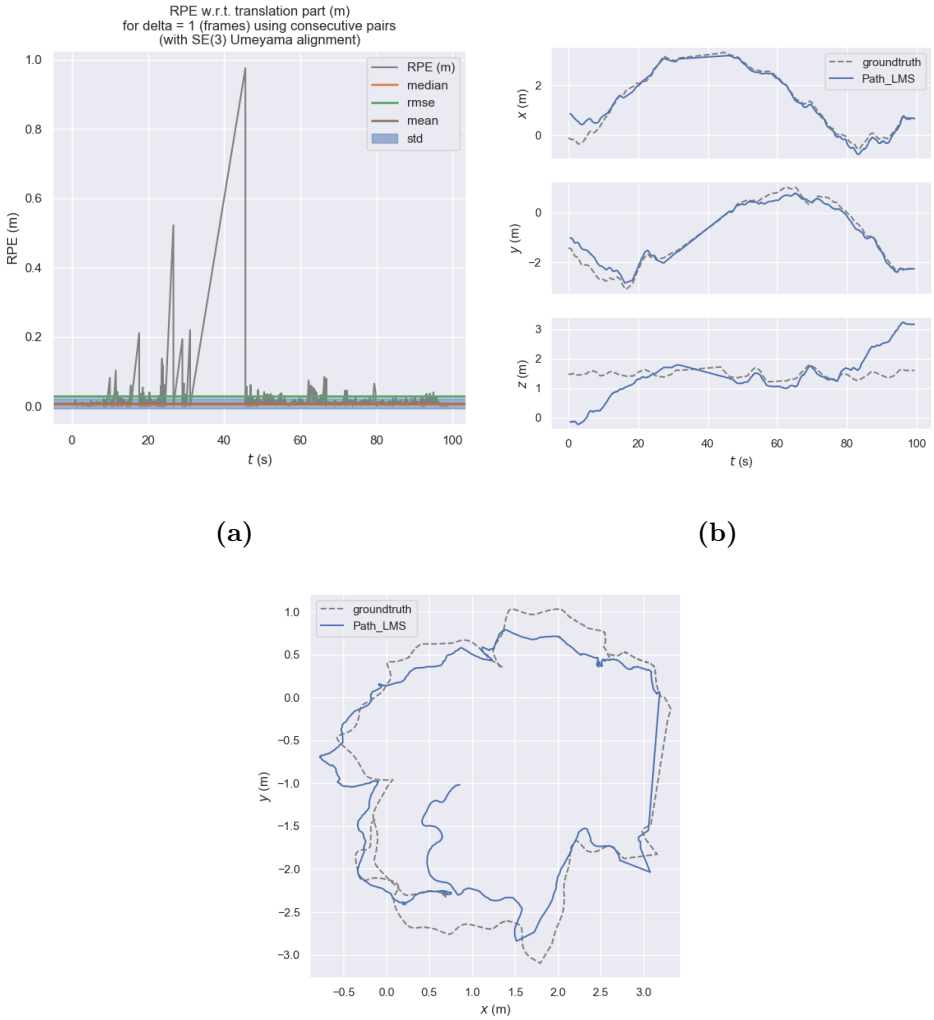


(a) **(b)**

(c)

Figura 5.10. *Estimación de trayectoria sin corrección por filtro. En (a) error de pose relativa, en (b) trayectoria por coordenada y en (c) trayectoria global de estimación por el sistema propuesto y la trayectoria real.*

Para la figura 5.10 se tiene una trayectoria con forma similar a la trayectoria real, pero con dimensiones más extendidas, donde los problemas que se presentan para que la trayectoria estimada se degenere son las rotaciones en los ángulos pitch y roll. Esto es debido a que las características se encuentran fundamentalmente esparcidas en el espacio del plano bidimensional xy, el cuál básicamente es el plano de la mesa en la oficina artificial, por lo que al momento de las estimaciones para rotaciones en pitch y roll son de bajo acierto.



(a) **(b)**

(c)

Figura 5.11. *Estimación de trayectoria con corrección por filtro LMS. En (a) error de pose relativa, en (b) trayectoria por coordenada y en (c) trayectoria global de estimación por el sistema propuesto y la trayectoria real.*

Teniendo como objetivo reducir los errores de estimación de la trayectoria con filtros adaptativos se implementó el fitro con algoritmo LMS, aprovechando su capacidad de inferir comportamientos futuros de acuerdo a comportamientos pasados, y de esta manera tener un sistema estable que rechaza cambios bruscos en una estimación secuencial reduciendo la degeneración de la trayectoria y minimizando el error acumulado de traslación del sistema de odometría visual.

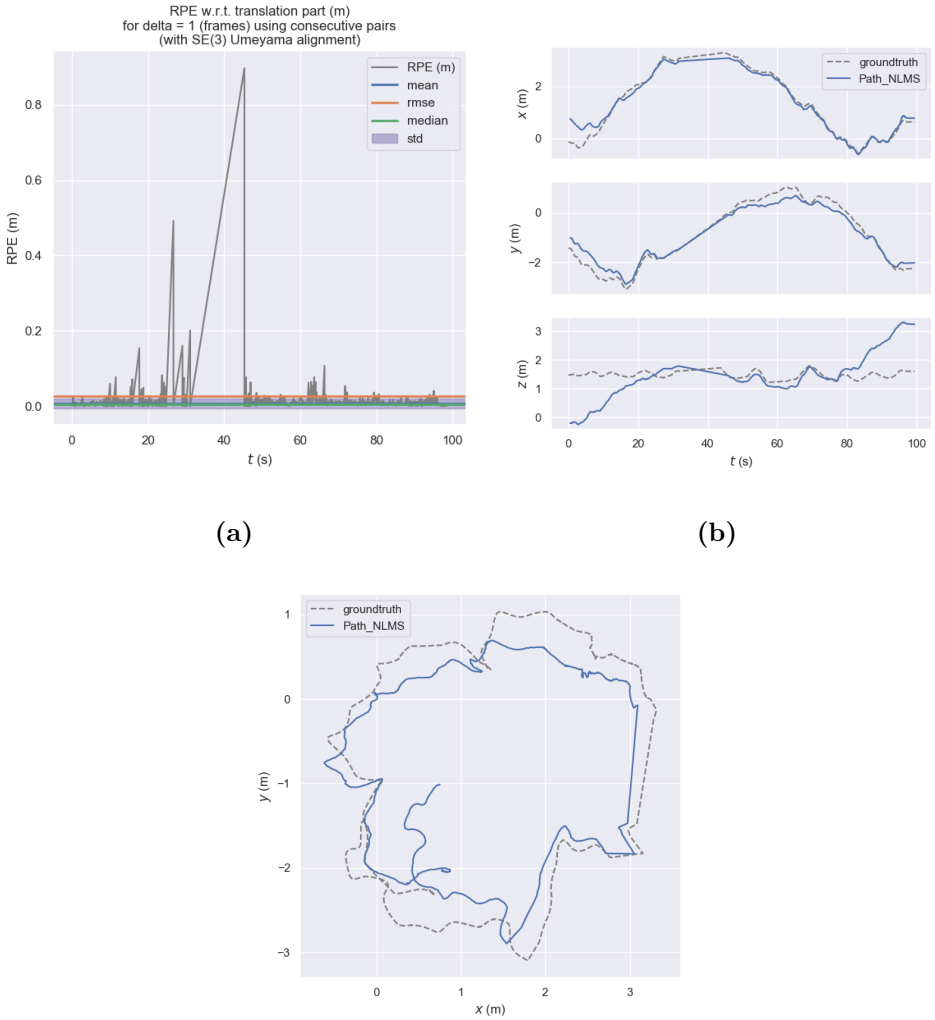


Figura 5.12. *Estimación de trayectoria con corrección por filtro NLMS. En (a) error de pose relativa, en (b) trayectoria por coordenada y en (c) trayectoria global de estimación por el sistema propuesto y la trayectoria real.*

Por estudios previos en los filtros con algoritmo LMS el cual tiene tendencias a inestabilizarse se implemento un filtro con algoritmo NLMS 5.12 que limita las estimaciones al realizar una modificación del paso de acuerdo a la energía de la señal de entrada, y así, aumentar la estabilidad del filtro ante cambios de amplitud de la señal. Los cuales en el caso del sistema de odometría visual se pueden presentar al momento de un aumento de velocidad de desplazamiento espacial del sensor o características correspondidas n frames después de la ultima estimación, lo que puede derivar en traslaciones y/o rotaciones considerables en

amplitud.

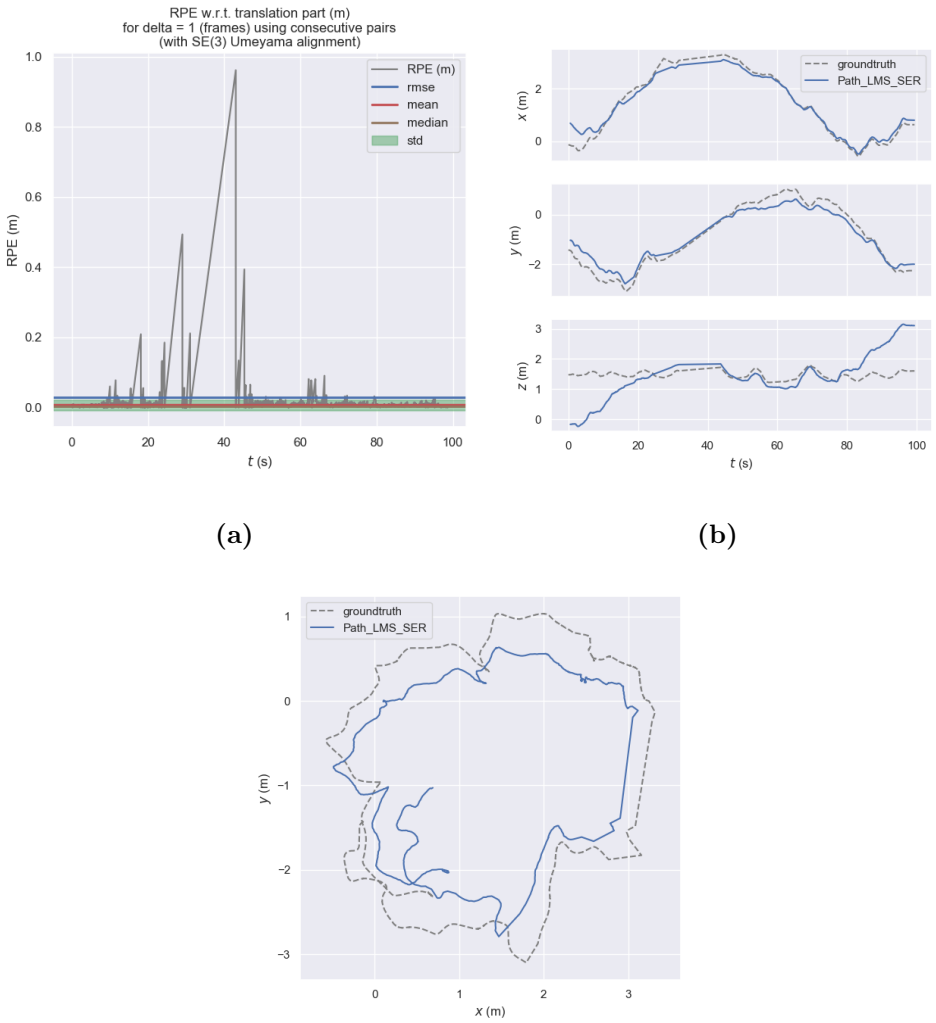


Figura 5.13. *Estimación de trayectoria con corrección por filtro con algoritmo LMS-SER. En (a) error de pose relativa, en (b) trayectoria por coordenada y en (c) trayectoria global de estimación por el sistema propuesto y la trayectoria real.*

Finalmente en la figura 5.13 se implementó un filtro predictor con algoritmo de regresión secuencial LMS para el cual, a pesar de no ser una mejora notoria en la correción comparado con el LMS y NLMs, su estabilidad ante el ruido o cambios muy marcados es mayor a los otros dos. En resumen de los errores del sistema de odometría visual con los filtros predictores se tiene la tabla 5.3.

Tabla 5.3. *Errores de pose relativa en estimación de trayectoria con correciones por filtro y, en el primer caso el resultado dado por el algoritmo para solucionar el problema de PNP sin filtro.*

RPE	**Max.**	**Min.**	**Media**	**Mediana**	**RMSE**	**Std**
OV	2.094805	0.000000	0.012257	0.006859	0.054145	0.052739
OV-LMS	0.975361	0.000000	0.008200	0.005135	0.029082	0.027902
OV-NLMS	0.897101	0.000000	0.007893	0.005073	0.027250	0.026082
OV-LMS-SER	0.961633	0.000000	0.007617	0.004727	0.030207	0.029231

Los resultados presentados en 5.3 permiten determinar que los filtros efectivamente hacen una correción general en el resultado final de la trayectoria. El filtro con algoritmo LMS-SER se presenta como el de mejor desempeño. A pesar de tener un sistema de odometría visual con características dispersas y minimización del error por técnicas de filtrado que permite tener una correción de las trayectorias estimadas haciendo estas más próximas a la trayectoria real, aún dista en gran medida la estimación de los valores "*reales*".

Capítulo 6

Conclusiones y trabajo futuro

6.0.1. Conclusiones

El desarrollo de esta tesis permitió tener un sistema de odometría visual con características dispersas que a partir de una solución iterativa al problema de cuerpo rígido desde el problema de perspectiva de n puntos, con extracción de características por SURF y correspondencia con el algoritmo KNN, hizo posible tener un sistema capaz de realizar estimaciones incrementales del movimiento de la cámara, logrando inferir la trayectoria desde un sistema óptico RGBD. El sistema se acopló con filtros adaptables con algoritmos LMS, NLMS y LMS-SER los cuales, con los parámetros configurados adecuadamente, disminuyeron el error de estimación de pose relativa contra la trayectoria real en comparación con el sistema de odometría visual sin correción alguna. obteniendose como resultado una trayectoria de mayor semejanza al reducir en 0.026895 metros en el mejor de los casos el RMSE, lo que hace más similar la estimación a la trayectoria real del conjunto de datos usado.

En un análisis particular de los módulos que componen el desarrollo de este trabajo se concluye que:

- En cuestiones de relación costo beneficio, el algoritmo SURF, es la mejor alternativa de extractor y descriptor de características de imagen para un sistema de odometría visual dispersa, gracias a la robustez de las características que entrega y su costo computacional.
- La eliminación de características con filtros de distancia reduce el error en la estimación de la matriz de transformación de cuerpo rígido, esto debido al rango estable de funcionamiento del sensor.

- El algoritmo de estimación de la matriz de transformación a partir de la solución del problema de perspectiva de n puntos, tiene un costo computacional reducido considerando que hace una mejora a la estimación directa de la solución por descomposición de valores singulares, aprovechando el conocimiento que se tiene de la matriz de profundidad de las características.
- La simplicidad y el bajo costo computacional de los filtros adaptables con variaciones del algoritmo LMS que en nuestro caso no pasan de doce coeficientes, es decir, un máximo de doce multiplicaciones y once sumas, algo sumamente económico para ser un sistema adaptativo bastante robusto para acoplar a una estimación de matriz de transformación de cuerpo rígido, logrando una reducción en el error de estimación de la trayectoria.

6.0.2. Trabajo futuro

Como trabajos futuros se propone agregar una etapa adicional de refinamiento de pose basándose en la etapa de optimización global o local de las transformaciones, esta etapa permitiría contrarrestar el ruido inducido por la ubicación espacial de las características tanto en el plano bidimensional de la imagen como en el espacio tridimensional de la escena, lo que conlleva a que los errores de las estimaciones incrementales se vean reducidos.

Se plantea la posibilidad de cambiar el método de correspondencia de características por un enfoque más robusto que permita tener un seleccionador de frames óptimos para la estimación del movimiento, lo que permitiría realizar estimaciones cuando se tenga un movimiento en el flujo de la imagen, evitando estar realizando cálculos para transformaciones mínimas, o en el caso extremo, con el sistema en total quietud.

Se propone realizar una investigación sobre el efecto de la información de geometría sobre extractores de características, incluyendo diferentes extractores actuales como ORB, FAST, SIFT y técnicas de flujo óptico.

Finalmente, se propone como trabajo futuro la creación de un repositorio con variaciones de métodos para todos los módulos principales en la metodología de odometría visual por características, adémas comprobar la funcionalidad de todo el sistema global con las combinaciones de los distintos algoritmos para cada submódulo con estadísticas de desempeño y de comportamiento ante entornos dinámicos y cambios sustancialmente considerables en los niveles de iluminación. Esto con el objetivo de robustecer el sistema para ser usado en entornos más reales.

Bibliografía

[1] Y. C. F. Tang Swee Ho and E. S. L. Ming, "Simultaneous localization and mapping survey based on filtering techniques," in *10th Asian Control Conference (ASCC)*, IEEE, 2015. xi, 13

[2] C. E. v. D. Wikus Brink and W. Brink, "Probabilistic outlier removal for robust landmark identification in stereo vision based slam," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2822–2827, October 2012. xi, 16

[3] Y. Z. Jingyi Gao, "An improved iterative solution to the pnp problem," in *International Conference on Virtual Reality and Visualization*, IEEE, 2013. xi, 19, 31, 32

[4] D. Scaramuzza and F. Fraundorfer, "Visual odometry part i: The first 30 years and fundamentals," *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 2011. xi, 20

[5] T. T. L. V. G. Herbert Bay, Andreas Ess, "Speeded-up robust features (surf)," in *Computer Vision and Image Understanding*, p. 346–359, 2008. xi, 22, 23, 24

[6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. xi, xi, xi, 28, 29, 45, 46

[7] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii - matching, robustness, and applications," vol. 19, pp. 78–90, 06 2012. xv, 22, 23, 31

[8] B. Siciliano, *Modelling, planning and control*. springer, 2010. 1

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics & Automation Magazine, IEEE*, vol. 13, pp. 99–110, 2006. 2

[10] H. C. Y. L. D. S. J. N. I. R. Cesar Cadena, Luca Carlone and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE TRANSACTIONS ON ROBOTICS*, vol. 32, December 2016. 4

[11] H. durrant Whyte and T. Bailey, "Simultaneous and mapping part i," *Robotics y Automation Magazine, IEEE*, vol. 32, p. 99–110, 2006. 11

[12] K. A. DeSouza, G.N., "Vision for mobile robot navigation: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 237–267, 2002. 13

[13] S. S. Shin, D.H., "Path generation for robot vehicles using composite clothoid segments.," tech. rep., The Robotics Institute, Internal Report CMU-RI-TR-90-31. Carnegie-Mellon University, 1990. 14

[14] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *Robotics and Automation, IEEE Transactions on*, vol. 7, pp. 376–382, 1991. 15

[15] C.-Y. C. Hsiang-Jen Chien, Chen-Chi Chuang and R. Klette, "When to use what feature? sift, surf, orb, or a-kaze features for monocular visual odometry," in *evaluated Image and Vision Computing New Zealand (IVCNZ)*, 2016. 16, 22

[16] H. J. A. Chiuso, P. Favaro and S. Soatto, "Structure from motion causally integrated over time," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, p. 523–535, 2002. 17

[17] O. N. D. Nister and J. Bergen, "Visual odometry," p. 652–659, 2004. 17, 20

[18] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979. 17

[19] H. P. Moravec, "Rover visual obstacle avoidance," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, (San Francisco, CA, USA), pp. 785–790, Morgan Kaufmann Publishers Inc, 1981. 18, 20

[20] L. C.-N. O.-K. Haralick, R. M. and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, p. 331–356, 1994. 20

[21] L. Quan and Z. Lan, "Linear n-point camera pose determination," *IEEE Transac-tions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, p. 774–780, 1999. 20

[22] L. V. Moreno-Noguer, F. and P. Fua, "Accurate non-iterative o(n) solution to the pnp problem," in *In Eleventh International Conference on Computer Vision (ICCV 2007)*, 2007. 20

[23] Y. Ma, S. Soatto, J. Koeck, and N. Y.-. p. . S.S. Sastry, Springer, *An invitation to 3-D vision from images to geometric models*. Springer, 2004. 20

[24] R. C. S. Lacroix, A. Mallet and L. Gallo, "Rover self localization in planetary-like environments," in *Int. Symp. Articial Intelligence, Robotics, and Automation for Space (i-SAIRAS)*, pp. 433–440, 1999. 20

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, 1981. 20

[26] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," in *IEEE Int. Conf. Vision Systems*, pp. 21–24, 2006. 20

[27] S. G. J. Stühmer and D. Cremers, "Real-time dense geometry from a handheld camera," in *In Joint Pattern Recog- nition Symposium* (Springer, ed.), p. 11–20, 2010. 20

[28] S. J. L. R. A. Newcombe and A. J. Davison, "Dtam:dense tracking and mapping in real-time," in *IEEE International Conference on Computer Vision (ICCV)*, p. 2320–2327, 2011. 21

[29] D. Scaramuzza and R. Siegwart, "Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles," *IEEE Trans. Robot*, vol. vol. 24, pp. 1015–1026, 2008. (Special Issue on Visual SLAM). 21

[30] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, vol. 2, p. 1150–1157, 1999. 22

[31] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 11 2011. 22

[32] H. Eddine Benseddik, O. Djekoune, and M. Belhocine, "Sift and surf performance evaluation for mobile robot-monocular visual odometry," vol. 2, pp. 70–76, 01 2014. 22

[33] K. Khoshelham, "Accuracy analysis of kinect depth data," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science*, 2011. 29

[34] H. J. E. N. S. J. C. D. Endres, F. and W. Burgard, "An evaluation of the rgb-d slam system," in *International Conference on Robotics and Automation*, pp. 1691–1696, 2012. 30