Reconstrucción de mapas por segmentación de rectas en la tarea de navegación de un robot móvil con escáner láser RPLIDAR

Jorge López Ortega. Leslie Janet Carboney Palafox. Francisco Alejandro Chávez Estrada

jlo.lopez.ortega@gmail.com, leslie_carboney@hotmail.com, falexchavez@hotmail.com

Centro de Innovación y desarrollo tecnológico en Cómputo CIDETEC-IPN

Abstract

Actualmente los robots móviles se encuentran involucrados en múltiples tareas que deben llevarse a cabo de manera eficiente, precisa y segura, por ello, es importante el desarrollo de algoritmos robustos que entreguen una correcta interpretación de los datos del entorno adquiridos por los sensores. En el presente trabajo se desarrolla una interfaz gráfica en MATLAB, donde se grafican los datos de mapeo del ambiente obtenidos mediante un escáner láser 360° RPLIDAR montado sobre un robot móvil en la reconstrucción de mapas 2D, combinando y comparando los algoritmos de clustering y extracción de rectas más populares encontrados en la literatura a fin de obtener el algoritmo con mejores resultados.

El objetivo es realizar una reconstrucción total del entorno por segmentación de rectas y no una reconstrucción parcial por nube de puntos en técnicas de SLAM (Simultaneous Localization And Mapping), mejorando así el rendimiento en la navegación al no permitir ambigüedades en la toma de decisiones en posibles caminos a explorar durante la navegación, el reconocimiento de patrones en la búsqueda de esquinas como puntos de referencia para ajuste, o en la búsqueda de objetos o espacios particulares.

Introducción

Los métodos de navegación son importantes en la autonomía de un robot móvil; para que éste interactúe con su entorno y pueda desplazarse de manera eficiente y precisa, éstos métodos deben contener estructuras y formas geométricas que modelen las características del ambiente y aporten información del ambiente en busca de puntos de referencia para la localización y ajuste de errores, así como en la toma de decisiones sobre trayectoria y colisiones durante la navegación.

Actualmente el SLAM es una de las técnicas principales de navegación que involucra la reconstrucción de mapas. Dentro de este campo se han realizado diversas investigaciones para resolver la asociación de datos en la extracción de características del ambiente. Problemas como la correcta interpretación de datos adquiridos por sensores, ha motivado a desarrollar diversos algoritmos dedicados a extracción de rectas, es por ello que en este artículo se prueban diferentes algoritmos dedicados a la tarea de construcción de mapas mediante algoritmos de *clustering* y extracción de rectas, a fin de reforzar los métodos propuestos realizando una comparativa en busca de un algoritmo óptimo que combine eficiencia, robustez y bajo procesamiento computacional.

La Figura 1 muestra algunos robots móviles dedicados a la navegación con ayuda de un escáner láser como herramienta de mapeo del ambiente.

Figura 1. Robots móviles dedicados a la navegación autónoma.

Método de Mínimos Cuadrados

Este método calcula la ecuación que mejor se ajusta a un conjunto de puntos por pares ordenados, de manera que se encuentre la mejor aproximación con mínimo error cuadrático. En este caso, es implementado en la segmentación de rectas para la reconstrucción de mapas, ajustándose a la mayor cantidad de puntos sobre secciones que representen muros, muebles, puertas, ventanas, etc.

No obstante al aplicarse sobre una nube de puntos como representación de un plano en 360 grados presenta desventajas como: si entre mediciones se llega a encontrar puntos erróneos *outliers* (puntos distantes de un conjunto definido) la recta estimada será incorrecta así como en presencia de puntos que generen una recta vertical ya que se vuelve nula la suma de desviaciones de los puntos, por lo que no se obtiene una recta en dichas condiciones.

Por esta razón este método no es usado para la reconstrucción de mapas, sino únicamente como herramienta de ajuste sobre puntos previamente agrupados y filtrados. La Figura 2 a) muestra el error obtenido para una sección de puntos sin clasificar y b) la desviación por presencia de *outliers*, ambas rectas color rojo respecto a las rectas verdes sin desviación.

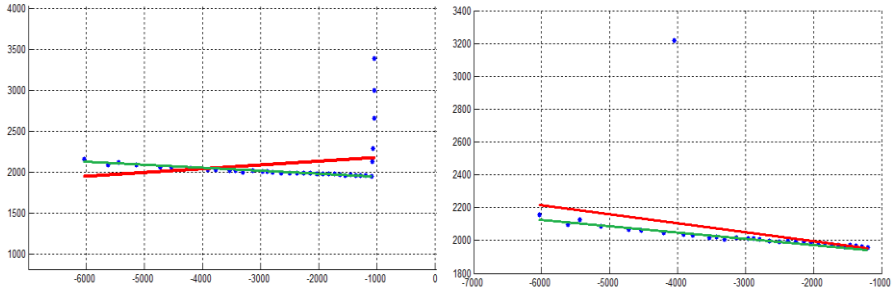

Figura 2. Ajuste por mínimos cuadrados de a) puntos sin clustering b) puntos outliers.

Algoritmo *Line Tracking*

La popularidad de este algoritmo se debe a su simplicidad al ser un proceso secuencial y rapidez en la clasificación de puntos colineales. El algoritmo calcula la ecuación de la recta para un grupo reducido de puntos y mide la distancia entre los puntos sucesivos a la recta. Si los puntos se encuentran dentro de un umbral establecido T_{max}, se incorporan a dicho grupo; en caso contrario se crea un nuevo grupo y se repite el proceso evaluando los puntos.

El algoritmo obtiene los parámetros de la recta que mejor se ajusta por método de mínimos cuadrados, por lo que se encuentra la mejor recta. Sin embargo, elegir un umbral T_{max} adecuado es difícil debido a que puede generar errores en la reconstrucción, formando más segmentos de los requeridos o rectas donde no existen puntos. La Figura 3 a) muestra segmentos de recta erróneos para un umbral de 150 mm, por lo que puntos dentro del grupo dejan de ser colineales, b) con umbral de 30 mm, menos puntos se consideran parte de la recta por lo que no hay distinción, y c) con umbral de 55 mm se obtiene mejor agrupación.

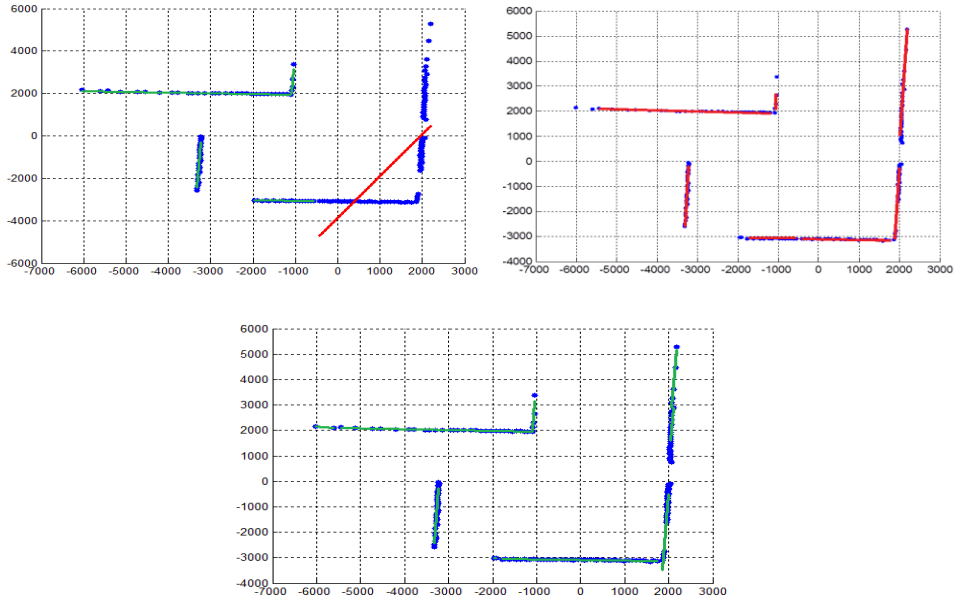

Figura 3. Algoritmo LT con límite de umbral a) 150 mm, b) 30 mm y c) 55 mm

Algoritmo *Iterative End Point Fit (IEPF)*

La ventaja de este algoritmo es su bajo proceso en la obtención de la recta por dos puntos, el punto inicial y final de cada sección a través de la definición de la ecuación de distancia del punto a la recta, y no en el cálculo de ajuste de recta por conjunto de puntos establecidos por la ecuación de distancia del punto a la recta (algoritmo Line Tracking LT). Sin embargo, tiene la misma sensibilidad a puntos *outliers* estableciendo rectas erróneas, como la construcción en secciones donde se encuentra puertas y ventanas abiertas como se muestra en la Figura 4 en color rojo.

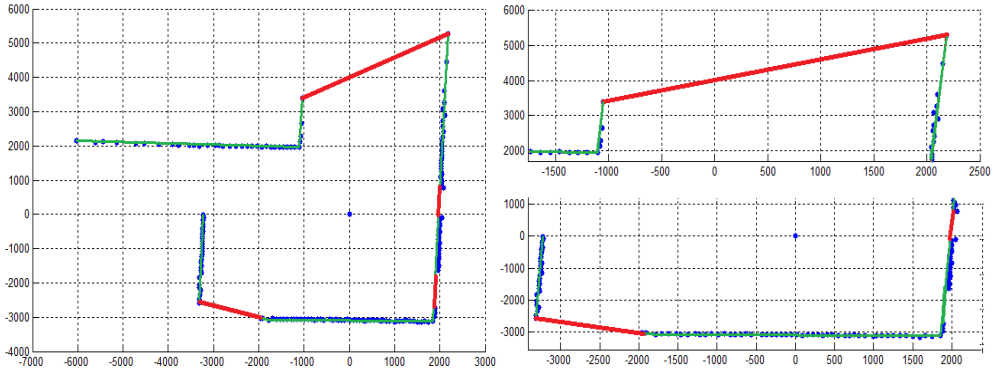

Figura 4. Algoritmo IEPF.

El algoritmo tiene gran robustez en la clasificación de puntos colineales, sin embargo, no es apto aplicarse en un conjunto global de puntos por la injerencia de espacios vacíos como de puertas y ventanas abiertas y pasillos largos.

Algoritmo *Split And Merge*

Este algoritmo es uno de los más populares como herramienta en técnicas de navegación de robots basada en un sistema de medición por láser. Este método es una modificación al algoritmo IEPF dividiéndolo en dos etapas. La primera etapa radica en reducir el grupo de puntos en secciones de puntos colineales lo cual realiza IEPF, y la segunda etapa se encarga de unir las secciones por rectas que se ajusten a dicho grupo de puntos como método de mínimos cuadrados a diferencia de IEPF que une las rectas por el punto inicial y final de cada sección.

Sin embargo, aunque las rectas tienen un ajuste con base en la distribución de los puntos, al definirse la ecuación de la recta, las líneas no están bien definidas por una pendiente elevada y la ordenada al origen cuando se trabaja en escala de 10^3. La Figura 5 muestra el mismo comportamiento que el algoritmo IEPF pero con errores en el ajuste de la recta debido a la pendiente elevada (vertical).

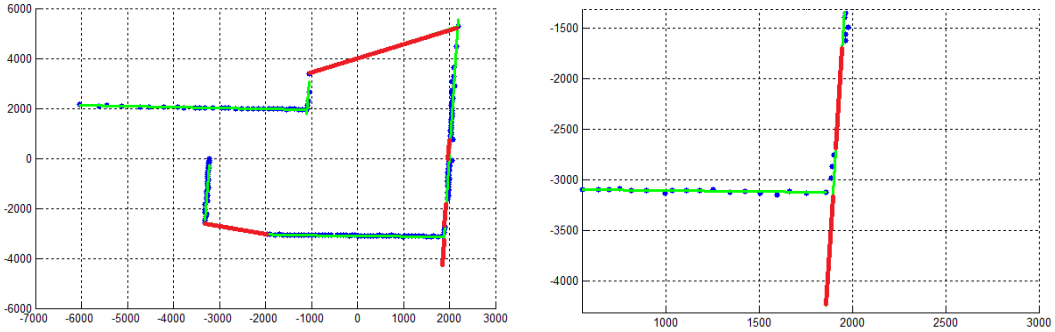

Figura 5. Algoritmo Split and Merge.

La Figura 6 muestra la diferencia en una misma sección de puntos. Mientras que en *Iterative End Point Fit* el segmento se une por los puntos inicial y final, en *Split and Merge* la recta está definida por su ajuste.

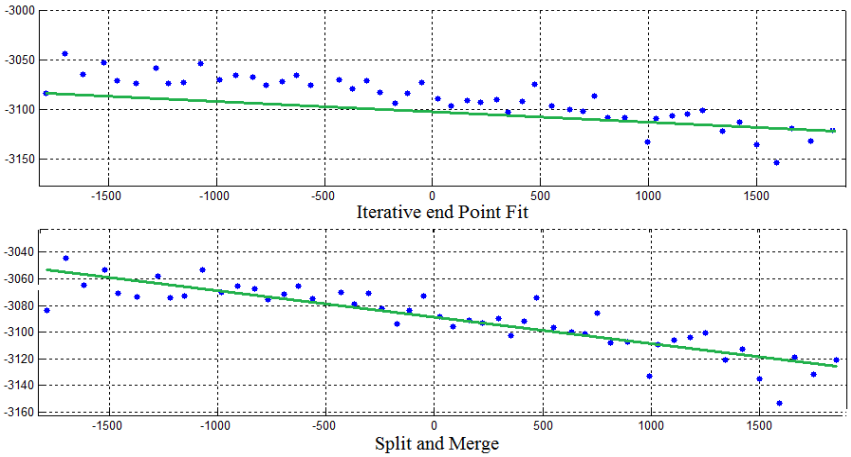

Figura 6. IEPF por unión de puntos extremos y SAM por ajuste de puntos.

Algoritmo *Hough Transform*

Diseñado en detección de fronteras en procesamiento de imágenes, el algoritmo utiliza la ecuación paramétrica de una recta dada por la Ecuación 1 donde ρ es la distancia del origen (0,0) a la línea a lo largo de un vector perpendicular y θ el ángulo entre el eje x y dicho vector ρ. Una línea es representada por un punto en el espacio de coordenadas (ρ, θ).

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \qquad \text{Ecu. 1}$$

El programa aquí implementado es una combinación del algoritmo IEPF para la agrupación de puntos que forman una única sección, y el algoritmo de Hough que encuentra la mejor aproximación a la recta para una serie de puntos que satisfacen la ecuación de la recta con los valores de ρ y θ donde se encuentra el mayor número de intersecciones. La Figura 7 a) muestra la gráfica de distintas secciones del mapa y la recta obtenida a partir de ρ y θ, y en b) la representación paramétrica de dichos puntos donde se obtiene ρ y θ por intersección.

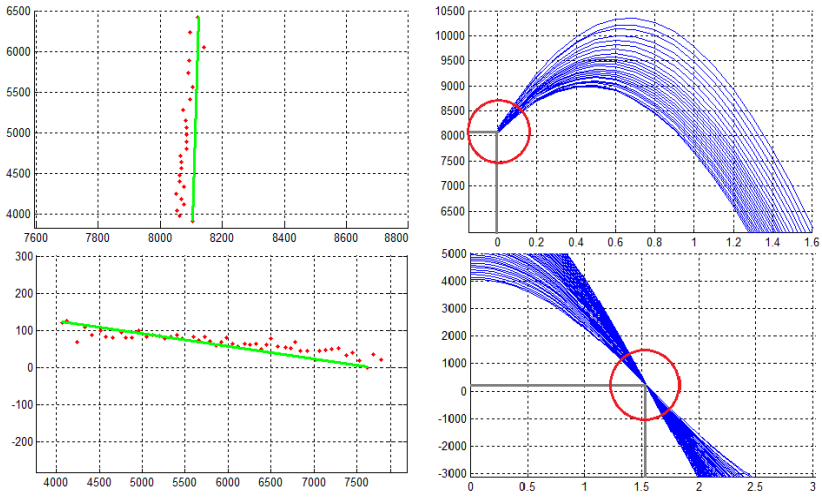

Figura 7. Obtención de rectas por algoritmo de Hough.

Se realiza una ventana por cada grupo de puntos con ancho y alto definido por los puntos máximos y mínimos de cada grupo, obteniendo ρ para ángulo θ en 360º encontrando la coordenada (ρ, θ) de cruce. La Figura 8 muestra la recta definida para cada ventana con respecto al plano completo de construcción.

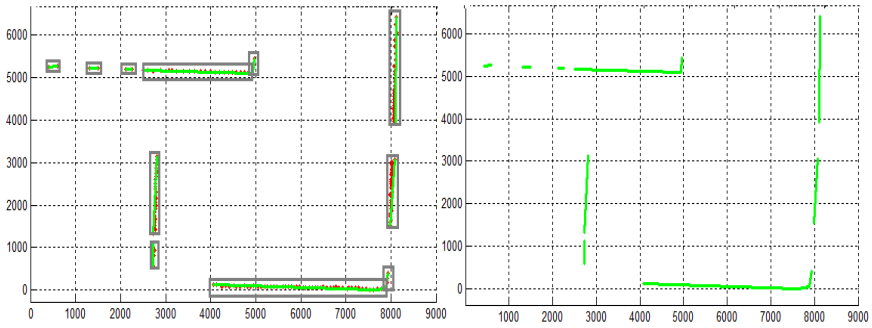

Figura 8. Reconstrucción de mapas por ecuación paramétrica.

El algoritmo requiere de la integración de un clasificador (**cluster**) para la agrupación de puntos por discontinuidad, mostrando así rectas definidas por puntos de una única sección de superficie (Figura 9).

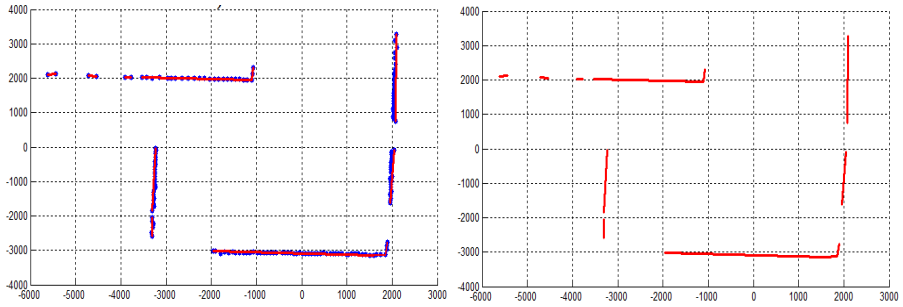

Figura 9. Reconstrucción de mapas por Clasificador y algoritmo IEPF.

Conclusiones

Al realizar el análisis de los cuatro algoritmos implementados en este trabajo, se encontró que el **Algoritmo de Hough** combinado con *clustering* básico entrega los mejores resultados debido a su robustez al encontrar la recta óptima y no ser afectado por puntos de ruptura *outliers*. Sin embargo, el costo computacional es elevado en un 400% más respecto a otros algoritmos dedicados únicamente a puntos de mapeo 2D, afectando tiempos de ejecución durante la navegación.

Una solución alternativa es el **Algoritmo Iterative End Point Fit**, este algoritmo entrega excelentes resultados debido a que está dedicado a puntos de medición 2D previamente ordenados y tiene un error de medición mínimo respecto a otros sensores descartando puntos *outliers*, extrayendo rectas con excelente aproximación sin aplicar ajuste de puntos en un menor tiempo de ejecución.

Referencias y Recursos electrónicos

Caramés, C. F. (2007). Técnicas de Navegación de Robots Basadas en Sistemas de Medición por Láserr. *Departamento de Informática y Automática, Universidad de Salamanca, Salamanca, España*.

Narváez, Y., Javier, F., Tipantaxi, N., & Julio, V. (2013). Diseño e Implementación de un Sistema de Localización y Mapeo Simultáneos (SLAM) para la Plataforma Robótica ROBOTINO® (Doctoral dissertation, QUITO/EPN/2013).

Berrio, J. S., Ordoñes, S. A. O., & Bravo, E. F. C. (2012). Extracción de Líneas a partir de Escaneos Láser integrando Transformada de Hough, Mínimos Cuadrados Totales y Seguimiento de Bordes Sucesivos. Ingeniería, 17(1), 49-59.

Borges, G. A., & Aldon, M. J. (2004). Line extraction in 2D range images for mobile robotics. Journal of intelligent and Robotic Systems, 40(3), 267-297.