

Diseño e implementación de un sistema de visión artificial usando una técnica de mapeo y localización simultánea (SLAM) sobre una plataforma robótica móvil

Design and implementation of an artificial vision system using a slamming technique on a mobile robotic platform

Garzón Obregón, Luz Karime; Forero Rincón, Luis Albero; Duque-Suárez, Oscar Manuel

Luz Karime Garzón Obregón
Universidad de Pamplona, Colombia
Luis Albero Forero Rincón
Universidad de Pamplona, Colombia
 **Oscar Manuel Duque-Suárez**
omduke@hotmail.com
Universidad de Pamplona, Colombia

Mundo Fesc
Fundación de Estudios Superiores Comfanorte, Colombia
ISSN-e: 2216-0388
Periodicidad: Semestral
vol. 8, núm. 16, 2018
revista_mundofesc@fesc.edu.co

Recepción: 23 Febrero 2018
Aprobación: 01 Junio 2018

URL: <http://portal.amelica.org/ameli/journal/170/1704827001/>

Resumen: El SLAM es uno de los algoritmos que han nacido con la intención de ser capaz de posicionar al robot en un entorno desconocido a la misma vez que navega por él y usando dicha información precedente de sus sensores construir un mapa del ambiente en el que se encuentra. Para lograr el mapeo, diversas investigaciones han usado sensores de tipo ultrasonido, LIDAR, hasta cámaras usando visión artificial. El objetivo de esta investigación está centrado en aplicar una técnica de SLAM sobre una plataforma robótica móvil. Para lograr dicho objetivo se realiza un estudio de diferentes metodologías de SLAM existentes que pudieran cumplir con las expectativas resultando elegido el LSD-SLAM, la cual con una cámara adecuada se logra obtener un mapa semi-denso 3D del ambiente en el cual se trabaja. Además se desarrolla en ROS (Robot Operating System) definido como sistema operativo, que cada vez debería ser tenido en cuenta para futuras investigaciones gracias a sus grandes prestaciones. Una vez investigado, seleccionado e implementado se dio inicio a la programación del sistema por medio de herramientas y librerías las cuales tienen soporte, en cuanto a documentación confiable y previamente programadas para su uso.

Palabras clave: visión artificial, sistema de percepción, SLAM, software, LSD_SLAM.

Abstract: The SLAM is one of the algorithms that were born with the intention of being able to position the robot in an unknown environment while navigating through it and using the above information from its sensors to build a map of the environment in which it is located. To achieve the mapping, several investigations have used sensors of ultrasound type, LIDAR, even cameras using artificial vision. The objective of this research is focused on applying a SLAM technique on a mobile robotic platform. To achieve this objective, a study is made of different existing SLAM methodologies that could meet the expectations resulting in the LSD-SLAM, which with a suitable camera is able to obtain a semi-dense 3D map of the environment in which it works.

It also develops in ROS (Robot Operating System) defined as operating system, which should be taken into account for future research thanks to its great features. Once researched, selected and implemented, the programming of the system was started by means of tools and libraries which have support, in terms of reliable documentation and previously programmed for their use.

Keywords: Static analysis, 3D printing, transradial prosthesis, Myoelectric system.

Introducción

“Las Tecnologías de la Infomación y la Comunicación surgen como apoyo a través de software en los que se interactúa desde cualquier computador con acceso a internet” [1]. Según Gualdrón “la revolución tecnológica actual y el interés multidisciplinar en desarrollar metodologías y productos que mejoren los resultados obtenidos en el reconocimiento de patrones en su aplicación en el reconocimiento y clasificación de características de rostros y de esta manera perfeccionar la emulación del reconocimiento de rostros propios de los seres humanos a fin de ser incorporados en múltiples aplicaciones tales como la interacción robótica social y la seguridad; entre otras” [2-3].

En la robótica, el principal problema se enfrenta en la localización a la hora de conseguir robots autónomos capaces de navegar por un espacio [4]. La mayoría de los sistemas SLAM estaban centrados en la construcción de mapas exclusivamente en 2D, consiguiendo buenos resultados, sin embargo, recientes investigaciones han demostrado una gran utilidad de los métodos de SLAM basados en cámaras monoculares, como estéreo para aplicaciones que tienen como objetivo obtener en tiempo real la posición en 3D.

El problema del SLAM surge cuando el robot desconoce tanto su posición, como el mapa de la zona en la que se encuentra, únicamente tiene acceso a la información suministrada por sus sensores, por lo tanto, cuando se plantea este problema, el robot tiene que ir construyendo un mapa del entorno al mismo tiempo que se localiza en el espacio. La localización es un problema que podría resolverse de manera independiente al mapeado empleando la odometría. Sin embargo, la odometría suele presentar errores, por lo que el robot no estará donde se cree. Por este motivo se aborda el problema de la localización junto con el mapeado del entorno.

El trabajo busca el diseño de un sistema de visión artificial mediante el desarrollo de una técnica capaz de proporcionar la localización y mapeo sobre una plataforma móvil en entornos no estructurados ni conocidos, para realizar labores de inspección y generación de mapas en 3D [5].2.

Materiales y métodos

Sistema de visión artificial

El sistema de percepción es también denominado sistema de visión artificial el cual permite que el robot sea capaz de hacer frente a situaciones cambiantes del entorno, así como a reaccionar ante posibles eventos inesperados mientras

navega, lo que exige la utilización de un sistema sensorial que suministre la información del entorno. Esta información requerida debe permitir al robot realizar tres tareas fundamentales: estimar su posición y orientación, mantener actualizado el mapa del entorno y detectar los posibles obstáculos.

Así, mientras que para la estimación de la posición y la construcción (o actualización) del mapa del entorno se tienen en cuenta sobre todo características como la precisión, resolución espacial, alcance, entre otras. Según [6], se puede decir que la virtualización, “es una abstracción de los recursos tecnológicos en donde se puede llegar a utilizar un servidor o muchos servidores siendo invisible para el usuario final”.

Es importante realizar una selección adecuada del sistema de percepción debido a que esta enlazado con la navegación y la reconstrucción del mapa, pues la apropiada elección garantizó el éxito de la tarea designada.

A continuación, se presentan los criterios de diseños establecidos para la selección del sistema de visión artificial:

- Alcance
- Rango
- Robustez ante vibraciones
- Consumo
- Economía (costo)
- tiempo de respuesta

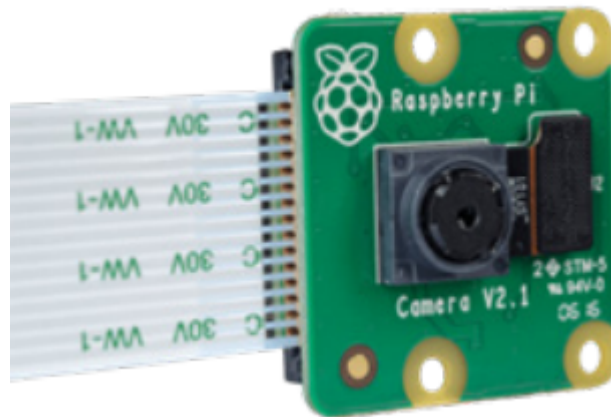


Figura 1
Cámara Raspberry pi versión 2
[4]

Raspberry pi 3 modelo B

Para usar la raspicam versión 2 fue necesario el uso de una tarjeta de procesamiento raspberry pi 3 modelo B (figura 2). A continuación, se profundizó sobre las principales características de esta tarjeta.



Figura 2
Raspberry pi 3 modelo B Fuente 5
[5]

Características

- Procesador a 1.2 GHz de 64 bits con cuatro núcleos
- 1 GB de memoria
- Bluetooth 4.1
- 802.11n Wireless LAN
- 4 puertos USB.
- 40 pines GPIO.
- Puerto Full HDML.
- Puerto Ethernet.
- Conector combo compuesto de audio y vídeo de 3,5 mm.
- Interfaz de la cámara (CSI).
- Interfaz de pantalla (DSI).
- Ranura para tarjetas microSD (ahora push-pull en lugar de push-push).

Selección del software

Existen diferentes métodos de mapeo, definiéndose el “SLAM (Simultaneous Localization and Mapping)” como el método usado cuando generamos el mapa mediante una exploración del entorno [7]. Como su nombre lo dice, requiere de realizar en paralelo al proceso de mapeo, un proceso de localización ya que la posición es una variable necesaria a la hora de generar el mapa con este método. Pero todo este proceso se conecta directamente con el software debido a que este, procesa toda la información, de manera que, es importante haber seleccionado con base en ello.

A continuación, tenemos las características más importantes que comprende la selección de un software, cuya matriz se basó en la familia de normas ISO/IEC 25000 el cual, se usó ISO/ IEC 9126 cuyo objetivo principal es guiar el desarrollo de los productos de software mediante la especificación de requisitos y evaluación de características de calidad. A continuación se señalan los criterios:

- Funcionabilidad

- Exactitud, Seguridad
- Datos de entrada
- Modularidad
- Usabilidad
 - Ayuda y documentación
 - Facilidad de comprensión
 - Tiempo de aprendizaje
- Fiabilidad
 - Recuperabilidad
 - Tolerancia a fallos
- Portabilidad
 - Adaptabilidad
 - Instalabilidad
- Requerimientos
 - Capacidad del software
 - Técnicos
 - Documentación del sistema
 - Seguridad del sistema
 - Confiabilidad del sistema
 - Sistema operativo
 - Rapidez de compilación
- Lenguajes de programación

Los entornos de trabajo que fueron evaluados son los siguientes:

- Matlab
- ROS
- OpenCV

Como resultado a esta matriz de selección el entorno de trabajo con la mayor puntuación es ROS.

ROS es considerado como un sistema operativo de código abierto, y como tal aporta los servicios que caben esperar de un sistema, incluyendo abstracción de hardware, controladores de dispositivos a bajo nivel, librerías, herramientas de visualización, comunicación por mensajes, administración de paquetes mediante comandos desde terminal, compilación y ejecución de archivos bajo la licencia open source BSD [8].

Presenta gran flexibilidad en cuanto al uso de distintos lenguajes de programación, como lo son, Python, C++, Lisp, Lua o JAVA. Puede ser ejecutado sobre máquinas tipo Unix, como Ubuntu y MAC OS X [8].

Selección de la técnica

Existen múltiples técnicas, métodos o algoritmos para realizar SLAM desarrollados hasta el día de hoy, los cuales se realizan con diversos tipos de sensores, al ser elegido anteriormente como sistema de percepción una cámara monocular se debe realizar una selección de la técnica de este tipo.

Las técnicas monoculares existentes, como LSDSLAM, PTAM, ORB-SLAM, SVO, DTAM, y aunque existen muchas más, estas son las más relevantes [9].

Por esta razón la técnica que seleccionó fue LSD_SLAM porque cuenta con las siguientes características y mayor puntaje.(Tabla I).

Tabla 1 Características

Tabla 1
Características

Probabilístico	No
Hilos de ejecución	3
Emparejamiento	Métodos directos
Puntos 3D con incertidumbre	Sí
Mapa inicial	Incertidumbre
Keyframes	Sí
Puntos en el mapa	Miles
Mapa denso	Sí
Gestión de mapas grandes	Sí
Relocalización	Sí
Rechazo de espurios	Sí
Cierre de bucle	Sí

LSD_SLAM

LSD-SLAM (Large Scale Direct Monocular SLAM) también conocido como SLAM Monocular Directo de Larga Escala, es un método directo, no utiliza características de la imagen, sino que usa toda la información disponible, además estima la posición (localización de la cámara) de manera precisa basada en el alineamiento directo de las imágenes para luego reconstruir el ambiente en 3D en tiempo real como un conjunto de cuadros clave que tienen asociados mapas semi-densos de profundidad [10].

El método se puede dividir en tres componentes:

- Rastreo: se mantiene estimando la posición de la cámara respecto al cuadro clave actual, utilizando la posición del cuadro anterior como inicialización.
- Estimación del mapa de profundidad: se utilizan los cuadros rastreados previamente para refinar o reemplazar el cuadro clave actual, luego la profundidad se obtiene a partir de las comparaciones de la secuencia de imágenes adquiridas.
- Optimización del mapa: se toma en consideración el cuadro clave refinado o reemplazado para incorporarlo al mapa global, el cual se encarga de detectar si se volvió a la posición inicial (cierre de trayectoria), también busca corregir el error de estimación a través una transformación de similitud 3D [11].

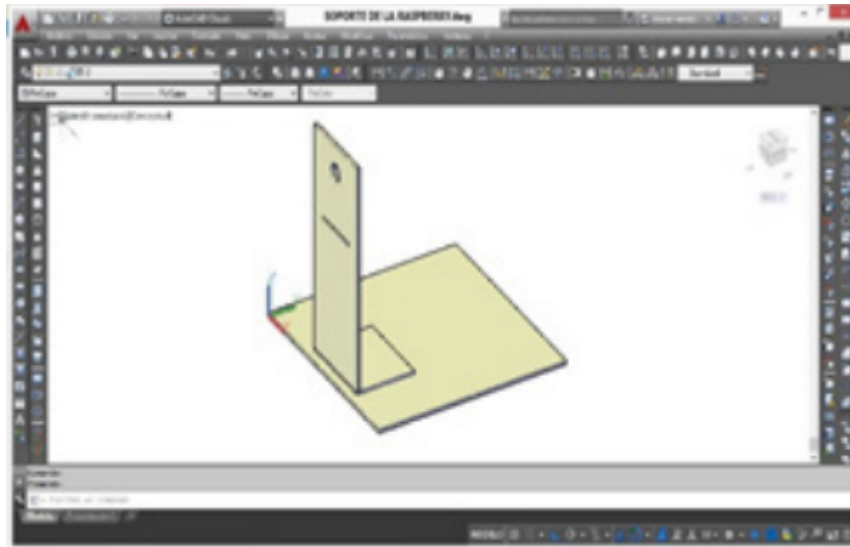


Figura 3
CAD del soporte de la Raspicam

Por otra parte, la calibración es un proceso que permite obtener una serie de parámetros que definen la forma en la que una cámara interactúa con el entorno, el modo en que se ve afectado por las imperfecciones presentes en el hardware (lentes, sensor), y como corregir estas alteraciones.

Anteriormente se hizo la selección del sistema de percepción que es un módulo de cámara Raspberry Pi versión 2, El objetivo principal es percibir todo lo que se encuentre en su entorno para ir creando un mapa en una perspectiva tridimensional.

El nodo cameracalibration.py en ROS admite la calibración de una cámara monocular. Para calibrarla se utilizó un tablero de ajedrez de 6x4 cuya dimensión característica (lado de un cuadrado del tablero) es de 0.03 m., por lo que la instrucción de la consola se muestra en la figura 4.

```
rosrun camera_calibration cameracalibrator.py --size 6x4 --square 0.03
image:=/camera/image camera:=/camera
```

Figura 4
Código de calibración

Esto abrirá la ventana de calibración que resaltará el tablero de ajedrez, no verá ninguna imagen en la ventana de calibración hasta que esté presente un tablero de ajedrez. Figura 5.



Figura 5
Calibración en ROS

Desarrollo del Algoritmo de mapeo y localización simultánea (SLAM). El algoritmo se divide en tres grandes componentes: seguimiento de cuadros (Tracking), estimación del mapa de profundidad (Depthmap estimation) y optimización del mapa como se muestra en la figura 6.

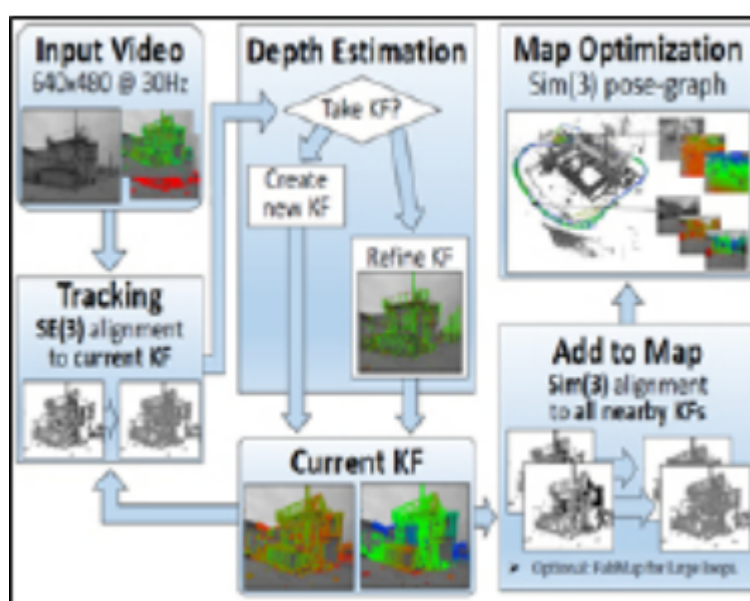


Figura 6
Algoritmo LSD-SLAM
[12]

- El componente de seguimiento (Tracking) realiza un seguimiento continuo de las nuevas imágenes de la cámara. Es decir, estima la posición de su cuerpo rígido ξ $SE(3)$ (espacio euclídeo 3 dimensiones) con respecto al fotograma clave actual, utilizando la posición del fotograma anterior como inicialización. Para el seguimiento de los Poses se utiliza una mejora del espacio euclídeo especial $SE(3)$. La mejora consiste en

aplicar el algoritmo de Gauss-Newton con pesos sobre ese espacio para obtener el espacio sim (3).

Los cálculos de pasar de un espacio a otro no representan una sobrecarga adicional de cómputo.

Para el cierre de lazo (loopclosure) se utilizan varios posibles cuadros para comparar.

Una vez se tienen los nodos programados y ejecutándose se pueden visualizar de manera más detallada con la herramienta `rqt_graph` de la línea de comandos de ROS.

para evitar falsos cuadros de cierre de camino se compara también de forma estadística el error.

Finalmente se usa el de menor error para cerrar el camino.

- El componente de estimación del mapa de profundidad (Depthestimation) usa los frames obtenidos para refinar o sustituir el actual keyframe. La profundidad se refina filtrando muchas comparaciones estéreo de línea de base pequeña por píxel. Si la cámara se mueve demasiado lejos del mapa existente es necesario crear un nuevo keyframe del frame mas reciente que se haya obtenido en el bloque de tracking. Una vez se ha elegido el frame que se convertirá en fotograma clave, su intensidad se inicializa proyectando los puntos desde el anterior keyframe sobre este. Por último, se reemplaza el fotograma y este se usa para el seguimiento de los nuevos frames. Aquellos frames que se obtienen pero que no se convierten en keyframe se utilizan para refinar el keyframe actual. El resultado se incorpora al mapa actual de intensidad, lo que significa que se añaden al mapa nuevos píxeles. Una vez que un fotograma clave se sustituye como referencia de seguimiento - y por lo tanto su mapa de profundidad no se refinará más - se incorpora en el mapa global por el componente de optimización de mapas.

Paquetes, nodos y temas

Se utilizaron varios paquetes que permitieron el desarrollo de la técnica de LSD_SLAM, partiendo inicialmente del robot móvil diferencial, se usó el paquete `rosserial` para la comunicación entre ros y el Arduino, `learning_joy` para la Teleoperacion. Para el mapeo en 3D, se usó la paquetería de `lsd_slam_viewer` y `lsd_slam_core`, todo esto en conjunto con el nodo `tesis_principal` como se muestra en la figura 7, figura 8.

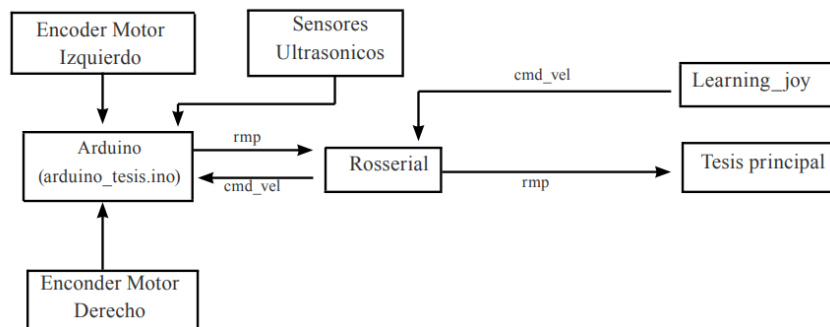


Figura 7
Diagrama de flujo de la paquetería de Ros del robot móvil
[12]

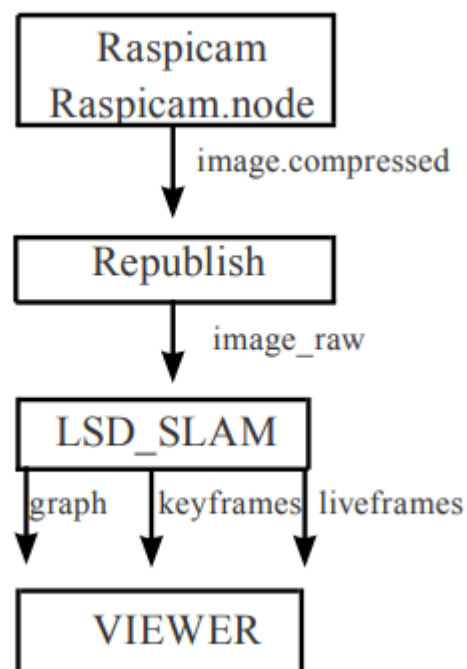
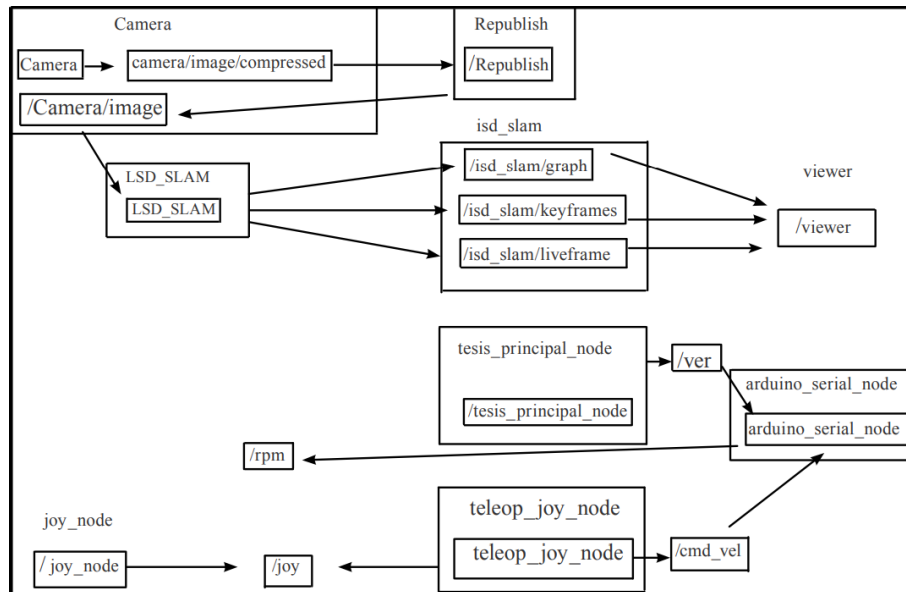


Figura 8
Diagrama de flujo de la paquetería de Ros del LSD_SLAM

**Figura 9**

Nodos y tópicos del LSD_SLAM y el robot móvil

En la figura 9 se visualiza el proceso que lleva a cabo el LSD_SLAM junto con el robot móvil, en ella se observa todos los nodos y los tópicos publicados y suscritos. El nodo “camera” se encarga de publicar la imagen comprimida en JPEG de la raspicam y también publica la imagen recibida luego de ser descomprimida por el nodo “republsh”. El nodo “LSD_SLAM” es el encargado del procesamiento del algoritmo de SLAM, el cual publica los tópicos (lsd_slam/keyframes, lsd_slam/graph y lsd_slam/liveframes), dichos tópicos son suscritos por el nodo “viewer” quien se encarga de visualizar la nube de puntos.

Por otra parte se observa también en el diagrama el nodo “joy_node” quien publica el tópico /joy, el cual entrega las instrucciones que le indican al nodo “teleop_joy_node” que velocidad debe publicar el topic /cmd_vel, en pocas palabras esta parte es la encargada de determinar la velocidad lineal y velocidad angular de cada llanta.

El nodo “arduino_serial_node” se encarga de realizar el control PID de velocidad de los motores y publica las rpm actual de cada llanta, esta información es suscrita por el nodo “tesis_principal_node”, en el cual se realiza el cálculo de la edometría mecánica y publica la posición en X, la posición en Y y la rotación, mediante el tópico /ver, esta información se suscribe a su vez al nodo de arduino.

Resultados y Discusión

En esta sección se presentaron los resultados obtenidos de las pruebas experimentales que se le realizaron al sistema de visión artificial para optimizarlo de la mejor forma, las pruebas realizadas se evidenciaron en capturas de pantalla.

Máquina de prueba

Las especificaciones técnicas de la maquina utilizada para la realización de esta investigación

Tabla 2 Máquina de Prueba

Tabla 2
Máquina de Prueba

Nombre	Características
Procesador	Intel Core i7- 4500 U CPU 1.80 GHz x 4
RAM	8 GB
Año	2013
Distribución	Ubuntu 14.04 LTS x86_64
Kernel Linux	4.4.0-119
Versión de ROS	Indigo

prueba realizada LSD_SLAM

Para la prueba 1 se tomó el siguiente escenario mostrado en la figura 10. Se tomaron varias capturas del resultado del mismo entorno pero con diferentes perspectivas. En la figura 48 se visualizan dos ventanas, la imagen izquierda muestra el mapa de puntos generado (PointCloudViewer) y la imagen derecha es el mapa de profundidad. Se trabajo con una resolucion de 640 x 480 con una velocidad de 30 fps.

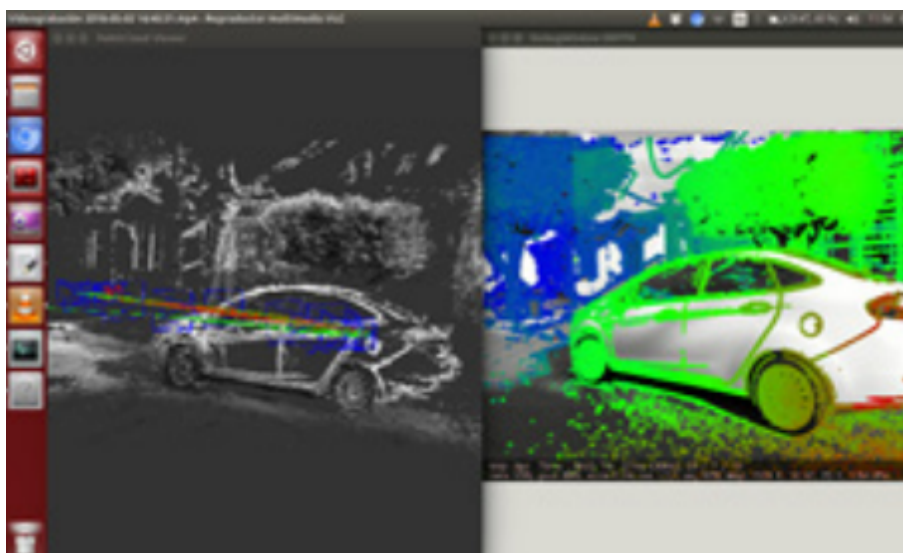


Figura 10
Resultado de la técnica LSD_SLAM

En la figura 10 se observa en la parte derecha una buena toma de puntos de referencia de profundidad, lo que nos da como resultado un mapa semi-denso con una buena representación 3D observable en la parte izquierda de la imagen, en él también se observa las poses de la cámara y su trayectoria.

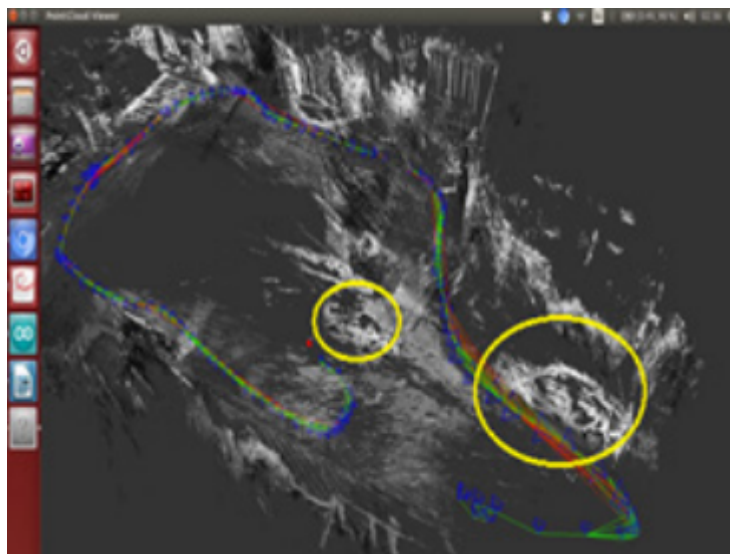


Figura 12

Vista superior del mapa generado después del loopclosure.

En la figura 12 se evidencia, como mediante el “loopclosure” (cierre de ciclo), se logra corregir el mapeo y obtener una mejor representación 3D de la escena.

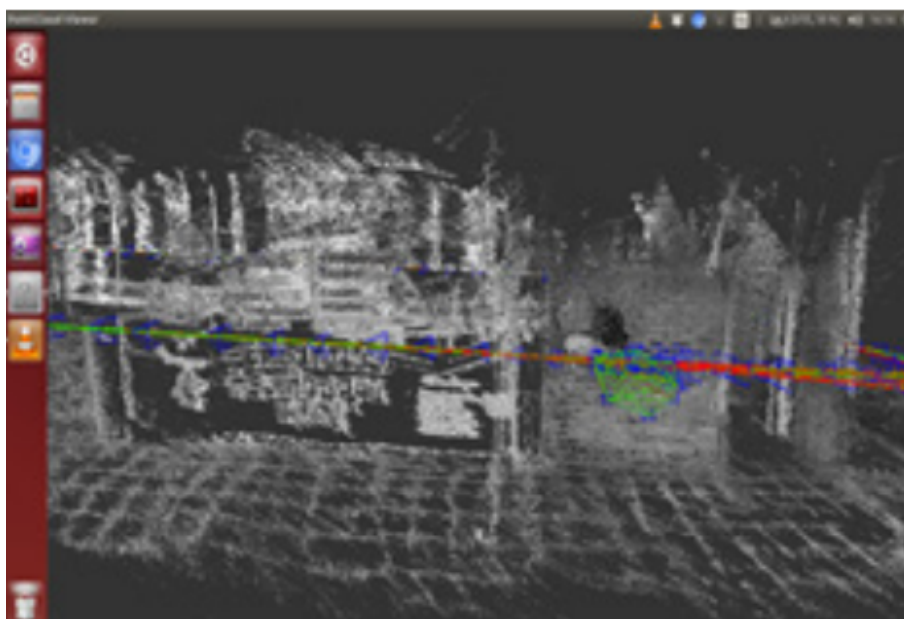


Figura 13

Prueba de la técnica de LSD_SLAM

La prueba 2 se realizó en el pasillo del primer piso del bloque PT de la universidad de pamplona. En la figura 13 se observa el resultado obtenido de la prueba 3, en ella se logra identificar parte del mural a las afueras del laboratorio PT-102 y la entrada a este.

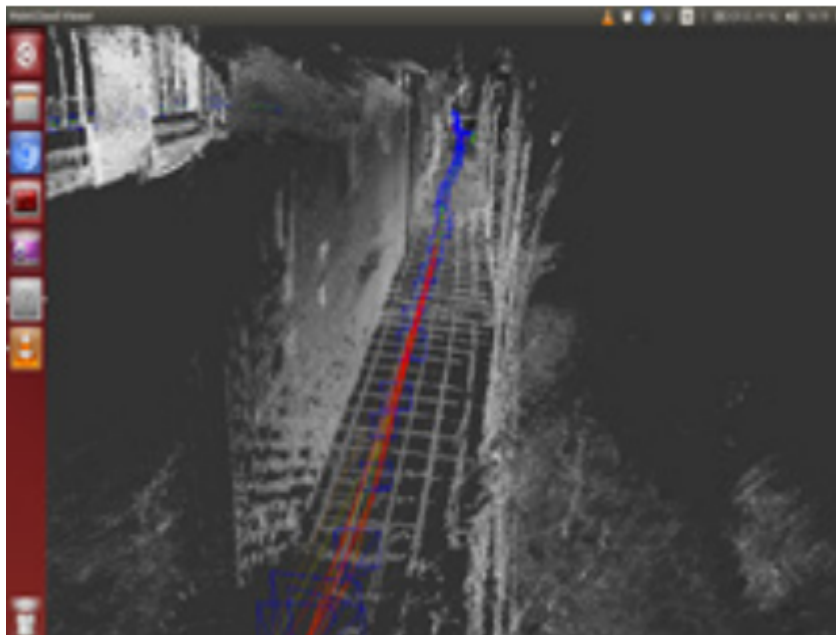


Figura 14.

Prueba de la técnica de SLAM en el pasillo

En la figura 14 se evidencia las poses calculadas por el algoritmo además de la trayectoria que realizó al momento del mapeo. La imagen se refiere al pasillo que queda en frente de los laboratorios pt-104 y pt-102 del primer piso del bloque patios.

La realización de este mapeo tardó alrededor de 14 minutos recorriendo cerca de 60 metros, logrando obtener una representación 3D en un solo circuito. se pueden detallar en la tabla II.

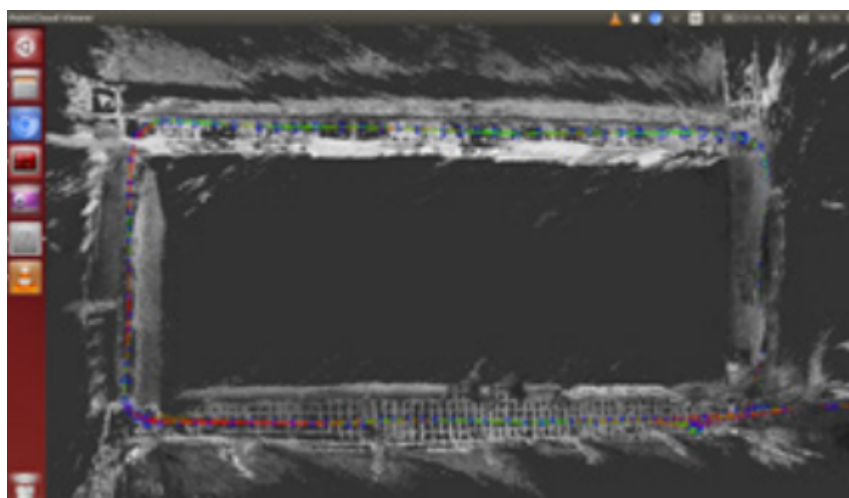


Figura 15

Vista superior del mapa terminado

En la figura 15 se muestra la vista superior del mapa obtenido de la prueba número 3, en la cual se logra identificar la trayectoria calculada de la cámara y las poses de esta todo esto gracias al algoritmo de la edometría que realiza el método LSD-SLAM.

Conclusiones

Una buena selección de los criterios de diseño basados en una norma reconocida como lo es ISO/IEC 9126 permite tener una base sólida para seleccionar el entorno de trabajo más acorde para el sistema de visión. De la misma manera se realizó para el sistema de percepción (cámara) pero basándose en las características técnicas, económicas y de disponibilidad.

Realizar una matriz de selección es importante porque permitió elegir correctamente el sistema de percepción y software, evitando resultados erróneos, donde se condensa toda la investigación encontrada. La matriz se dividió en las características más importantes, cada ítem se le dio un peso para que el proceso fuera más transparente arrojando así buenos resultados, pues no hubo pérdida de tiempo en pruebas con otros elementos que no se acoplaran a la necesidad del proyecto.

Al comparar los resultados obtenidos en este trabajo con un proyecto de investigación realizado en por [14], se comprueba que al tener una mayor capacidad de cómputo, se obtienen mapas con mejores cualidades y además el tiempo de ejecución disminuye.

El algoritmo LSD-SLAM presenta desventajas como la sensibilidad a la pérdida de su localización frente a giros fuertes o movimientos rápidos, además de la pérdida de escala en el mapeo, pero este último lo logra compensar con el cierre de bucle.

El entorno de trabajo en el que se implementó la técnica LSD-SLAM fue ROS (Robot OperatingSystem) Indigo [15],[16] permitió desarrollar funciones de control, SLAM, calibración de la cámara, de manera eficiente y rápida. Esta herramienta tiene un impacto positivo en proyectos que utilicen vehículos aéreos no tripulados o vehículos en general no solo porque disminuye los costos de producción, sino también tiempo. Además, se encuentra con diversidad de lenguajes de programación permitiendo hacer uso de las ventajas de cada uno dependiendo de lo que se vaya a usar.

Con los avances en la computación, la electrónica y la robótica cada vez se realizan robots más pequeños con grandes funcionalidades, en este proyecto se usó un robot hecho en base al robot Turtlebot 2, el cual posee grandes dimensiones en cuanto a su tamaño, siendo esta una limitante y por eso se quiso direccionar hacia el fin de realizar el mismo trabajo con plataformas mucho más pequeñas que ahorren espacio con el uso de tarjetas de procesamiento como la raspberry y tele-operación con ROS.

La calidad de mapeo que se obtuvo no fue excelente, pero se consiguió una gran representación de la escena real, comparada con otras técnicas de mapeo que solo logran extraer algunas características del entorno. Dichos resultados se podrían mejorar utilizando una cámara de mayor ángulo de visión, mayor velocidad de frames, y en un computador con procesamiento más rápido.

Referencias Bibliográficas

1. E. Rúa, A. Barrera, y N. Moreno, "Aprendizaje interactivo de Termodinámica de fluidos apoyado en las tecnologías de la información y comunicación", *Respuestas*, vol. 19, no. 2, pp. 41- 50, 2014.

2. O.E Gualdrón, O.M Duque-Suárez y M.A. Chacón-Rojas, “Diseño de un sistema de reconocimiento de rostros mediante la hibridación de técnicas de reconocimiento de patrones, visión artificial EIA,enfocado a la seguridad e interacción robótica social, Mundo Fesc, Vol. 3, No. 6, pp. 16-28, 2013.
3. Y. Medina y H. Miranda, “Comparación de algoritmos basados en la criptografía simétrica DES, AES y 3DES”, Mundo FESC, vol. 5, no. 9, pp. 14-21, 2015.
4. A. Ollero-Baturone, Robótica: manipuladores y robots móviles, Barcelona, España: MARCOMBO S.A, 2001.
5. V.R Barrientos, J.R García Sánchez, y R. Silva Ortigoza, “Robots Móviles: Evolución y Estado del Arte”. Polibits, vol. 35, pp. 12-17, 2007. doi: 10.17562/PB-35-3.
6. N. Hernández y A. Flórez-Fuentes, "Computación en la Nube", Mundo Fesc, vol. 4, n.º 8, pp. 46-51, dic. 2014.
7. J. Engel and T. Schops, “LSD_SLAM desarrollado en la Universidad Técnica de Múnich (TUM)”, 2014. [En línea]. Disponible en: <http://tracserver.labc.usb.ve/trac/visualslam/wiki/An%C3%A1lisisComparativo>
8. “ROS”, ROS.ORG, 2018. [Online]. Available: <http://wiki.ros.org/ROS/Introduction>
9. J. Engel, “Visual SLAM and VO: Surreal Vision, Oculus Research”, 2015 [presentation in People @EECS, UC Berkeley]. Available: <https://people.eecs.berkeley.edu/~chaene/cvpr17tut/SLAM.pdf>.
10. B. Davison, “Real-time simultaneous localisation and mapping with a single camera”, Proceedings Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 1403-1410, 2003. doi: 10.1109/ ICCV.2003.1238654
11. M. Shridhar and K.Y Neo, “Monocular SLAM for Real-Time Applications on Mobile Platforms”, 2015. [Online]. Available: <https://pdfs.semanticscholar.org/>
12. J. Zijlmans, “LSD-slam and ORB-slam2, a literature based explanation”, 2017. [Online]. Available: <https://medium.com/@j.zijlmans/lsd-slam-vs-orb-slam2-a-literature-basedcomparison-20732df431d>
13. E. Jakob, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM”, 2017. here: DSO: Direct Sparse Odometry, <https://vision.in.tum.de/research/vslam/lsdslam?redire>.
14. A. Villarroel, “Implementación de un sistema de mapeo 3D para inspección de sitios remotos utilizando un Cuadricoptero”, trabajo de fin de grado, Universidad Simón Bolívar, Sartenejas, 2015.
15. J. García-Merino, “Sistema avanzado de detección de obstáculos y navegación autónoma para vigilancia y protección basado en flota de vehículos aéreos no tripulados”, tesis doctoral, Universidad de Málaga, 2016.
16. Canónigo, “El aprendizaje en la era digital. Perspectivas desde las principales teorías”, Aibi revista de investigación, administración e ingeniería, vol. 5, n° 2, pp. 29-33., 2017