

Navindoor: Software de simulación para el algoritmos localización

Jesus Oroya
DeustoTech Mobility
Universidad de Deusto, Bilbao
Email: djoroya@deusto.es

Resumen—En este artículo se presentará la herramienta Navindoor. Esta es un paquete para MATLAB que permite el desarrollo de algoritmos de posicionamiento en interiores, así como la simulación de la señales asociadas al movimiento del objetivo, como puede ser las señales de radio frecuencia, señales inerciales, norte magnético, presión atmosférica, entre otras. Navindoor contiene la implementación de filtros de Kalman con capacidad de contener nuevos algoritmos de todo tipo. De esta forma se reduce el tiempo de desarrollo en la investigación de posicionamiento.

Index Terms—localización en interiores, procesamiento de señales

I. INTRODUCCIÓN

Las metodologías utilizadas para la sistemas de localización son tan variadas como la diversidad de sensores existentes. Sin embargo, aunque los algoritmos sean diversos, existe una serie de pasos comunes que se deben realizar antes que desarrollar un algoritmo de posicionamiento. Estas son, la toma de medidas experimentales de las señales, así como medidas de la propia trayectoria. Esta recolección de datos debe ser variada, ya que la colección en pocas casuísticas puede agregar sesgos no deseados a nuestros algoritmos. Es por ello que lo ideal es realizar pruebas en varios entornos con distintas trayectorias. Sin embargo, esto hace que el tiempo de desarrollo se alargue.

Navindoor propone una solución a este problema mediante la simulación del proceso de posicionamiento. El escenario, la trayectorias y las señales son simulados mediante modelos correspondientes. Además de permitir al usuario crear nuevos modelos si así lo quisiese. De esta forma Navindoor se convierte en un simulador muy versátil.

Navindoor esta diseñado para poder desarrollar tres tipos de algoritmos.

1. **Algoritmos de simulación de trayectorias:** Los modelos de movimiento de un objetivo deberá reproducir las velocidades y aceleraciones. Este modelo puede cambiar si el objetivo es un robot o una peatón, es más el modelo de movimiento puede cambiar según el esenario.
2. **Algoritmos de simulación de señales:** En Navindoor se contemplan dos tipos de señales, estas son las que dependen de balizas para ser generadas y las que son solo dependientes de la trayectoria seguida (Sección 1). Navindoor contiene modelos señales *RSS*, *ToF*, *AoA*,

Barometer, *Magnetometer*, etc. que pueden ser modificados o creado otros a partir de ellos.

3. **Algoritmos de posicionamiento:** Algoritmos de posicionamiento ya concidos como los filtros de Kalman ya estan implementados en Navindoor por defecto. Y de la misma forma que en los dos puntos anteriores, la implementación de nuevos algoritmos es muy fácil.

Gracias a que la herramienta contiene modelos por defecto en cada uno de sus frentes, podemos desarrollar de forma independiente a los demás modelos.

En este articulo, describiremos el estado del arte en materia de simuladores, además de la arquitectura de software debajo de navindoor, la funcionalidad del software en algunos ejemplos concretos y por último la dirección de los futuros desarrollos.

II. ESTADO DEL ARTE

En la comunidad del posicionamiento se esta buscando una solución para estandarizar los desarrollos. Es por ello que se han realizado trabajos relacionados. En este apartado mencionaremos algunos trabajos y sus principales características.

II-A. SMILe

SMILe [1] es un software que propone una solución de simulación completa y unificada que ayude al desarrollo y evaluación de los métodos de localización basados en ToF. El objetivo es proporcionar una herramienta de simulación bien definida y altamente configurable, donde se puedan evaluar varios métodos de localización de manera reproducible. SMILe permite al usuario configurar un ambiente interior artificial, donde se pueden configurar varios factores que afectan significativamente el rendimiento general de la localización. Estos factores incluyen: despliegue de diferentes nodos, capacidades de radio, inexactitud de relojes de hardware. SMILe, nos provee herramientas necesarias para la comparación de algoritmos de estimación de la posición a partir de señales ToF, sin embargo por ahora otras tecnologías no estan implementadas.

II-B. PyLayers

PyLayers [2] es un simulador de radio frecuencia. Se ha diseñado para evaluar el rendimiento del algoritmo de localización a través de la simulación de parámetros dependientes de la ubicación (LDP) en redes de radio móviles heterogéneas. El canal de radio se sintetiza mediante el uso de un método de

trazado de rayos basado en gráficos, de esta forma es capaz de simular el efecto de la reflexión de las ondas. PyLayers, esta desarrollado en python, y esta pensado para ser independientes de los algoritmos de procesamiento.

II-C. Sensor Fusion and Tracking Toolbox

Sensor Fusion and Tracking Toolbox [3] es un toolbox desarrollado por mathworks, que incluye algoritmos y herramientas para el diseño, simulación y análisis de sistemas que fusionan datos de múltiples sensores para mantener la posición, la orientación y el conocimiento de la situación. Los ejemplos de referencia proporcionan un punto de partida para implementar componentes de sistemas navegación.

Con este toolbox se puede importar y definir escenarios y trayectorias, transmitir señales y generar datos sintéticos para sensores activos y pasivos, incluidos sensores de RF, acústicos, EO / IR y GPS / IMU. También puede evaluar la precisión y el rendimiento del sistema con puntos de referencia estándar, métricas y gráficos animados.

Esta toolbox es bastante reciente,, por lo que por ahora contiene pocos ejemplos, sin embargo contiene un gran potencial.

III. ARQUITECTURA DE SOFTWARE

El desarrollo de Navindoor se ha dividido en módulos claramente diferenciados, que siguen los pasos para el posicionamiento mencionado en la introducción. A continuación describiremos cada una de ellas.

III-A. Planimetría

Para la definición del escenario se ha definido clases en MATLAB que representarán los distintos objetos necesarios para crear una planimetría. Ya que en un principio las trayectorias se definen en un edificio, la clase más grande que contiene los demás elementos, es la clase *building* (figura 1). Estos objetos pueden contener objetos *level* que representa las plantas. A su vez, los objetos *level*, contienen objetos *walls*, *doors*, *beacons*, *stairs* y *elevators*. Todos estos objetos contiene la posición y otras relaciones entre ellos para definir completamente la planimetría.

Con estos objeto podemos determinar que trayectorias son posibles en un objeto *building*, así como el tipo de camino esta siguiendo el peatón, (muy util a la hora de simular la trayectoria), además de definir la distribución de los *beacons*.

Debido a que el proceso de creación de la planimetría, Navindoor contiene una interfaz gráfica capaz de generar un objeto *building* (figura 2). De esta forma podemos cargar nuestras imagenes de la planimetría de interés, y guardarlo en un archivo tipo *.mat*, que luego puede ser utilizada por la propia interfaz o por en la propia consola de MATLAB.

Debemos notar que la clase *building* es una abstracción de la planimetría de un edificio, por lo que la traslación a otros formatos como *XML* o *JSON* son posible, y ya se estan explorando por parte del equipo de desarrollo. Este desarrollo permitirá comunicaciones con plataformas como *Open Street Maps*.

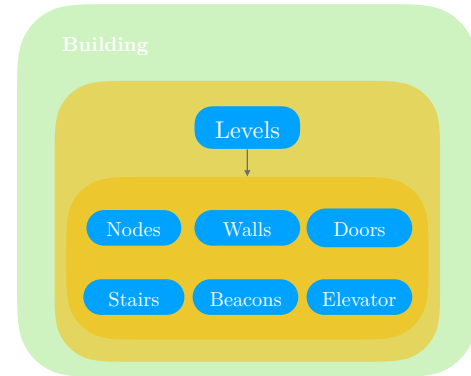


Figura 1. Esquema de la clase *building*

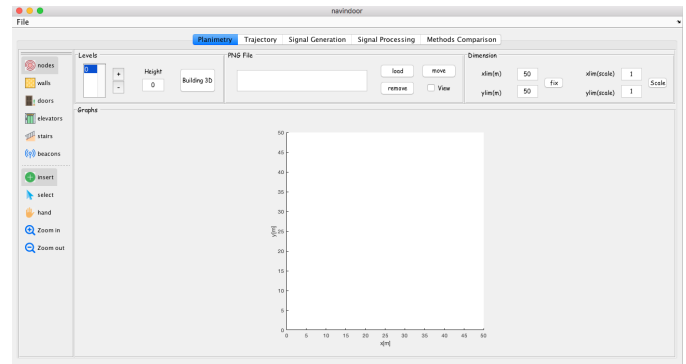


Figura 2. Interfaz gráfica para el diseño de la planimetría *building*

III-B. Trayectorias

III-C. Señales

Para la representación de las señales se ha creado un objeto MATLAB ...

III-D. Procesamiento

III-E. Comparación

IV. SIMULACIÓN

Veremos una ejemplo para ver la capacidades de navindoor en un ejemplo real. Para ello previamente, hemos creado la planimetría del edificio de la facultad de ingeniería de Deusto, con ayuda de la interfaz gráfica *navindoor*. Un vez creada podemos guardar esta en un archivo *.mat* y utilizar las distintas funciones implementadas para la clase edificio. Un ejemplo muy util es utilizar la funcion plot sobre el objeto creado. Obtendremos la figura IV

V. CONCLUSIÓN

- Contiene todos sus modulos en el mismo lenguaje de programación, lo que hace que sea facilmente instalable y mejora sus mantenimiento

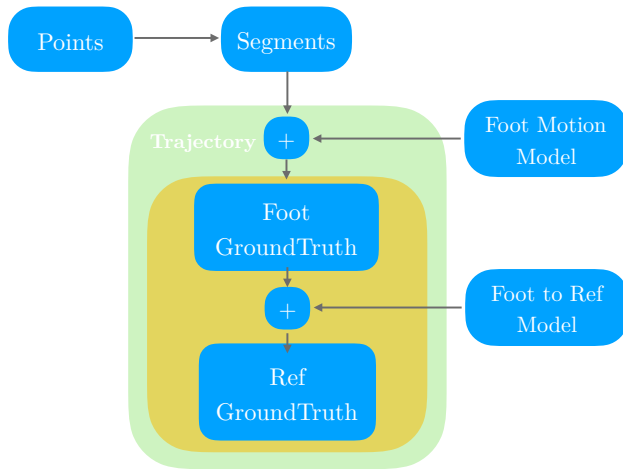


Figura 3. Visualización de la Facultad de ingeniería

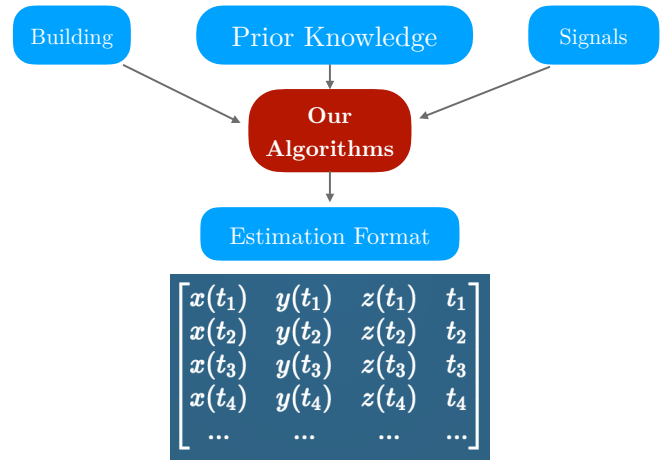


Figura 6. Visualización de la Facultad de ingeniería

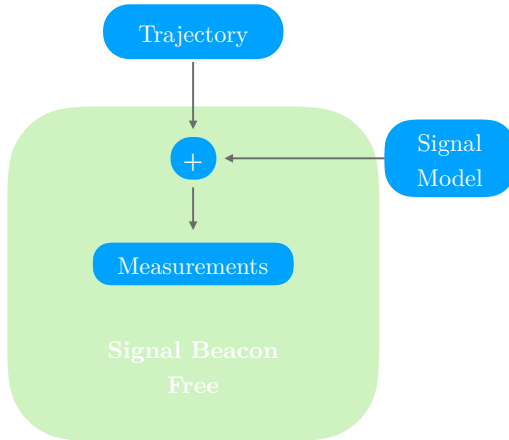


Figura 4. Visualización de la Facultad de ingeniería

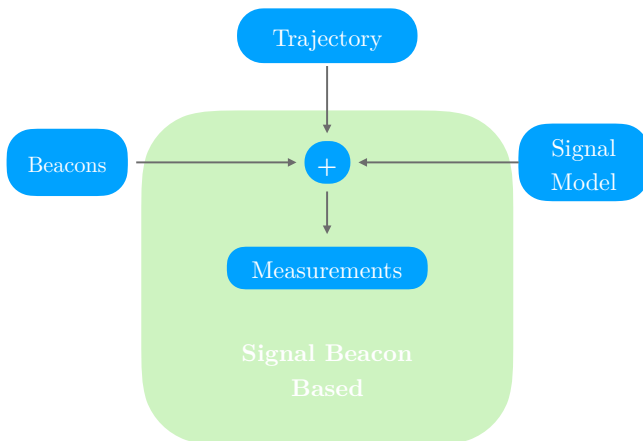


Figura 5. Visualización de la Facultad de ingeniería

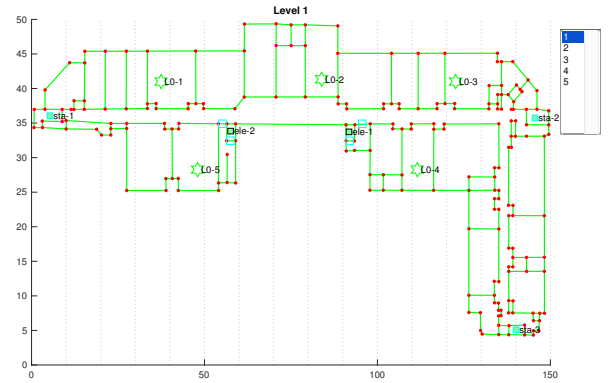


Figura 7. Visualización de la Facultad de ingeniería

- Aunque todos sus módulos están en MATLAB puede contener simuladores programados en otros lenguajes gracias a los MATLAB APIs.
- La navegación en exteriores será desarrollada mediante la importación de mapas desde la plataforma *Open Street Maps*
- NAVINDOOR mezcla la experiencia GUI (interfaz gráfica) y CLI (línea de comandos) en el desarrollo de los algoritmos, lo que hace que la curva de aprendizaje sea menor.

REFERENCIAS

- [1] Jankowski T, Nikodem M. SMILe – Simulator for Methods of Indoor Localization. 2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2018;(September):24–27.
- [2] Amiot N, Laaraiedh M, Uguen B. PyLayers: An open source dynamic simulator for indoor propagation and localization. 2013 IEEE International Conference on Communications Workshops, ICC 2013. 2013;(July 2014):84–88.
- [3] Mathworks. Sensor Fusion and Tracking Toolbox; 2018. Available from: <https://es.mathworks.com/help/fusion/>.