

2016

**Propuesta para la
Implementación de Algoritmos
Rapidly Exploring Random
Tree (RRT), en la planificación
de trayectorias por medio de
coordenadas GPS, en
entornos desconocidos bajo el
Sistema Operativo Android**

OSCAR EDUARDO GUACANEME
COTRINO.

Fundación Universitaria Los Libertadores
2016

PROPUESTA PARA LA IMPLEMENTACIÓN DE ALGORITMOS RAPIDLY
EXPLORING RANDOM TREE (RRT), EN LA PLANIFICACIÓN DE
TRAYECTORIAS POR MEDIO DE COORDENADAS GPS, EN ENTORNOS
DESCONOCIDOS, BAJO EL SISTEMA OPERATIVO ANDROID.

OSCAR EDUARDO GUACANEME COTRINO

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
BOGOTÁ, D.C.

PROPUESTA PARA LA IMPLEMENTACIÓN DE ALGORITMOS RAPIDLY
EXPLORING RANDOM TREE (RRT), EN LA PLANIFICACIÓN DE
TRAYECTORIAS POR MEDIO DE COORDENADAS GPS, EN ENTORNOS
DESCONOCIDOS, BAJO EL SISTEMA OPERATIVO ANDROID.

OSCAR EDUARDO GUACANEME COTRINO

Trabajo de grado para optar al título de Ingeniero Electrónico

Director

Alejandro Caicedo

Ingeniero Electronico.

FUNDACIÓN UNIVERSITARIA LOS LIBERTADORES

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

BOGOTÁ, D.C.

2016

Nota de aceptación

Firma del presidente del jurado

Firma del jurado

Firma del jurado

BOGOTA DC (01,03, 2016)

Las directivas de la Fundación Universitaria Los Libertadores, los jurados calificadores y el cuerpo docente no son responsables por los criterios e ideas expuestas en el presente documento. Estos corresponden únicamente al autor

DEDICATORIA

Dedico este trabajo a Dios y a mí familia quien gracias a su apoyo y comprensión permitió que yo pudiera finalizar mis estudios y de esta manera afrontar todos los retos consecuentes a la culminación de los mismos.

AGRADECIMIENTOS

A mis padres quienes incondicionalmente estuvieron siempre apoyándome, a mis profesores quienes mostraron comprensión y me ayudaron en cada una de las etapas de mi carrera, al Ingeniero Alejandro Caicedo, ya que con su guía y apoyo pude lograr la terminación de este trabajo.

Y en general a todas las personas que de alguna manera estuvieron involucrados en mi proceso.

ABSTRACT

In the area of current life moving from one place to another is essential in our lifestyle, it is necessary either to go to work, resupply or just for fun, although the management of technology makes these tasks simpler in many cases you do not have easy access to this technology, and therefore can not be required to cover all distances in the shortest possible time.

This graduation project is focused on performing, through current technologies development that allows path planning in an unfamiliar environment, giving the possibility to reach a destination through its GPS coordinates.

Planning a movement it is performed with the goal of bringing a body from an initial position to a final position, within a space free of any type of collision (Latombe, 1991; Laumont et al.1994; Muñoz, 1995).

For this we will focus on information coordinate start and end coordinate by GPS and thus find the most suitable route for navigation in unknown environment where the user is located.

In the larger the environment, or greater degrees of freedom that have or more kinematic restrictions arise, the complexity for solving these paths generates more limitations to the methods and solution times. (2).

With the information obtained through the implementation of RRT algorithm and OOP we reach a minimum viable product on which they can perform (experimental) tests. Navigability and ways of implementing the technology that surrounds us now.

INTRODUCCIÓN

En el ámbito de vida actual el desplazamiento de un lugar hacia otro es esencial en nuestro estilo de vida, es necesario ya sea para ir a trabajar, reabastecernos o simplemente por la diversión, aunque el manejo de la tecnología hace que estas tareas sean más simples, en muchas ocasiones no se cuenta con la facilidad de acceso a esta tecnología, y por lo tanto no se pueden cubrir todas las distancias requeridas en el mínimo tiempo posible.

Este proyecto de grado está enfocado en realizar, por medio de las tecnologías actuales un desarrollo que permite la planificación de trayectorias en un entorno desconocido, dando la posibilidad de llegar a un destino por medio de sus coordenadas GPS.

La planificación de un movimiento se realiza con el objetivo de llevar un cuerpo de una posición inicial a una posición final, dentro de un espacio y libre de cualquier tipo de colisión (Latombe, 1991; Laumont et al.1994; Muñoz, 1995).

Para esto nos enfocaremos en obtener información de coordenadas de inicio y la coordenada final por medio del GPS y de esta manera buscar la ruta más adecuada para la navegación en el entorno no conocido en donde se encuentre el usuario.

En cuanto mayor sea el entorno, o mayores los grados de libertad que se tengan, o se presenten más restricciones de cinemática, la complejidad para la solución de estas trayectorias genera más limitaciones en cuanto a los métodos y los tiempos de solución. (2).

Con la información obtenida y por medio de la implementación del algoritmo RRT y programación orientada a objetos llegaremos a un producto mínimo viable sobre el cual se puedan realizar pruebas (experimentales). De la navegabilidad y formas de implementación en la tecnología que actualmente nos rodea.

CONTENIDO

1	PLANTEAMIENTO	13
1.1	Resumen del Proyecto	13
1.2	Identificación del problema	13
1.3	Justificación	14
2	OBJETIVOS	15
2.1.1	Objetivo General	15
2.1.2	Objetivos Específicos	15
3	MARCO REFERENCIAL	16
3.1	Estado del arte	16
3.1.1	Métodos Para la planificación de trayectorias	16
3.1.2	Grafos de visibilidad.	17
3.1.3	Diagrama de Voronoi.	18
3.1.4	Modelado del espacio libre	18
3.1.5	Campos potenciales	19
3.1.6	Mapas probabilísticos	20
3.1.7	Algoritmos Genéticos	20
3.1.8	Colonia de hormigas (Ant Colony Optimization ACO).	21
3.1.9	Enjambre de partículas (Particle Swarm Optimization PSO).	22
3.2	Arboles de exploración rápida (Rapidly Exploring Random Tree RRT).	23
3.2.1	RRT Básico.	24
3.2.2	RRT Bidireccional.	26
3.2.3	RRT- Ext-Ext	27
3.2.4	Simulaciones del algoritmo RRT en java	29
3.2.5	Geo posicionamiento	35
3.2.6	Uso de las coordenadas GPS en Google Mapas.	37
3.2.7	Android.	37
4	DESARROLLO DEL PROBLEMA	38
5	IMPLEMENTACION.	39

5.1	<i>Uso del Algoritmo RRT.</i>	39
5.2	<i>Variables del algoritmo.</i>	40
5.3	<i>Ejecución de pruebas.</i>	41
5.3.1	<i>Diseño de Pruebas</i>	41
5.3.2	<i>Funcionamiento del Aplicativo.</i>	42
5.4	<i>Resultados.</i>	43
5.4.1	<i>Tabla de Resultados</i>	47
6	<i>Conclusiones, limitaciones y planes futuros.</i>	48
7	<i>Bibliografía</i>	50

TABLA DE ILUSTRACIONES

Ilustración 1. Modelo RRT - Crecimiento del árbol. (13)	24
Ilustración 2. Modelo RRT - Expansión del árbol para un móvil omnidireccional. (14)	25
Ilustración 3. Simulación algoritmo RRT sin obstáculos. Fuente Propia	30
Ilustración 4. Simulación algoritmo RRT con un obstáculo. Fuente propia	30
Ilustración 5. Simulación algoritmo RRT con varios obstáculos. Fuente propia....	31
Ilustración 6. Simulación algoritmo RRT con obstáculos Complejos. Fuente propia	32
Ilustración 7. Simulación algoritmo RRT con obstáculos Complejos. Fuente propia	32
Ilustración 8. Simulación algoritmo RRT con obstáculos Complejos. Fuente propia	33
Ilustración 9. Simulación del comportamiento del algoritmo de RRT con obstáculos complejos y sin solución. Fuente propia	34
Ilustración 10. Simulación del comportamiento del algoritmo de RRT con obstáculos complejos y sin solución. Fuente propia	34
Ilustración 11. GPS Fuente: http://www.gps-auto.org/navegador-gps/funcionamiento-gps.html	36
Ilustración 12. Simulación del comportamiento del algoritmo de RRT con mala parametrización. Fuente Propia	41
Ilustración 13. Imagen del inicio del programa (Prototipo). Fuente propia	43
Ilustración 14. Imagen con lugar de origen y lugar de destino simulación. Fuente propia.....	43
Ilustración 15. Imagen Camino más cortó hallado por el programa. Fuente propia	44
Ilustración 16. Simulación de un obstáculo en el camino. Fuente propia.....	44
Ilustración 17. Segundo camino trazado por el programa. Fuente propia.....	45
Ilustración 18. Llegada al punto final. Fuente propia.....	45
Ilustración 19. Llegada al punto final #2. Fuente propia.....	46

Ilustración 20. Simulación con varios caminos #1- Fuente propia	46
Ilustración 21. Simulación con varios caminos #1. Fuente propia.....	46

1 PLANTEAMIENTO

1.1 Resumen del Proyecto

Este proyecto consiste en la implementación del algoritmo Rapidly Exploring Random Tree (RRT) de planificación de trayectorias en un aplicativo Android, en el cual se pueda realizar una navegación en un entorno desconocido. El algoritmo elegido se emplea por su característica de funcionamiento para la exploración de rutas en entornos desconocidos.

1.2 Identificación del problema

Como una primera medida se requiere el estudio de los algoritmos de planificación de rutas, y de una manera particular los algoritmos de Rapidly Exploring Random Tree (RRT), aprovechando sus grandes ventajas para exploración de terrenos o espacios geográficos no conocidos.

Se pretende realiza un aplicativo móvil (app), en la cual se integraran los algoritmos RRT, con el uso de sensores, puntos del gps el cual permita que por medio de un dispositivo Android sea posible la exploración de espacios geográficos desconocidos o cambiantes de manera rápida y constante

Esta aplicación será un producto mínimo viable (PMV) por el cual se pretenden mostrar las ventajas del uso de la tecnología ten la exploración y el uso de los espacios geográficos en el que nos encontremos, por medio de esta aplicación se busca llegar desde un punto de inicio a un punto de llegada por medio de coordenadas GPS, poder generar una ruta y recalcularla constantemente por la cual podamos evadir cualquier tipo de obstáculo que se presente.

1.3 Justificación

En campos específicos de la ingeniería como lo son la robótica, la automatización, la aeronáutica, de igual manera en acciones cotidianas de la vida es necesario ir de una posición a otra, punto a al punto b, para poder realizar esta acción mucho más eficaz, este desplazamiento es fácil y rápido cuando se tienen las cualidades físicas, o las herramientas necesarias para lograrlo, en el caso que se tenga alguna dificultad en la cual se nos impida conocer un lugar el desplazarnos de un punto a a uno b es mucho más complicado.

En el caso de vehículos no tripulados o personas con discapacidad visual la cual no les permita conocer la geografía de un lugar y que además de esto pueda ser cambiante constantemente es necesario generar formar de planificar una trayectoria en tiempo real obteniendo informaciones constantes del entorno, la cual es posible de realizar por el uso correcto del algoritmo Rapidly Exploring Random Tree (RRT), el cual por medio de diferentes ramas es capaz de explorar todo el terreno para generar un camino efectivo desde un punto de inicio a un punto final.

2 OBJETIVOS

2.1.1 Objetivo General

Desarrollar un aplicativo de planificación de trayectorias en entornos geográficos desconocidos implementando algoritmos Rapidly Exploring Random Tree (RRT) y empelando sensores y coordenadas GPS bajo el S.O Android.

2.1.2 Objetivos Específicos

- Consultar los algoritmos de planificación de trayectorias Rapidly Exploring Random Tree (RRT) para su implementación.
- Obtener y verificar la información necesaria respecto al funcionamiento y usos los algoritmos de planificación de trayectorias Rapidly Exploring Random Tree (RRT).
- Realizar pruebas en ambientes controlados bajo un simulador y obtener información práctica del funcionamiento del algoritmo (Rapidly Exploring Random Tree) RRT.
- Realizar la implementación del algoritmo Rapidly Exploring Random Tree RRT en S.O Android realizando las alteraciones respectivas para su adecuación y funcionamiento, obteniendo información fiable del dispositivo móvil con el uso de los sensores como el Acelerómetro, el sensor geomagnético, el sensor de giro, GPS a usar para la ejecución desarrollo (PMV).
- Realizar pruebas experimentales del funcionamiento del aplicativo (PMV) desarrollado.

3 MARCO REFERENCIAL

3.1 Estado del arte

Se presenta el estado del arte de los temas relacionados con el proyecto.

La primera parte consiste en una reseña de los métodos generales para la planeación de trayectorias.

En la segunda parte nos centraremos en el método para planificación elegido en este caso algoritmos (Rapidly Exploring Random Tree) RRT

En la tercera parte nos centraremos en las tecnologías que se usaron para la implementación del aplicativo

3.1.1 Métodos Para la planificación de trayectorias

Planificar es prever y decidir hoy las acciones que nos pueden llevar desde el presente hasta un futuro deseable, no se trata de hacer predicciones acerca del futuro sino de tomar las decisiones pertinentes para que ese futuro ocurra. (3)

La idea general en la planificación de trayectorias consiste en desvincular el problema de la cinemática y dinámica del robot de la obtención de una ruta (óptima o no) libre de obstáculos. (3)

Existiendo un gran número de métodos efectivos para resolver el problema de la planificación en tiempo real, tales como campos potenciales, grafos de visibilidad, diagramas de Voronoi, etc. (2)

Podemos realizar una clasificación de los modelos de planificación de trayectorias. (4)

- Determinísticos
 - Basados en grafos
 - Grafos de visibilidad.
 - Diagramas de Voronoi.

- Modelado del espacio libre.
- Descomposición en celdas.
 - Campos potenciales artificiales.
- Probabilísticos y aleatorios (algoritmos de búsqueda aleatoria)
 - Planeador aleatorio de trayectorias (Randomized Path Planner RPP).
 - Mapas probabilísticos (Probabilistic Road Maps PRM).
 - Árboles de exploración rápida (Rapidly Exploring Random Tree RRT).
- Basados en optimización
 - Algoritmos genéticos (Genetic Algorithms GA).
 - Colonia de hormigas (Ant Colony Optimization ACO).
 - Enjambre de partículas (Particle Swarm Optimization PSO).
 - Quimiotaxis bacteriana (Bacterial Chemotaxis BC).

3.1.2 Grafos de visibilidad.

El gráfico de visibilidad es un conjunto de obstáculos poligonales de no intersección en el plano, es un grafo no dirigido cuyos vértices son los vértices de los obstáculos y cuyos bordes son pares de vértices de tal manera que el segmento de línea abierta entre cada dos vértices no intersectan cualquiera de los obstáculos.

El problema principal en el diseño de la gráfica visibilidad es la determinación de las porciones visibles del mapa. Esta operación el tiempo y el espacio que consume, incluso para las computadoras modernas. (6)

Para determinar la visibilidad de polígonos muchos algoritmos han sido desarrollados. Todos ellos tienen una gran complejidad computacional. Se trata de una familia de algoritmos que convierten de forma unívoca una serie temporal en un grafo o red. Por construcción, la red extraída por estos métodos hereda o captura propiedades estructurales de la serie de partida. Así, por ejemplo, si la serie temporal es periódica, el grafo asociado es regular; si la serie es aleatoria, la red asociada presenta una distribución de conectividades exponencial; o si la serie es fractal, la red correspondiente es libre de escala. (5)

3.1.3 Diagrama de Voronoi.

El diagrama de Voronoi es una estructura que captura la información de proximidad de un conjunto de puntos P descomponiendo el plano en regiones poligonales convexas. Esta estructura tiene un gran interés para este proyecto por las aplicaciones geométricas que posee. Entre sus aplicaciones geométricas, se encuentra la construcción de la triangulación de Delaunay dualizando el diagrama de Voronoi, o la búsqueda del punto $p_i \in P$ más cercano a un punto q del plano. Entre sus aplicaciones no geométricas, se encuentran usos ecológicos, como la de determinar la supervivencia de organismos en competencia por alimentos o luz (por ejemplo, árboles en un bosque), y usos en redes sociales para representar las relaciones entre personas. (7)

3.1.4 Modelado del espacio libre

El modelado de espacio libre se lleva a cabo por cilindros rectilíneos generalizados (CRG). Al igual que los diagramas de Voronoi, con el uso de los CRG se pretende que el vehículo navegue lo más Obstacle Diagrama Voronoi, con el uso de los CRG se pretende que el vehículo navegue lo más alejado de los obstáculos. De forma que la ruta que lleve al robot desde una configuración inicial hasta otra final estará compuesta por una serie de CRG interconectados, de tal modo que la configuración de partida se encuentre en el primer cilindro de la sucesión y la final en el último.

Espacio libre es la colección de posturas que un vehículo puede de manera segura asumir, con el fin de maniobrar, un robot móvil requiere poca descripción de los límites para liberar espacio.

Modelado del espacio libre determina el conjunto de posiciones y orientaciones que el robot puede asumir sin "chocar" con otros objetos.

Posición es la estimación que permite al robot situarse con respecto a las metas y conocimiento sobre el medio ambiente.

Infortunadamente, los dispositivos de detección que están disponibles para los robots móviles a menudo fallan en una variedad de circunstancias. Esto es especialmente cierto de los dispositivos menos costosos como el rango ultrasónico y de infrarrojos sensores. La combinación de DAT; y de varios sensores y de un pre-almacenada modelo del dominio proporciona una forma de mejorar la fiabilidad de un sistema de percepción. Esa combinación puede ser acoplada por la integración de mediciones de distancia en un modelo geométrico del ambiente local. (8)

3.1.5 Campos potenciales

El método de campos potenciales es una de las técnicas más populares en la generación de trayectorias para robots móviles, por ser una forma práctica de solucionar este problema. Entre los primeros desarrollos, existe el presentado por Khatib O, que propone que los obstáculos y el robot tengan una carga eléctrica del mismo signo, con la finalidad de generar una repulsión, mientras que la meta tiene asociada una carga eléctrica de signo opuesto, para atraer el robot al punto destino. La propuesta presentada por Khatib O, se denominó FIRAS (sigla en francés que se traduce al inglés como The Force Involving and Artificial Repulsion from the Surface Function).

En modo convencional, los potenciales atractivos y repulsivos se generan de forma separada y el potencial total del espacio de trabajo se obtiene como la suma de aquellos. Otra propuesta con este mismo enfoque, fue desarrollada por Krogh B. A. y es llamada GPF (Generalized Potential Field). Para generar la trayectoria con estos potenciales, sólo se requiere calcular los gradientes; sin embargo, los campos potenciales pueden ser no convexos, presentando mínimos locales que dejaría estancado al robot en un punto diferente al objetivo. Otra desventaja es la estimación de los parámetros del campo potencial para lograr una adecuada evasión de obstáculos.

Modificaciones al planteamiento original de campos potenciales se realizan, considerando las anteriores limitaciones; Volpe y Khosla, proponen las funciones potenciales súper-cuadráticas (Super-quadric Artificial Potential Functions), donde se emplea un amplio conjunto de formas desde rectángulos hasta elipses que están descritos por fórmulas súper-cuadráticas. (4)

3.1.6 Mapas probabilísticos

El algoritmo “Probabilistic RoadMaps” se diseñó con su propio nombre en el año 1996 por L. E. Kavraki, P. Svestka, J.-C. Latombe y M. H. Overmars , aunque es el resultado de una investigación que inicio Overmars en 1992. Es un algoritmo de planificación por muestreo y de consulta múltiple.

Se divide en dos fases, pre procesado del entorno y consulta que se detallan a continuación.

El pre procesado es la parte más costosa computacionalmente y en esta fase se trata de generar un grafo llamado mapa de carreteras (Roadmap) lo suficientemente complejo para resolver cualquier problema de movimiento. No conviene hacerlo demasiado complejo ya que el coste de añadir un nodo y sobre todo un camino en el grafo es muy grande, ya que se debe llamar al detector de colisiones numerosas veces.

Los nodos son generados de forma aleatoria, esa idea fue innovadora e interesante ya que así se conseguía huir de los mínimos locales. Es un tema de debate el sí hacerlo de esta forma es conveniente.

En la fase de consulta se usa el grafo para obtener un camino óptimo para un problema determinado. Normalmente se suelen incluir en el grafo el estado inicial y final.

Para determinar el camino óptimo normalmente se suele utilizar el algoritmo de Dijkstra. (9)

3.1.7 Algoritmos Genéticos

Los Algoritmos Genéticos son métodos adaptativos, generalmente usados en problemas de búsqueda y optimización de parámetros, basados en la reproducción sexual y en el principio supervivencia del mas apto.

Más formalmente, y siguiendo la definición dada por Goldberg, “los Algoritmos Genéticos son algoritmos de búsqueda basados en la mecánica de selección natural y de la genética natural. Combinan la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas”. Para alcanzar la solución a un problema se parte de un conjunto inicial de individuos, llamado población, generado de manera aleatoria. Cada uno de estos individuos representa una posible solución al problema. Estos individuos evolucionan tomando como base los esquemas propuestos por Darwin sobre la selección natural, y se adaptan en mayor medida tras el paso de cada generación a la solución requerida.

En términos muy generales se podría definir la computación evolutiva como una familia de modelos computacionales inspirados en la evolución. Más formalmente el término de computación evolutiva se refiere al estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas basadas en los principios de la evolución natural. (10)

3.1.8 Colonia de hormigas (*Ant Colony Optimization ACO*).

La meta heurística ACO está inspirada en la observación del comportamiento de hormigas reales. Se presentan observaciones hechas en experimentos con hormigas, estos experimentos inspiraron el diseño de la meta heurística.

Las hormigas son insectos sociales, los cuales viven en colonias y cuyo comportamiento está dirigido más hacia la supervivencia de la colonia como un todo que al de un simple componente individual de la colonia. Los insectos sociales han capturado la atención de muchos científicos por el gran nivel de estructuración que alcanzan sus colonias, especialmente cuando se lo compara con la simplicidad relativa de los componentes de la colonia.

Algunos científicos investigaron experimentalmente este comportamiento de depositar feromona y seguir rastros de feromona para entenderlo mejor y poder cuantificarlo. Diseñaron un experimento llamado “experimento del puente binario”. Ellos usaron hormigas *Linepithema humile* (hormigas originarias de Argentina).

El nido de las hormigas fue conectado a una fuente de comida por dos puentes de igual longitud. Las hormigas podían elegir libremente que brazo del puente usar cuando buscaban comida y la traían de vuelta al nido. Su comportamiento fue observado por un periodo de tiempo.

En este experimento, inicialmente no hay feromona en ninguno de los dos puentes. Las hormigas empiezan explorando los alrededores del nido y eventualmente cruzan uno de los puentes, alcanzando la fuente de comida. Al caminar hacia la fuente de comida y de vuelta, las hormigas depositan feromona en el puente que usaron. Inicialmente, cada hormiga elige aleatoriamente uno de los puentes. Sin embargo, debido a cambios aleatorios, después de algún tiempo habrá más feromona depositada en uno de los puentes que en el otro. Debido a que las hormigas tienden a preferir (en probabilidad) seguir un rastro de feromona más fuerte, el puente que tiene más feromona atraerá más hormigas.

Esto a su vez hace que el rastro de feromona crezca más, hasta que finalmente la colonia de hormigas converge hacia el uso del mismo puente. (11)

3.1.9 Enjambre de partículas (*Particle Swarm Optimization PSO*).

Un Algoritmo basado en Cúmulos de Partículas o Particle Swarm Optimization es una técnica metaheurística basada en poblaciones e inspirada en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces. PSO fue originalmente desarrollado por el psicólogo sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart en 1995, basándose en un enfoque conocido como la metáfora social, que describe a este algoritmo y que se puede resumir de la siguiente forma: los individuos que conviven en una sociedad tienen una opinión que es parte de un conjunto de creencias (el espacio de búsqueda) compartido por todos los posibles individuos. Cada individuo puede modificar su propia opinión basándose en tres factores:

- Su conocimiento sobre el entorno (su valor de fitness).
- Su conocimiento histórico o experiencias anteriores (su memoria).
- El conocimiento histórico o experiencias anteriores de los individuos situados en su vecindario.

Siguiendo ciertas reglas de interacción, los individuos en la población adaptan sus esquemas de creencias al de los individuos con más éxito de su entorno. Con el

tiempo, surge una cultura cuyos individuos tienen un conjunto de creencias estrechamente relacionado.

El principio natural en el que se basa PSO es el comportamiento de una bandada de aves o de un banco de peces: supongamos que una de estas bandadas busca comida en un área y que solamente hay una pieza de comida en dicha área. Los pájaros no saben dónde está la comida pero sí conocen su distancia a la misma, por lo que la estrategia más eficaz para hallar la comida es seguir al ave que se encuentre más cerca de ella. PSO emula este escenario para resolver problemas de optimización. Cada solución (partícula) es un “ave” en el espacio de búsqueda que está siempre en continuo movimiento y que nunca muere. (12)

3.2 Árboles de exploración rápida (Rapidly Exploring Random Tree RRT).

El algoritmo RRT basado en la generación de Árboles de Exploración Rápida, por lo que no es necesario usar funciones de potencial, ya que se trata de una técnica puramente probabilística, y es fácilmente extensible a sistemas con varios grados de libertad y asumir diferentes restricciones. (13), no precisan el establecimiento de un campo de potencial lo que lo lleva a tener un gran ahorro en el procesamiento de información al mismo tiempo este algoritmo es capaz de asegurar la exploración total de todo el espacio de configuraciones, y fácil extensión a escenario. (2)

El RRT es un algoritmo de planificación de caminos que calcula un camino continuo conectando una configuración inicial, q_{ini} , con una configuración final, q_{fin} , utilizando técnicas probabilísticas y teniendo en cuenta los obstáculos y las restricciones. Cada configuración determina la posición y orientación del vehículo en un escenario bidimensional o tridimensional. Si $\mathcal{C}$ es el conjunto de configuraciones posibles del vehículo, independientemente de los obstáculos presentes, $\mathcal{C}$ libre denotará al subconjunto de $\mathcal{C}$ formado por las configuraciones libres de colisión. Se supondrá que conjuntamente con $\mathcal{C}$ se define una determinada métrica. El algoritmo RRT básico es una estructura de datos que comienza en la configuración q_{ini} y determina dónde deben añadirse las nuevas configuraciones, las cuales aparecen como los vértices de un árbol que crece explorando el espacio.

El objetivo es dirigir la exploración hacia zonas inexploradas libres de colisiones, para lo cual el algoritmo escoge una configuración aleatoria denotada por q_{rand}

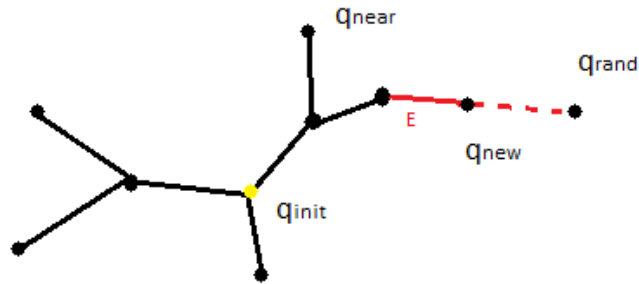


Ilustración 1. Modelo RRT - Crecimiento del árbol. (13)

3.2.1 RRT Básico.

A pesar de su corta vida, existen muchas aplicaciones en las que se han usado esta familia de algoritmos, la mayoría de las veces formando parte de un planificador más extenso que hace uso de la exploración rápida del RRT de algún espacio de configuraciones.

Aunque hay algunas variantes prometedoras de RRT de otros investigadores, la gran mayoría de los avances en el desarrollo de algoritmos RRT, así como la base teórica del algoritmo principal se deben a Steve M. La Valle y sus colaboradores.

Algoritmo General del RRT Básico

```

T.init(xinit);
for k = 0 to K do
  xrand ← EstadoAleatorio( );
  xnear ← NodoMásCercano(xrand, T )
  u ← MejorActuación(xrand, xnear, & xnew, & success)

```

```

if success then
 $x_{new} \leftarrow \text{Integra}(x_{near}, u, \Delta t)$ 
T .añade nodo( $x_{new}$ );
T .añade lado( $x_{near}, x_{new}, u$ );
return T

```

El fundamento de los denominados Rapidly Exploring Random Trees (RRTs) se puede observar en el anterior algoritmo. Se trata de un algoritmo que explora aleatoria y densamente el espacio de configuraciones. A continuación se explica su funcionamiento.

Partiendo de un estado inicial x_{init} y durante un número determinado de iteraciones, el árbol va creciendo, añadiendo nuevos nodos y conectándolos al árbol. En cada iteración, se elige un estado aleatorio x_{rand} dentro del espacio de configuraciones, así como el nodo más cercano del árbol x_{near} a dicho nodo. El nodo que se añadirá al árbol al ser resultado de integrar las ecuaciones de estado desde x_{near} con la entrada que produzca el resultado más cercano posible a x_{rand} . En el caso de un robot omnidireccional sin restricciones holonomicas, las líneas 5 y 6 del algoritmo se sustituirían por un sencillo paso de longitud ρ en la dirección de x_{rand} , que puede realizarse algebraicamente, o incluso podría añadirse directamente x_{rand} al árbol.

(14)

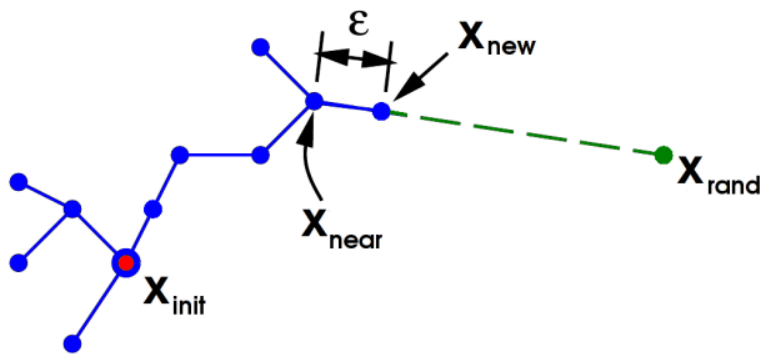


Ilustración 2. Modelo RRT - Expansión del árbol para un móvil omnidireccional. (14)

3.2.2 RRT Bidireccional.

Es una adaptación del algoritmo *RRT* para conseguir conectar una configuración inicial con una final. Éste se basa en la generación simultánea de dos árboles cuyas raíces son las configuraciones inicial y final. Dichos árboles crecen explorando el espacio vacío y buscándose entre sí hasta conectar, obteniendo entonces una trayectoria como resultado. El método utilizado para ello varía, dando lugar a distintos algoritmos.

Este algoritmo comienza con la generación de dos árboles que parten de los puntos origen y destino.

El algoritmo concluye en cuanto dichos árboles conectan. Si alcanzado el valor de K_{max} ambos árboles no han coincidido, se devuelve un mensaje de error.

En cada iteración uno de los árboles agregará una nueva rama si es posible en dirección al punto aleatorio generado. De esta forma dicho árbol tendrá una función de exploración del espacio vacío, dado que q_{rand} es una configuración que puede surgir en cualquier punto del escenario, marcando una dirección de crecimiento para el nodo más próximo.

Algoritmo General del RRT Bidireccional

Arbol_a[0]= q_{ini}

Arbol_b[0]= q_{fin}

Para $k = 1$ hasta K_{max}

q_{rand} = Configuración_Aleatoria();

Si Extiende(Arbol_a, q_{rand}) ≠ "rechazado" Entonces

Si (Extiende(Arbol_b, q_{new})= "alcanzado" Entonces

Devuelve Camino(Arbol_a, Arbol_b)

Intercambiar

(Arbol_a, Arbol_b)

Siguiente k

Devuelve Error

Si ha habido una nueva rama, entonces existe un vértice nuevo q_{new} . Ahora le toca el turno al segundo árbol que tomará como meta no el punto aleatorio q_{rand} , sino el nuevo vértice generado q_{new} . De esta forma dicho árbol no explora el espacio vacío, sino que intentará encontrar a su homólogo si es posible.

Obsérvese cómo se hace uso de la función Intercambiar, que alterna el orden de los árboles de manera que el reparto de sendas funciones sea equilibrado. Así pues, las acciones resultantes de dos iteraciones consecutivas serían:

- Árbol A crece hacia q_{rand} (1a iteración).
- Árbol B crece hacia q_{new} de A.
- Árbol B crece hacia q_{rand} (2a iteración).
- Árbol A crece hacia q_{new} de B.

De esta forma cada árbol invierte la mitad de su tiempo en explorar el espacio libre, y la otra mitad, en buscar a su compañero.

3.2.3 RRT- Ext-Ext

Este algoritmo posee una mejora la cual permite agilizar la conexión entre los árboles y se logra con una pequeña variación, ya que es posible el poder atribuirle a cada árbol una capacidad para que pueda dirigir su crecimiento hacia su homólogo lo que permite que sea mucho más rápido frente a otros algoritmos puramente aleatorios.

Algoritmo General del RRT Ext-Ext

```
RRT-Ext-Ext (qini, qfin)
{
  ArbolA[0]= qini;
  ArbolB[0]= qfin;
  Para k = 1 hasta Kmax
  {
    qrand= ConfiguraciónAleatoria( );
    Si (Extiende(ArbolA, qrand)=rechazado )
    {
      Si (Extiende(ArbolB, qnew)=alcanzado )
      {
        Devuelve Camino(ArbolA, ArbolB);
      }
    }
    Intercambiar(ArbolA, ArbolB);
  }
  Devuelve Error;
}
```

Este sutil cambio con respecto al algoritmo anterior es posible observarlo donde el punto *qrand* se sustituye en el segundo árbol por *qnew*, es decir, el punto recién agregado al árbol precedente. Finalizada la iteración, la función Intercambiar se ocupa de que la próxima vez, sea el último árbol el que crezca hacia *qrand* y el primero hacia *qnew*. Así pues, las acciones resultantes de dos iteraciones consecutivas serían:

1. ArbolA crece hacia *q_{rand}* (1a iteración).
2. ArbolB crece hacia *q_{new}* de ArbolA.
3. ArbolB crece hacia *q_{rand}* (2a iteración).
4. ArbolA crece hacia *q_{new}* de ArbolB.

De esta forma cada árbol invierte la mitad de su tiempo en explorar el espacio libre, y la otra mitad, en buscar a su compañero.

3.2.4 Simulaciones del algoritmo RRT en java

Simulaciones realizadas en un entorno de trabajo con diferentes obstáculos y con diferentes configuraciones de iteraciones y mejoras del algoritmo RRT.

- Simulación del comportamiento del algoritmo de RRT sin obstáculos entre los puntos de inicio y llegada, Ver figura [3] para observar el funcionamiento del algoritmo en un espacio libre.

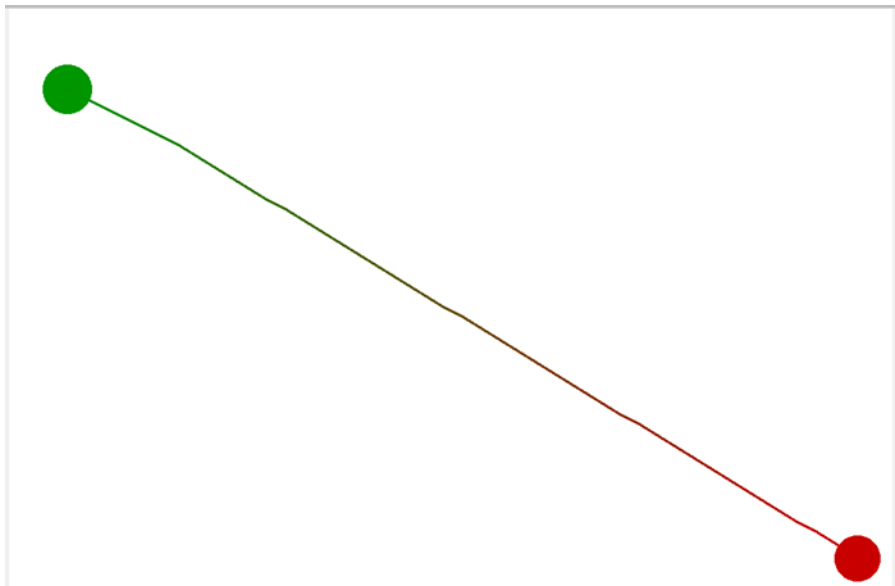


Ilustración 3. Simulación algoritmo RRT sin obstáculos. Fuente Propia

- Simulación del comportamiento del algoritmo de RRT con un obstáculo, entre los puntos de inicio y llegada, ver figura [4], en esta simulación podemos apreciar el comportamiento del crecimiento del árbol mientras realiza un exploración del espacio donde se encuentra, para encontrar el camino hacia su punto de llegada

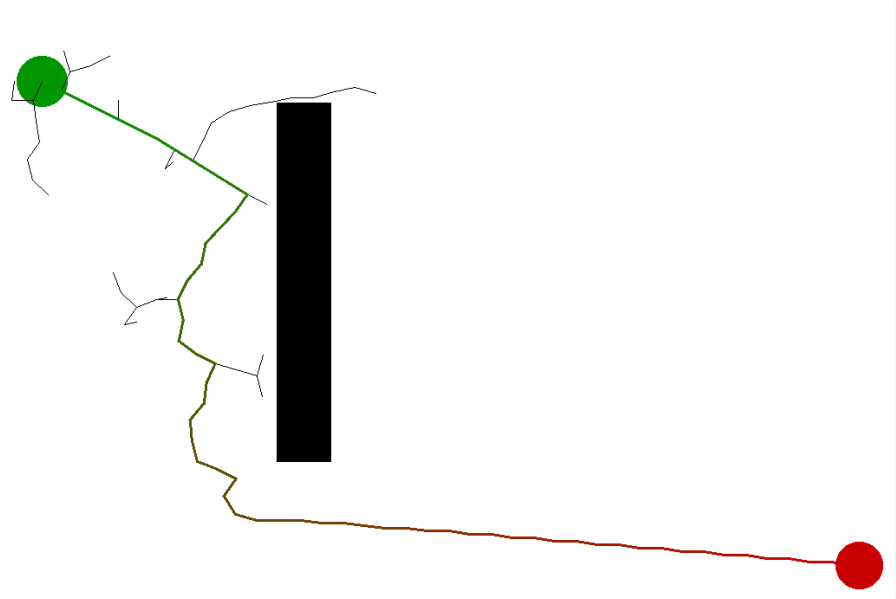


Ilustración 4. Simulación algoritmo RRT con un obstáculo. Fuente propia

- Simulación del comportamiento del algoritmo de RRT con varios obstáculos, entre los puntos de inicio y llegada, Ver figura [5] en esta simulación podemos apreciar el comportamiento del crecimiento del árbol más evidente además de observar las ramificaciones que se van generando al medida que realiza la exploración del espacio



Ilustración 5. Simulación algoritmo RRT con varios obstáculos. Fuente propia

- Simulación del comportamiento del algoritmo de RRT con obstáculos complejos, Ver figura [6], Ver figura [7], Ver figura [8] entre los puntos de inicio y llegada, en esta simulación podemos apreciar la forma en la cual el algoritmo de RRT usado es capaz de hallar el camino teniendo obstáculos de diferentes geometrías, también observamos como el crecimiento de las ramas del árbol realizan la exploración a el espacio muestra para de esta manera buscar los caminos para su llegada.

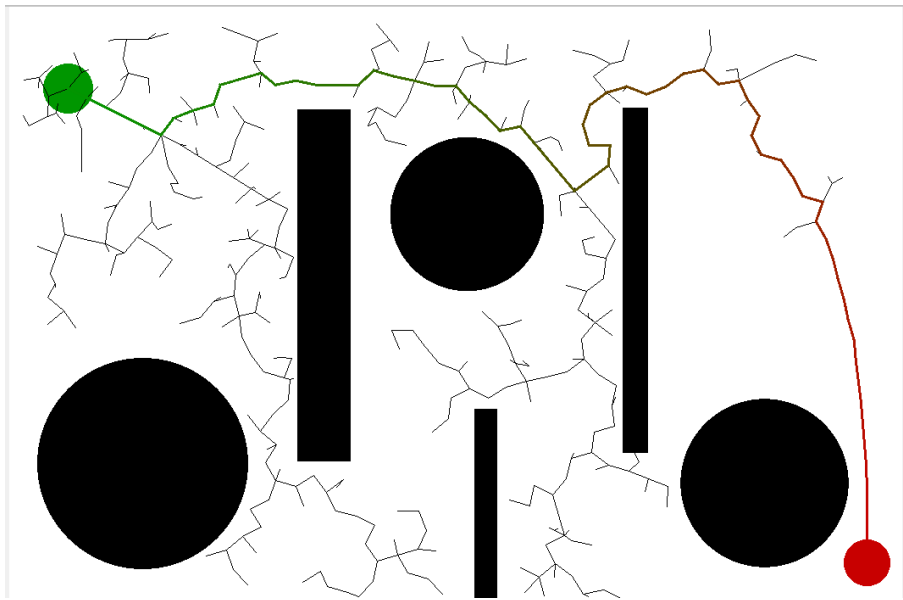


Ilustración 6. Simulación algoritmo RRT con obstáculos Complejos. Fuente propia



Ilustración 7. Simulación algoritmo RRT con obstáculos Complejos. Fuente propia

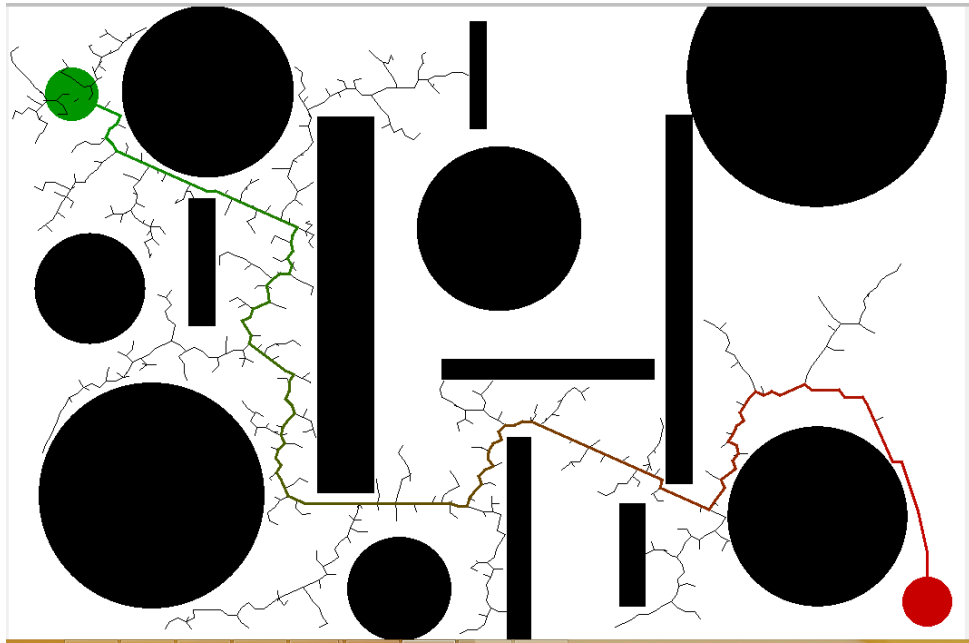
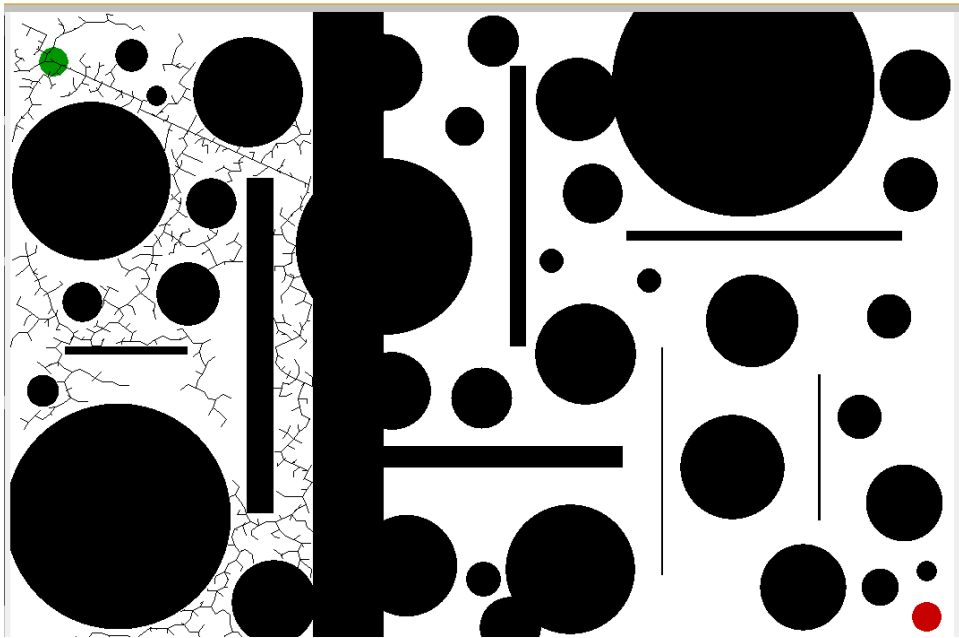
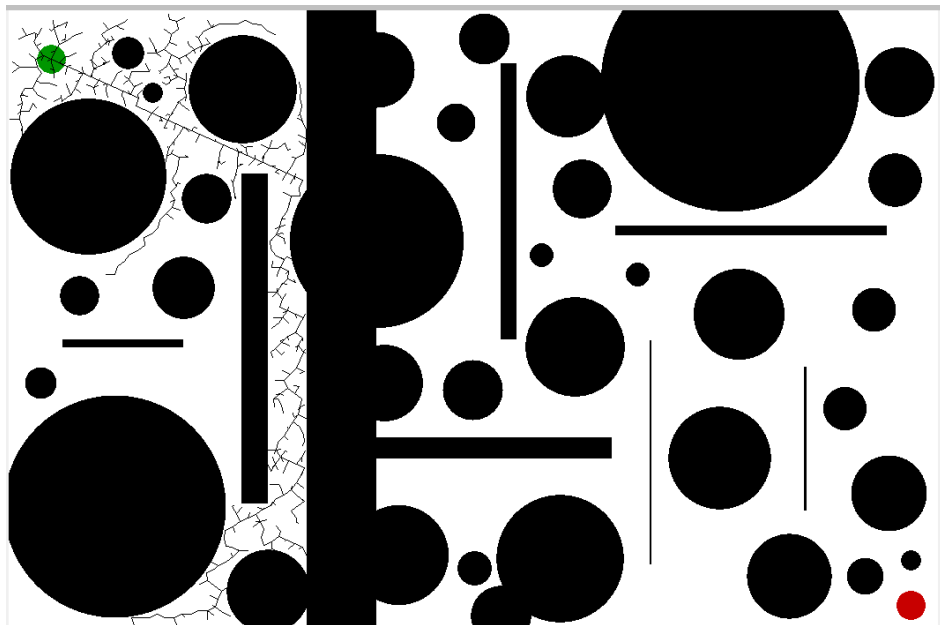


Ilustración 8. Simulación algoritmo RRT con obstáculos Complejos. Fuente propia

- Simulación del comportamiento del algoritmo de RRT con obstáculos complejos y sin solución, Ver figura [9], Ver figura [10], entre los puntos de inicio y llegada, en esta simulación podemos apreciar la forma en la cual el algoritmo de RRT usado intenta de hallar el camino teniendo obstáculos de diferentes geometrías, también observamos como el crecimiento de las ramas del árbol realizan la exploración a el espacio muestra para de esta manera buscar los caminos para su llegada en esta ocasión no se encuentra una solución.



*Ilustración 9. Simulación del comportamiento del algoritmo de RRT con obstáculos complejos y sin solución.
Fuente propia*



*Ilustración 10. Simulación del comportamiento del algoritmo de RRT con obstáculos complejos y sin solución.
Fuente propia*

3.2.5 Geo posicionamiento

La implementación del programa NAVSTAR, GPS (Navigation System Timing And Ranging, Global Positioning System) fue efectivamente iniciada en diciembre de 1973. El 22 de febrero de 1978 fue lanzado el primer satélite de una serie de cuatro.

La responsabilidad del desarrollo y mantenimiento del sistema recae en el Departamento de Defensa de los Estados Unidos, División Sistema Espacial. Esa dependencia se debía a que el sistema fue concebido, para uso militar.

GPS es un sistema que tiene como objetivo la determinación de las coordenadas espaciales de puntos respecto de un sistema de referencia mundial. Los puntos pueden estar ubicados en cualquier lugar del planeta, pueden permanecer estáticos o en movimiento y las observaciones pueden realizarse en cualquier momento del día. Para la obtención de coordenadas el sistema se basa en la determinación simultánea de las distancias a cuatro satélites (como mínimo) de coordenadas conocidas. Estas distancias se obtienen a partir de las señales emitidas por los satélites, las que son recibidas por receptores especialmente diseñados. Las coordenadas de los satélites son provistas al receptor por el sistema

Desde el punto de vista geodésico-topográfico, el Sistema GPS responde a dos requerimientos básicos:

Planteo directo o levantamiento: Se tiene en el terreno un punto materializado, un pilar con placa y marca, un mojón, etc. Se piden sus coordenadas en un sistema de referencia prefijado.

Planteo inverso o replanteo: Se dan las coordenadas de un punto en un sistema de referencia determinado y se pide la localización de dicho punto, que, de no estarlo ya, será materializado en el terreno.

La operatividad del sistema no implica un compromiso legal del gobierno de Estados Unidos. Por lo tanto la Agencia Cartográfica del Departamento de Defensa, NIMA (National Imagery and Mapping Agency) puede modificar sin previo aviso su funcionamiento alterando, por ejemplo, el denominado mensaje de navegación (en el que está incluida información esencial para el cálculo como son las coordenadas de los satélites), limitando el acceso a uno o más componentes de la señal, alterando el estado de los relojes, degradando la precisión de las órbitas, entre otros.

De todos modos el acceso a las señales que emiten los satélites es de carácter público, no requiriéndose licencia o autorización alguna, al menos hasta el año 2005.

En la actualidad, el uso civil de GPS ha sobrepasado largamente el uso militar, convirtiéndose de hecho en un servicio público de carácter mundial de enorme importancia y con innumerables aplicaciones.

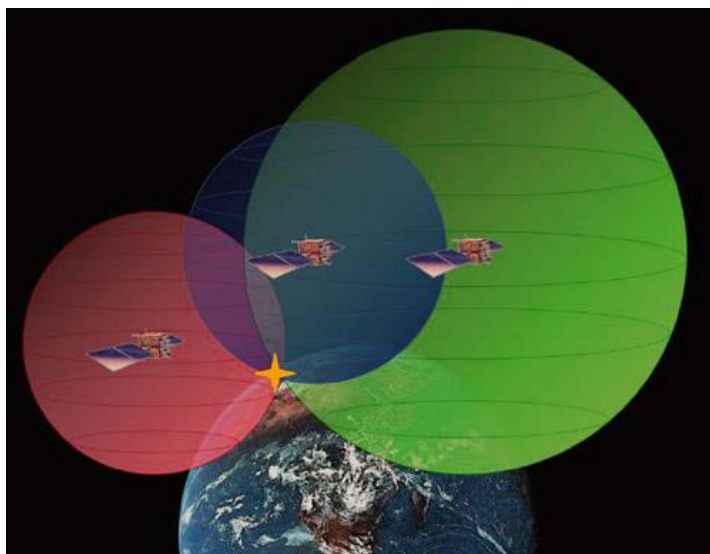


Ilustración 111. GPS Fuente: <http://www.gps-auto.org/navegador-gps/funcionamiento-gps.html>

3.2.6 Uso de las coordenadas GPS en Google Mapas.

Las coordenadas GPS están formadas por dos componentes que son latitud y longitud.

La posición norte-sur de un punto está dada por su latitud, mientras que su longitud brinda información de su posición este-oeste.

Latitud

La latitud de un punto es la medida del ángulo formado por el plano ecuatorial con la línea que une a éste punto al centro de la tierra.

Por regla general está comprendido entre -90° y 90° . Los valores negativos son para ubicaciones en el hemisferio sur, y el valor de la latitud es de 0° en el ecuador.

Longitud

La longitud tiene el mismo principio, con la diferencia en que no existe una referencia natural como lo es el ecuador para la latitud. La referencia para la longitud ha sido establecida arbitrariamente en el Meridiano de Greenwich, y la longitud de un punto es la medida angular formada por el semiplano del eje de la tierra que pasa por el meridiano de Greenwich, y el semiplano del eje de la tierra que pasa por el punto.

Un tercer componente

Los lectores meticolosos habrán notado que hace falta un tercer elemento para localizar un punto, este es la altitud. En la mayoría de los casos las coordenadas GPS son necesarias para ubicar lugares en la superficie terrestre, lo que le resta importancia a dicho parámetro. Sin embargo, es igual de necesario que la latitud y longitud para poder definir una ubicación GPS de manera completa y precisa.

3.2.7 Android.

Uno de los sistemas de operación para dispositivos móviles más usados a nivel mundial¹, sacando al mercado aplicaciones, ocasionó que las aplicaciones que existían solo para computadores (programas) también comenzaran a migrar y

1 Android el Sistema para dispositivos móviles más usado *Android*. (7 de agosto de 2013). Obtenido de <http://www.xatakandroid.com/mercado/android-eleva-su-cuota-de-mercado-hasta-el-80>

evolucionar a las características de los dispositivos móviles, siendo además compatibles casi con todo tipo de dispositivos; video llamadas en las redes sociales que inicialmente solo compartían publicaciones de imágenes.

4 DESARROLLO DEL PROBLEMA

A continuación se dará a conocer la solución del problema planteado, así mismo se expondrá los diferentes pasos y técnicas que se tuvieron en cuenta para realizar el desarrollo del aplicativo, y la forma en la cual se llevó a cabo la implementación del algoritmo para que este fuera funcional y efectivo en un dispositivo móvil.

A continuación se dará a conocer la solución del problema planteado, así mismo se expondrá los diferentes pasos y técnicas que se tuvieron en cuenta para realizar el desarrollo del aplicativo y la forma en la cual se llevó a cabo la implementación del algoritmo para que este fuera funcional y efectivo en un dispositivo móvil.

4.1 Software y Librerías

Las aplicaciones o proyectos de desarrollo del Android se pueden efectuar bajo los entornos de Windows y Linux,

4.1.1 Jdk

(Kit Development java) este software facilita las herramientas de trabajo y desarrollo en java, necesaria para usar con una aplicación adicional (editor de código) donde es llamada para creación de aplicaciones java.

4.1.2 Netbeans.

Uno de los entornos de desarrollo integrado (IDE) de preferencia para desarrollo de aplicaciones y proyectos en lenguajes de programación como C++, Java y Android, disponible en la página oficial de eclipse¹; para poder usar eclipse es necesario tener instalada la JDK.

4.1.3 Adt.

(Android Development Tools) son las herramientas con las cuales se trabaja en Android, es un plugin necesario para poder realizar desarrollo de Android en eclipse¹, dependiendo de la versión de eclipse se descargan todos los plugin, unas de las versiones de eclipse son Índigo, Juno que es el más recomendado, Kepler y Luna.

4.1.4 Sdk.

(Software Development Kit) es una de las herramientas que da las bibliotecas API también la encargada de la creación, prueba y depuración de los proyectos Android, también puede descargar un paquete completo que contiene Android Studio, ADT, CDT y SDK para desarrollo de Android.

4.1.5 Android Studio.

Android Studio, un IDE especializado para desarrollo en Android, Android Studio está basado en IntelliJ IDEA, un IDE para Java de JetBrains, con código nativo. La NDK se encuentra disponible en la página principal de Android Developers.

5 IMPLEMENTACION.

5.1 Uso del Algoritmo RRT.

Uno de los problemas para realizar una exploración de un terreno desconocido es el tipo de algoritmo a ejecutar, ya que por medio de este es posible obtener la información del terreno a explorar.

Para escoger el algoritmo más eficiente en el desarrollo de la aplicación; se hicieron estudios de los desarrollos y evoluciones más significativos a nivel lógico para el ámbito de exploración de terrenos, sobre los cuales se encontraron algoritmos que funcionan de diferentes maneras para realizar la exploración y determinar el camino más eficaz de un punto a otro entre estos están los determinísticos, Probabilísticos y aleatorios (algoritmos de búsqueda aleatoria), basados en optimización, los cuales requieren ciertas condiciones para su funcionamiento y

eficacia. Basándose en el estudio de cada uno de sus tipos y de su forma de implementación se tomó como mejor opción para el desarrollo algoritmos de tipos probabilísticos y aleatorios y entre sus tipos el más idóneo, por su funcionamiento y su forma de implementación fue el algoritmo RRT el cual puede generar el camino más rápido de un punto a otro son conocer el terreno por medio de la exploración del mismo.

El algoritmo RRT funciona realizando iteraciones desde el punto de inicio hasta el punto de llegada, si la ruta la cual se designó de manera aleatoria no se puede completar por algún tipo de motivo este genera otra iteración hacia una nueva ruta, haciendo poco a poco una exploración del espacio de donde se encuentra y constantemente revisando si en cada una de las nuevas iteraciones se encuentran o no se encuentran obstáculos que interfieran en el camino, cada nueva iteración genera una nueva rama de esta manera se va construyendo un árbol que poco a poco realiza la exploración de espacio maestro hasta que se haya la ruta para poder llegar desde el punto de inicio al punto final.

5.2 Variables del algoritmo.

Qini: Es el punto inicial de donde se inicia en el espacio maestro está punto se da por medio de dos coordenada latitud y longitud.

Qfin: Es el punto Final de donde se llega en el espacio maestro está punto se da por medio de dos coordenada latitud y longitud.

Edge Length: Es la distancia máxima en la cual se va a generar el nuevo punto aleatorio esta distancia nos permite limitar el crecimiento del árbol.

Iteración: esta variable nos permite dar un limitación a la cantidad de iteraciones que pueda hacer el algoritmo ya que este seguiría creciendo sin fin al no encontrar una solución el control de las iteración nos permite finalizar el crecimiento del árbol y hace evitar un desbordamiento en la memoria.

Cuando no se realiza una correcta parametrización del algoritmo de RRT puede llevar a un mal comportamiento del mismo lo que puede causar grandes errores en el su funcionamiento, y esto dará caminos erróneos, y poco fiables en sus caminos, además que esto evitaría llegar al punto final, podemos evidencia esto en la siguiente figura Ver figura [12].



*Ilustración 12. Simulación del comportamiento del algoritmo de RRT con mala parametrización.
Fuente Propia*

5.3 Ejecución de pruebas.

5.3.1 Diseño de Pruebas

1	Pruebas funcionamiento del GPS dentro del aplicativo móvil desarrollado
2	Ubicar un lugar de origen y el de destino mediante el aplicativo móvil y obtener una primera traza del aplicativo ejecutando el algoritmo de RRT el cual aplico en el desarrollo del aplicativo.
3	Seguir el Camino marcado en el mapa y generar obstáculos para que el algoritmo de RRT genere un nuevo camino para llegar a la ubicación del destino.
4	Realizar los pasos 2 y 3 repetidamente, observar el comportamiento del programa.

En la realización de las pruebas se hará de manera independiente, en diferentes lugares de inicio y de destino.

EL algoritmo del RRT de las simulaciones fue modificado para que pudiera ser funcional en un dispositivo móvil, por lo que las pruebas en el móvil varían de las pruebas de escritorio ya que el algoritmo podría tener un comportamiento distinto.

5.3.2 Funcionamiento del Aplicativo.

El aplicativo nos muestra dos tipos de interfaz distintas en la primera nos da la opción de controlar el vehículo por medio de una serie de botones los cuales pueden ser programables para que envíen una señal (en esta interfaz no se realizaron grandes manejos ya que se profundizo en la interfaz de manejo autónomo esta simplemente es una propuesta) Ver figura [13]

Para la interfaz de manejo autónomo tenemos un mapa el cual se ubica en la posición actual del equipo, esta interfaz nos permite tocar en cualquier parte del mapa para seleccionar la ubicación de destino. Ver figura [14]

Al seleccionar la ubicación de destino se debe dar clic en el botón de iniciar, el obtiene las coordenadas de los puntos de inicio y de llegada y empieza a trazar la ruta más corta por medio del algoritmo RRT, y es mostrada por medio de trazos en el mapa. Ver figura [15]

Para simular que se encuentra un obstáculo en el camino se incluye un botón en el cual indica al algoritmo que no se puede continuar en la trayectoria actual, Ver figura [16] de esta manera el algoritmo genera una nueva ruta, Ver figura [17], Ver figura [18] para poder continuar y evitar el obstáculo en nuestro medio para por ultimo llegar a nuestro destino. Ver figura [19], Ver figura [20] Ver figura [21]

5.4 Resultados.

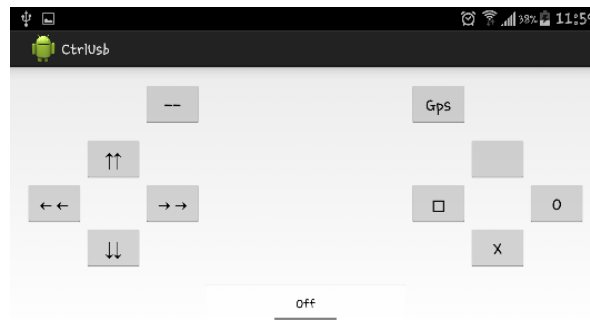


Ilustración 13. Imagen del inicio del programa (Prototipo). Fuente propia



Ilustración 14. Imagen con lugar de origen y lugar de destino simulación. Fuente propia

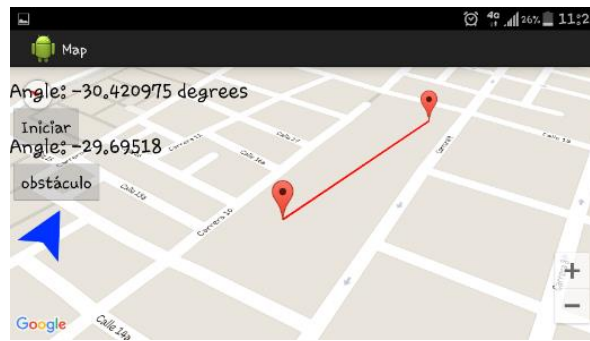


Ilustración 15. Imagen Camino más cortó hallado por el programa. Fuente propia

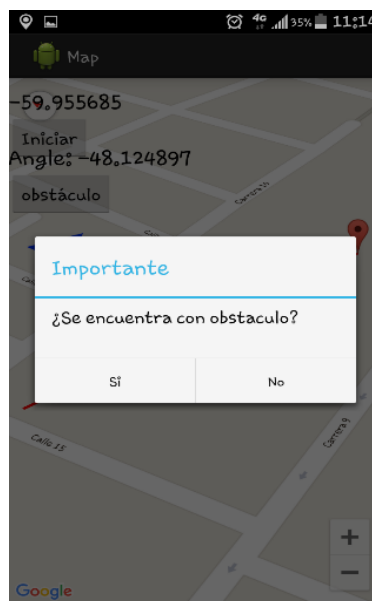


Ilustración 16. Simulación de un obstáculo en el camino. Fuente propia

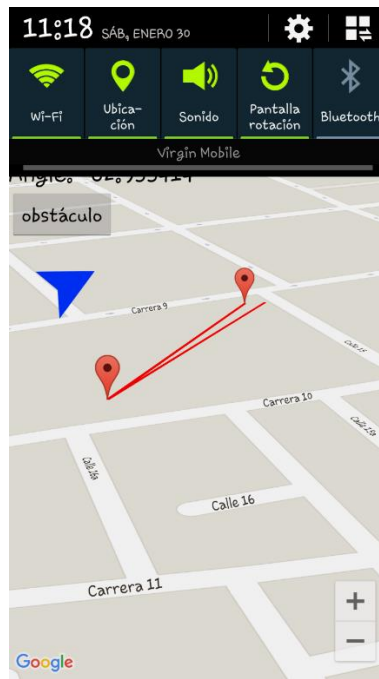


Ilustración 17. Segundo camino trazado por el programa. Fuente propia

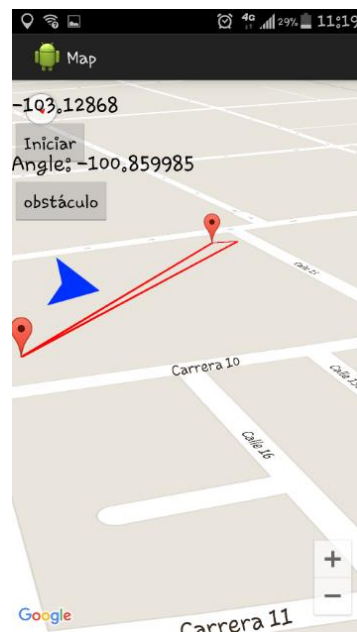


Ilustración 18. Llegada al punto final. Fuente propia

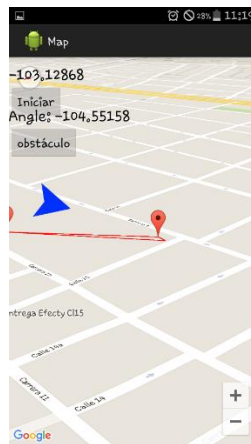


Ilustración 19. Llegada al punto final #2. Fuente propia

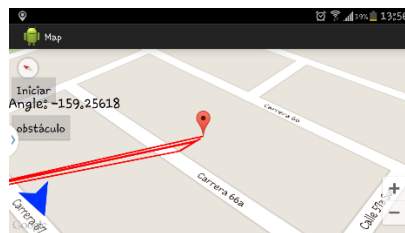


Ilustración 20. Simulación con varios caminos #1- Fuente propia

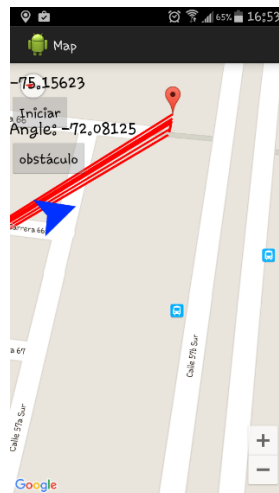


Ilustración 21. Simulación con varios caminos #1. Fuente propia

5.4.1 Tabla de Resultados

N° Intento	N° Obstáculos	N° Rutas
1	0	1
1	1	2
1	2	3
1	3	4
2	0	1
2	1	2
2	2	3
3	0	1
3	1	2
3	2	3
3	3	4
4	0	1
4	1	2
4	2	3
4	3	4

6 Conclusiones, limitaciones y planes futuros.

El objetivo de este trabajo es desarrollar una aplicación móvil, la cual por medio de algoritmos RRT pueda generar una ruta para llegar de un punto a otro en un espacio geográfico muestral no identificado, para lograrlo se realizó una implementación en S.O Android con el uso del GPS, Google Maps, y un algoritmo de RRT.

Las consultas realizadas a los diferentes tipos de algoritmos de planificación de trayectorias y en especial la consulta realizada a los algoritmos de RRT, nos permitieron usar sus cualidades en la exploración de terrenos geográficamente desconocidos, permitiéndonos realizar una exploración del terreno y generar rutas alternas para sortear obstáculo, y gracias a la información suministrada tanto por la manipulación directa del móvil, y el uso de sus sensores.

Este trabajo muestra un comportamiento de un algoritmo RRT en un dispositivo móvil, considerando que es una nueva forma de implementación de este algoritmo para su uso dentro de un dispositivo este puede comportarse de manera errónea por lo que los resultados obtenidos son una referencia para futuras implementaciones.

Es posible implementar este algoritmo para guiar a diferentes tipos de vehículos no tripulados, en sistemas holónomos, no holónomos como drones, autos, entre otros, para realizar tareas de mensajería, entrega de mercancía en lugares de difícil acceso podemos ver algunos ejemplos en los vehículos tipo Ackerman.

Realizar mejoras en la ejecución de algoritmo RRT en el aplicativo la depuración constante y el manejo más preciso de las variables del mismo podrá hacer de las dependientes del mismo ayudara a que el cálculo de las rutas se pueda hacer de manera más eficiente.

Mejoramiento de la interfaz, lo que ayudara al usuario que tenga contacto con el desarrollo a que sea más eficiente y eficaz en su trabajo, además podría proveer de muchas más información como velocidad de desplazamiento, altura entre otros.

Realizar un log en el cual puedan ser guardadas y consultadas rutas ya recorridas para su posterior uso, entre un mismo punto de origen y de llegada ayudando a mejorar la velocidad de desplazamiento del vehículo, teniendo en cuenta que si en este recorrido ya no se pudo completar por alguna circunstancia externa este log pueda ser actualizado y nuevamente guardado.

7 Bibliografía

1. *Android Developers*. (s.f.). <https://developer.android.com/index.html>
2. [D. Lopez, F. Gomez-Bravo, F. Cuesta, A. Ollero, PLANIFICACION DE RAYECTORIAS CON EL ALGORITMO RRT. APLICACION A ROBOTS NO HOLONOMOS, 2006]
3. [Miguel Ángel Fernández Lancha, David Fernández Sanz, Carlos Valmaseda Plasencia, PLANIFICACION DE TRAYECTORIAS PARA UN ROBOT MOVIL Universidad Complutense Madrid 2009]
4. [Helbert Eeduardo Espitio Cuchango, PROPUESTA DE UN ALGORITMO PARA LA PLANEACION DE TRAYECTORIAS DE ROBOTS MOVILES EMPLEANDO CAMPOS POTENCIALES Y ENJAMBRES DE PARTICULAS ACTIVAS BROWNIANAS UNIVERSIDAD NACIONAL DE Colombia, 2011]
5. [Barloto Luque, Fernando Ballesteros S, Jordi Luque Y Juan Carlos Nuno DEPARTAMENTO DE MATEMÁTICA APLICADA ETSI AERONÁUTICOS, UNIVERSIDAD POLITÉCNICA DE MADRID, UN MAPEO ENTRE SERIES TEMPORALES Y REDES COMPLEJAS: EL GRAFO DE VISIBILIDAD]
6. [Hrvoje Kaluder, Misel Brezak, y Ican Petrovic A VISIBILITY GRAPH BASED METHOD FOR PATH PLANNING, FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING, UNIVERSITY OF ZAGREB MIPRO 2011, May 23-27, 2011, Opatija, Croatia].
7. José Luis Moreno Duran, Sergio Ordoñez Pérez. Diagramas de Voronoi de alcance limitado. 19 de Junio de 2009.
8. [James L. Crowley, DYNAMIC MODELING OF FREE-SPACE FOR A MOBILE ROBOT, Institut National Polytechnique de Grenoble, 1989].
9. [David Alejo Teissiere, APLICACIÓN DEL ALGORITMO PRM A LA PLANIFICACIÓN DE CAMINOS DE ROBOTS MÓVILES, 5 de octubre de 2007].

- 10.[Marcos Gestal Pose, INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS Depto. Tecnologías de la Información y las Comunicaciones Universidad de Coruña]
- 11.[Franco Luis Alejandro Arito, Algoritmos de Optimización basados en Colonias de Hormigas aplicados al Problema de Asignación Cuadrática y otros problemas relacionados. Universidad Nacional de San Luis Facultad de Ciencias Físico Matemáticas y Naturales Departamento de Informática, San Luis – Argentina Abril de 2010].
- 12.[José Manuel García Nieto, Algoritmos Basados en Cúmulos de Partículas Para la Resolución de Problemas Complejos, septiembre de 2006].
- 13.[F. Gómez-Bravo*, A. Ollero, D. López*, F. Cuesta, M. del Toro*, P. Gil, F. Real, RRT-D: PLANIFICACIÓN DISTRIBUIDA DE CAMINOS BASADA EN LA INFORMACIÓN DE UNA RED DE SENSORES WIRELESS. Departamento de Ingeniería de Sistemas y Automática. Escuela Superior de Ingenieros. Universidad de Sevilla]
- 14.[Conde Ojeda, Roberto, APLICACIÓN DEL ALGORITMO RRT A LA PLANIFICACIÓN DE CAMINOS PARA ROBOTS MÓVILES EN CONFIGURACIÓN ACKERMAN,]
- 15.A Neiw Algorith foor GPS-ased Vehicle Navigation System Han Baomin and Qu Guoqing School of Architecttue and Engineering, Shandong Unixersity of Technology, Zhangzhou road 12, Zibo, Shandong province, China, 255049