

# ALGORITMOS PARA EL CONTROL DE ESPACIOS EN LAS ABEJAS ROBÓTICAS

Juan Andres Arroyave  
Universidad Eafit  
Colombia  
[jaarroyavs@eafit.edu.co](mailto:jaarroyavs@eafit.edu.co)

Juan C. Iguaran Fernandez  
Universidad Eafit  
Colombia  
[jciguaranf@eafit.edu.co](mailto:jciguaranf@eafit.edu.co)

Gustavo Lopez  
Universidad Eafit  
Colombia  
[galopezg@eafit.edu.co](mailto:galopezg@eafit.edu.co)  
[o](#)

Mauricio Toro  
Universidad Eafit  
Colombia  
[mtorobe@eafit.edu.co](mailto:mtorobe@eafit.edu.co)

## RESUMEN

Queremos aportar al futuro problema de escases de abejas, sin ellas no se puede llevar a cabo el proceso de polinización y esto afectaría gravemente el medio ambiente. Para solucionar esto se está intentando hacer abejas robot que puedan cumplir con estas funciones. El proyecto necesita mejorar ciertos aspectos que es en lo que nos enfocaremos como por ejemplo el control de espacios que hace que estás no colisionen entre sí.

## 1. INTRODUCCIÓN

Es la justificación de las condiciones en el mundo real que llevan al problema. En otras palabras, es hablar sobre qué va a tratar el documento e incluir la historia de este problema. (Queremos entrar en el problema de la escases de abejas, en un futuro habrán muy pocas abejas lo cual hará un gran daño ambiental ya que estas son las que hacen el proceso de polinización, nosotros en el amplio concepto de las abejas robóticas, queremos enfocarnos en el control de espacios que estas deben de tener.)

## 2. PROBLEMA

El problema es el control del espacio en las abejas, estas tienen que tener ciertos algoritmos que impida que colisionen entre ellas o que polinicen las mismas flores.

## 3. TRABAJOS RELACIONADOS Aquí deberán explicar

### 3.1 Funciones Bash

El algoritmo hash se designa para minimizar la posibilidad de que dos entradas tengan el mismo valor de hash, que se denomina colisión. La función hash participa en la localización de puntos cercanos entre sí. La función hash busca las celdas contiguas en la cuadrícula, evitando que se repitan esas celdas. Lo primero que tenemos que hacer para implementar el algoritmo es crear el hash espacial, luego proporcionarle un parámetro que será el tamaño de la celda. Toma un punto como parámetro, divide cada elemento por el tamaño de celda y lo convierte en entero. El tamaño de la celda será el tamaño del terreno y el resultado de los datos divididos es en donde se pararán las abejas evitando su colisión, y esos puntos son las diferentes celdas en donde la abeja se parará. Esta función bash es muy interesante y sirve de mucha ayuda

### 3.2 Quadtree

Se utiliza para describir clases de estructuras de datos

jerárquicas cuya propiedad común es que están basados en el principio de descomposición recursiva del espacio. Al insertar un nuevo elemento, el espacio queda dividido en

cuatro cuadrantes. Al repetir el proceso, el cuadrante se divide de nuevo en cuatro cuadrantes, y así sucesivamente. La familia Quadtree se usa para representar puntos, áreas, curvas, superficies y volúmenes. Los puntos se insertan de una manera similar a la usada en el Point Quadtree. Primeramente, se hace una búsqueda para encontrar el subcuadrante a que el nodo pertenece (el nodo echa hojas). Si este subcuadrante ya está ocupado para otro nodo B con las coordenadas diferentes, entonces este cuadrante debe subdividirse en las partes necesarias para que los nodos A y B no ocupen el mismo cuadrante. Esto puede causar un gran número de subdivisiones, sobre todo si los dos puntos se contienen en un bloque muy pequeño del espacio. Los nodos interiores no guardan los datos, desde el punto que ocupó el cuadrante previamente (el punto B) si se vuelve el hermano del nuevo punto (el punto A). Como resultado se observa que todos los nodos grises poseen a dos descendientes que contienen los datos como mínimo.

### 3.3 Árboles AABB

los árboles AABB son estructuras de datos de subdivisión espacial y un conjunto de algoritmos usados principalmente para realizar intersecciones y calcular distancias de forma eficiente en objetos geométricos en 3D con un número finito de puntos. El árbol AABB es construido a partir de un conjunto inicial de elementos (puntos, triángulos, polígonos, etc.). Este conjunto es ordenado a lo largo del mayor de los ejes de coordenada de esta caja. Los elementos del conjunto inicial son separados en dos subconjuntos iguales o al menos balanceados. Este método es aplicado recursivamente hasta llegar al caso base. Se suelen tomar varios casos bases, el tamaño de las cajas, la cantidad de subdivisiones, pero generalmente se usa la cantidad de elementos en una caja; aunque también se pueden combinar varios de estos casos bases.

## REFERENCIAS

Myopic Rhino. Spatial Hashing. Gamedev. Recuperado en Febrero 17, 2019

<https://www.gamedev.net/articles/programming/general-and-gameplay-programming/spatial-hashing-r2697/>

James. Introductory Guide to AABB Tree Collision Detection. Azure From The Trenches. Recuperado Febrero 17, 2019

<https://www.azurefromthetrenches.com/introductory-guide-to-aabb-tree-collision-detection/>

Quadtree, Wikipedia. Recuperado Febrero 17, 2019

<https://es.m.wikipedia.org/wiki/Quadtree>

IBM PureData System for Analytics. IBM, Version 7.1, recuperado Febrero 17, 2019

[https://www.ibm.com/support/knowledgecenter/es/SSULQD\\_7.1.0/com.ibm.nz.sqltk.doc/c\\_sqlxt\\_hashing.html](https://www.ibm.com/support/knowledgecenter/es/SSULQD_7.1.0/com.ibm.nz.sqltk.doc/c_sqlxt_hashing.html)

Árbol AABB, Wikipedia. Recuperado Febrero 17, 2019

[https://es.m.wikipedia.org/wiki/%C3%81rbol\\_AABB](https://es.m.wikipedia.org/wiki/%C3%81rbol_AABB)