

Proyecto (Entrega 2): Localización de abejas robóticas (Estructura de datos)

Universidad Eafit

Cristian Darío Ceballos Rodríguez
cdceballor@eafit.edu.co
Mauricio Toro

Universidad Eafit

Sebastian Loaiza Correa
sloaizac@eafit.edu.co
Mauricio Toro

RESUMEN

Se hace necesario encontrar una estructura de datos eficiente que implementar para el tratamiento de datos de ubicación de abejas robóticas para la prevención de colisiones entre ellas. Es importante encontrar una estructura efectiva para la solución de este problema ya que existe una falta de efectividad en el tratamiento de cantidades de datos de tamaños considerables entonces se hace necesario dividir o clasificar estos datos para poder construir un algoritmo mucho más efectivo para darle solución a este problema. Otros problemas relacionados a este podrían ser las colisiones en los videojuegos tanto entre personajes como con los diferentes objetos del entorno de juego, entre otros.

Palabras claves

Abejas; estructuras de datos; posicionamiento; tamaño; distancia; algoritmos; manipulación.

Clasificación de la ACM

- Algoritmo de ubicación
- Dominio de Software y contextualización
- Robótica

1. INTRODUCCIÓN

En el momento en el cual nos dedicamos a solucionar o tratar este tipo de problemas nos damos cuenta que cada vez se hace más complicado ya que la cantidad de datos va en un aumento constante y las soluciones pierden efectividad ya que lo que se necesita es que este tipo de cálculos y análisis de datos deben hacerse de manera rápida y de una manera constante.

En este documento enunciaremos con más detalle el problema sobre detección de colisiones y/o prevención de estas, también se darán a conocer un poco los trabajos en el mundo de la computación relacionados con este tipo de problemas que nos puedan brindar una alternativa de solución.

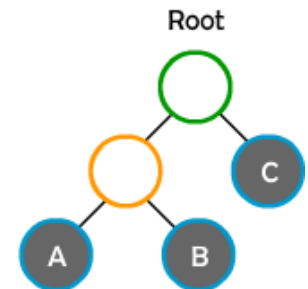
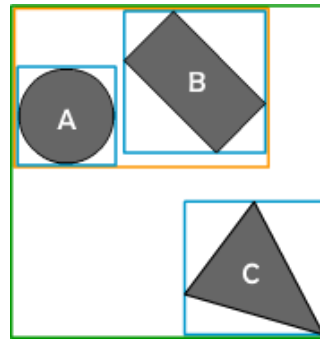
2. PROBLEMA

El problema por resolver es la implementación de una estructura de datos la cual nos permita manejar y calcular de forma eficaz los datos de posición en este caso de abejas y como se relacionan entre ellas en concreto calcular la posibilidad de colisiones entre las abejas se tratará de solucionar este problema para reducir a lo sumo el número de cálculos y comparaciones necesarias para poder detectar dichas colisiones.

3. TRABAJOS RELACIONADOS

3.1 Clase AABB

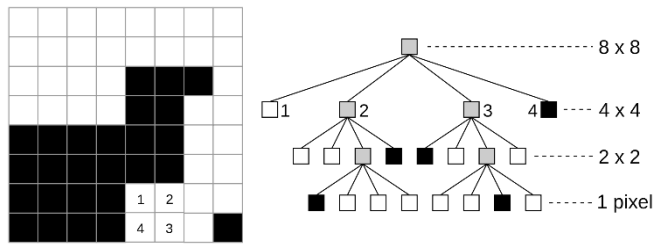
Esta estructura de datos se puede usar para representar límites aproximados de objetos, en situaciones donde se pueden evitar cálculos detallados a nivel de triángulo. En los sistemas físicos, los cuadros delimitadores se usan como una prueba de salida temprana eficiente para las consultas de intersección de geometría.



3.2 Quadtree

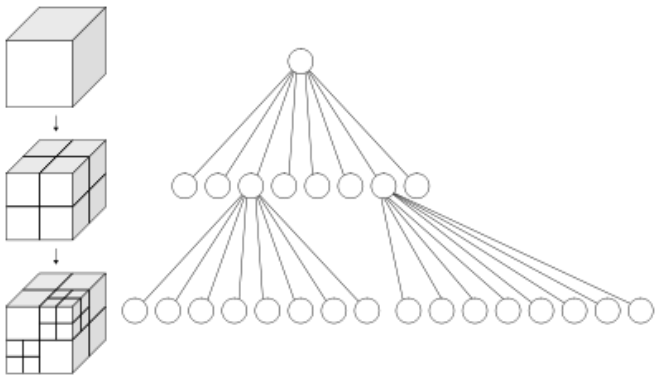
Se utiliza para describir clases de estructuras de datos jerárquicas cuya propiedad común es que están basados en el principio de descomposición recursiva del espacio. En un QuadTree de puntos, el centro de una subdivisión está siempre en un punto. Al insertar un nuevo elemento, el espacio queda

divido en cuatro cuadrantes. Al repetir el proceso, el cuadrante se divide de nuevo en cuatro cuadrantes, y así sucesivamente.



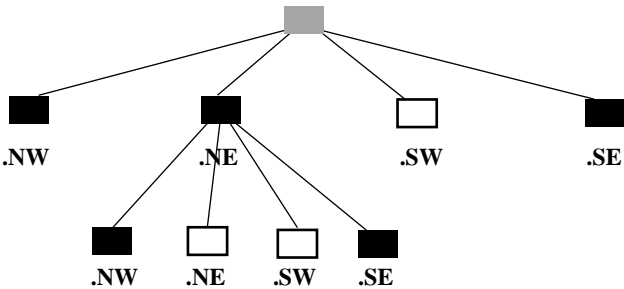
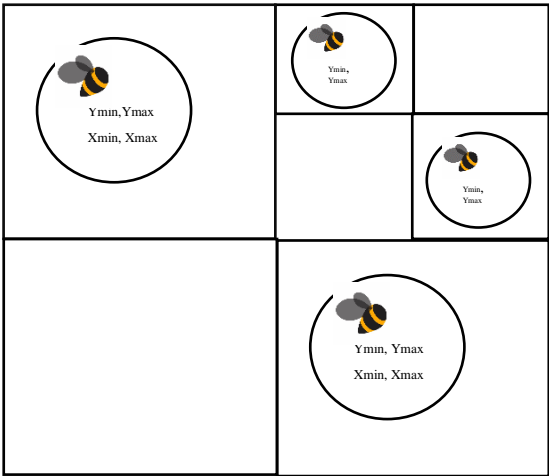
3.3 Octree

Es una estructura en "árbol" de datos en la cual cada nodo interno tiene exactamente 8 "hijos". Las estructuras octree se usan mayormente para particionar un espacio tridimensional, dividiéndolo recursivamente en ocho octantes. Las estructuras octree son las análogas tridimensionales de los quadtree bidimensionales



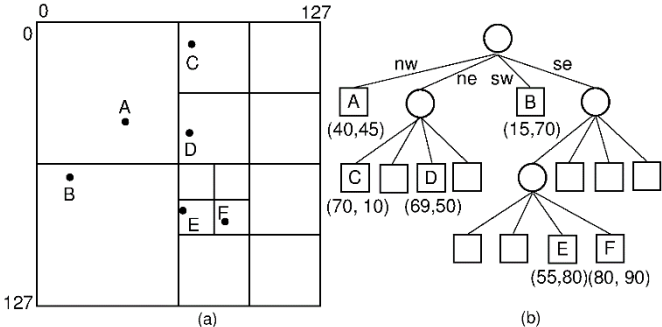
4. QUADTREE

El término Quadtree, o árbol cuaternario, 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. En un QuadTree de puntos, el centro de una subdivisión está siempre en un punto. 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.



4.1 Operaciones de la estructura de datos

Insert.



4.2 Complejidad.

En caso tal de haber una gran cantidad de números para evaluar, el algoritmo comienza a ser $O(n)$ lo cual le va aumentando la complejidad asintótica con cada operación. Podríamos afirmar que la complejidad asintótica se define con respecto al total de datos a manejar.

REFERENCES

<https://en.wikipedia.org/wiki/Octree>

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

http://clb.demon.fi/MathGeoLib/nightly/docs/AABB_summary.php

- I. Posicionamiento.
http://catarina.udlap.mx/u_dl_a/tales/documentos/msp/ra_mirez_o_md/capitulo_2.html
- II. Algoritmo para detectar colisiones entre rectángulos.
<https://alfffa.wordpress.com/2010/12/25/algoritmo-paradetectar-colisiones-entre-rectangulos/>