

PROYECTO FINAL: LOCALIZACIÓN DE ABEJAS ROBÓTICAS EN 3D

(Optimización considerando el tiempo de algoritmo que verifica la posible colisión de abejas robóticas)

Pablo Alberto Osorio Marulanda
Universidad Eafit
Medellín, Colombia
paosoriom@eafit.edu.co

Verónica Mendoza Iguarán
Universidad Eafit
Medellín, Colombia
vmendozai@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

RESUMEN

Debido a que la implementación de abejas robóticas es algo nuevo para la población, probablemente, la predicción de los movimientos de estas pueda hacerse difícil; esto a motivado la creación de un algoritmo que proporcione la localización de las abejas robóticas que se encuentran a 100 metros de distancia o menos, para evitar las colisiones entre ellas; con el fin de evitar que aquello que en un principio se pensó como solución al gran daño que ha causado la disminución de población de abejas, se convierta en un daño mayor a los agricultores e incluso a la población en general. El proyecto se enfocará en la optimización de un algoritmo que ejecute tal función.

1. INTRODUCCIÓN

Dada la gran necesidad de una nueva forma de preservación del medio ambiente y de la supervivencia del ser humano, mediante la polinización de cultivos con abejas robóticas que suplen las necesidades de los sistemas globales al igual que abejas reales, nos vemos en la necesidad de prever y anticipar posibles problemas futuros que estas puedan causar.

Un problema a considerar son las posibles colisiones entre las abejas, es decir, las aglomeraciones y la mala distribución de estas, generando así, posiblemente otros graves daños futuros. Este proyecto se realiza con el fin de implementar un algoritmo que permita solucionar el problema mencionado, mediante la búsqueda de problemas algorítmicos similares y sus soluciones, que proporcionen diferentes percepciones para una mejor y más completa solución de este.

2. PROBLEMA

El problema que se busca solucionar mediante la ejecución de este proyecto es la instauración y optimización (priorizando el tiempo) de un algoritmo que, a través de las coordenadas geoespaciales de cada abeja, prevenga las colisiones entre ellas, evaluando las abejas que se encuentran a 100 o menos metros entre ellas. Este problema sugiere la solución a los posibles que puedan surgir al momento de polinización de cultivos y otros procesos ejecutados por las mismas.

3. TRABAJOS RELACIONADOS

3.1 Quadtree

El “Quadtree” es un algoritmo que usa un formato de representación 2D para codificar imágenes. Su idea básica es dividir sucesivamente un plano en ambas direcciones para obtener cuadrantes, llegando así a una subdivisión recursiva del espacio[3]. Cuando un cuadrante contiene datos se denomina como área negra, los vacíos como área blanca y las áreas que contienen a ambos son áreas grises[3]. De esta manera, se verifica si cada cuadrante se encuentra lleno, vacío, o con contenidos de ambos tipos, todo esto dependiendo que tanto el área intersecta el cuadrante. Así, los cuadrantes parcialmente lleno se subdivide recursivamente de nuevo en cuadrantes llenos y vacíos, todo esto hasta un límite determinado. Lo anterior es ejecutado mediante una subdivisión arbórea, donde las estructuras jerárquicas y las propiedades de cada nodo describen el cuadrante que estas representan. Con esto, conteniendo un grupo de puntos se pueden ejecutar tareas tales como determinar si un punto selecto se encuentra contenido en un grupo dado o finalmente seleccionar un grupo de puntos que se relacionen con algún criterio[1].

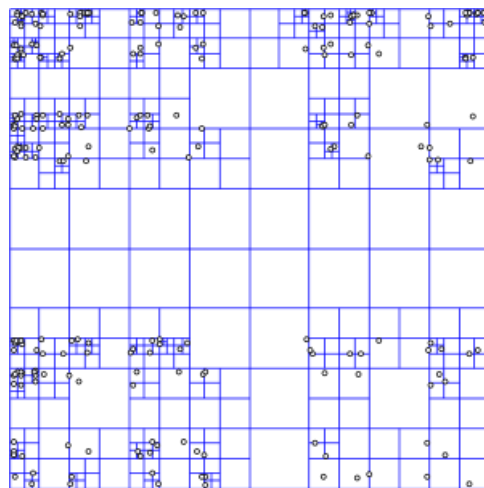


Imagen tomada de:
https://en.wikipedia.org/wiki/File:Point_quadtree.svg

3.2 Bounding Boxes-Sprites

Este algoritmo se desarrolla mediante la sobre-inscripción de figuras geométricas sobre los objetos, desarrollando una especie de “hit-boxes”. De esta manera, cada figura (por lo general círculos y rectángulos) representan las dimensiones del objeto en el algoritmo de detección. Este es un

algoritmo basado en la serie de McLaurin, donde se calcula la distancia entre dos puntos mediante derivadas aproximaciones matemáticas, con una precisión cercana al 97%. [3]

Tales figuras tienen entonces lados paralelos a los ejes x e y (z para 3 dimensiones), que son lo suficientemente grandes para que el modelo completo quede dentro. El centro de estos determinan su posición. Con la posición de cada objeto en el espacio se puede hallar la distancia entre ellos y teniendo en cuenta que, para que no hayan chocado, esta distancia debe ser mayor a la suma de las distancias del centro a una arista en cada figura. La selección de si cada objeto vendrá representado por círculos o rectángulos no es arbitraria, por lo que se recomienda que “Usa rectángulos para aquellos sprites que no necesiten rotaciones, usa círculos para todo lo demás.” [4]

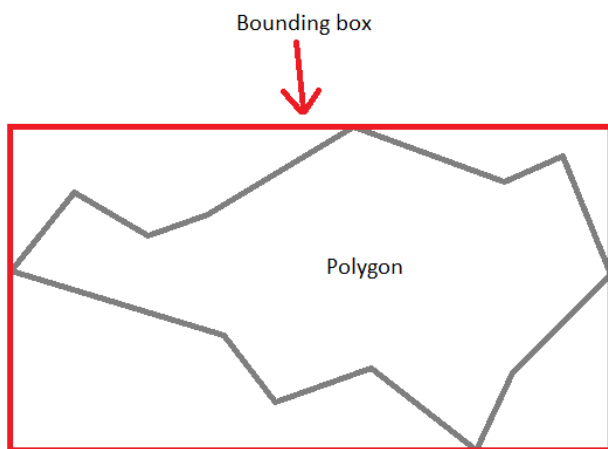


Imagen tomada de: <https://i.stack.imgur.com/wljel.png>

3.3 DES (Data Encryption Standard)

DES es un algoritmo de cifrado, es decir, un método para cifrar información, escogido como un estándar FIPS en los Estados Unidos en 1976, y cuyo uso se ha propagado ampliamente por todo el mundo. [2]

Este lleva un proceso de cifrado por bloques, donde se toma un texto en claro de longitud fija de bits y se transforma mediante una serie de operaciones en otro texto cifrado de la misma longitud. [2] El tamaño del bloque de DES es de 64 bits, y la longitud de clave también es de 64 bits, aunque en realidad solo son 56 bits los que pertenecen a la clave ya que los otros 8 bits son de comprobación de paridad, y se descartan. [2]

Hoy en día, DES se considera inseguro para muchas aplicaciones. Esto se debe principalmente a que el tamaño de clave de 56 bits es corto; las claves de DES se han roto en menos de 24 horas.

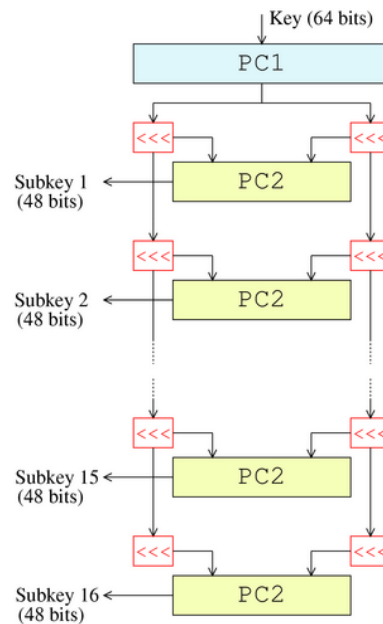


Imagen tomada de: <https://upload.wikimedia.org/wikipedia/commons/thumb/0/06/DES-key-schedule.png/250px-DES-key-schedule.png>

3.4 Árboles BSP (Binary Space Partitioning)

Los árboles de partición binaria del espacio se desarrollaron gracias a Fuchs y Naylor. [5] Este es un método eficiente utilizado para calcular las relaciones de visibilidad entre un grupo de polígonos (estáticos) observados desde una perspectiva arbitraria [5]. El árbol se genera tomando cualquier polígono como la raíz. Desde ese subespacio se divide el espacio total en dos cuadrantes, donde uno contiene los polígonos delante de él y otros los que se encuentran tras él. Si un polígono pertenece a ambos subespacios es dividido por el plano, y si otro polígono pertenece al plano raíz se pone en cualquier subespacio [5]. De esta manera, los subespacios “hijos” se dividen nuevamente de forma recursiva para la creación de nuevos subespacios. El nodo tiene llamados recursivos hasta que cada polígono tiene un nodo, es decir, un subespacio único. Así se podrían evaluar los nodos que no son únicos en cierto momento de la recursión, verificando

que estos, dado que comparten un subespacio específico, por lo tanto, no les corresponde un subespacio único.

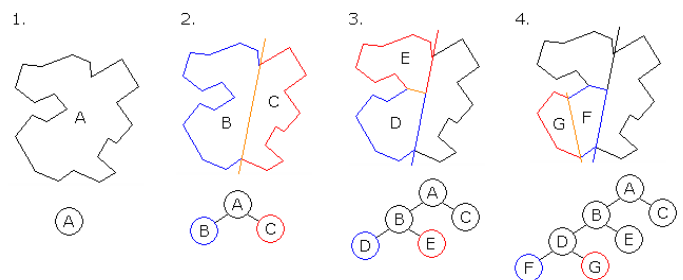


Imagen tomada de:
http://www.esacademic.com/pictures/eswiki/66/Binary_space_partition.png

REFERENCIAS

1. Artículo tomado de :
<https://es.wikipedia.org/wiki/Quadtree#Uso>
2. Artículo tomado de
https://es.wikipedia.org/wiki/Data_Encryption_Standard
3. Gallego, J and Pradas, S *Colisiones 2D en los videojuegos*, DCCIA (Departamento de ciencia de la computación e Inteligencia artificial) , 2002.
4. Nickname: ADRIGM , *Teoría de colisiones 2D: Conceptos básicos* , Genbeta (weblog), 29 de junio 2013
5. Universidad de las américas, *Árboles BSP*, Cap. 5, Puebla, México.