# ALGORITHMIC CONTROL AND SUPERVISION FOR THE PREVENTION OF COLLISION BETWEEN ROBOTIC BEES

Hamilton Smith Gómez Osorio
Universidad EAFIT
Colombia
hsgomezo@eafit.edu.co

Santiago Isaza Cadavid
Universidad EAFIT
Colombia
sisazac@eafit.edu.co

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

## ABSTRACT

Robotic bees are the future of agriculture in their process of production, that is why makes necessary having control and supervision of the exact location of these in order to achieve an optimal behavior.

## KEYWORDS

Sorting and Searching, Data structures, Hashing, Data Management.

## ACM CLASSIFICATION System

CCS → Theory of computation → Design and analysis of algorithms → Data structures design analysis → Sorting and searching.

## 1. INTRODUCTION

Faced with the decline of the population of bees that is happening nowadays and the importance of these in the process of pollination and in the agricultural sector it is possible to say that there is a risk in agricultural crops, so it is necessary to find a solution to this problem. This is how the idea of creating robotic bees was born, which can help in this process and, to supervise them, and control their behavior, develop and implement and algorithm that prevents their collision.

## 2. PROBLEM

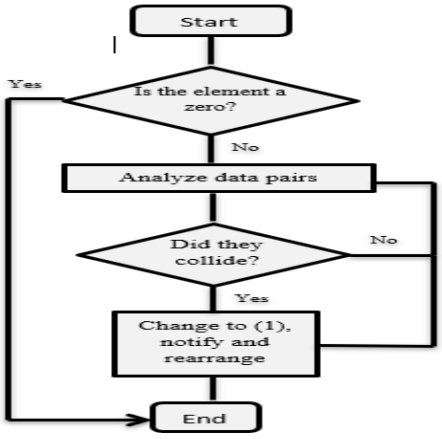
The implemented robotic bees in agriculture for the pollination process can collide if they are less than 100 meters from other bees, that is why is so important solve the problem in order to achieve and optimal behavior and an improvement in their processes.

## 3. RELATED WORKS

### 3.1 Radio Frequency Identification System (RFIS)

When there are several labels (used to store the information) and readers (read, change and verify the information on the label) in the same channel and signal transmission, a collision problem occurs due to mutual interference between the labels and the readers.

**Solution**: an anti-collision algorithm based on a matrix and a coding scheme. The decoded data is established in a matrix and then the reader is responsible for processing the data row by row, analyzed in pairs and finding a collision is replaced by a value (1), otherwise set a zero (0). After replacing the rows, the collisions are extracted and the following rows are analyzed until the algorithm finishes.



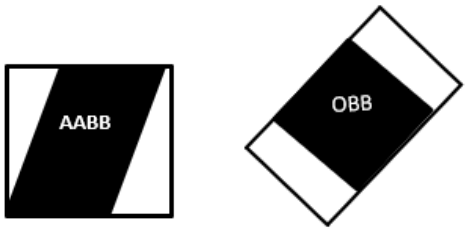

[1]

Figure 1: RFID Flowchart

### 3.2 Bounded volumes to detect collisions

Very used in graphic computing when making videogames. It is based on the use of basic geometric shapes bounding more complex figures and using the intersection of these to determine when a collision occurs; thanks to the figures you have control of the objects when there is movement or perspective changes, a solution used in AABB (Axis-

aligned bounding box) and OBB (Oriented bounding box).




[2]

Figure 2: Bounding figures

### 3.3 Octree

To analyze collisions between multiple elements in a plane, this method is used, which subdivide a delimited space into rectangles of equal size and then divides the space again until it has a relatively smaller area in which is easier and faster to compare positions of one object and another, considering its volume and coordinates. [3]

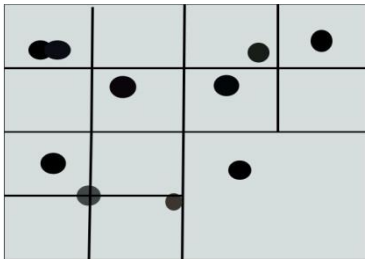


Figure 3: Octree subdivisions

### 3.4 Data structure spatial hashing

It consists of dividing a zone into cubes with a specific measure, considering the maximum and minimum value of coordinates, and then organizing the objects that are inscribed within them in a list of reference to the index of the box in which it is located. [4]

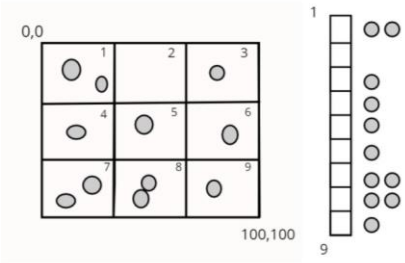



Figure 4: Subdivision and classification by boxes.

### 4. Structure to use: Spatial Hashing

### 4.2 Design criteria for the data structure

This structure is more efficient compared to the others presented because in the construction of the algorithm we can define the size of each cell so we do not need to analyze every single position and compare to the others (brute force) because the fact that two bees are in the same cell means that there is a collision risk. So, with this structure it is possible to save memory, avoiding any type of classification and query, as well as making the algorithm faster.

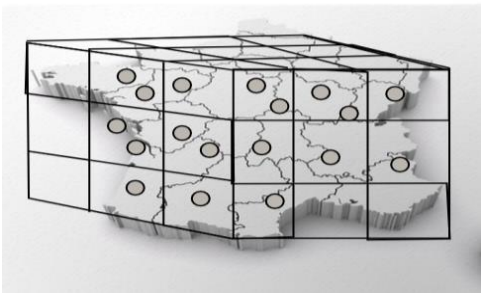


Figure 5: surface bounded by boxes

### Implementation approach

In the Java Programming language, we seek to implement an optimal solution with the Spatial Hashing data structure, which, using cubes with a volume of 10m3, delimiting the territorial extension of the municipality of Bello, inscribing the robotic bees located there, classifying them in a list and then return those bees that are in the same cube and recognize the ones that at risk of collision.

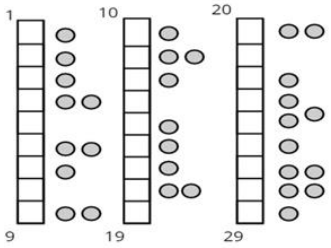



Figure 5: Organization of bees according to their position

**ACKNOWLEDGEMENTS**

We are especially grateful to the national education project Ser Pilo Paga of the Government of Colombia and to the Fundación scholarship of Universidad EAFIT for financially support our education.

We would also like to thank Daniel Mesa, instructor of the Data Structures and Algorithms I course at Universidad EAFIT, for his advice in the learning process, for his contributions in class and his guidance in the realization of this project

**REFERENCES**

[1]. Liu, B. and Su, X. An Anti-Collision Algorithm for RFID Based on an Array And Encoding Scheme. Information, 2018, 2078-2489. Retrieved August 25, 2018 from Eafit University: https://bit.ly/2PEzPhx

[2]. Dinas, S. and Bañón J. M. A literature review pf bounding volumes hierarchy focused on collision detection. Ingeniería Competitiva, 2015, 49-62. Retrieved August 25, 2018, from Eafit University: https://bit.ly/2BMI9sD

[3]. Nevala, E. Introduction to Octrees. GameDev.net, 2018

https://gamedev.net/articles/programming/general-and-gameplay-programming/introduction-to-octrees-r3529/.

Accessed September 23, 2018.

[4]. Spatial hashing implementation for fast 2D collisions. The mind of Conkerjo, 2013. https://conkerjo.wordpress.com/2009/06/13/spatial-hashing-implementation-for-fast-2d-collisions/. Accessed September 23, 2018.

[5]. How to efficiently remove duplicate collision pairs in spatial hash grid? Stack Overflow, 2015. https://stackoverflow.com/questions/31124702/how-to-efficiently-remove-duplicate-collision-pairs-in-spatial-hash-grid. Accessed September 23, 2018.