

# AVOIDING COLLISION: A SPATIAL STUDY OF POSITION ON AUTOMATED ROBOTIC BEES

Rafael Mateus Carrión  
Universidad Eafit  
Colombia  
rmateusc@eafit.edu.co

Daniel Otero Gómez  
Universidad Eafit  
Colombia  
doterog@eafit.edu.co

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

## ABSTRACT

The purpose of this research paper is to propose a solution to avoid the collision of automated robotic bees. Georeferences as geodesic coordinates are essential to recognize and solve this problem, therefore, they will be present throughout the report. Moreover, an analysis of the subproblems is included in the first section, as well, as possible solutions for themselves.

## 1. INTRODUCTION

Undoubtedly, technology is each time more present in our daily life, and automated robots are a clear example of it. Robotic bees are now a reality, and even though their functioning is still controlled by humans, automated robotic bees could be a possibility in the near future. For this to happen, preventing the collision between them is an important aspect to be solved first.

There are some challenges in order to solve this problem accurately. It is important to take into account that this is a three-dimensional problem, hence, the chances of collision are innumerable. Consequently, the algorithms that will be implemented should be analyzed in detail, considering that there are high odds of inaccuracy.

## 2. PROBLEM

In a few words, explain the problem, the impact that has in society and why is important to solve the problem.

The main problem is to avoid collisions between near bees. Within it, we can find another set of problems while trying to fix the central one, such as locating the other bees position with accuracy and retracing the movement of the bees in order to make sure that they do not crash.

Solving this problem can have a great impact on the agriculture sector, and may lead to a technology revolution all around the environment. Likewise, it is probable that this improvement contributes to other problems involved in automated robots.

## 3. RELATED WORK

### 3.1 Collision detection

It is the detects when an object hits something or when two objects hit each other. In this specific case, the objective is to avoid the collision between two non-stationary objects. This problem can be solved either mathematically or graphically. For this problem, solving it mathematically is

more efficient as there will be a mathematical model that describes the position of the objects, in this case, the bees.

If we imagine bees as points, it will only be necessary state that the distance between two points should be a non-zero result, in order to avoid a collision, therefore, if we apply a mathematical formula to do so, the result will be:

$$\text{dist}(P_0, P_1) = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2}$$

Being P0 and P1 the bees and x,y,z the coordinates of each one respectively. And the result of the distance between both points has to be different than zero, as zero will mean that both points are on the same coordinate, which means they are colissioning.

### 3.2 Obstacle avoidance

In case of a possible collision between two bees, an algorithm for avoiding a collision is important, each bee should be able to dodge if a crash is imminent.

Velocity, distance and the size of each bee is important to make the algorithms as well as the mathematical calculation on how to not crash between them. Algorithms of aggregation and disaggregation can be implemented, in orden to change the position or the velocity of each bee. If the position is changed, the path that the bees were following will be changes, hence, the collision will be avoided, and if the velocity is changed, it will be easier to change its direction. It is important to understand that online calculation should be made by the bee to avoid all crashes.

### 3.3 Spatial Partition

It is a technique used to simplify the process of identifying movement and collisions in video games. This procedure consists on dividing a given space on quadrants in a way in which such subdivisions will not overlap with each other. This kind of approach of solving the problem is a great way of making the program faster when it has a great number of entities in it.

The purpose of subdividing the space is to avoid comparing the location of an entity with every other entity present in the environment and determining which entities are nearby. It simplifies the process by only comparing the position of the entities located in the same quadrant. As this is a very common way of approaching the problem, there are several

data structures that solve the problem of dividing the space in their own way, as some of them may result efficient to a specific kind of circumstances.

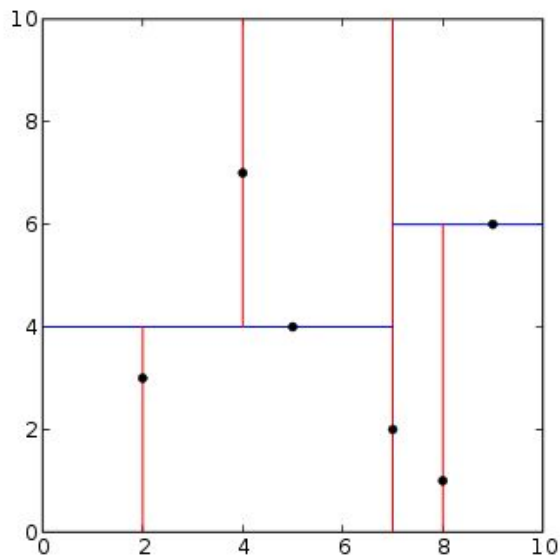
The K-D Trees are a type of data structure that make space partition in a way that range searches, and nearest neighbor searches are done in a more efficient way. A K-D Tree is a class of binary space partition tree, that works as it follows:

Each leaf of the tree acquires the function of a k-dimensional point. Each non-leaf node can be identified as a subdivision of the plane. Every point located at the left of the hyperplane will be located in the left subtree of the node. Likewise, the points located at the right of the hyperplane will be located at the right of the subtree.

### 3.4 Retracing Position

Identifying when two or more bees are nearby is only half of the work. Our main goal is to prevent them from colliding between each other. In order to do that is necessary to adjust the position of the bees and locate them in a another in which there is a fewer possibility of colliding with one another. To make the solution as simple as possible we will like to involve the K-D tree again.

As it was said before, the K-D Tree divide the plane in several subdivision depending on the root point. A more graphical way to see it is the one shown below.



The diagram is composed of several squares formed by the vertical and horizontal subdivisions created by the position of the bees.

Let's assume that the diagram shown above is a real case scenario of the position of some bees. We propose to establish a minimum distance between bees to avoid crashes. Let's suppose that this distance will be two units.

As you may observe the bees located at the position (7,2) and (8,1) are extremely close, to the point of violating the 2 unit range established by us before, which means that they could collide in a certain point.

As we can observe, this points are surrounded by squares. One between them, other between the right border and the (8,1) bee, and other at the left of the (7,2) bee. The rectangles at the sides represent the area in which they could move without colliding with other bee. So the perfect solution to the problem would be to move the position of one bee towards the square who has a greater area.

Correspondingly, the (7,2) bee will change its position in a way that the distance of every other bee located in the axis of any of the two subdivisions of the plane denoted by itself violates the minimum distance range.

### REFERENCES

Reference sourced using ACM reference format. Read ACM guidelines in <http://bit.ly/2pZnE5g>

As an example, consider this two references:

1. Thompson, J. Collision detection. 2019.
2. Xiangru, C, Fengxia, L, Zhaohan, L. and Yue, Y. Obstacle avoidance behavior of swarm robots based on aggregation and disaggregation method. School of Computer Science, Beijing Institute of Technology, China, 2017.
3. [https://en.wikipedia.org/wiki/Space\\_partitioning](https://en.wikipedia.org/wiki/Space_partitioning)
4. [https://en.wikipedia.org/wiki/K-d\\_tree](https://en.wikipedia.org/wiki/K-d_tree)