Robot bees as new pollination bees

Manuela Zapata Giraldo
Universidad EAFIT
Colombia
mzapatag1 @eafit.edu.co

Luis Bernardo Zuluaga
Universidad EAFIT
Colombia
lbzuluagag@eafit.edu.co

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

ABSTRACT

The decreasing number of pollinators is a major concern to the world. Significant crops, including cotton, potatoes and avocado are pollinator-dependent, which means that there is no way, other than pollination, to produce seeds to said crops. Additionally, this problem doesn't only affect the agriculture industry but the food industry and the clothing industry as well. What will fast food chain restaurants do without potatoes? What will clothing brands do without cotton?. If we don't address this problem soon, we may be facing a major crop deficiency that could drive many industries into bankruptcy.

1. INTRODUCTION

Pollination is one of the most important processes carried out by arthropods in nature, more specifically bees. It consists of transferring pollen from the male anthern of a flower to the female stigma, which results in the production of a seed. However, given that bees are the main participants in this process, and they are unfortunately becoming extinct because of numerous reasons, we must find a way to fill the void in pollination which directly affects the agriculture industry. There have been many potential solutions for this problem, including ecological farming, providing bee habitat, supporting beekeepers and robot bees.

2. PROBLEM

The decreasing bee population is directly affecting the seed production of significant crops in the agricultural industry. This means that there could be a serious shortage of certain foods and supplies in the market, which will not only directly affect the food and clothing industry, but us customers too. In order to keep crops from disappearing, we must use technology to come up with a solution that will not only keep pollination going but will also be environmentally friendly.

3. RELATED WORK

3.1 Quad trees

A quad tree is a data structure made up by nodes, each node has exactly four children that act as nodes themselves. Each node works like a bucket in which, when it surpasses a k amount of elements, it divides into four new nodes. Said process will be done every time a node reaches that k amount.

The purpose of this process is to keep the program from comparing each object to all the other objects to see if they will run into each other. This way, the program will only compare objects to the ones close by, saving memory and unnecessary processes.

3.2 Axis-aligned bounding boxes (AABB)

This method is one of the easiest and fastest ones to check if two objects are near each other. It consists of wrapping the entities in a box, if the box of an object overlaps an other box, then there is a collision between the two objects.

The issue with this method is that it is not very accurate since the box cannot rotate with the object. It also consumes a lot of resources if there are many objects, since each object has at least a position (x,y,z) and a box with eight vertices, and the system has to keep track and change each parameter since the objects are always moving.

3.3 Bounding Volume

Hierarchy

This method consists of a set of geometric objects in a tree-like structure. Each object is wrapped in bounding volumes that form the nodes of the tree. If the volumes of two nodes do not overlap, neither will the objects.

There are two ways for the tree to be built: top-down and bottom-up. In the top-down method we divide input set into two (or more) subsets. Bounding them in a defined volume and then keep on dividing them recursively until each subset consist of a leaf node (simplest form). On the other hand, in the bottom-up method, we begin with the input set as the leaves of

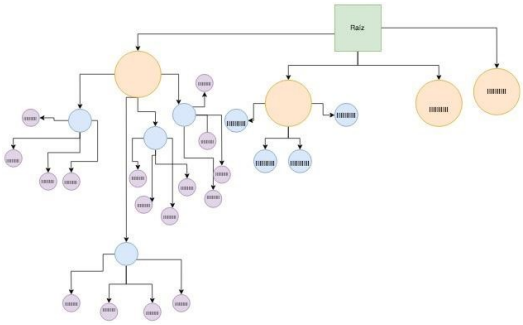
the tree and bind them together to form a new node. We proceed in the same way until everything has been binded to a single group of objects, that is considered the root of the tree.

3.4 Sweep and prune

This method is based on the arrangement of objects along every axis. From each object we take its minimum and maximum value in the x axis. Then we arrange them according to their minimum x value and select the objects that might overlap in the x axis. From the selected pairs we repeat the process but with the (y) axis, then the (z) axis. If a pair of objects is true to the three tests, they collided.

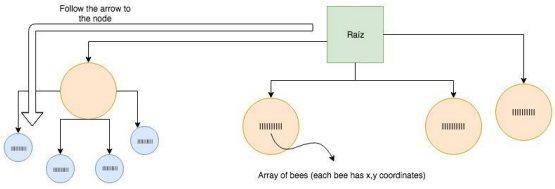
4. Data structure

The data structure we chose is a Data structure based on the Quad tree with some little additions in order to adapt the tree to this problem

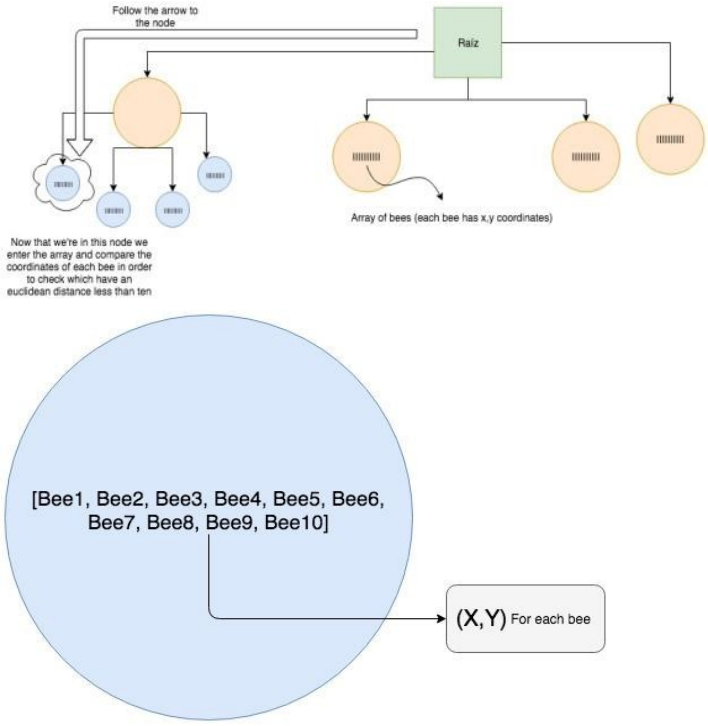


4.1 Operation of the data structure

The main operation of this System is comparing the distance between 2 objects due to the fact that comparing each and every one of the we designed a tree like data structure to minimize the operation ass seen in the first image, the first thing to do is to go to the each of the last nodes.



Once it gets to the node, the system starts to compare each and every one of the bees to each other to see if there is any collision between the bees, it returns if there is a collision, if so, which bees and the coordinates of said bees.



4.2 Criteria of the data structure

We chose this data structure based on the search time of the tree structure, since the complexity in the worst case scenario will be O(n), and we need to compare the distance between the bees, we based our data structure on the Quad tree with a slight difference since every node has an array with an amount of bees, the bees will only be compared with the bees within the array, this will bring the time needed to compare the bees that are close to collide, the fact that we will only be comparing a few bees with each other is the main reason to why we chose the data structure.

4.3 Complexity analysis

Method	Complexity
Add bees to list	O(n)
Sort bees into nodes	O(log4(n))
Detect collision	O(m(log4(n)))

Table 1: Table to report complexity analysis

4.4 Execution time

	1.000 bees	10.000 bees	100.000 bees
Load operation	0.036596838 (seconds)	0.061967095 (seconds)	0.086280895 (seconds)
Sort operation	0.014982947 (seconds)	0.021908294 (seconds)	0.01228422 (seconds)
Detect collision operation	0.036596838 (seconds)	0.061967095 (seconds)	0.086280895 (seconds)

Table 2: Execution time of the operations of the data structure for each data set.

4.5 Memory used

	1.000 bees	10.000 bees	100.000 bees
Load operation	3665592 (bytes)	3064736 (bytes)	2446656 (bytes)
Sort operation	166400 (bytes)	181080 (bytes)	183496 (bytes)
Detect collision operation	0 (bytes)	0 (bytes)	0 (bytes)

Table 3: Memory used for each operation of the data structure and for each data set data sets.

5. Quad Tree

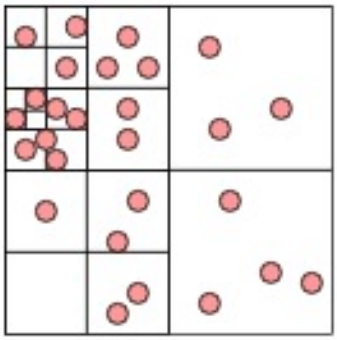


Figure 1: Quad Tree of bees. A bee is a class that contains x, y coordinates.

5.1 Operations of a Quad Tree

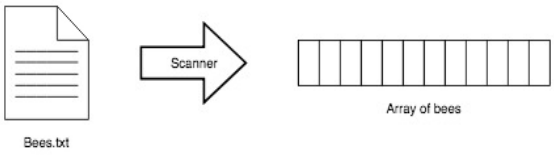


Figure 2: Load operation of a Quad Tree

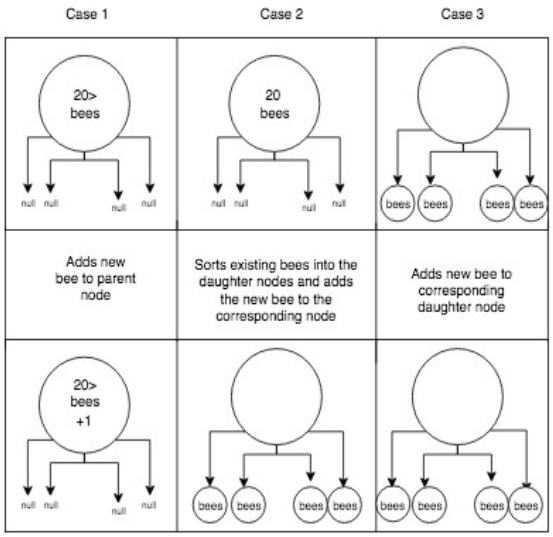


Figure 3: Sort operation of a Quad Tree

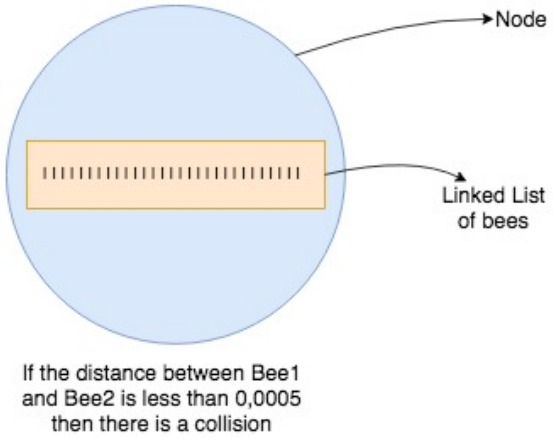


Figure 4: Detect collision operation of a Quad Tree

Analysis of the results

6. CONCLUSIONS

The decreasing pollinator population is not something to not be worried about. We must act now and take advantage of all the technological resources within our reach. With data structures and algorithms we can design a way to keep track of our alternative bees that will pollinate flowers for us and keep the food chain going.

Quad Trees serve as an exceptionally substantial solution to compare data of multiple objects from the same class in an efficient way, in our case, bees and their coordinates. Using Quad Trees one does not have to compare every bee to each other, only to the ones in the same node, or quadrant. Whereas if one was to use arrays, seeking the same functionality as us in this project, they would have to compare every object one by one which, with big data, would consume many resources and take a lot of time. That solution would not be efficient since processing speed is essential to keep bees from crashing or getting lost. Cases like this are often seen with data structures different from Quad Trees.

In the future, we hope to be able to take this project further and prove with actual robot bees that Quad Trees are the way to go, and that the food chain can be saved with technology.

REFERENCES

1. Onegreenplanet.org Animals and nature. Retrieved February 23, 2018. https://www.onegreenplanet.org/animalsandnature/the-disappearing-bees-andwhat-you-can-do-to-help/
2. Sos-bees.org Solutions. Retrieved February 23, 2018. http://sos-bees.org/solutions/
3. Euclideanspace.com Bounding boxes. Retrieved February 23, 2018 http://www.euclideanspace.com/threed/animation/collisiondetect/index.htm
4. Wikipedia. sweep and prune. Retrieved February 23, 2018 https://en.wikipedia.org/wiki/Sweep_and_prune
5. Moodle2015-16.ua.es. Videojuegos2. Deteccion de colisiones retrieved February 23 2018 https://moodle201516.ua.es/moodle/pluginfile.php/12368/mod_resource/content/3/vii-07-colisiones.pdf
6. buildnewgames.com . Broad Phase Collision Detection Using Spatial Partitioning retrieved February 23, 2018 http://buildnewgames.com/broad-phasecollision-detection/