Robotic bees: Data structures and algorithms for collision detection and prevention.

Vincent Alejandro Arcila
Universidad EAFIT
Colombia
vaarcilal@eafit.edu.co

Isabel Piedrahita
Universidad EAFIT
Colombia
ipiedrahiv@eafit.edu.co

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

ABSTRACT

With the population of bees rapidly dropping in recent years, we are left with the task of finding ways in which to carry out tasks bees usually handle for us. One of such solutions might be the implementation of robotic bees, but with it arises the need to stop them from crashing into one another, resulting in a problem of collision detection in rigid yet constantly moving objects. **(To be completed in following deliverables)**

1. INTRODUCTION

In recent years the population of bees has been constantly decreasing . More than a billion of bees have dyed in Colombia alone in the span of the last four years[1], worldwide, bees are not doing any better. With the number of bees dwindling dramatically in the past 10 years it is time to consider the importance of bees in the ecosystem. Out of the 369,000 species of flowering plants, 90% depend solely on insect pollination[2], and taking into account that bees can visit anywhere between 50 and 1000 plants a day[3], it is undeniable that their role on this task is fundamental. Without bees not only would countless plants die, but the food chain would also be considerably damaged, since a sizable amount of primary producers would disappear. This urges us to find ways in which to cover up for the bees, which leads to solution such as the creation and deployment of robotic bees.

2. PROBLEM

A problem that will rise is how to get the robotic bees not to crash when they are pollinating. A small step to get with a solution is to find an efficient way to identify which bees are in risk to collide. In other words, alert bees when they are near to others.

3. RELATED WORKS

3.1 Better collisions and faster cloth for Pixar's Coco[4]

While making Pixar's Coco animators ran into a complication, it was nearly impossible to get the cloth of clothes to sit properly on the huge cast of esqueletal characters, whose bones pinched the fabric causing what producers referred to as “Skeleton Wedgies”. Because of this, their in-house cloth system, Fritz, had to implement a more efficient and easily systematized version of continuous collision detection.

In order to fix this, the Global Intersection Analysis, an algorithm that uses constant collision detection to identify when cloth has crossed over to the wrong side by analysing all the intersections between the digital mesh that composes the costumes. This was used to create a fast way to stop the cloth from going through and getting stuck between the small geometries that make up Coco’s skeletons.

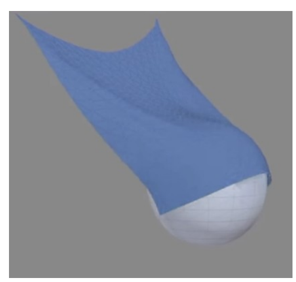

Fig. 1. Example of clothing

3.2: Search and tracking algorithms for swarms of robots: A survey[5]

This paper serves as an introduction to any scientist interested on understanding ways to keep an eye on groups of robots and to make them work more efficiently. It covers various algorithmic related problems in order to give the reader an idea of what to have in mind when designing systems that involves groups of robots.

One of the problems that needs to be solved in software is the problem of target search and tracking, understanding a target as an object with which the robots need to interact directly in order to succeed on their task. Understanding how to handle with software all the different environments that the robots might have to face is not as simple as one might think. The number of targets may be unknown, or may vary with time. Also, how to manage robots so that they do not act on the same target simultaneously, known as the task allocation problem is a thing to have in mind. Targets might not be static, which has to be considered.

Cooperation and coordination among the robots is an essential aspect that has to be taken into account when designing swarm robots systems. An ideal system would be the one on which all robots are exchanging information, combining measurements from multiple positions to accurately determine the targets position, and get the best synchronization possible.

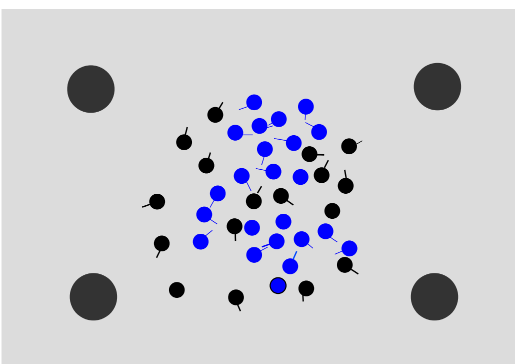

Fig. 2. Simulation screenshot. Experimental setup included a set of robots in search of four targets. When a robot detects an obstacle its color changes to blue. Once the robot has taken a new direction its color goes back to black.

3.3: Cooperative Path Planning of Robot Swarm Based on ACO[6]

Ant Colony Algorithm is the abstraction of the way that ants find food. It is based on the fact that they secrete pheromones when walking, so the more pheromone is on a path the more ants have used it, meaning that there is likely to be food. The problem arises when a group of ants finds a deadlock area, meaning they have to go back to try a different path.

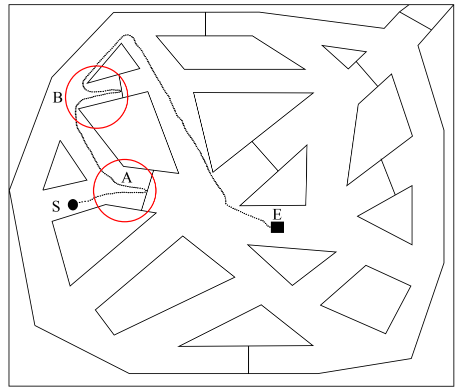

Fig. 3. The robot is in a deadlock area

Can this algorithm be improved? Clearly, if no pheromone were on the path that leads to a deadlock ants would find food faster. The solution given in this paper states a barely obvious but hard to implement way to fix this issue. If the individual that found a deadlock removed the pheromone left on that path other individuals would not waste time on that path.

So, being robots instead of ants, we get a great increase on performance when a robot communicates to its peers that there is no point going through certain areas, due to the lack of targets on certain points of an open field.

3.4: Real-Time Robot Motion Planning in Dynamic Environments[7]

Navigation is one of the main fields collision detection is used on. As soon as a robot can move on its own, ithas to decide where to, and this is where collision detection comes in very handy. Although there has already been extensive study of collision detection, further study is very important, because current algorithms understand the world around them via a group of queries, instead of using geometrical analysis.

Because of this, the process of collision detection is usually divided in two phases.

“Broad-Phase: Determines which geometric objects need to be checked for a certain query, ruling out distant objects based on bounding boxes or other partitionings of the space. Two data structures stand out throughout the broad-phase, namely, spatial hashing and spatial partitioning trees.

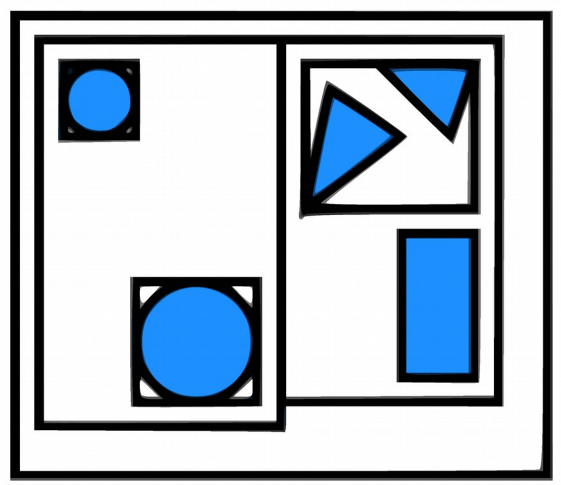

Fig 4. AABB tree

Narrow-Phase: For the objects that overlap in the broad-phase, a check is performed between the query and a single primitive object that represents an obstacle. Often objects correspond to rigid bodies."

In the narrow-phase some very useful algorithms stand out when we take into account that most of the structure is described by primitive objects such as spheres. These algorithms include checking a point q and radius r for collision, calculating the distance from q to the nearest point on an obstacle and Calculating the nearest point p on the obstacle to a query point q.

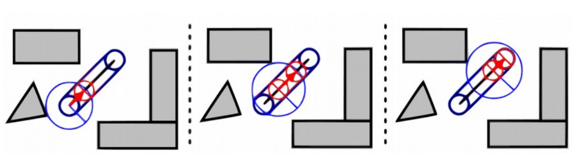

Fig. 5. Example swept circle obstacle check using only distance queries

REFERENCES

[1] RCN radio, Carlos Brand (2018) Más de mil millones de abejas han muerto en Colombia en los últimos tres años. (16, February 2018). Retrieved February 15, 2019 from https://www.rcnradio.com/recomendado-del-editor/mas-de-mil-millones-de-abejas-han-muerto-en-colombia-en-los-ultimos-tres

[2] BBC News, Rebecca Morelle (2016) Kew report makes new tally for number of world's plants. (10,May 2016). Retrieved February 15, 2019 from https://www.bbc.com/news/science-environment-36230858

[3] Nicola Bradbear BEES AND THEIR ROLE IN FOREST LIVELIHOODS: A guide to the services provided by bees and the sustainable harvesting, processing and marketing of their products. FAO, Rome 2009, 13-16.

[4] David Eberle. 2018. Better collisions and faster cloth for Pixar's Coco. In ACM SIGGRAPH 2018 Talks (SIGGRAPH '18). ACM, New York, NY, USA, Article 8, 2 pages. DOI: https://doi-org.ezproxy.eafit.edu.co/10.1145/3214745.3214801

[5] Madhubhashi Senanayake, Ilankaikone Senthooran, Jan Carlo Barca, Hoam Chung, Joarder Kamruzzaman, Manzur Murshed. 2015. Search and tracking algorithms for swarms of robots: A survey. *Robotics and Autonomous Systems 75, B (2016) 422-434.* DOI:https://doi.org/10.1016/j.robot.2015.08.010.

[6] Li Yong, Li Yu, Guo Yipei,,Cai Kejie. 2017. *Cooperative Path Planning of Robot Swarm Based on ACO*. Key Laboratory of Industrial Internet of Things and Intelligent Instrument, Ministry of Education. Industrial IoT Collaborative Innovation Center Chongqing Education Commission. Chongqing University of Posts and Telecommunications, Nanshan Scenic, .

[7] James Robert Bruce. 2006. Real-Time Motion Planning and Safe Navigation in Dynamic Multi-Robot Environments. Ph.D. thesis Dissertation. Carnegie Mellon University Pittsburgh, PA 15213.