Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems

Project Report

Vector based navigation of desert ants

Josua Graf, Noah Zarro

Zurich
December 6, 2018

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of COSS. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Josua Graf Noah Zarro

Contents

1 Abstract

2 Individual contributions

3 Introduction and Motivations

3.1 Motivation

Desert ants (*Cataglyphis*) has a difficult task to navigate since there natural habitat has not much landmarks helping to find there way. Different scientific papers (Müller and Wehner [1988]), (Collett et al. [1998]), (Knaden and Wehner [2005]) are postulating that desert ants use path integration as one component of there navigation. Because many technical applications are inspired by nature it is of high interest to have a clear idea how ants navigate in a unrecognizable landscape like desert. We could imagine a technical use in satellite steering or the navigation of evacuation robots in.

3.2 Base Paper

This project is based on the paper "Local and global vectors in desert ant navigation" (Collett et al. [1998]). Through simulating an ant which follows the concepts of local and global vector described by Collett et al. [1998] and comparison with the actual behavior we wanted to find out if the concept postulated is complete or needs adjustments. The goal was to implement the ant following the concepts as accurately as possible, then simulate the same test setup as it was used for the paper and compare the simulated plots with the plots from the experiments.

3.3 Hypothesis

description of base paper (who did it, roughly what was it all about)

4 Description of the Model

4.1 Behavior of the ant

how did Wehner think global and local vector worked (Collett et al. [1998])

4.2 Conducted experiments

description of the experiments Wehner conducted (explanation of channel, cylinder, etc.)

5 Implementation

with code examples

5.1 Main Concept

Our goal was to set up a simulation environment which was as general as possible, so we could easy switch between the test cases. The simulation is actually a huge loop, where in every iteration the ant does a step on a grid. It can just move horizontally, vertically and diagonal. This of course limits our ant, but if we simulate it with a big enough resolution it does not really matter. The ant can not reach every point on the map, there are several restrictions. First there is the channel, which can only be entered at specified channel exit or enter points. On the other hand the ant can not leave the map, so we constructed a virtual wall around the whole map, which it can not pass. Finally there is the nest, which we placed in the middle of the map.

In every iteration, the desired direction of the ant gets calculated according to the global and local vector in the `desired_direction` function. Then we calculate all possible directions in the `possible_direction` function. At the end the function `actual_step` lets the ant walk one step and the global vector gets adjusted. If the ant did a certain amount of steps the simulation stops and the path gets plotted.

Our whole simulation environment is highly configurable, in the `conifg.py` file one can import any test file, in which all things like the map configuration, the start position of the ant etc. can be specified.

5.2 Global Vector

The global vector starts with an initial value, which can be edited in the test file. In every iteration of the main loop the step conducted by the ant gets subtracted form the global vector. Additionally to model the imperfectness of the ants 0.1 mg brain we introduced two random factors. One gives credit to the fact, that a normal ant never walks in a straight line but rather zig-zags its way home. For this reason, before a step is executed by the ant, the step gets randomly rotated, but with a gaussian distribution around the straight forward direction. The second randomization mirrors the fact that an ant never can memorize exactly how many millimeter it walked north or east. Thus the global vector gets slightly rotated randomly after every step, which means the ant does not exactly remember which way it took to get to its current position.

5.3 Local Vector

The local vector gets calculated every iteration and depends on the close surroundings of the ant. The influence and the position of these surroundings can be configured in the test files. The channel exit for example pushes the ant to the south. All these objects have a circular area of influence and the influence lowers quadratically with the ants distance to the object. All influences of all objects get summed up to the local vector and are merged with the global vector to the desired direction with the following formula:

$$desired\_direction = local\_vector * local\_weight + global\_vector * (1 - local\_weight)$$

6 Simulation Results and Discussion

6.1 Simulation Results

what did we get, is it the same

6.2 Discussion

what was different, probably the randomization was no as in real life

7 Summary and Outlook

7.1 Summary

it worked pretty well, Wehner did a good job, his experiments corresponds to our simulation, which was based on his theory. We do not know if its biologically correct

7.2 Outlook

we probably will not continue our research about desert ants, maybe Wehner will

8 References

Martin Müller and Rüdiger Wehner. Path integration in desert ants, *Catglyphis fortis*. *Proc. Natl. Acad. Sci. USA*, 85, July 1988.

M. Collett, T. S. Collett, S. Bisch, and R. Wehner. Local and global vectors in desert ant navigation. *Nature Magazin*, 394, July 1998.

Markus Knaden and Rüdinger Wehner. Ant navigation: resetting the path integrator. *The Journal of Experimental Biology*, 209, 26-31, November 2005.

9 Latex-Stuff

Beispielverweisung:

Hier wird auf dieses Bild verwiesen (fig. 1)

Python Code:

Listing 1: Python code example

```
import numpy as np

# show numbers 0-9
print ('Numbers from 0-9')
for i in range(10):
        print(i)
```

Figure 1: Beispielbild, in Ordner images

Beispielliste:

- erster Eintrag
- zweiter Eintrag