# TERMES: Termite tracking in collaboration with Harvard University

Nikolaj Aaes, Niklas Schalck Johansson, and Hildur Uffe Flemberg
{niaa, nsjo, hufl}@itu.dk

IT University of Copenhagen
Rued Langgaards Vej 7
DK-2300 Copenhagen S

**Abstract.** 10 linjer
problem stilling
evaluation
resultat

• State the problem - There is no existing software that allows tracking of termites using the HP plotter provided by harvard as well as being able to extract relevant statistics.
• Say why it's an interesting problem - It will enable biologists to make better analyses based on more empricial data
• Say what your solution achieves - A way to control the plotter that tracks an ant/termite in real time and collects data that can be extracted as statistics
• Say what follows from your solution - While this is just a prototype (in a lab enviroment) it is a step towards making new and better field equipment for biologists along wiht corresponding software.

**Keywords:** Computer Vision, Image Processing, Termite Tracking, Biology, Computer Science.

# Table of Contents

# 1 Introduction

This report is written at the IT University of Copenhagen (ITU) in the fall term of 2013 as a project supervised by Kasper Støy from ITU and with additional guidance from Kirstin Petersen from Harvard University. The project was developed from 1. september to 16. december 2013 and was worth 15 ETCS points. The report is addressed to people interested in tracking ants and/or termites using computer vision libraries.

The Self-Organizing System Research Lab at Harvard University has created a autonomous robot for tracking African termites in a lab environment. Working with and tracking these termites is cumbersome as they only thrive in environments that resemble their native environment and only when they are together with other termites. Therefore automatic tracking of specific termites is necessary, as tracking up until now has been done manually. Hardware has been created to handle this, however it lacks proper software support.

This project aims to develop software that is able to track both ants and termites in a natural environment using a HP xy-plotter provided by Harvard University. While it is already possible to communicate with the plotter using the program called "Termite"(ref), there is a need for software that integrates tracking, plotter communication, statistics and a GUI. The end product is to be used by biologists in the field and this software will enable them to track ants/termites with more precision and with better data being produced.

A team at Harvard University previously experimented with using the plotter to track a singel ant/termite on a white background with some success. While the details of this effort were unknown to us we decided to start this project from scracth.

Since tracking of ants/termites is very hard to test systematically we have chosen to evaluate it by describing in what instances we expect the tracking to work as intended and in what instances we expect it not to. SKRIV NOGET OM OUTCOME.

• A summary (a “map”) of how the paper is organized.
TODO

# 2 Project Proposal

The Self-Organizing System Research Lab on Harvard has created a autonomous robot for tracking African termites in a lab environment. Working with and tracking these termites is cumbersome as they only thrive in environments that resemble their native environment and only when they are together with other

termites. Therefore automatic tracking of specific termites is necessary, as tracking up until now has been done manually. Hardware has been created to handle this, however it lacks proper software support.

The purpose of this project is to develop software, such that the hardware can be used to track these termites and analyse the output, as well as providing basic user input to the robot tracker.

There are three parts in this project:

1) Tracking termites using a low resolution camera. In order to track a termite, it is necessary to be able to determine its position and move the camera to its new position using image analysis on the camera output.

2) Interact with the tracking device and update the camera's position with the result from the tracking software.

3) Design and develop a user interface that can be used by biologists to retrieve statistical data from the tracking.

## 3 Background

Projektet er erstatning for GSD hvad betyder det for os? Hvilket udstyr havde vi at arbejde med? Umiddlebart ville vi gerne have logic code i C++ og GUI i Java så vidt muligt. Hvilke fordele giver det os? hvorfor? Hvad for nogen myrer havde vi og var der nogen udfordringer her? (tænk: hvis jeg skal gøre det igen, hvad skal jeg så huske om myrer) Hvordan var det med at male myrerne?

### 3.1 Assumptions

TODO

### 3.2 Scope

Kun test i et lab environment test på myrer, regner med at det kan generaliseres til termitter (andre størrelser) Kun test med de farver maling vi har Kun test med denne plotter Kontrollerede lysforhold Tracker kun een myre ad gangen

### 3.3 Requirements

Maa jeg foreslaa at i laver 2 modes (man kan f.eks. vaelge mode i menuen af toppen af programmet). 1. Den ene er kalibrerings mode hvor man kan se det direkte feed oeverst til venstre, samt billedbehandlet billede oeverst til hoejre, og sliders nedenfor. 2. Nummer to er tracking mode hvor man kan see feedet

fra kameraet taet paa oeverst til venstre, feedet fra overview-kameraet oeverst til hoejre, og saa statistik/grafer nederst til venstre, og valgmuligheder mht til statistik og hvor filerne skal gemmes mv. nederst til hoejre? + start knap. 3. Hvis i faar tid kan i tilfoeje en bias-mode hvor man kan klikke paa et punkt i skaalen som plotteren skal forsoege at traekke myren imod (ved hjaelp af mad).

1. Position i skaalen over tid (marker dens rute). + indstilling for hvor lang tid tilbage ruten skal vises. 1. Gennemsnitshastighed/tid (siden start), +indstilling for running average. 2. Heatmap over hvor myren opholder sig mest i skaalen, (groft firkantet grid er fint til at starte med). 3. Heatmap over hvor myren holder pauser i skaalen, +indstilling for hvor lang en 'pause' er, hvor lav den gennemsnitshastighed skal vaere. 3. Tael andre myrer den moeder/tid. +Indstilling for afstand af et 'moede'. 4. Hvor lang tid der gaar imellem den moeder andre/tid (siden start). 5. Hvor meget tid den pauser naer andre myrer/tid. +Indstilling for afstand+pause laengde 6. Hvor meget af skaalen den har undersoegt (Additivt) over tid. +Indstilling for radius af hovedet, (hvor langt fra center af hovedet den kan undersoege af gangen) 7. Hvor meget nyt areal den daekker hvert interval over tid. +instilling for interval: f.eks. 30s/1min/2min 8. Mean free path. (maal hvor lang tid der gaar mellem pauser. Pauserne defineres af lav gennemsnitshastighed)

Mht gui'en er et mere objekt-orienteret sprog fint med mig. Saa laenge at selve data- og billed behandlingen foregaar i C/C++. Nedenfor har jeg forsoegt at beskrive projektet lidt mere specifikt. Hvis i stadig er interesseret i projektet synes jeg at vi skal tage en skype-snak om helt noejagtige instruktioner.

1. UI: Brugervenligt, visning af live video, samt statistikker for opsamlet data. Bruger input til manuel styring af plotteren, aendring af billed behandlingsparametre i tilfaelde af at udstyret skal bruges til andre insekter i fremtiden, samt input af punkter i arenaen plotteren skal forsoege at styre imod, eller undgaa.

2. Tracking: Formentlig den stoerste udfordring. Plotteren skal kunne tracke en termit maerket med roed og groen maling ud fra et lav resolutions-kamera monteret paa manipulatoren. (Hvis i finder et bedre alternativ til det nuvaerende kamera, kan vi sagtens koebe et nyt). Termitten er hvid og brun, og dens baggrund roed/brun jord, derudover vil den vaere omgivet af mange andre (umaerkede) termitter. De bevaeger sig op til 2cm/s og manipulatoren skal helst bevares indenfor 0.5cm radius af hovedet (uden at ramme den eller andre termitter). Jeg vil foreslaa at i starter med at tracke myrer (de er nemmest at finde), paa hvid baggrund, og derefter paa jord.

3. Styring af manipulator: Manipulatorens position skal aendres afhaengigt af outputtet fra tracking-softwaret. Hardwaret kan modtage serielle kommandoer om ny oensket position. Hvis i oensker adgang til softwaren paa microcontrolleren som styrer plotteren kan det ogsaa sagtens arrangeres. Det kan blive noedvendigt at indfoere soft-start hvis plotteren skal bevaeges langt for at undgaa pludselige ryk i maskinen der kan skraemme insekterne.

5. Data behandling: Dette vil involvere mange maader at behandle den indsamlede data paa. F.eks. beregning af hastighed/tid, mean-free-path/tid, his-

togram af position i arenaen/tid, maengde af interaktion/tid, mm. Jeg sender gerne en laengere liste hvis i er interesserede.

# 4 Tracking

Gå ud fra at dette afsnit ikke har noget video input fra et kamera. Integration med kameraet kommer senere.

Teori først - hvilken slags image manipulation har vi brugt? hvad var alternativerne? Så framework - vi bruger OpenCV, why? hvad giver det os? hvad er drawbacks? så implementation - Hvad har vi implementeret? Hvordan var performance? hvad var vores alternativer? har vi eksperiementeret med nogen af de andre muligheder?

Image Segmentation - hvad giver det os/hvorfor er det smart? Thresholding Dilating Eroding Background detection and why we can't use it Alternatives and why we don't use them

# 5 Integration with camera

## 5.1 Communication with the plotter

Hvilken plotter er det og hvilket udstyr sidder på den Hvordan er den sat til computeren hvad vil vi gerne have den til hvordan gør vi dette hvilke "services" ender vi med at udstille til resten af programmet (flyt til koordinat, current coordinate?) Hvordan er performance? kan videoen køre i real time? hvis nej hvad betyder det for os?

* 0x01 = send coordinate to plotter * 0x01 + 2 bytes for x + 2 bytes for y (up to 10 bits) up to 0x03FF (y probably only up to 01F0) * 0x02 in response = sucesss * 0xFE in response = failure

## 5.2 Moving the camera

Now that we know how to move the plotter, how do we move the camera when the ant moves? what about the volume of the movement sound? what about the speed of the movement?

# 6 Graphical user interface

What was the reqs for the GUI? How did we want it to look? What tasks do we expect the users to do (normal work flow)? Are we satisfied with the GUI (eval)?

## 6.1 Statistics

What statistics do we extract? How do the users do this? How are they produced? Can they be improved (future work)?

# 7 Process

Hvad var process planen? Hvordan kommunikerede vi med Harvard + vejleder? Hvordan gik det (eval)? Havde vi nogen problemer med det? (tidsforskel, kultur forskel) Hvilke værktøjer brugte vi? Virkede de efter hensigten? Hvilke kommunikations kanaler brugte vi? Virkede de efter hensigten? ville vi gerne have haft mere eller mindre kommunikation? Ville vi gøre noget anderledes hvis vi skulle gøre det igen/hvad har vi lært?

Vi har modtaget video først for at kunne starte før vi fik plotteren.

Halls has defined cultures as being either high-context or low-context, which states the degree to which their culture is expressed. As an example, a low-context culture will have a lot of implicit actions, where a high-context culture will act more explicitly. - Olson, J. & G. Surprises in Remote Software Development Teams. p. 58

J. & G. Olson, Surprises in Remote Software Development Teams from lectures - `http://www.itu.dk/people/nh/BAAAP/2011/Olson,%20Culture%20surprises%20in%20Remote%20Software%20Development%20Teams.pdf`

# 8 Reflection

TODO

# 9 Evaluation

vis hvornår det virker og hvornår det virker. Se at det virker når vi regnede med at det virkede.

Hvad tester vi? Hvordan tester vi det? Virkede det efter hensigten? Hvorfor/hvorfor ikke? Er der nok testing? Hvordan kan man lave mere testing? Er der andre måder vi kunne have testet på? (fordele og ulemper ved det)

# 10 Threats to validity

Høj lyd / hastighed kan skræmme myrerne til at flytte sig hurtigere/mere. TODO

## 10.1 Threats to internal validity

Internal Validity is concerned with confirming that the correlation between the treatment and the outcome is indeed casual, and not accidental, or caused by some third variable that has not been observed. For example we may discover that all programmers in C were faster than programmers in Java, but forget that all the programmers in Java took the experiment very late at night, when they were tired.

### 10.2 Threats to external validity

External validity discusses how far the results are generalizable, or in other words how representative the sample of subjects and the circumstances of the experiment were, to be able to draw general conclusion. Do you expect the same results to be confirmed in somewhat modified conditions?

## 11 Related Work

TODO

## 12 Future Work

More statistics Bias mode Support for multiple plotters Plotter control as a library Tests with real termites Developer terminal/log (print was is sent to the plotter and the answer + the hard tracking coordinate data)

## 13 Conclusion

TODO

## 14 Defition of terms

TODO

## References

[1] TestTitle - TestPublisher - 1999

## A First Appendix