

---

# NXT - Localisering door middel van lijnen en blobs

---

*Auteurs: Tom Peerdeman &  
René Aparicio Saez*

*Datum: December 7, 2013*

## 1 Materiaal

Om de experimenten uit dit rapport te kunnen uitvoeren zijn de volgende materialen gebruikt:

- PC/Laptop met Matlab
- Boek: Autonomous Mobile Robots 2th Edition - Roland Siegwart et al.
- NXT-Robot
- Logitech Webcam
- Gloeilamp
- Zwarte tape

## 2 Introduction

Een autonome mobiele robot moet weten waar in de wereld hij momenteel is. Als er een kaart bekend is kan aan de hand van de omgeving die de robot momenteel detecteert bepaald worden waar in de wereld hij is. Dit kan gedaan worden aan de hand van de detectie van lijnen en blobs in zijn huidige omgeving en deze te vergelijken met de bekende blobs die opgeslagen zijn in de kaart. Gecombineerd zullen de lijnen en blobs een goede indicatie kunnen geven over de locatie van de robot.

## 3 Fouten in meegeleverde code

De meegeleverde code bevatte enkele fouten. Zo was het omzetten van het opgenomen beeld naar een zwart wit beeld 'geinverteerd' (zwart was wit en andersom). Tevens was het bestand `imflipud.m` aangepast. Het bestand `imflipud.m` zou een functie moeten leveren om een plaatje verticaal te spiegelen om zijn middelpunt. Het gegeven bestand bevatte echter een functie die helemaal niks deed met het plaatje.

## 4 Maken van een dataset

### 4.1 Foto's maken

Om experimenten uit te kunnen voeren is een dataset nodig. Deze dataset moet bestaan uit fotos die een afgeplakt parcours volledig omschrijven. Er moet hiervoor een parcours gemaakt worden met behulp van tape. Vervolgens moeten er voldoende fotos gemaakt worden zodat het gehele parcours gezien wordt. Voor de experimenten is het ook nodig dat het parcours gesloten is. Na het maken van de eerste dataset bleek er echter geen rekening gehouden te zijn met het maken van een tweede testset. Er moet opnieuw een dataset gemaakt worden op een nieuw parcours (het oude parcours is al weggegooid). Dit nieuwe parcours wordt gebruikt voor zowel de train-set als de test-set. In figuur 1 is te zien hoe er te werk is gegaan bij het maken van een dataset. In figuur 2 is te zien hoe de uiteindelijke plattegrond van de dataset er uit is komen te zien.

### 4.2 Parameters

De parameters moeten worden aangepast aan de nieuwe dataset. De camera is lichtelijk verschoven en moest verstevigd worden waardoor de positie van de lamp en het middelpunt van de camera is verplaatst ten op zichte van de vorige proeven. Het middelpunt heeft de coördinaten  $x = 545$ ,  $y = 402$ . De radius blijft hetzelfde. Er zijn nog enkele uitwendige niet gebruikte sensoren van de robot afgehaald waardoor de waarden voor  $R_{min}$  kan worden verlaagd. De waarde wordt verlaagd naar  $R_{min} = 115$ . De waarde voor  $R_{max}$  wordt aangepast vanwege de verplaatste locatie van de bolling ten opzichte van de camera. De waarde van  $R_{max}$  wordt:  $R_{max} = 190$ . De waarde van  $\alpha$  wordt geschat door te kijken naar de manier waarop het gemeten beeld wordt 'rechtgetrokken'. De waarde verandert naar  $\alpha = 140$ . Tot slot wordt de threshold voor de zwart-wit beelden aangepast. Dit wordt gedaan omdat de nieuwe opnamen geen verhoogd contrast hebben (dit was bij de vorige dataset wel het geval). De nieuwe waarde wordt  $threshold = 130$ .

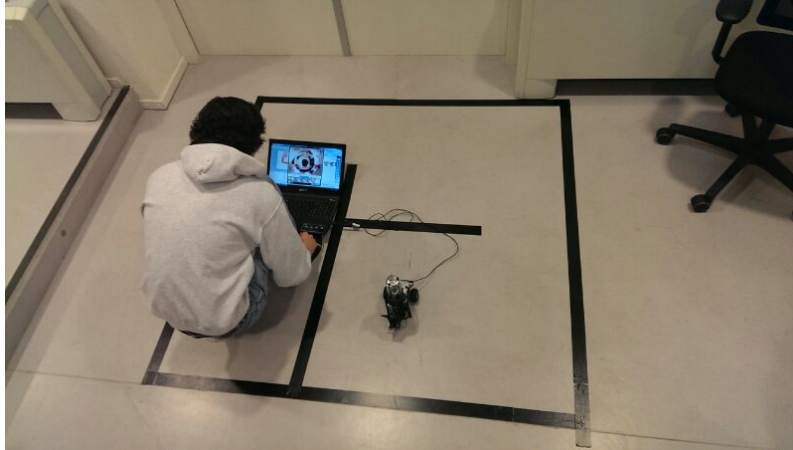


Figure 1: Het maken van een dataset (niet gebruikte dataset)

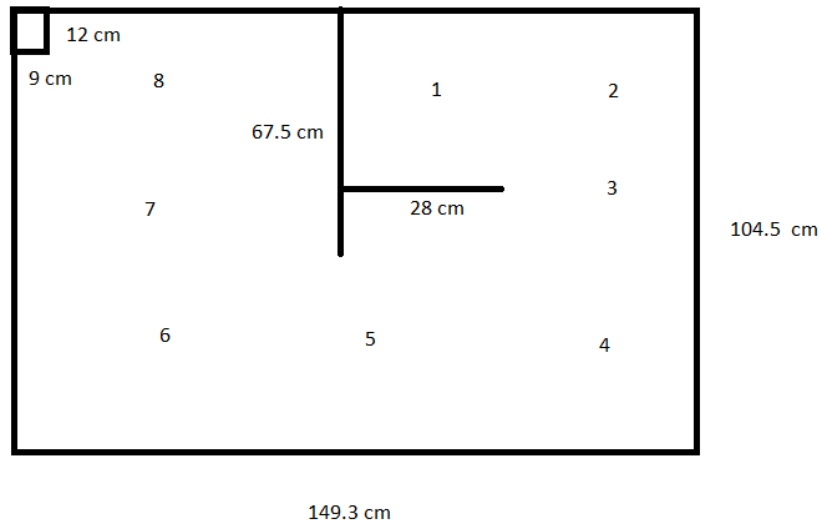


Figure 2: Plattegrond en locaties van de gemaakte fotos

## 5 Line fingerprints

Met behulp van de code uit het vorige labexperiment kunnen lijnen worden gedetecteerd. Met behulp van deze code kunnen deze gevonden lijnsegmenten worden opgebouwd. Dit wordt gedaan met een meegeleverd algoritme. De gecombineerde lijnsegmenten in een foto vormen een fingerprint voor de lijnen op die locatie. In figuur 3 is zo een fingerprint te zien. Door eerst te leren hoe de lijnsegmenten er uit zien. Als de robot vervolgens de testset bekijkt, worden de fingerprints die opgebouwd worden voor de test-set vergeleken met de fingerprints van de train-set. Hieruit kan de waarschijnlijkheid bepaald worden voor de huidige locatie van de robot. Dit kan worden weergegeven in een confusion matrix, zoals te zien is in figuur ??

## 6 Blob fingerprints

Met behulp van meegeleverde code wordt gezocht naar blobs. Er moet echter aangegeven worden naar welke kleurwaarden er gekeken moet worden. Aangezien er geen kleuren in het parcours zijn verwerkt moet dit worden aangepast. Door dit aan te passen zodat er alleen een onderscheidt

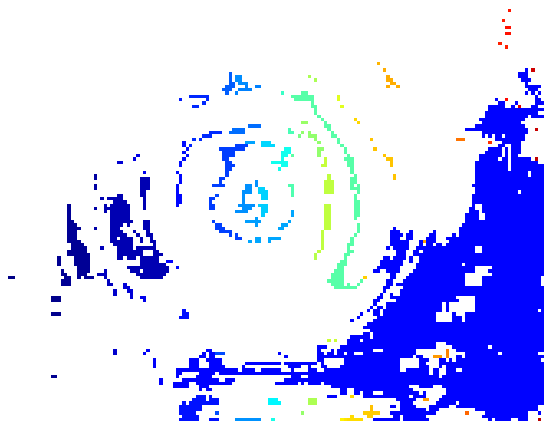


Figure 3: Fingerprint voor lijnsegmenten

wordt gemaakt tussen zwart en wit wordt dit overkomen. De combinatie van een de blobs in een beeld vormt een fingerprint, een voorbeeld is te zien in figuur 4. Net als bij de lijnsegmenten moet worden gezocht naar een match tussen de bekende fingerprints en de huidige gemeten set blobs. Er kan weer een waarschijnlijkheid worden bepaald voor de locatie van de robot en deze kan worden weergegeven met behulp van een confusion matrix.

## 7 Gecombineerde fingerprints

## 8 Verbeteringen

Momenteel worden blobs gedetecteerd die niet per se altijd in de wereld aanwezig zijn. Denk aan mensen die in de train-set als blob zijn gedetecteerd, die hoeven niet de hele tijd op dezelfde locatie te blijven staan. Daarnaast zijn de lijnen allemaal zwart. Als elke lijn een eigen kleur had gekregen had de blobdetectie waarschijnlijk beter gewerkt.

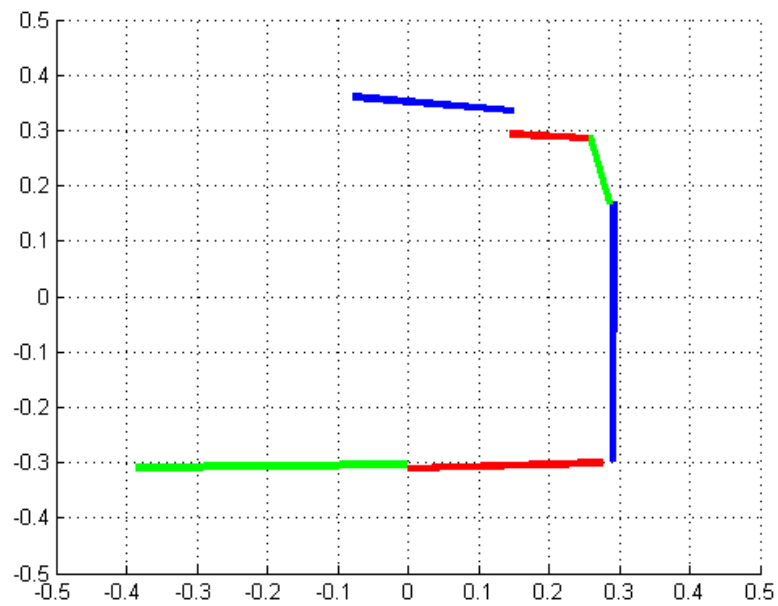


Figure 4: Fingerprint voor blobs