

Semantische positietracking van een mobiele robot in ziekenhuisgangen d.m.v. visie

Eindpresentatie

Olivier Van den Eede

KU Leuven - De Nayer

25 Juni 2019

0 Inhoud

1 Probleemstelling

2 Uitwerking

3 Resultaten

4 Besluit

1 Inhoud

1 Probleemstelling

2 Uitwerking

3 Resultaten

4 Besluit

1 Probleemstelling

Ziekenhuizen kampen met volgende problemen:

- ▶ Personele tekorten
- ▶ Hoge werkdruk voor zorgpersoneel

Onder andere veroorzaakt doordat personeel verantwoordelijk is voor textiellogistiek en goederenstroom.

1 Mogelijke oplossing

- ▶ Automatisatie
 - Moeilijk door infrastructuur en bestaand logistiek materiaal.
- ▶ **Autonomo Geleid Voertuig (AGV)**
 - Zelfstandige navigatie doorheen ziekenhuisgangen
 - Uitgerust met sensoren en een RGB camera
 - Semantische kaart voor navigatie

1 Doel masterproef

- ▶ Onderzoeken aanwezige visuele objecten/features in ziekenhuisgangen
 - Gangen zijn zeer monotoon
 - Weinig visuele kenmerken aanwezig
- ▶ Zoeken naar geschikte beeldverwerkingstechnieken voor detectie van deze features
- ▶ Tracken van de robotpositie op basis van:
 - De gedetecteerde visuele features uit 1 RGB camera
 - De informatie op een semantische kaart

1 Doel masterproef



Figuur: Voorbeeld van een gang uit de dataset.

2 Inhoud

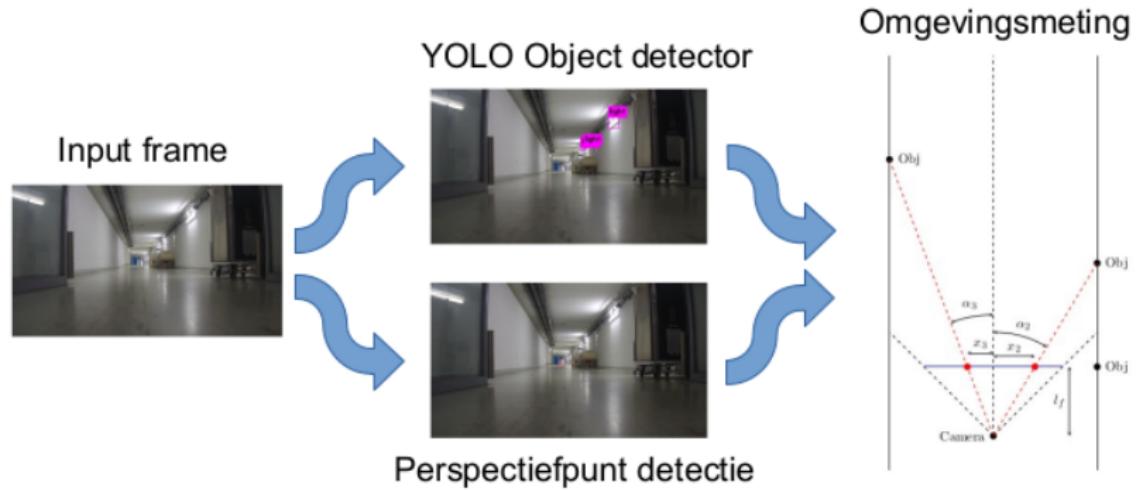
1 Probleemstelling

2 Uitwerking

3 Resultaten

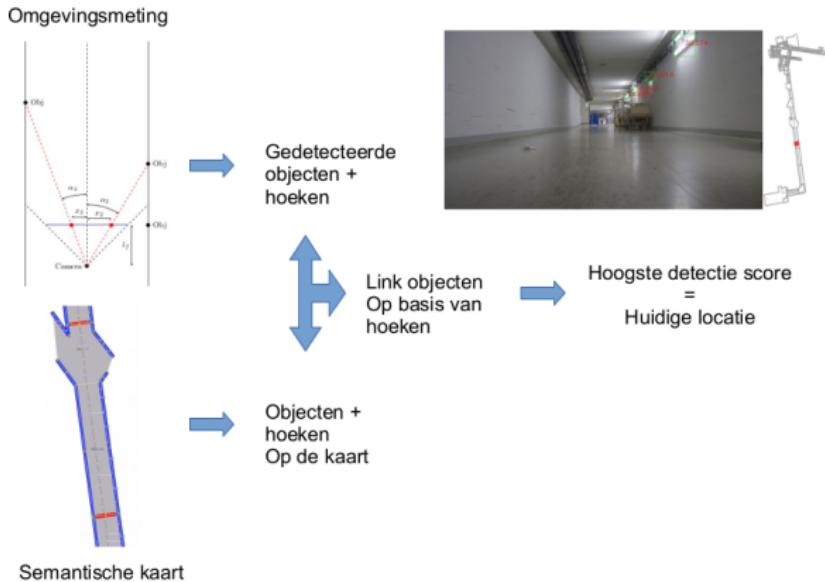
4 Besluit

2 Uitwerking



Figuur: Overzicht van het programma(1).

2 Uitwerking



Figuur: Overzicht van het programma(2).

2 Uitwerking

- ▶ Object detectie
- ▶ Perspectiefpunt detectie
- ▶ Omgevingsmeting
- ▶ Semantische kaart
- ▶ Robot positie tracking

2 Object detectie

- ▶ YOLOv2
 - Veel gebruikt
 - Single shot detectie + classificatie CNN
 - Kan real-time werken op GPU
- ▶ Datasets
- ▶ Annotatie beeldmateriaal
- ▶ Training YOLOv2

2 Datasets

Kenmerken datasets:

- ▶ 2 video's opgenomen met RGB camera
- ▶ Beeldmateriaal van opeenvolgende gangen
- ▶ Eerste deel is validatieset, 2de deel is trainingsset
- ▶ 1920x1080 pixels @30Hz
- ▶ Herschaald naar 1280x720 pixels @15Hz

Visuele kenmerken in dataset:

- ▶ Lampen
- ▶ Rookdetectoren
- ▶ (Deurklinken)
- ▶ Nooduitgang bordjes

2 Annotatie beeldmateriaal

- ▶ Bounding-box met class label
- ▶ Nodig voor training van CNN
- ▶ Manuele annotatie d.m.v. CVAT
- ▶ Conversie CVAT XML-formaat naar YOLO-formaat

Tabel: Annotaties in training en validatie set

	Aantal frames	Rookmelder	Deurklink	Pictogram	TL-lamp	Totaal
Trainingsset	899	1016	147	340	5260	6763
Validatieset	711	1130	180	408	992	2710

2 Training YOLOv2

Aanpassen YOLOv2 hyperparameters:

- ▶ 4 detectieklassen
- ▶ Batch size van 64
- ▶ Learning rate van 0.0001
- ▶ Maximum van 45000 batches

2 Uitwerking

- ▶ Object detectie
- ▶ **Perspectiefpunt detectie**
- ▶ Omgevingsmeting
- ▶ Semantische kaart
- ▶ Robot positie tracking

2 Perspectiefpunt detectie

Perspectiefpunt nodig in volgende stap van de pipeline. Het perspectiefpunt kan gedetecteerd worden d.m.v. 3 verschillende methoden:

- ▶ Hoogste vloerpixel segmentatie
- ▶ Vloerlijn kruising
- ▶ Perspectieflijn kruising

2 Hoogste vloerpixel segmentatie

- ▶ ResNet segmentatiennetwerk
 - Elke pixel krijgt label
 - Enkel vloerpixels worden gebruikt
- ▶ In theorie is hoogste vloerpixel perspectiefpunt
 - Fouten in segmentatieoutput beïnvloeden hoogste pixel

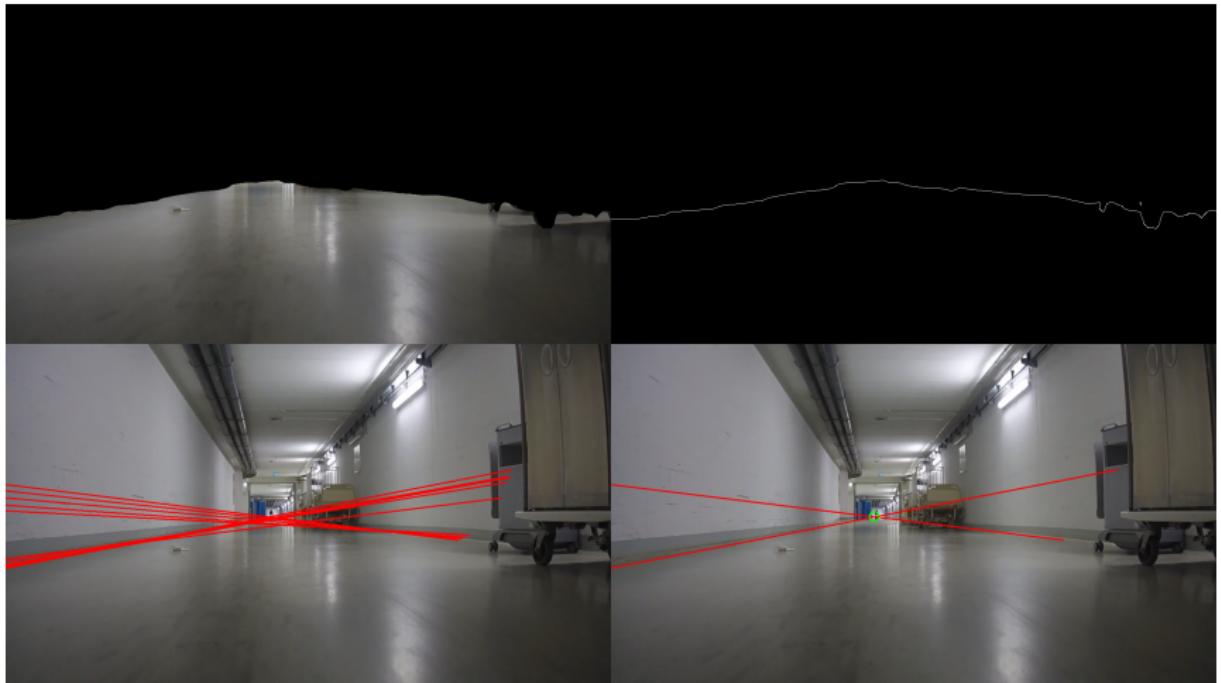
2 Hoogste vloerpixel segmentatie



2 Vloerlijn kruising

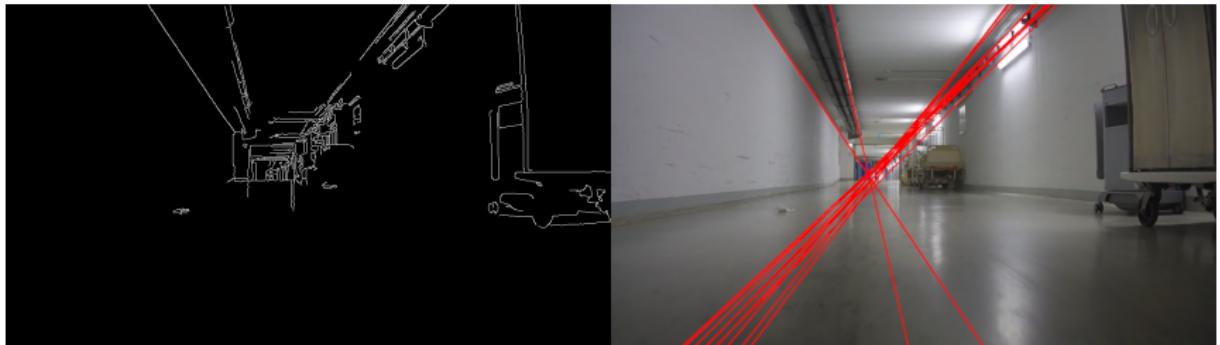
- ▶ ResNet segmentatienetwerk
- ▶ Edge detectie op masker van vloerpixels
- ▶ Zoeken van perspectieflijnen d.m.v. de Hough-transformatie op masker
- ▶ Kruising van perspectieflijnen is perspectiefpunt
 - Dezelfde lijnen filteren op basis van de richtingscoëfficiënt.
 - Ideaal als 1 lijn langs elke kant van de vloer
- ▶ Probleem:
 - Vloer volgt niet de perspectieflijnen door obstructies

2 Vloerlijn kruising



2 Perspectieflijn kruising

- ▶ Geen vloersegmentatie
- ▶ Canny-edgedetectie op hele grijswaardeafbeelding
- ▶ Variabele threshold op basis van de grijswaardemediaan.
- ▶ Hough-transformatie om perspectieflijnen te vinden
- ▶ Filtering van horizontale en verticale lijnen
- ▶ Kruispunt van rechten is perspectiefpunt



2 Uitwerking

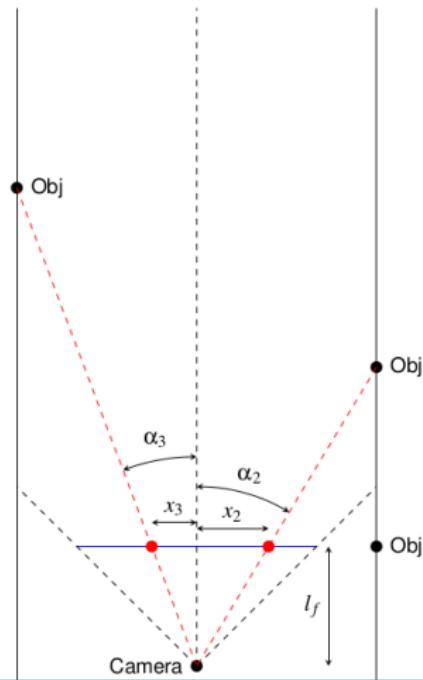
- ▶ Object detectie
- ▶ Perspectiefpunt detectie
- ▶ **Omgevingsmeting**
- ▶ Semantische kaart
- ▶ Robot positie tracking

2 Omgevingsmeting

- ▶ Elk object heeft unieke locatie in ruimte en beeld
- ▶ Transformatie van afstand tot perspectiefpunt naar hoeken
- ▶ Bepaal hoek α_i voor elk object

Verband tussen x_i en α_i

$$\alpha_i = \tan^{-1}\left(\frac{x_i}{l_f}\right)$$

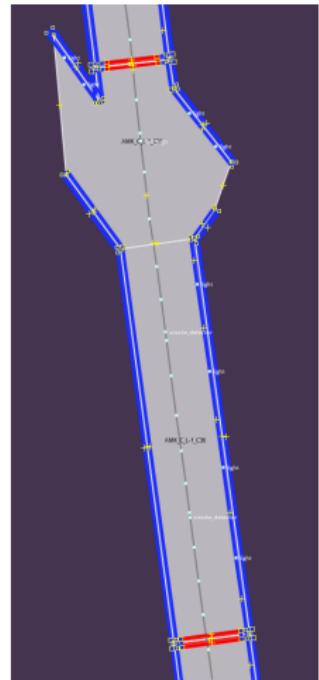


2 Uitwerking

- ▶ Object detectie
- ▶ Perspectiefpunt detectie
- ▶ Omgevingsmeting
- ▶ **Semantische kaart**
- ▶ Robot positie tracking

2 Semantische kaart

- ▶ Representatie van alle visuele objecten binnen de ruimtes
- ▶ OpenStreetMap (OSM) formaat
- ▶ Node
 - Basis OSM object
 - Parameters in key-value formaat
 - Object en locatie nodes
- ▶ Way
 - Basis OSM object
 - Gesorteerde lijst van opeenvolgende (locatie) nodes
 - Beschrijft een route
- ▶ Inlezen via eigen parser



2 Uitwerking

- ▶ Object detectie
- ▶ Perspectiefpunt detectie
- ▶ Omgevingsmeting
- ▶ Semantische kaart
- ▶ **Robot positie tracking**

2 Robot positie tracking

- ▶ Corrigeer perspectiefpunt naar midden van afbeelding om rotatie van robot op te vangen
- ▶ Robot vertrekt op gekende locatie en volgt de gedefinieerde route vanop de kaart
- ▶ Robot kan niet springen en enkel vooruit rijden
 - Enkel de actuele en volgende locatie-nodes worden vergeleken
- ▶ Vergelijken omgevingsmeting en hoeken op de kaart
 - Bereken score hoeveel het huidige frame lijkt op de 2 theoretische locatie-nodes
 - Koppel gedetecteerde objecten aan objecten uit de kaart d.m.v. de hoeken
- ▶ Verschillende wegingsfactoren

2 Robot positie tracking

Berekening van de score

$$score = \sum_{i=1}^n [a - |\alpha_{mi} - \alpha_{di}|] \cdot f_w(\alpha_{mi}) + \sum_{i=1}^m [b \cdot f_w(\alpha_{mi})]$$

Met n het aantal gelinkte objecten, en m het aantal niet gelinkte objecten. En a een positieve belijgingsfactor en b een afstraffactor.

Berekening van de wegingsfactor

$$f_w(\alpha) = \left\{ \begin{array}{ll} 1 & \text{for } 30 < |\alpha| \\ \frac{7\alpha+10}{220} & \text{for } 0 \leq |\alpha| \leq 30 \end{array} \right\}$$

3 Inhoud

① Probleemstelling

② Uitwerking

③ Resultaten

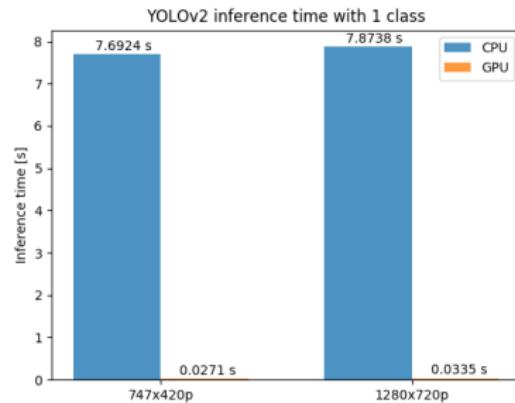
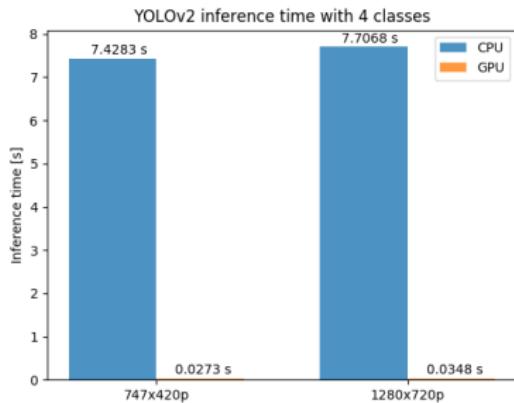
④ Besluit

3 Resultaten

- ▶ **Object detectie**
- ▶ Perspectiefpunt detectie
- ▶ Lokalisatie

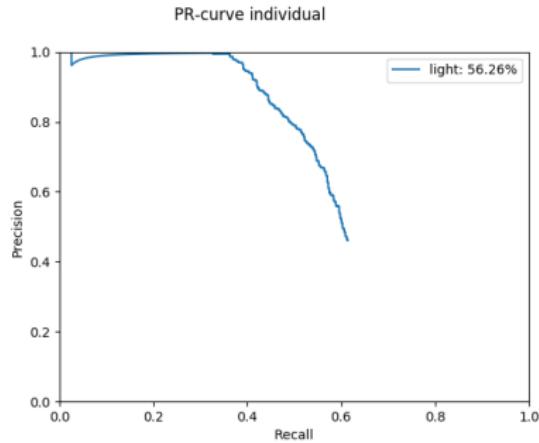
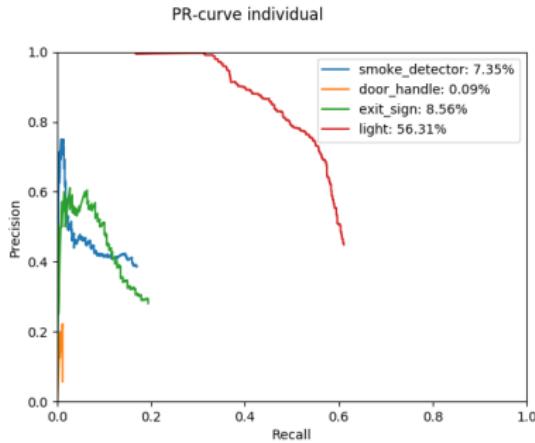
3 Object detectie

- ▶ Inferentietijd CPU vs GPU
- ▶ Invloed resolutie inputafbeeldingen
- ▶ Invloed aantal detectieklassen



3 Object detectie

- ▶ Nauwkeurigheid op basis van validatieset
- ▶ Precision-recall-curve
- ▶ Mean Average Precision (mAP)

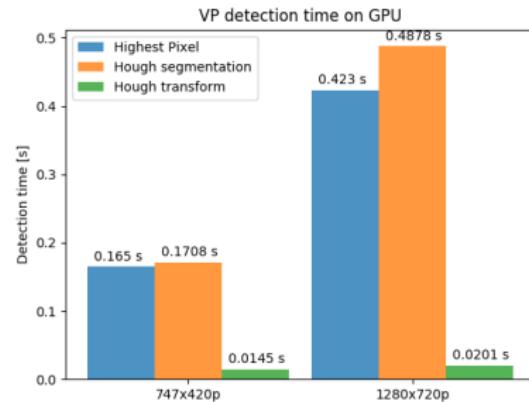
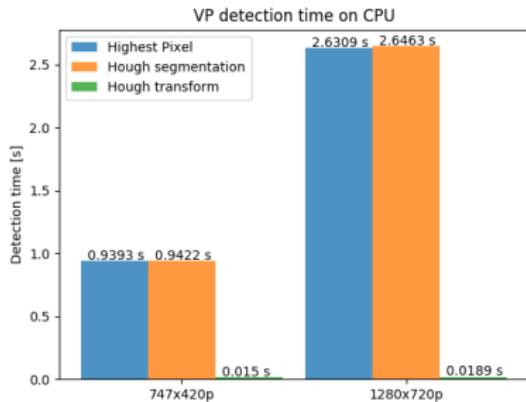


3 Resultaten

- ▶ Object detectie
- ▶ **Perspectiefpunt detectie**
- ▶ Lokalisatie

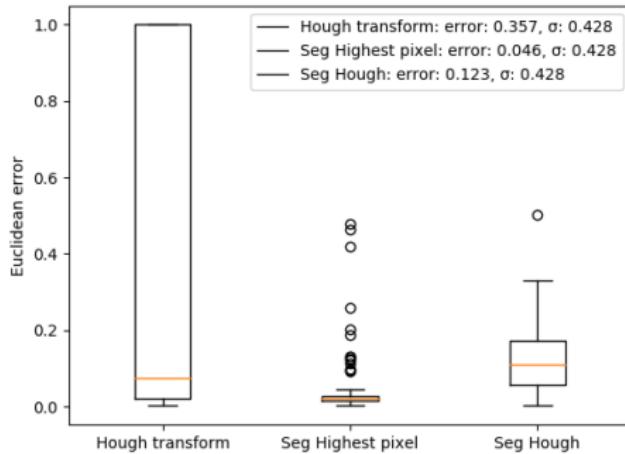
3 Perspectiefpunt detectie

- ▶ Detectietijd CPU vs GPU
- ▶ Vergelijking 3 detectietechnieken
- ▶ Invloed resolutie inputafbeeldingen



3 Perspectiefpunt detectie

- ▶ Nauwkeurigheid van 3 technieken
- ▶ Vergelijking met manuele annotatie van perspectiefpunt voor 100 frames

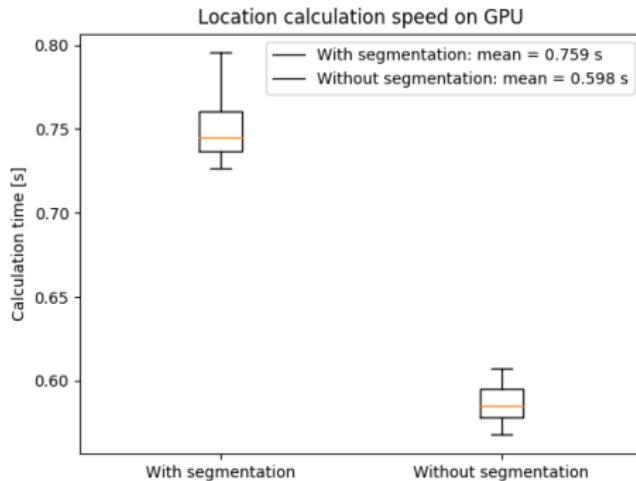


3 Resultaten

- ▶ Object detectie
- ▶ Perspectiefpunt detectie
- ▶ **Lokalisatie**

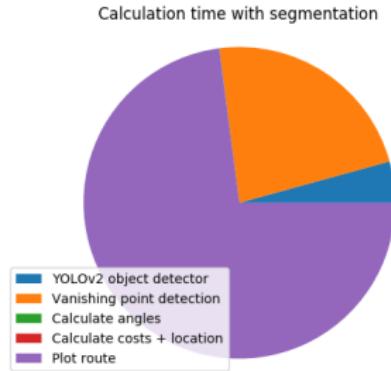
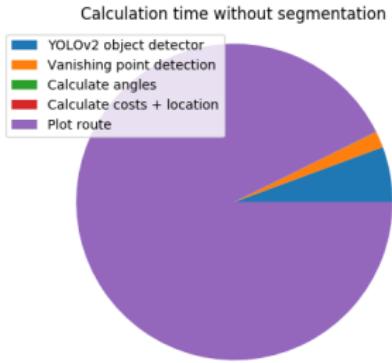
3 Lokalisatie

- ▶ Iteratietijd per frame
- ▶ Verschil tussen perspectiefpuntdetectie met of zonder segmentatie
- ▶ Gemiddelde rekentijd hoog t.o.v. vorige resultaten



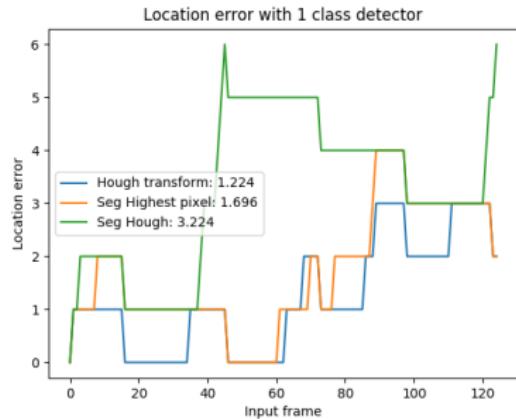
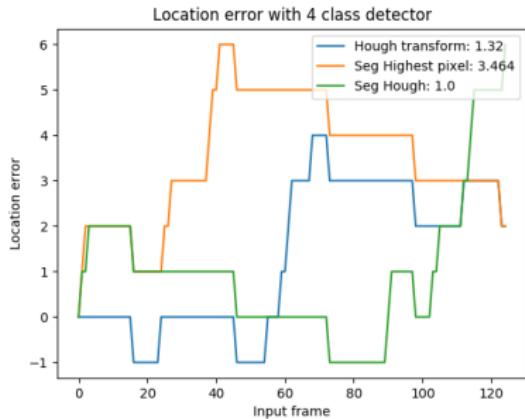
3 Lokalisatie

- ▶ Profilering van de implementatie
- ▶ Visualisatie van kaart neemt meeste tijd in beslag



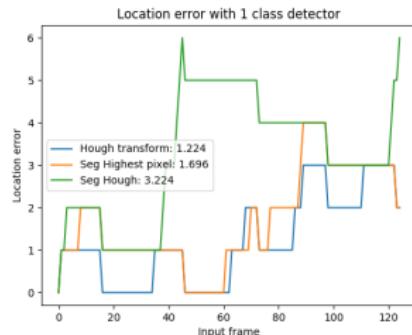
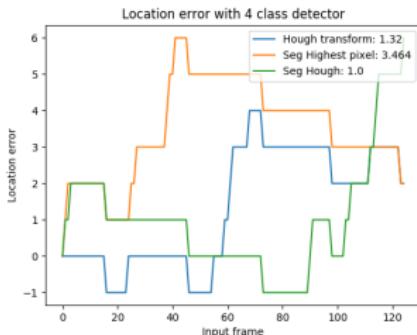
3 Lokalisatie

- ▶ Locatie manueel geannoteerd voor 125 frames
- ▶ Afwijking van detector t.o.v. actuele locatie



3 Lokalisatie

- ▶ Algoritme wil te snel gaan
- ▶ Keuze objectdetector beïnvloed nauwkeurigheid
- ▶ Keuze perspectiefpuntdetector beïnvloed nauwkeurigheid
- ▶ Resultaten niet accuraat omdat semantische kaart opgesteld is op basis van schattingen uit het beeldmateriaal
- ▶ Geen exacte gegevens beschikbaar



4 Inhoud

1 Probleemstelling

2 Uitwerking

3 Resultaten

4 Besluit

4 Besluit



4 Besluit

- ▶ Positie traking van een mobiele robot op basis van RGB camera en semantische kaart
- ▶ Vergelijking van verschillende detectietechnieken
- ▶ Opstellen en implementatie van pipeline en algoritme
- ▶ Resultaten van lokalisatie minder nauwkeurig door gebrek aan data
- ▶ YOLO object detector is mits genoeg training zeker bruikbaar in deze toepassing
- ▶ Door finetuning en combinatie van perspectiefpunctdetectie methoden kunnen deze ook toegepast worden
- ▶ Toekomstig werk
 - Andere bibliotheek gebruiken voor visualisatie van de kaart om geheel te versnellen
 - Op basis van echte gegevens de nauwkeurigheid opnieuw bepalen

Vragen?