

Urbinn

URBINN is het LearningLab rondom autonoom rijdend vervoer binnen stedelijke gebieden (last mile). Binnen URBINN wordt een autonoom rijdend voertuig ontwikkeld dat als basis zal dienen om steeds verder te ontwikkelen en waar allerlei onderzoek op en mee gedaan zal gaan worden. URBINN is als LearningLab gelieerd aan het Lectoraat Smart Sensor Systems, terwijl de ontwikkeling van het voertuig geheel wordt gecoördineerd en gefaciliteerd binnen de Betafactory. Het onderzoek van URBINN sluit daarbij aan bij het onderzoeksplatform : “Goed Bestuur in een Veilige Wereld”.

# Milestones

* Milestone 1 - Orientatie project
* Milestone 2 - ORB localization gang Slinger/KITTI
* Milestone 3 - Object detection gang Slinger/Kitti
* Milestone 4 - Volledige semantische map (testcase)
* Milestone 5 - Volledige semantische map (Delft)

# Portfolio

In de portfolio geef ik per sprint aan wat ik allemaal voor het project heb gedaan. Hier komen issues, opdrachten en courses die ik heb gemaakt aan bod.

**Sprint 1**

De milestone van sprint een was om te oriënteren op het project Urbinn.

De eerste week hadden we een scrum workshop waar we de principes van scrum konden leren. We moesten ook een opdracht ervoor maken. In deze opdracht heb ik beschreven hoe scrum aan de ‘Principles behind the Agile Manifesto’ voldoet. (Zie scrum opdracht)

We gingen naar Delft om de locatie te onderzoeken. We konden dan een beter beeld schetsen van de eisen waaraan de semantische kaart moet voldoen.

Tijdens deze sprint was ik met de volgende issues bezig.

* + Issue 27 : SVO Paper bestuderen

**Sprint 2**

De milestone van sprint 2 was om een point cloud te generen van de Slinger en Kitti dataset.

Tijdens deze sprint was ik met de volgende issues bezig.

* + Issue 32 : Camera Calibratie,
  + Issue 40 : ORB2 save/load map,
  + Issue 44 : Installeren ORB2 SLAM server

Het was niet gelukt om een point cloud te maken van stereo beelden van de slinger. Dit omdat we de kalibratie van onze zelfgemaakte stereocamera niet goed konden uitvoeren.

Met de mono beelden van Slinger en de stereobeelden van Kitti konden we wel point clouds genereren .

**Sprint 3**

De Milestone van sprint 3 was om Object detection uit te voeren op de Slinger en Kitti dataset.

Tijdens deze sprint was ik met de volgende issues bezig.

* + Issue 55 : Object detection papers lezen.
  + Issue 58 : Beschrijven data structuur MapPoint in ORB2
  + Issue 60 : Beschrijven structuur KeyFrameDatabase in ORB2
  + Issue 71 : Resutaten ORB2 bin exporteren

Tijdens deze sprint was ik ook bezig met de presentaties en het updaten van het blog.

## Scrum

In deze opdracht beschrijf ik hoe scrum aan de ‘Principles behind the Agile Manifesto’ voldoet.

Scrum maakt gebruikt van sprints waar aan het einde van een sprint een werkende product opgeleverd wordt. De derde principe “Deliver working software frequently” wordt hiermee voldaan. Het gebruik van sprints is dan ook het maatstaf waarmee de voortgang wordt bepaald.

Na elke sprint zijn er specifieke momenten om feedback te geven. Hier kunnen de requirements ook bijgesteld worden. Zo gebruikt scrum flexibel requirements als voordeel voor de opdrachtgever.

Daily standup help ervoor dat er communicatie is tussen alle betrokkenen. Ook help het voor het persoonlijk overleggen.

Tijdens de sprint review wordt er bepaald wat er de volgende keer beter kan. Hier wordt het telkens proces verbeterd.

## Issues

Hier worden de issues beschreven waar ik aan heb gewerkt.

**Issue 27 : SVO Paper bestuderen**

Om het beste SLAM Algoritme te selecteren hebben we verschillende papers doorgenomen.

Ik moest samen met Isa de SVO (Semi-direct Visual Odometry) Algoritme doornemen.

Een grote nadeel van SVO is dat het geen loop closure heeft. We hebben een samenvatting gemaakt van de werking van SVO. Dit is als bijlage opgenomen.

Daarna hebben we ook een overzicht gemaakt van alle mogelijke algoritmes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Naam** | **Geïnstalleerd** | **Open Source** | **Configuratie** | **ROS benodigd** | **Up-To-Date** | **Licentie** |
| ORB-SLAM2 | Ja | Ja | Alles | Nee | Ja | GPL V3 |
| SVO | Nee | Ja | Monoculair | Nee | Nee | GPL V3 |
| SVO2 | Nee | Nee | Alles | Nee | Ja | - |
| Monocular LSD-SLAM | Ja | Ja | Monoculair | Ja | Nee | - |
| Stereo LSD-SLAM | Nee | Nee | Stereo | Ja | Nee | - |
| DSO | Nee | Ja | - | - | Ja | GPL V3 |

**Issue 32 : Camera Calibratie**

Voor het kalibreren van de camera hebben we naar verschillende mogelijkheden gekeken. Een veelgebruikte standaard is de OpenCV Camera Calibratie framework. We hebben deze in de omgeving van ROS gebruikt.

**Issue 40 : ORB2 save/load map**

De save/ load map feature is door een andere developer gemaakt . Het is gemaakt met behulp van de BOOST C++ framework. De boost serializatie zorgt ervoor dat de Map en de KeyframeDatabase classen geserializeerd worden. Als de applicatie opnieuw geopend wordt kunnen de classen in dezelfde staat geïmporteerd worden. Op deze manier is het mogelijk een pointcloud te updaten met verschillende runs. Deze feature maakt een map.bin file met de classes.

**Issue 44 : Installeren ORB2 SLAM server**

Ik heb ORB SLAM2 samen met Jeffrey geïnstalleerd op een centrale server. (Eerste server)

**Issue 51 : ORB2 extracten points naar csv file**

Om de evaluatie uit te voeren hebben we de gegenereerde point cloud nodig om de nauwkeurigheid van de algoritme te bepalen. De functionaliteit was toegevoegd en gemerged met de master branch.

**Issue 55 : Object detection papers lezen.**

Voor het selecteren van een object detection framework moesten we twee papers doornemen.

We zijn tijdens de group reading sessions snel achter gekomen dat YOLO (You Only Look Once) het beste was.

**Documentatie ORB SLAM2:**

ORB SLAM2 is een open source SLAM algoritme met een loop closure mogelijkheid. Dit is een van de belangrijke redenen waarom we ervoor hebben gekozen. De source code was volledig gedocumenteerd. Ik heb documentatie toegevoegd aan de source code.

**Issue 58 : Beschrijven data structuur MapPoint in ORB2**

**Issue 60 : Beschrijven structuur KeyFrameDatabase in ORB2**

**Issue 71 : Resutaten ORB2 bin exporteren**

ORB SLAM2 is in c++ geschreven. Om het import tool te maken moest ik een functie maken in c++ die van ORB SLAM 2 geïmporteerd kon worden in een andere programma.