

ZOEKEN, STUREN EN BEWEGEN

**Autonome registratie van
beweging en reactie op beweging
door een AR-drone quadcopter**

Auteurs:

Haitam BEN YAHIA, *10552359*

Harm MANDERS, *10677186*

Tony NGUYEN, *10646973*

Arend VAN DORMALEN, *10615199*

Jeroen VRANKEN, *10658491*

Supervisor:

Toto VAN INGE

Universiteit van Amsterdam

4 juli 2014

Inhoudsopgave

1	Introductie	2
1.1	Onderzoeksvraag	2
1.2	Hypothese	2
2	Materiaal en Methode	2
2.1	Materiaal	2
2.2	Methode	3
3	Resultaten	4
3.1	Beweging	4
3.2	Tracking	5
4	Conclusie	5
5	Bibliografie	6
A	Labboek	6
A.1	Vrijdag 20 juni	6
A.2	Maandag 23 juni	6
A.3	Dinsdag 24 juni	6
A.4	Woensdag 25 juni	7
A.5	Donderdag 26 juni	7
B	Specificaties Software	7
B.1	Programmeertalen	7
B.2	Programma's	7
B.3	Besturingssysteem	8
C	Specificaties Hardware	8

1 Introductie

1.1 Onderzoeksvraag

In dit onderzoek gaat een quadcopter(AR-drone) een mens achtervolgen. De proefpersoon zal op zijn rug drie stippen dragen ter herkenning. De stippen worden in een gelijkbenige driehoek geplaatst, waarvan twee stippen boven op een lijn boven één onderste stip staan.

Uit voorgaand onderzoek blijkt dat het mogelijk is om een quadcopter een vliegend object te laten volgen [2]. Ook is het mogelijk om een AR-drone bewakingsopdrachten uit te laten voeren door middel van bewegingsdetectie[3]. Hierbij kan de drone objecten volgen op een grote hoogte.

In dit onderzoek wordt er op deze onderzoeken voortgebouwd en zal de drone een object volgen dat op de grond beweegt. Hierbij zal de drone op gelijke hoogte als het object voortbewegen en zich dus ongeveer anderhalve meter boven de grond bevinden.

1.2 Hypothese

Op basis van de voorgaande onderzoeken lijkt het dat deze toepassing mogelijk is. Dankzij de oppervlakte van de driehoek van stippen kan de afstand van de quadcopter tot de proefpersoon bepaald worden. Dankzij de symmetrie van de stippen kan de positie van de proefpersoon bepaald worden. Mede door deze feiten is het de verwachting dat dit onderzoek haalbaar is.

2 Materiaal en Methode

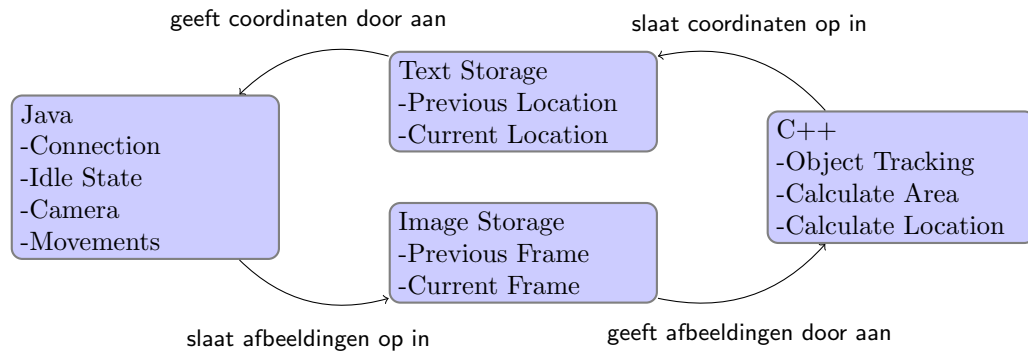
2.1 Materiaal

Er is eerst gedacht aan het gebruik van Node, dit is software om de drone in Javascript te besturen. Qua code was dit ideaal aangezien de drone makkelijk te besturen was met enkele regels. Zo is het ons gelukt om de drone op de juiste hoogte te krijgen om hem vervolgens te laten flippen. Node bleek niet makkelijk te combineren met OpenCV, omdat OpenCV geen Javascript support. OpenCV wordt gebruikt voor het beeldverwerken en is essentieel voor het onderzoek. Er is overgestapt naar Javadrone voor de besturing, aangezien de groep in Java het meeste programmeerervaring heeft. Dit programma is dus als vervanger van Node gaan werken.

De quadcopters waarvan in dit onderzoek gebruik is gemaakt zijn twee versies van de AR drone. Deze quadcopters zijn dankzij hun afmetingen, hun wendbaarheid en het feit dat ze camerabeelden op kunnen nemen erg geschikt voor deze toepassing.

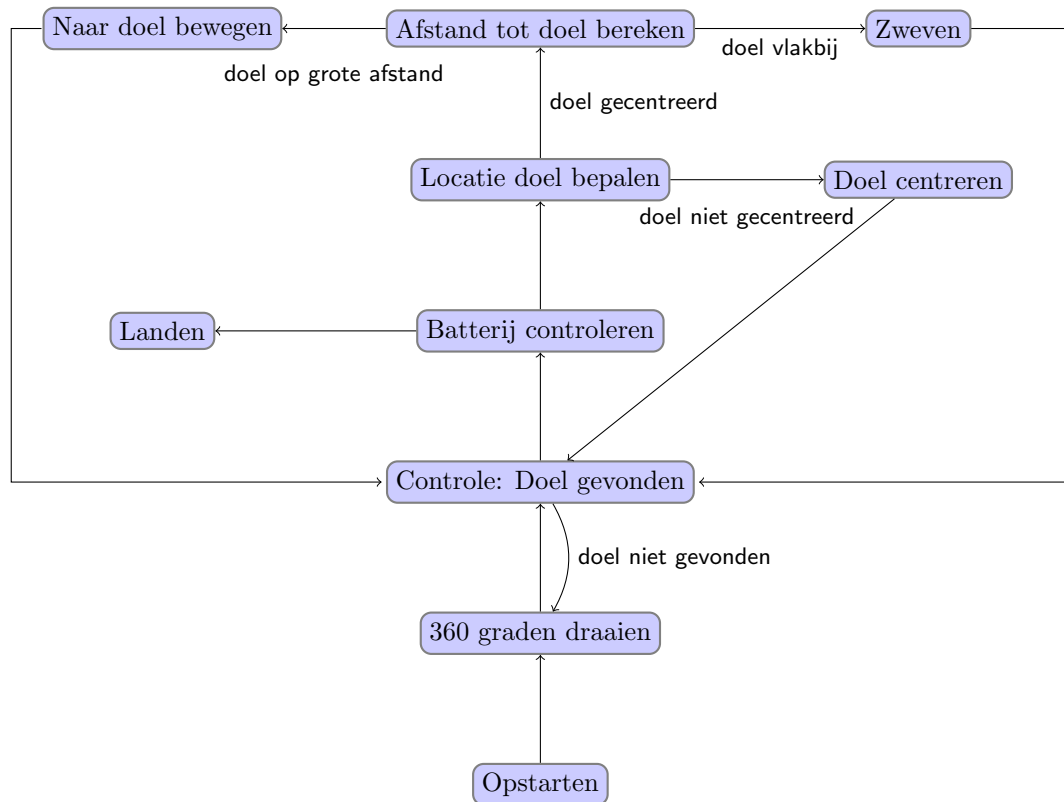
2.2 Methode

Voor dit project werd een structuur uitgedacht waarin twee programmeertalen worden gebruikt. Dit komt doordat sommige taken beter implementeerbaar leken in C++, terwijl een ander deel van de code geschikter leek in Java. Hierbij zijn de taken opgesplitst onder de verschillende programmeertalen (zie figuur 1). Ook is aangegeven wat er opgeslagen moet worden om de twee talen samen te laten functioneren.



Figuur 1: Taakverdeling onder programmeertalen

De uitvoering van de taken verloopt in een vaste volgorde. Eerst steigt de quadcopter op volgens een commando dat in de documentatie van de AR drone meegeleverd was. Hierna draait de drone 360° om op zoek te gaan naar het doel op de rug van de proefpersoon. Wanneer het doel gevonden is wordt er gekeken naar waar het doel zich bevindt ten opzichte van het opgenomen beeld. Als het doel zich niet in het midden van het scherm (met enige marge) bevindt, wordt er gecorrigeerd totdat dit wel zo is. Als het doel zich in het midden van het scherm bevindt, wordt de grootte van het doel geanalyseerd. Als deze grootte onder een bepaalde waarde is, zal de drone naar voren vliegen om het doel te achtervolgen. Als de grootte boven deze waarde is, betekent dit dat het doel dichtbij is en blijft de drone op zijn plek zweven. Hiernaast zal de drone na elke actie zijn batterijcapaciteit testen. Wanneer de batterij bijna leeg is zal de drone een landing inzetten. Net als het bij het opstijgen gaat het hier om een commando wat al meegeleverd was. Zie voor de architectuur figuur 2



Figuur 2: Architectuur Taken

Voor het object tracking is gebruik gemaakt van een bestand op OpenCV [4]. Hierbij werden de RGB-waarden (Red,Green,Blue) omgezet naar HSV-waarden (Hue, Saturation, Value). Op basis van filtering in de HSV-waarden kon een binaire afbeelding gecreerd worden. Onze toevoeging op dit programma is het verscherpen van deze binaire afbeelding, door eerst een minimale pixelgrootte in te stellen, waardoor losse pixels of *white noise* weg werd gefilterd. Deze methode heet *erode*. Daarna werd het overgebleven beeld weer vergroot, waardoor het weer duidelijk zichtbaar was. Deze methode heet *dilate*. Verder is het programma aangepast zodat het naar de juiste HSV-waarden zoekt, die in dit geval toebehoren aan de kleur rood.

3 Resultaten

3.1 Beweging

De AR drone (1.0) kon goed worden bestuurd met behulp van een computer. Het opstijgen ging recht omhoog met zo nu en dan een afwijking naar rechts

of naar links. Bij het zweven bleef de AR drone vrijwel op zijn positie, bij een kleine stootje corrigeerde de AR drone dit vanzelf en bleef het in evenwicht. De 2.0 had echter een sterke afwijking naar voren bij het opstijgen. Het autonoom bewegen op basis van de doorgekregen beelden verliep goed.

3.2 Tracking

Het is niet gelukt om de drie stippen te herkennen met OpenCV. Het is versimpeld tot het herkennen van een rood vlak, waarbij de oppervlakte werd gebruikt als maat voor de afstand. Dit is gelukt en de AR drone kon dit goed herkennen. Het nadeel van het gebruik van een rood vlak was dat de hoek tussen de AR drone en het rode vlak niet berekend kon worden. De AR drone kon dus geen bepaalde hoeken vasthouden ten opzichte van het vlak, maar kon wel parallel aan het vlak bewegen. De AR drone deed dit vrij snel en had minimale afwijkingen ten opzichte van het vlak. Wanneer de AR drone het vlak niet meer in beeld had, bleef het op zijn positie zweven tot dit wel het geval was.

4 Conclusie

De resultaten zijn gunstig, het is uiteindelijk gelukt om de drone een object te laten volgen, de methode met de drie stippen vergde echter meer tijd om te implementeren. Nu het mogelijk is om 1 object te tracken is er een weg gebaad voor verder onderzoek met 3 objecten (stippen).

Achteraf bleek het grootste struikelblok de communicatie tussen java en C++ te zijn. Wellicht was het beter geweest om alles in één taal te schrijven. In dit geval zou C++ erg geschikt zijn geweest. Aangezien onze groep weinig ervaring heeft met C++ is er bewust gekozen voor een combinatie met Java om te zorgen dat het onderzoek haalbaar bleef. Bij een volledige implementatie met C++ zou het leren van de taal te veel tijd hebben gekost. Er is ook gezocht naar een oplossing met Java als enige taal, maar zonder succes. Ook is er veel tijd verloren aan het opzoeken van alternatieve software.

De drones bleken niet allemaal even goed te werken. Een aantal drones zijn vermoedelijk een paar keer hard gevallen waardoor enkele propellers niet even snel draaiden als de rest. Dit leidde tot instabiel opstijgen van de drone. In een potentieel vervolgonderzoek kan hier rekening mee gehouden worden en zou de instabiliteit gecorrigeerd kunnen worden.

Een laatste feit wat voor vervolgonderzoek belangrijk kan zijn, is dat er binnenkort een nieuwe versie van de AR drone uitkomt; de AR drone 3.0. Deze drone heeft verbeteringen ten opzichte van de AR drone 2.0, zoals een scherper beeld en een betere stabiliteit. Hiermee kan het mogelijk zijn om betere resultaten te halen voor een vergelijkbaar onderzoek dan met de oudere versies van de AR drone.

5 Bibliografie

- [1] A. van Inge, *Search, Actuate, and Navigate Lab course*, 2013.
- [2] Zhu, Yuke and Song, Zhao, *Vision-based Autonomous Flying Target Following*, 2012, Simon Fraser University, Canada
- [3] Bernardini, Sara and Fox, Maria and Long, Derek, *Planning the Behaviour of Low-Cost Quadcopters for Surveillance Missions*, 2014, Twenty-Fourth International Conference on Automated Planning and Scheduling.
- [4] Hounslow, Kyle, *ObjectTrackingTut.cpp*, 2013.

A Labboek

A.1 Vrijdag 20 juni

Op vrijdag zijn we begonnen met het project, aangezien alle vijf leden van de groep hun opdracht van die week af hadden. Eerst zijn we naar het robotlab gegaan, waar we via een Android-app een paar testvluchten met de drone hebben gemaakt. Hierdoor kregen we een idee van de mogelijke bewegingen en de struikelpunten van de drone. Er waren reparaties nodig aan de drone, we hebben een rotor vervangen en een paar breuken in het frame afgeplakt met tape. Jeroen, Arend en Harm hebben lang geprobeerd om de drone via de computer live bestuurbaar te krijgen, maar dit was zonder resultaat. Ook lukte het hierna niet om de drone via een Xbox-controller te besturen.

A.2 Maandag 23 juni

Op maandag hebben we de benodigde software geïnstalleerd om de drone bestuurbaar te krijgen en om te kunnen coderen. Haitam deed onderzoek naar Node.js, software om de drone te besturen met javascript. Tony installeerde Javadrone, hiermee lukte het om programma's te schrijven waarmee de drone bestuurbaar was, het lukte maandag nog niet om beeld terug te krijgen via Javadrone. Er werd veel tijd besteed aan een poging tot het koppelen van deze twee programmas, dit bleek echter te veel werk. Arend en Jeroen zijn bezig geweest met het downloaden en installeren van OpenCV. Dit bleek op Linux helaas veel problemen te geven. Bij Arend bleek er een gebrek aan schijfruimte te zijn, waardoor een nieuwe partitie gemaakt moest worden. Ook bij Jeroen ging het installeren niet vlekkeloos en kostte het veel moeite om OpenCV werkend te krijgen.

A.3 Dinsdag 24 juni

Op dinsdag hebben Tony en Haitam vooruitgang geboekt met het verkrijgen van beelden van de drone. Dit lukte via zowel Javadrone als Node.js. Javadrone kreeg de beelden dit keer wel terug, maar alleen van de AR drone 1.0 en niet de

2.0 versie. Node.js kon een videobestand opslaan van een vlucht, maar dit bleek moeilijk in OpenCV te importeren. Tony is het gelukt om zowel een sequentie van plaatjes op te slaan als een 2d array met RGB waarden tijdens een vlucht. Dit zou handig kunnen zijn als OpenCV ook in Java werkt. Daarna hebben Arend, Harm en Jeroen RGB waarden om weten te zetten naar HSV waarden en lukte het om via OpenCV objecten in een bepaalde kleur te kunnen identificeren en volgen. De verbinding naar de RGB-waarden die de drone registreerde werd nog niet gemaakt.

A.4 Woensdag 25 juni

Op woensdag hebben Haitam en Jeroen ervoor gezorgd dat video- en fotobestanden in OpenCV leesbaar werden en objecten in deze bestanden herkenbaar werden. Tony had met OpenCV een gezichtsherkenningsmodule geïmplementeerd. Ook probeerde Tony om herkenning van een driehoek mogelijk te maken, dit lukte echter niet. Mede hierdoor hebben we afgezien van het idee met drie stippen in een driehoek. Harm is op zoek gegaan naar manieren om Java en C++ te combineren. Socket communication leek hiervoor de meest gangbare oplossing. Arend heeft een opzet aan het verslag gemaakt.

A.5 Donderdag 26 juni

Op donderdag begonnen we met het aanpakken van het probleem dat veroorzaakt werd door de verschillende talen. Het object tracking programma werkt in C++, terwijl de aansturing van de drone en de meeste andere functionaliteiten in Java zijn geprogrammeerd. Arend is doorgeshaan met onderzoek naar mogelijk gebruik van socket communication, dit bleek echter te lastig om in twee dagen te implementeren. Jeroen, Haitam en Tony zijn tegelijkertijd begonnen met het omschrijven van het C++ bestand naar Java. Haitam is vervolgens begonnen met het schrijven van een programma wat correcties maakt op basis van de locatie van het object in het beeld.

B Specificaties Software

B.1 Programmeertalen

- Java versie 1.7
- C++ versie 11

B.2 Programma's

- Javadrone versie 1.3
- OpenCV versie 2.4.9

B.3 Besturingssysteem

- Ubuntu 12.04

C Specificaties Hardware

Tijdens het onderzoek zijn twee versies van de AR drone gebruikt. Uiteindelijke resultaten zijn geboekt met de AR drone 1.0.

- Parrot AR drone 2.0
- Parrot AR drone 1.0