

UNIVERSITEIT VAN AMSTERDAM

KUNSTMATIGE INTELLIGENTIE

ZOEKEN, STUREN EN BEWEGEN

Shapecopter

Figuren tekenen met een AR.Drone

Auteurs:

Alex Khawalid

Erik Stammes

Max Bos

Ragger Jonkers

Supervisor:

Toto van Inge

July 4, 2014

Abstract

Dit onderzoeksverslag beschrijft een manier om de AR Drone te laten tekenen. Het ultieme doel was om de quadcopter letters en woorden te kunnen laten schrijven met behulp van beeldherkenning. In dit verslag is dit beperkt tot basisvormen zonder beeldherkenning. De software, javadrone[1], is naast het besturen van de quadcopter ook goed voor beeldherkenning. Als tekenmiddel is gekozen voor een spons, die in tegenstelling tot een pen niet te zwaar en dus niet het pad van de drone beïnvloedt, maar wel onder zijn eigen gewicht kan tekenen. Deze spons werd achter de drone aangesleept via een touwtje. Zonder beeldherkenning is het gelukt om een basisvorm te vliegen met de drone. In een vervolgonderzoek zouden er met beeldherkenning meer ingewikkelde vormen te maken kunnen zijn.

1 Inleiding

Shapecopter is een samenvoeging van *shape* en *quadcopter*. Met *shape* wordt een figuur bedoeld die een quadcopter zou kunnen tekenen. Een quadcopter is een helicopter met vier propellers. Om een vorm te tekenen met een quadcopter zijn er verschillende subproblemen die daarvoor opgelost moeten worden, hiervoor zijn onderzoeksgebieden als inverse kinematica en lineaire algebra belangrijk.

Quadcopters zijn al in staat om bepaalde voorgeprogrammeerde bewegingen te maken, tevens kunnen ze objecten herkennen met de aanwezige camera's.

De hoofdvraag van dit onderzoek is of de quadcopter in staat is om op een nauwkeurige manier een figuur te tekenen. Hiervoor dient ook uitgezocht te worden wat het beste gebruikt kan worden voor het tekenen. Het tekenen van een random figuur ansich is een simpele opgave, maar om op een nauwkeurige manier een gepland figuur te tekenen vergt meer aandacht. Met genoeg tijd en middelen is dit echter mogelijk. Om de figuren te tekenen wordt er vanuit gegaan dat het makkelijk is om dit met een touwtje en een pen of stift te doen.

Voor het bewegen van de quadcopter zijn er al verschillende bibliotheken[1][2][6] aanwezig met een-

voudige functies, hier worden dus geen problemen verwacht.

Met de resultaten uit het onderzoek zijn quadcopters niet alleen in staat figuur te tekenen, maar kan ook de volgende stap gezet worden door complexere figuren en zelfs tekst te tekenen.

Een quadcopter vliegt uit zichzelf niet altijd recht, er zijn veel factoren die invloed hebben op de manier van vliegen. Door bijvoorbeeld wisselende batterijen en de invloed van wind moet een quadcopter vaak correcties uitvoeren. Hierdoor kan het zijn dat er meer dan simpele vliegbewegingen nodig zijn om een figuur te tekenen, bijvoorbeeld het herkennen van getekende lijnen met de naar beneden wijzende camera. Daarnaast kan het zijn dat een pen of stift te zwaar is voor de quadcopter en dat deze vervangen moet worden door iets lichters.

2 Materiaal en methode

De gebruikte quadcopter is een AR.Drone 1.0[5] van Parrot, deze quadcopter heeft meerdere sensoren en twee camera's. Het gebruikte materiaal om de figuren te tekenen is een stukje touw van 100cm, met daaraan een stukje spons bevestigd. De communicatie met de quadcopter is gedaan met behulp van javadrone, een Java-bibliotheek waarmee de quadcopter bestuurd kan worden. Tevens verzorgt deze bibliotheek het videosignaal. Voor de beeldherkenning is gebruik gemaakt van OpenCV[4], een C++ bibliotheek.

2.1 Het tekenen van een lijn

Het tekenen van figuren kan lastig zijn, dus is het handig om eerst een aantal figuren uit te proberen. Het eerste figuur dat getest werd is een lijn, dit omdat het simpel is en meteen een goed idee geeft van de problemen die zich voor zouden gaan doen. Bij het trekken van een lijn worden nog veel factoren die later wel belangrijk zijn buiten beschouwing gelaten. Het gaat bij de lijn erom dat het fundamentele idee van een lijn op papier krijgen werkt. De lengte van het touwtje kan licht variëren doordat deze niet elke keer precies hetzelfde aan de drone geknoopt wordt. Dit hoeft niet een groot probleem te zijn in de prak-



Figure 1: Parrot AR.Drone 1.0

tijk als het touw elke keer ongeveer hetzelfde wordt vastgemaakt. Verder kan de voor de offset van het touw een variabele gebruikt worden en dit kan voor gebruik gec calibreerd worden.

De spons heeft een touwtje van 25, 50 en 100cm aan het bevestigingspunt geknoopt gekregen. Met deze verschillende lengtes touw kan geprobeerd worden om de spons te bevestigen aan de quadcopter. Een andere lengte van het touwtje heeft invloed op hoeveel groter de baan van de quadcopter moet zijn om de figuur te tekenen.

2.2 Teken met de quadcopter in praktijk

Het eerste wat uitgezocht moest worden was hoe de drone bestuurd werd. Met Node.js en een AR Drone library kan de quadcopter met high-level functies zoals `takeoff()`, `stop()` en `move()` bewegen. Tegelijkertijd is er uitgezocht wat de beste methode van tekenen met de drone is. Er is gekeken naar de volgende methoden:

Case 1: De stift

Een stift rechtstreeks op de onderkant van de drone plakken

Dit zou voor het tekenen van vormen niet gunstig zijn. Elke keer dat de stift het tekenoppervlak zou raken zou dit ervoor zorgen dat hij weer een beetje opstijgt omdat hij gewicht verliest. Voor het tekenen van lijnen en figuren is dit niet handig. Als het pointillisme als tekenstijl gehanteerd werd, zou dit wel kunnen werken.

Case 2: De pen

Een pen aan een touwtje achter de drone aan slepen

De lijn is niet zo dik maar er moet druk gezet worden om een spoor achter te laten. Na praktisch testen bleek ook dat de pen te zwaar is hiervoor.

Case 3: Het sponsje

Een sponsje doppen in inkt of waterverf en deze aan een touwtje achter de drone aan slepen

Dit zou gaan lekken en problemen veroorzaken met inkt op het papier. Dit zou wel opgelost kunnen worden door het op en neer laten gaan van de drone zodat de extra inkt eraf geschud wordt, of de drone even op de zelfde plek laten vliegen zodat het overschot er vanzelf afdruipt. Anders kan de inkt/verf ook nog van tevoren worden aangebracht, en tevens worden afgedropen .

3 Resultaten

Herkenning van tekengebied

Bij het herkennen van het tekengebied, in dit geval een papiertje, werd OpenCV[4] gebruikt. Hierbij is de OpenCV library gebonden aan NodeJs met `opencv-node`[3]. In de documentatie van `nodejs` staat al code wat dit doet, en de basis vormt voor het programma waarmee de drone getest zou worden. Na een aantal tests bleek dat NodeJs geen videostream ondersteunt van de AR drone 1 en de onderste camera van de AR drone 2 ook niet. Dit gecombineerd met het slecht werken van de basisfuncties en de af en

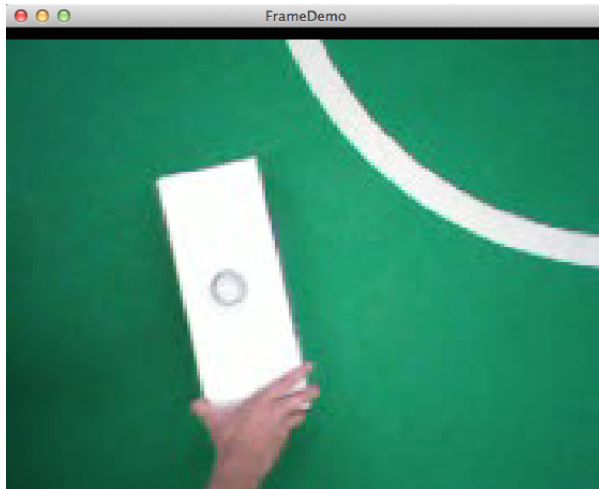


Figure 2: De video output van de quadcopter naar javadrone

toe willekeurige uitkomsten van het runnen van programma's heeft ertoe geleid dat er van framework overgestapt moest worden. Het besluit viel op het YADrone framework[6]. Deze leek goed te werken en de basisbewegingen waren stabiel. Alleen de video werkte niet, het bleek dat de video decoder niet meer bestaat en dus viel YADrone ook af. Hierna werd er gewerkt met javadrone[1]. Hierbij werkte alles na een tijdje, zelfs de video.

De drone was met javadrone tijdens de demonstratie in staat om een vierkant te vliegen. De spons zelf was bijna op de correcte hoogte, en kwam eigenlijk alleen een heel klein stukje van de grond bij het van richting veranderen van de drone, bij bijvoorbeeld een pitch- of rollbeweging.

4 Discussie

Bij de hypothese van ons onderzoek hebben we het feit dat de drone niet super accuraat is, enigszins onderschat. Dit had mede te maken met de software keuze. Er zijn namelijk erg veel verschillende libraries voor de AR Drone die allemaal hun eigen voordelen en nadelen hebben. Zo is het met de ene library gemakkelijk om de drone besturen, maar werken de

compatibiliteiten van de camera niet goed, of heeft het ingebouwde predicaat voor optijgen een instabiele uitvoering.

Het is echter wel gelukt om gecontroleerd een basisvorm te vliegen met de quadcopter. Hierbij hebben we een touw met vaste lengte gebruikt, waarmee we ook een vorm kregen met vaste grootte. Om dit variabel te maken, moet gekeken worden naar de extra afstand die de quadcopter moet vliegen om het sponsje mee te slepen. In theorie zou deze afstand, die mede afhangt van de touwlengte, berekend kunnen worden. In de praktijk is de offset niet altijd hetzelfde, omdat de quadcopter soms wat onnauwkeurig is. Het idee was daarom om op het moment zelf deze correcties te herstellen door middel van beeldherkenning. Op deze manier konden we bijvoorbeeld inaccuraat opstijgen verhelpen door de drone het tekenpapier te laten herkennen, of de drone de spons laten herkennen, zodat precies vanaf de spons weer verder getekend kan worden. Gezien de beperkte tijd was het echter niet mogelijk dat allemaal op tijd werkend te krijgen.

Met functionaliteiten als beeldherkenning zou het in het vervolg zelfs mogelijk zijn om geavanceerdere vormen, of zelfs letters te maken met de quadcopter. Dit zou weliswaar op iets grotere schaal moeten gebeuren in verband met de onnauwkeurigheid van de drone.

References

- [1] javadrone. <http://code.google.com/p/javadrone/>.
- [2] node-ar-drone. <https://github.com/felixge/node-ar-drone>.
- [3] node-opencv. <https://github.com/peterbraden/node-opencv>.
- [4] OpenCV. <http://opencv.org>.
- [5] Parrot AR.Drone. <http://web.archive.org/web/20100727045028/http://ardrone.parrot.com/parrot-ar-drone/en>.
- [6] YADrone. <https://github.com/MahatmaX/YADrone>.