KU Leuven
Departement Computerwetenschappen

P&O: Computerwetenschappen

Tussentijds verslag

Team:
Zilver

Bram Vandendriessche (Coördinator)
Arne Vlietinck (Secretaris)
Matthias Van der Heyden
Jef Versyck
Vincent Vliegen
Laura Vranken

Academiejaar 2016-2017

Samenvatting

Auteurs: ; redactie: Bram Vandendriessche, Laura Vranken & Arne Vlietinck

Inhoudsopgave

Inleiding

Auteurs: Laura Vranken & Arne Vlietinck

Drones zijn de laatste jaren enorm in populariteit toegenomen en ondervinden bijgevolg ook een evolutie op technologisch vlak. Ze worden tegenwoordig gebruikt voor talloze toepassingen. De bekendste toepassing bevindt zich binnen Defensie, die drones gebruiken om informatie te verkrijgen over vijandelijk gebied zonder mensenlevens te moeten riskeren. Daarnaast hebben ook grote bedrijven (o.a. Amazon[1]) de weg naar deze technologie gevonden. De toekomst brengt echter nog veel meer voordelen. Enkele voorbeelden [1] zijn veiligheidsinspectie van windturbines of elektriciteitslijnen, luchtsteun bij zoek- en reddingsoperaties, bewaking en luchtfotografie.
Wanneer een drone autonoom functioneert, is een betrouwbare aansturing door de Autopilot van levensbelang. Hij moet namelijk bestand zijn tegen allerlei externe factoren (bv. wind, obstakels...).

Dit verslag behandelt de autonome aansturing van een drone, meer bepaald een quadcopter. Er wordt uitgegaan van een drone waarop twee voorwaarts gerichte camera's bevestigd zijn. Op basis van deze beelden moeten afstand en positie tegenover het doel ingeschat worden en nieuwe bewegingsopdrachten voor de drone gegenereerd worden. Deze bewegingen worden weergegeven in een Virtual Testbed. Dit is een softwaresysteem dat een fysieke opstelling van een drone en camera's simuleert. De simulator genereert beelden van de drone uit verschillende standpunten a.d.h.v. de verkregen bewegingsopdrachten van de Autopilot.

De Autopilot en het Virtual Testbed moeten zo ontworpen worden dat de drone in staat is om zijn doel te lokaliseren en er naar toe te vliegen. Dit semester is dat doel een polyhedron, willekeurig gegenereerd door het Testbed, bestaande uit verschillende driehoeken. In de opgave en sectie 1 staat meer specifiek uitgelegd aan welke HSV-combinaties de driehoeken moeten voldoen. Daarnaast wordt verwacht dat windinvloeden uit willekeurige richtingen teniet worden gedaan. Ook het ontwijken van obstakels behoort tot een van de vereisten. Bovendien moet de drone rond een object kunnen vliegen en hertekenen wat hij waarneemt. Tenslotte moet er een generator ontwikkeld worden die nieuwe werelden kan genereren en een editor die deze werelden kan bewerken.

De tekst is als volgt opgebouwd:
Eerst wordt het ontwerp van de Autopilot en Virtual Testbed verder uitgediept (sectie **??**). Vervolgens wordt er ingegaan op de gebruikte algoritmes (sectie **??**) en wordt de opbouw van onze software verduidelijkt (sectie **??**). Tot slot worden de uitgevoerde testen besproken (sectie 4) en wordt dit aangevuld met een kort besluit.

1 Beeldverwerking en grafische voorstelling

1.1 Beeldverwerking Polyhedra

Auteur: Laura Vranken

Om naar polyhedra te kunnen vliegen, moet de drone ze eerst kunnen herkennen op de ontvangen beelden. Het analyseren van de beelden gebeurt als volgt.
Ten eerste worden alle pixels gegroepeerd per kleur. Dan worden de kleuren opgesplitst in verschillende lijsten volgens hun soort, nl. binnen- en buitendriehoek van respectievelijk target en obstakel. Zie tabel 1 voor de exacte voorwaarden.
Om het zwaartepunt van de driehoek te kunnen bepalen, wordt eerst naar de hoekpunten gezocht. Hiervoor wordt de buitenste pixel van elke zijde van de driehoek berekend. Aangezien het kan zijn dat twee hoekpunten op dezelfde lijn pixels liggen, worden twee pixels van elke zijde bepaald (de minimum en maximum pixel). Op deze manier worden acht pixels gevonden. Figuur 1a geeft een voorbeeld van deze methode. Natuurlijk heeft een driehoek geen acht hoeken, dus worden de overeenkomstige buitenste pixels samengenomen tot één hoekpunt.

[1] Amazon Prime Air

	H	S	V
Buitendriehoek target	?	> 0.55	> 0.55
Binnendriehoek target	?	< 0.45	> 0.55
Buitendriehoek obstakel	?	> 0.55	< 0.45
Binnendriehoek obstakel	?	< 0.45	< 0.45

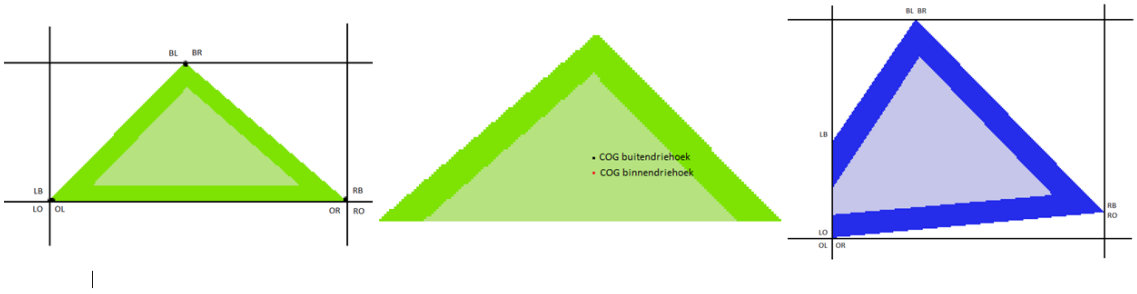
Tabel 1: Combinatie HSV-waarden om de verschillende driehoeken te herkennen. Hue waarde mag willekeurig gekozen worden.

Een volledige driehoek behoudt drie hoekpunten. Ook kan een driehoek die evenwijdig met een zijde door een andere polyhedra of door de rand van het beeld afgesneden wordt, drie hoekpunten overhouden. Indien niet evenwijdig met een zijde wordt afgesneden, blijven er vier of meer hoekpunten over. Zie Figuur 1b voor het eerste geval en Figuur 1c voor het tweede geval. De figuren met meer dan drie hoekpunten worden niet meer bekeken; ze zijn niet volledig.

Om onderscheid te maken tussen volledige en afgesneden driehoeken met drie hoekpunten, wordt de buitendriehoek samen met de overeenkomstige binnendriehoek verder onderzocht. Indien hun zwaartepunten samenvallen, zijn ze volledig. Indien ze niet matchen, kan geconcludeerd worden dat een deel van de driehoek is weggevallen. Zwaartepunten kunnen bepaald worden door Formule 1:

$$(x_{zpt}, y_{zpt}) = (\ \frac{1}{3}(x_1 + x_2 + x_3),\ \frac{1}{3}(y_1 + y_2 + y_3)\ ) \tag{1}$$

Dit gebeurt voor beide camera's en vervolgens worden de gevonden zwaartepunten gecombineerd en doorgestuurd naar de fysische component van de Autopilot om omgezet te worden naar 3D-coördinaten in het wereldassenstelsel. Uit deze reële punten zal de dichtstbijzijnde polyhedron gevonden worden en als doel ingesteld worden.



Figuur 1: Bepalen van de buitenste 8 punten van mogelijke driehoeken.
a) Een volledige driehoek.
b) Evenwijdig afgesneden driehoek met 3 hoekpunten. De zwaartepunten (COG) liggen echter op een andere plaats.
c) Willekeurig afgesneden driehoek met vier hoekpunten.
Afkortingen: LO = links onder, LB = links boven, RO = rechts onder, RB = rechts boven, OL = onder links, OR = onder rechts, BL = boven links, BR = boven rechts.

1.2 Visuele voorstelling bij de Autopilot

Auteur: Bram Vandendriessche

Voor de grafische weergave van de gescande objecten, is een iets andere aanpak gevolgd dan bij de Polyhedra in het Testbed om een onderscheid te verkrijgen tussen presentatie en representatie. *TriangleAPData* bestaat louter uit de data van een Triangle, d.w.z. de coördinaten van de punten

en de kleuren. Een Polyhedron is in twee delen opgesplitst: een data-object (*PolyhedronAPData*) dat de data-driehoeken bevat en een visueel object dat zal zorgen voor het tekenen van de driehoeken van de Polyhedron. Het visuele object bevat hiervoor een *TriangleDrawer*, die d.m.v. zijn draw-methode een meegegeven *TriangleAPData*-object visueel zal voorstellen.

Net als het Testbed, werkt ook de Autopilot met een World, die voor de implementatie van de OpenGL-functies zorgt. De wereld werd opgesplitst in een data-object en een visueel . Dit laatste zal met de gegevens uit de data-wereld een visuele voorstelling kunnen verzorgen.

Om te kunnen scannen, wat nog niet functioneel is, zullen de hoekpunten van de driehoeken omgezet worden in 3D-coördinaten met behulp van de beeldherkenning. Op die manier kan een polyhedron worden samengesteld uit alle gescande driehoeken. De beeldherkenning zoekt per kleur de hoekpunten van de buitendriehoek in de linker- en rechtercamera. Om de 3D-coördinaten van de hoekpunten in het wereldassenstelsel te kunnen berekenen, moeten de overeenkomstige hoekpunten uit beide beelden samengenomen worden. Het vinden van die overeenkomstige hoekpunten gebeurt door een algoritme, gelijkend op hetgeen dat controleert of een driehoek volledig is. Het zoekt eveneens naar uiterste punten en kan zo de combinaties terugvinden.

2 Vliegstrategie

2.1 Ruimtelijke voorstelling d.m.v. vectoren

Auteur: Vincent Vliegen

De Autopilot krijgt via de vernieuwde interface nu een positie en een volledige oriëntatie mee. Deze maken het mogelijk voor de Autopilot om accuraat verplaatsingen en rotaties in de ruimte te beschrijven. De extra gegevens zorgen ervoor dat de beweging van de drone, de krachten in de ruimte en andere variabelen als vectoren beschreven kunnen worden. Bovendien is er voldoende informatie aanwezig om een goede benadering te maken van de effecten die de onbekende windkrachten veroorzaken.

De voorstelling met vectoren voorziet de mogelijkheid om het vliegtraject van de drone te bepalen en bij te sturen naar wens. De positie van de drone bepaalt enkele cruciale gegevens: de snelheid, verplaatsingsrichting en een onbekende translatie door de wind. De oriëntatie daarentegen is nodig om de rotaties te berekenen om de vliegrichting te corrigeren, alsook om de in- en uitwendige krachten op de drone juist te oriënteren en om de eventuele invloed van windrotaties in te perken. Met al deze gegevens is de Autopilot in staat om de gewenste thrust en rotatiesnelheden te bepalen, om op een zo efficiënt mogelijke manier naar een gegeven targetpositie te vliegen.

Daarnaast worden objecten nu ook voorgesteld met vectoren. Aan de hand van de camerabeelden wordt de positie van het object relatief tegenover de drone bepaald. Wanneer de oriëntatie en positie van de drone in rekening worden gebracht, is ook de positie van het object in de ruimte bekend. Deze kan dan gebruikt worden als targetpositie of voor andere doeleinden.

2.2 Rotatiematrix

Auteur: Arne Vlietinck

Na een uitgebreide wijziging van de opgave werden de rotatiematrices opnieuw opgesteld. De conventies in de nieuwe opgave komen niet overeen met de algemene conventies in de literatuur. Hierdoor werden de rotatiematrices met extra zorg opgesteld. Het is een cruciaal element voor de oriëntatie en bewegingen van de drone. De RPY-matrices worden vermenigvuldigd in omgekeerde volgorde (YPR) dit door de conventies van de rotatiematrices. Ook de inversie rotatiematrix kan berekend worden.

$$\begin{bmatrix} cos(r) & sin(r) & 0 \\ -sin(r) & cos(r) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(p) & sin(p) \\ 0 & -sin(p) & cos(p) \end{bmatrix} \begin{bmatrix} cos(y) & 0 & -sin(y) \\ 0 & 1 & 0 \\ sin(y) & 0 & cos(y) \end{bmatrix}$$

Tabel 2: Bovenstaande matrixen geven respectievelijk de yaw-, pitch- en rollmatrix weer.

2.3 Vliegstrategie

Auteur: Vincent Vliegen

De Autopilot maakt gebruik van vectoren om de drone en andere objecten in de ruimte te situeren. Dit maakt het mogelijk om de thrustgrootte en rotatiesnelheden te berekenen, zodat de drone volgens een optimaal pad kan vliegen.

De drone start met een gegeven positie en oriëntatie. De Autopilot heeft nu twee opties: ofwel stelt deze in dat de drone naar een gegeven positie in de ruimte vliegt, ofwel dat de camera's anders georiënteerd worden.
Omwille van de uitwendige krachten op de drone, vliegt de drone niet in de richting van de thrustkracht. De zwaartekracht, drag en wind, berekend en opgeteld met de thrust, resulteren samen in de verplaatsingsrichting van de drone. Wanneer er een targetpositie is meegegeven, kan de thrustgrootte zo ingesteld worden dat de drone zo goed mogelijk in de richting van het doel blijft vliegen.

Daarna wordt een gewenste oriëntatie berekend. Hiervoor bestaan meerdere opties. Enerzijds kunnen de camera's een gevraagde richting uit kijken met de thrust zo gericht dat de drone quasi stil hangt. Anderzijds kan de thrust zo ingesteld worden dat de drone met een gewenste versnelling naar een doel vliegt met de camera's zo goed mogelijk kijkend volgens het vliegtraject.
Wanneer de gewenste oriëntatie bekend is, kunnen de nodige rotatiesnelheden berekend worden om de drone bij te sturen. Met behulp van de rotatiematrices worden de nog af te leggen rotaties bepaald. Wanneer hierbij het benaderde rotationele effect van de wind mee in rekening wordt gebracht, kunnen de rotatiesnelheden voor de yaw, pitch en roll vastgelegd worden.

2.4 Windcorrectie

Auteur: Vincent Vliegen

De wind zorgt voor een verandering van de positie en oriëntatie van de drone. Aangezien de Autopilot wordt opgeroepen met een constant tijdsinterval, kunnen er voorspellingen worden gemaakt over de ruimtelijke situering van de drone, wanneer er een bepaalde hoeveelheid tijd is verlopen.

De Autopilot stelt de thrust en rotatiesnelheden in. Met behulp van de uitwendige krachten en rotaties kunnen de verplaatsingsrichting en verandering van oriëntatie bepaald worden. Wanneer ook het constante tijdsinterval in rekening wordt gebracht, kan de verwachte positie benaderd worden aan de hand van de bewegingsvergelijking (zie Tabel 3 voor uitleg symbolen):

$$\frac{(\vec{T} + \vec{G} + \vec{D} + \vec{W})}{2m} * (\Delta t)^2 + \vec{v_0} * \Delta t + \vec{x_0} = \vec{x}_{exp} \tag{2}$$

De verwachte yaw, pitch en roll, die samen de oriëntatie bepalen, worden als volgt berekend:

$$\theta_0 + (\omega + \omega_{wind}) * \Delta t = \theta_{exp} \tag{3}$$

De Autopilot controleert in het begin van elke cyclus of de verwachte positie en de eigenlijke positie overeenkomen, respectievelijk de verwachte en eigenlijke oriëntatie. Zo niet, is de afwijking te verklaren als een verandering van de windtranslatie en -rotatie.
Het verschil in positie is het gevolg van een onnauwkeurige benadering van de windtranslatie. Deze kan gecorrigeerd worden door de kracht te berekenen die de afwijking heeft veroorzaakt, en dan

op te tellen bij de huidige windkracht.

$$\frac{(\vec{W}_{corr})}{2m} * (\Delta t)^2 = \vec{x} - \vec{x}_{exp} \tag{4}$$

De afwijking in rotatie duidt een rotatieverandering van de wind aan. Ook hier kan een correctie berekend worden voor de yaw, pitch en roll, die vervolgens opgeteld wordt met de huidige windrotaties.

$$\omega_{corr} * \Delta t = \theta - \theta_{exp} \tag{5}$$

$\vec{T}$: Thrust	Δt: tijdsverschil tussen twee frames	θ_0: initiële hoek
$\vec{G}$: Gravity	$\vec{v_0}$: startsnelheid	ω: hoeksnelheid
$\vec{D}$: Drag	$\vec{x_0}$: beginpositie	ω_{wind}: windhoeksnelheid
$\vec{W}$: Wind	$\vec{x}_{exp}$: verwachte positie	θ_{exp}: verwachte hoek
$\vec{W}_{corr}$: Windcorrectie	ω_{corr}: hoeksnelheidscorrectie	

Tabel 3: Verduidelijking van de gebruikte notaties.

2.5 Afremmen

Auteur: Matthias Van der Heyden

De Autopilot is zodanig ontworpen dat de drone tot stilstand kan komen in de coördinaten van het opgegeven target. Dit doet hij door voortdurend de targetpositie en maximale toelaatbare acceleratie mee te geven. Deze acceleratie wordt berekend aan de hand van de drone zijn huidige snelheid en afstand tot de targetpositie.

Afhankelijk van zijn snelheid ten opzichte van de afstand tot het doel moet de drone afremmen (bij $\frac{snelheid}{afstand} \geq empirischefactor$) of versnellen (bij $\frac{snelheid}{afstand} < empirischefactor$). De grootte van de acceleratie of deceleratie is steeds de maximaal mogelijke waarde, rekening houdend met de huidige wind, tenzij de afstand tot het doel kleiner is dan een bepaalde afstand. In dat geval verkleint de waarde met een afstandsafhankelijke factor.

3 Virtual Testbed

3.1 Polyhedra en het bestaande ontwerp

Auteur: Bram Vandendriessche

Het ontwerp van de simulator zorgt ervoor dat de uitbreiding ervan met polyhedra vrij eenvoudig in te voeren is. Door het nieuwe type Polyhedron dat erft van de *WorldObjects*, moesten in *World* zelf slechts enkele extra methodes worden geïmplementeerd m.b.t. het toevoegen en verwijderen van een Polyhedron aan de bestaande wereld.
Het ontwerp van het Testbed heeft de grote zwakte dat het erg moeilijk is om individuele elementen en methodes te testen. Dit komt doordat de objecten vaak erg afhankelijk zijn van de wereld waarin ze bestaan. Die wereld is bovendien altijd afhankelijk van de *gl*-component van OpenGL, waardoor het noodzakelijk is een volledige wereld op te zetten voor de test kan worden uitgevoerd. Een oplossing hiervoor zou zijn om de objecten en wereld zo op te splitsen dat presentatie en representatie gesplitst worden, zodat het representatie-gedeelte onafhankelijk kan gebruikt worden. Daarnaast zou dan enkel de wereld weet moeten hebben van haar objecten en niet andersom, zodat objecten onafhankelijk van de wereld kunnen worden getest.

De Polyhedron zelf is zo ontworpen dat die bestaat uit verschillende Triangles. De tekenfunctie van de Polyhedron draagt elke Triangle die tot de Polyhedron behoort, op om zichzelf te tekenen. Een

Triangle bestaat uit coördinaten voor zijn drie hoekpunten en een kleur voor het buitenste gedeelte van de driehoek. Op basis daarvan worden de binnenste hoekpunten berekend en er wordt een kleur afgeleid uit de gegeven kleur zodat de driehoek voldoet aan de gegeven beperkingen m.b.t. de kleur, zie tabel 1, en de oppervlakte.
Voor het testen van de Autopilot zijn verschillende figuren ontworpen. Zij vallen onder de *PredefinedPolyhedron*, een subklasse van Polyhedron die ook een positie mee kan krijgen bij creatie. Hierdoor kan de figuur op een gewenste positie worden geplaatst. De standaard Polyhedra krijgen geen positie mee. Hun positie wordt gezet op het massapunt van de polyhedron (de gemiddelde 3D-coördinaat van alle hoekpunten). Op die manier wordt gezorgd dat de positie logisch is t.o.v. de hoekpunten en het niet mogelijk is dat een figuur bijvoorbeeld rond (20,0,0) wordt gedefinieerd, maar wel een positie van (0,0,-40) krijgt toegewezen. De figuren variëren van een eenvoudige met vier hoekpunten tot meer complexe zoals een kubus met een opening in, waarbij de Autopilot zal moeten weten dat het voor de drone niet mogelijk is hierdoor te vliegen.

3.2 Generator en editor

Auteur: Jef Versyck

Dit semester wordt gevraagd om een wereld generator zelf te implementeren. Elk bestand gegenereerd door deze generator, moet voldoen aan de opgestelde eisen. Op dit moment zijn er twee generators, één die een wereld met bollen maakt en één die een wereld met Polyhedra maakt. Deze laatste kiest uit verschillende vooraf aangemaakte vaste Polyhedra, zoals hierboven vermeld.

Verder is de editor meer gedetailleerd uitgewerkt. Men selecteert eerst een object met de muis. Met behulp van de toetsen i en k (bewegen volgens positieve en negatieve x-as), j en l (bewegen volgens positieve en negatieve z-as), u en o (bewegen volgens positieve en negatieve y-as), kan het aangeklikte object nu verplaatst worden in de wereld. Er bestaat op dit moment geen manier om polyhedra toe te voegen. Elk object kan verwijderd worden met behulp van de DELETE knop.

3.3 Collision Detection

Auteur: Jef Versyck

Aangezien polyhedra niet per definitie mooie cirkels zijn, moet de collision detection veranderd worden. Eerst wordt er vergeleken of de afstand tussen de twee centra van de objecten (een polyhedron en de drone) kleiner is dan beide hun radii opgeteld. De radius van een polyhedron wordt gedefiniëerd als de grootste afstand tussen het centrum van de polyhedra en zijn punten.
Vervolgens wordt er getest of de loodrechte afstand op het vlak van een Triangle vanuit het middelpunt van de drone kleiner is dan de straal van de drone. Hierbij wordt het vlak waarin de Triangle zich bevindt, berekend aan de hand van het kruisproduct van twee vectoren van de Triangle en een hoekpunt ervan. Vervolgens wordt de loodrechte projectie op het vlak bepaald. Dit punt heet P. In formulevorm:

$$a * x + b * y + c * z = d$$

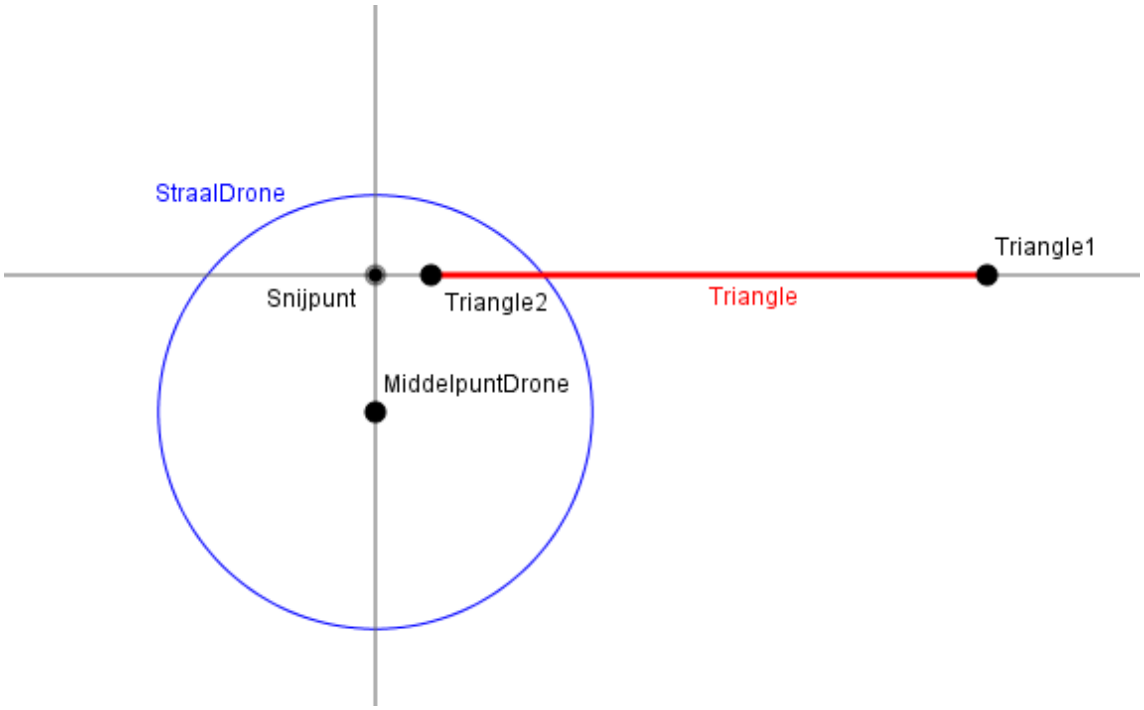
$$t = -\frac{a * x_{drone} + b * y_{drone} + c * z_{drone} - d}{a^2 + b^2 + c^2}$$

$$P = \begin{Bmatrix} a * t + x_{drone} \\ b * t + y_{drone} \\ c * t + z_{drone} \end{Bmatrix}$$

Dit is echter niet genoeg. De loodrechte projectie van het massacentrum kan zich buiten de Triangle bevinden en zo een foutief resultaat geven. Om te zien of het punt zich binnen de Triangle bevindt, wordt aan de hand van barycentrische coördinaten gecontroleerd. Het principe hierachter is dat elk punt P in de driehoek geschreven kan worden als een lineaire combinatie van de hoekpunten (A, B en C) van een driehoek. De som van alle coëfficiënten moet gelijk zijn aan 1 en alle coëfficiënten moeten groter zijn dan 0. In formulevorm:

$$P = u * A + v * B + w * C$$
$$0 \leq u, v, w \leq 1$$
$$u + v + w = 1$$

Dit lost echter niet alle problemen op. De drone kan de polyhedron raken, maar de loodrechte afstand kan buiten de triangle liggen. Hiervoor werd ten tijde van schrijven nog geen oplossing gevonden. Zie Figuur 2 voor verduidelijking. Het snijpunt ligt duidelijk buiten de Triangle, maar toch snijdt de drone de Triangle.



Figuur 2: Een voorbeeld van het probleem met de huidige collision detection.

4 Testen

4.1 Testen beeldverwerking

Auteur: Laura Vranken

Om de beeldverwerking van de polyhedra te testen, zijn verschillende afbeeldingen in het project ingeladen. Hierop zijn verschillende testen uitgevoerd. Er wordt onder andere gecontroleerd of de omzetting van integer waarden naar HSV waarden juist gebeurt. Bovendien is ook de functie getest die het zwaartepunt van elke zichtbare volledige driehoek teruggeeft. Dit bleek vrij accuraat. De laatste test controleert of het bepalen van de 3D coördinaten van de hoekpunten juist gebeurt. Dit is geverifieerd met de coördinaten van het Testbed.

Besluit

Auteurs: ; redactie: Arne Vlietinck

Referenties

[1] M. GMBH, *Microdrone-applications: aerial, photography, mapping, surveying, etc.* `https://www.microdrones.com/en/applications/`, 2016. [Geraadpleegd op 30 oktober 2016].