

Indoor Navigatiesysteem

Research and Development 1

Niklas Weber
Philipp Jakubeit
Anne Westerhof

July 6, 2010

1 Inleiding

Je stelt je thuisadres in. Je stelt in waar je naar toe wilt en de route wordt berekend vanaf waar je op dat moment bent. Het enige wat je nog hoeft te doen om bij je bestemming aan te komen is de instructies op het apparaat te volgen. Veel mensen kunnen tegenwoordig niet meer zonder navigatiesysteem in hun auto om te komen waar ze willen zijn. Dit bracht ons op het idee om te kijken of dit niet ook binnen kan. Vaak kom je in je leven in een groot gebouw wat je nog niet kent. Een sollicitatiegesprek, een afspraak in het ziekenhuis of de eerste keer in een nieuwe school of universiteit, er zijn altijd situaties waarin je een groot gebouw in loopt met in je ene hand een papiertje met het nummer van de kamer of het lokaal waar je naartoe moet en als je geluk hebt een mobieltje met telefoonnummer in je andere hand voor als je echt de weg kwijt raakt. Het zou natuurlijk veel makkelijker zijn als we in plaats van een nood telefoonnummer een programmatje op dat mobieltje hadden wat hetzelfde doet als die navigatiesystemen waar we zo gewend aan zijn geraakt. Hierbij is het wel zo dat er natuurlijk niet boven elk gebouw een satelliet hangt en dus kunnen niet exact dezelfde methodes als bij de auto navigatiesystemen gebruikt worden, omdat de GPS satellieten hier bij lange na niet gedetailleerd genoeg voor zijn. Maar toch zou het al een hele stap zijn als je aan kan geven waar je in het gebouw bent, waar je naar toe wilt om vervolgens de route op je scherm geprint te krijgen.

Met dat idee in ons hoofd zijn wij aan de slag gegaan. We hebben een prototype van een programma gemaakt wat het bovenstaande doet en dit vervolgens zorgvuldig uitgegetest. Het volgende verslag verteld onze methodes van het testen van het product en de resultaten hiervan. De vraag, de probleemstelling, die we hierbij in ons hoofd hielden was: "Is het gebruik van dit programma in de praktijk haalbaar?". Met andere woorden, is het prototype op dit moment goed genoeg om verspreid te worden en gebruikt te worden door mensen of zouden er nog veel dingen moeten veranderen. Deze probleemstelling sloot het meest aan bij wat we gemaakt hebben en liet de weg open voor een aantal subvragen die we door testen konden beantwoorden. Deze zullen verderop beschreven worden. Tijd om op weg te gaan.

2 Theoretisch Kader

2.1 Navigatiesystemen

Het is wel duidelijk dat ons product gebaseerd is op de al bestaande navigatiesystemen voor auto's. Deze navigatiesystemen maken gebruik van het zogenaamde Global Positioning System. Hierbij zit er een ontvanger in het systeem die signalen van satellieten rond de aarde opvangt om aan de hand hiervan te kunnen bepalen waar het systeem (en dus jij, waarschijnlijk) zich bevindt.[4] Ondanks dat we dit natuurlijk al wisten hebben we het toch nog een keer nagelezen om te kijken of we iets van de werking van deze systemen zouden kunnen gebruiken voor ons systeem. Echter bleek al snel dat dit niet zou lukken. Ondanks dat er ook in telefoons GPS ontvangers zitten tegenwoordig zijn deze veel te onnauwkeurig om in gebouwen te werken en daarbij kan men vaak in gebouwen niet eens een signaal opvangen.[3][2] Wel zijn we erachter gekomen dat de navigatiesysteem in auto's gebruik maken van een shortest path algoritme, om van de ene locatie naar de andere een pad te kunnen vinden. Echter is niet duidelijk wat voor een representatie er precies gebruikt wordt. Het gaat natuurlijk om hele grote oppervlakken waarover steeds berekend moet worden, maar helaas ontbreken er goede websites/boeken die precies uitleggen hoe een navigatiesysteem werkt. Zelfs een emailtje naar een aantal merken van deze systemen heeft niets opgeleverd en dus hebben we het grootste gedeelte van ons project zelf bedacht en gemaakt.

2.2 Theorie achter ons prototype

Zoals al gezegd hebben we een programma geschreven wat je binnen een gebouw van de ene locatie naar de andere locatie leid. Er zijn een aantal dingen die belangrijk zijn om te weten voordat we ons verdere onderzoek

bekend maken om de resultaten goed te kunnen begrijpen. Het eerste wat we ons afvroegen is of het mogelijk was iemand's positie te bepalen. Zoals al gezegd is het echter niet mogelijk om dit te doen aan de hand van GPS zoals makkelijk zou zijn geweest, maar moesten we op zoek naar andere oplossingen. Het zou bijvoorbeeld mogelijk zijn door in het gebouw bepaalde ontvangers te plaatsen op strategische plaatsen en iedereen die het gebouw binnenkomt een soort zendertje te geven en daarmee continu de route te bepalen, maar aangezien het idee was dat iedereen dit programma op zijn mobiele telefoon zou kunnen downloaden en vervolgens gebruiken is dit niet mogelijk. Een andere optie was gebruik te maken van een draadloos netwerk wat eventueel al in het gebouw aanwezig is, maar hetzelfde probleem was dat niet alle gebouwen een draadloos netwerk hebben. Aangezien tegenwoordig de meeste gebouwen dit wel hebben, hebben we geprobeerd uit te vinden hoe dit zou werken, maar kwamen tot de conclusie dat dit toch ook nog niet zo precies was als we het wilden hebben en we niet uit konden vinden hoe we dat zouden moeten implementeren. Uiteindelijk hebben we dan ook voor een simpelere versie gekozen die meer lijkt op de routeplanners online. Na het opstarten van het programma geef je aan waar je bent en waar je naar toe wilt, en vervolgens wordt je route gegeven. Ten tweede moesten we nadenken over de representatie van de gebouwen binnen ons programma. Deze keus was snel gemaakt. We hebben hierbij gekozen voor een graaf representatie zoals die bij veel programma's wordt gebruikt om afstanden te bepalen. Om ruimte te besparen hebben we daarbij gekozen voor een lijst representatie, waarbij we dus bij elk punt opslaan welk punt eraan vast zit in plaats van een matrix te gebruiken. Ook hebben we een coördinaten systeem gebruikt (op dit moment in twee dimensies) om gewichten tussen de verschillende punten te kunnen bepalen zonder deze expliciet bij te houden. Om vervolgens paden te kunnen berekenen konden we nu al een bestaand algoritme kiezen, namelijk het Dijkstra algoritme voor shortest paths[5], die we een klein beetje hebben aangepast naar onze representatie via ons coördinaten systeem. De graafrepresentatie van zo'n gebouw kan via een bepaald formaat in een tekstbestand worden gezet wat vervolgens wordt ingelezen door het programma. Om verder te weten hoe het programma werkt raden we aan om de bijbehorende documentatie te lezen.

2.3 OnderzoeksvARIABLEN

De vraag of het gebruik van dit prototype in de praktijk haalbaar is, is niet simpel te beantwoorden. Om deze vraag te kunnen beantwoorden hebben we een aantal onderzoeksvARIABLEN gekozen en die in twee groepen opgedeeld.

Gebruiksvriendelijkheid voor de gebruiker:

- Snelheid (Het programma moet snel zijn in gebruik)
- Efficiëntie (Het programma mag geen instructies geven die overbodig zijn)
- Duidelijkheid (De verschillende instructies moeten makkelijk uit te voeren zijn)

Gemak voor gebouweigenaren:

- Mobiliteit (Werkt het programma op alle mobieltjes?)
- Gemak kaarten te maken (Het gebouw moet in korte tijd om te zetten zijn in een geschikte representatie voor het programma)

Dit zijn voor ons de belangrijkste punten waarop we willen beoordelen of dit programma in de praktijk gebruikt kan worden. Hieronder zullen we per punt zowel een verduidelijking geven van het punt, de reden waarom we dit zo belangrijk vinden en een concrete manier om het punt te testen.

Snelheid

Met snelheid bedoelen we vooral dat het berekenen van de route niet te lang mag duren. Hetzelfde geldt natuurlijk voor het opstarten van het programma en het geven van de instructies, maar dit is lastiger en minder relevant om te beoordelen. We hebben snelheid als belangrijk punt gekozen, omdat niemand een programma wil gebruiken wat langzaam is. Als men in de tijd dat het programma de route berekent diezelfde route ook inmiddels al wel zelf had kunnen vinden is het programma niet meer nuttig. De reden dat we niet naar de snelheid van het opstarten van het programma kijken is dat we dit niet goed kunnen beoordelen. Als iemand een hele langzame telefoon heeft dan zal het natuurlijk een tijdje duren, maar het programma op zich is heel klein en zal niet veel tijd kosten om op te starten. Ook kijken we niet naar de tijd van het geven van de instructies, omdat deze al bij het berekenen van de route gegenereerd worden en de reden dat ze niet allemaal gelijk weer worden gegeven is dat er gewacht wordt op input van de gebruiker om deze zo goed mogelijk langs de route te kunnen leiden.

Het belangrijkste wat dus tijd zou kunnen kosten is het berekenen van de route. Zoals al gezegd hebben we hiervoor het Dijkstra algoritme gebruikt. Dit algoritme heeft een complexiteit van $O(N^2)$ voor samenhangende grafen en we gaan ervanuit dat de grafen die wij gebruiken samenhangend zijn, omdat er altijd een route is

van een punt naar een ander punt. Ondanks dat we de complexiteit weten willen we toch weten hoe lang het programma er over doet om een route te berekenen om snelheid te kunnen garanderen.

Efficientie

Het is belangrijk dat dit programma geen overbodige code uitvoert, omdat dit alleen maar tijd zou kosten, maar het is net zo belangrijk dat het probleem geen instructies geeft die nergens voor nodig zijn. Stel je namelijk voor dat het programma de volgende instructies geeft:

Ga naar links en loop drie meter.

Ga rechtdoor en loop vijf meter.

Dit is voor de gebruiker zinloos omdat hij net zo goed naar links had kunnen gaan en gelijk acht meter kunnen lopen. Deze en soortgelijke overbodige instructies mogen niet voorkomen, omdat ze storend zouden zijn voor de gebruiker.

Duidelijkheid

Met duidelijkheid bedoelen we simpelweg dat er geen verwarring mag bestaan over de instructies. Het zou tenslotte vervelend zijn als het programma goed werkt, maar de instructies zo onbegrijpbaar zijn dat mensen ze niet kunnen volgen. Dit kunnen problemen zijn zoals instructies die zeggen dat je tien meter moet lopen en je niet weet wat tien meter is. Of een probleem zoals loop tien meter en je komt midden in een gang uit waar helemaal niks is om af te slaan. Om dit soort dingen te voorkomen moeten de instructies duidelijk en simpel zijn.

Mobiliteit

De bedoeling van ons programma is dat iedereen het kan downloaden en via hun eigen telefoon gebruiken zonder dat de eigenaren van een gebouw iets speciaals in hun gebouw hoeven te plaatsen of te veranderen. Hiervoor is het belangrijk dat het programma draait op verschillende mobiele telefoons (het liefst op alle telefoons) omdat het programma anders alsnog door niemand gebruikt kan worden en het gebruik van dit programma nooit zal kunnen plaatsvinden op deze manier.

Gemak kaarten te maken

Het laatste punt is dat het voor de eigenaren van gebouwen die vinden dat dit programma voor hun gebouw gebruikt moet kunnen worden gemakkelijk moet zijn om een bestand te maken dat door het programma ingelezen kan worden. Deze bestanden zijn tekstbestanden die voor elk punt van de graaf een regel hebben met de benodigde informatie. Het is gemakkelijk te zien dat dit voor grote gebouwen veel tijd kan gaan kosten om zo'n bestand te maken. Dit zou er toe kunnen leiden dat niemand kaarten levert waardoor het programma nog steeds niet bruikbaar zou zijn. Dat is ook gelijk de reden dat we dit punt opgenomen hebben voor onze tests.

3 Methode

Aan de hand van de genoemde vijf variabelen zijn we gaan bepalen hoe we deze 5 konden operationaliseren, zodat we ze konden onderzoeken. Daarbij hebben we ook de keus gemaakt dat we alleen besluiten dat het programma inderdaad in de praktijk kan worden uitgevoerd als elk van de variabelen positief getest worden. Hoe we elke variabele getest hebben en wanneer we besluiten dat een variabele positief getest is of negatief vertellen we hieronder per variabele.

Snelheid

Hoewel we al besloten hebben dat de snelheid vooral gebaseerd is op het bepalen van de route en we daarbij ook al de complexiteit van Dijkstra's algoritme kennen wilden we toch een exacte waarde kunnen geven voor de snelheid van het programma. Hiervoor hebben we een test ingebouwd in het programma die na het bepalen van de route aangeeft hoe lang hij hier over gedaan heeft. Vervolgens hebben we een criterium bepaald om de snelheid een score te kunnen geven. Dit criterium is: "Bij het berekenen van 25 routes in verschillende grootte grafen mag de gemiddelde tijd hiervoor niet boven de 0.5 seconde uitkomen." Dit om te garanderen dat ook bij gebouwen zo complex dat wij niet op het idee kwamen het programma snel genoeg draait. Om dit verder te testen hebben we een verschillend aantal grafen gemaakt met N het aantal punten in zo'n graaf. Per graaf stonden alle punten met elkaar in verbinding om maximaal gebruik te maken van het Dijkstra algoritme. We hebben een graaf gebruikt met $N = 10$, met $N = 75$ en met $N = 500$.

Efficientie

Het is lastig te zeggen wanneer een instructie overbodig is of samengevoegd had kunnen worden met een andere instructie. Ook mag het nooit zo zijn dat je een instructie had kunnen overslaan. We hebben een criterium voor dit probleem opgesteld hoewel we er niet helemaal tevreden over zijn. We hebben besloten dat het programma efficient genoeg is als "bij vijf verschillende routes geen enkele instructie overgeslagen had kunnen worden of

bij de stap ervoor uitgevoerd had kunnen worden.” Hiervoor hebben we bij onze testkaart vijf verschillende routes gegenereerd en deze op de kaart gevolgd. Waarbij we elke route net zo vaak gelopen hebben als het aantal instructies in de route keer 2. Per keer lieten we of een instructie weg, of voegden hem samen met de instructie ervoor om te kijken of we nog steeds op dezelfde plek uit zouden komen. Belangrijk was dat bij het samenvoegen van de instructies eerst beide richtingen werden uitgevoerd en daarna de meters bij elkaar opgeteld.

Duidelijkheid

Om dit punt positief te kunnen beoordelen hebben we besloten een 15-tal willekeurige mensen een route binnen de universiteit te laten lopen. Hierbij beoordelen we duidelijkheid als positief op het moment dat alle personen op de plek van bestemming aankomen door het volgen van de instructies. (We vertellen ze van te voren niet, naar welk lokaal ze toe moeten.) Maar ook zijn we geïnteresseerd in hun mening over de instructies. Voor het beoordelen hiervan hebben we ze een kleine vragenlijst voorgelegd na het lopen van de route en met elk nog een paar minuten gesproken om feedback te krijgen. Zie de vragenlijst bij de bijlagen achterin. Om deze test uit te kunnen voeren hebben het programma op een tablet pc laten draaien, hoewel dit niet het doel is, moesten we iets mobiels hebben.

Mobiliteit

Het is vrijwel onmogelijk om te testen of het programma op alle telefoons draait, daarom hebben we de test teruggebracht tot een test waarbij we kijken of het bestand draait op een telefoon die java aankan. In dit geval een nokia die draait op Symbian S60 en een test om te kijken of het programma draait op een windows telefoon. Aangezien de meeste mensen echter een telefoon hebben die NIET op windows draait, moet het programma op een telefoon draaien die wel java aankan, maar geen windows heeft om dit punt positief te kunnen beoordelen. Wat we hiervoor hebben gedaan is simpelweg het programma op de Nokia laden en op een telefoon met windows en vervolgens geprobeerd het programma te draaien. Het programma draait goed als ”een tekstbestand in het goede formaat goed wordt ingelezen, de lokatie en bestemming kunnen worden ingevoerd en de instructies in de goede volgorde worden weergegeven.”

Gemak kaarten te maken

Het kost tijd om kaarten te maken aangezien op dit moment de benodigde bestanden voor het programma nog allemaal met de hand moeten worden gemaakt. Onafhankelijk daarvan (hoe de kaarten ook gemaakt worden) moet dit snel zijn om dit programma in de praktijk te kunnen gebruiken. Ook hiervoor hebben we een criterium opgesteld wat bepaald of het makkelijk genoeg is om de kaarten te maken. We hebben besloten dat voor een willekeurig gebouw het minder dan een minuut moet kosten om tien punten op papier te zetten. Als een gebouw nu 1000 punten heeft zou het nog steeds anderhalfuur duren om zo'n kaart te maken, maar op dit moment vinden we dit acceptabel aangezien zo'n bestand maar n keer gemaakt hoeft te worden. Om dit te testen hebben we alledrie een andere vleugel van het Huygensgebouw gekozen en daar tien minuten van in een bestand gezet en daarvan de tijd gemeten.

4 Resultaten

Snelheid

Grootte	Ms
10	1
10	1
10	0,5
75	32
75	34
75	31
500	210
500	276
500	239
Gemiddeld	91,6

Efficientie

Aantal routes in totaal gecheckt	Aantal routes goed gelopen
96	0

Duidelijkheid

	ja	nee
Op plek van bestemming aangekomen	15	0

	ja	nee
Aangekomen zonder fouten	15	0

	Heel erg oneens	Oneens	Eens noch oneens	Eens	Heel erg eens
De instructies waren eenvoudig	0	0	2	9	4
De instructies waren eenduidig	1	7	6	1	0
De instructies waren duidelijk	1	2	0	7	5
Ik kon de instructies zonder moeilijkheden volgen	1	2	4	7	1
Ik zou dit programma zelf willen gebruiken in de toekomst	0	3	7	5	0

Veel gehoorde statements achteraf:

"Het is lastig meters goed in te schatten."

"Waarom vertelt hij niet precies waar je naar toe moet, bijvoorbeeld de eerste afslag rechts of naar de lift?"

"Lekker simpel programma, maar toch zou grafisch misschien mooier zijn."

"Kan hij de route niet op een kaart weergeven?"

Mobiliteit

Het programma was niet op te starten op de Nokia telefoon.

Het programma was ook niet op te starten op de windows telefoon.

Gemak kaarten maken

Wie	Minuten
Philipp	13.2
Niklas	15.7
Anne	19

5 Discussie

Zoals al eerder gezegd hangt het antwoord van de vraag of dit programma te gebruiken is afhangen van de verschillende onderzoeken die we gedaan hebben. Gelukkig is het eerste punt waar veel gebruikers zich vaak zo veel zorgen over maken erg goed te beoordelen. Volgens onze criterium moest de gemiddelde tijd die het koste om een route te berekenen onder een halve seconde liggen. Het gemiddelde is echter nog minder dan een tiende van een seconde en bij kleinere gebouwen kost het zelfs praktisch geen tijd. Het is dan ook aardig om te zeggen dat dit punt ruimschoots goedgekeurd is.

Het tweede punt is efficiëntie en daar kunnen we zelfs nog korter over zijn. Aan onze tests is te zien dat het op geen enkele manier mogelijk was om een route anders te lopen (korter te lopen) dan aangegeven. Volgens de criteria die we zelf gesteld hebben, kunnen we nu besluiten dat ons programma zeker efficiënt genoeg is. Het kan niet vrkomen dat je een instructie krijgt die je samen met een andere had kunnen uitvoeren. Daarom is ook dit punt goedgekeurd ten opzichte van de probleemstelling.

Het derde punt was een van de belangrijkste, omdat het om gebruikers ervaringen gaat. Dit zorgt er ook voor dat het moeilijker was om een eenduidig criterium op te stellen. Dat was ook de reden dat we besloten hadden om alleen te kijken naar het feit of mensen aankwamen op de plek van bestemming, maar we wilden de rest van hun reacties wel graag meenemen in onze beoordeling van het programma. Aan de resultaten is te zien dat iedereen zonder problemen zijn bestemming heeft bereikt en alleen daarom al vinden we op dit punt het programma goed genoeg om in de praktijk te kunnen werken. Maar toch komen er twee dingen duidelijk naar voren. De eerste is dat veel van de test gebruikers vonden dat de instructies niet altijd eenduidig genoeg waren. De reacties die hierop gegeven werden is meestal dat het probleem zit in het gebruik van meters voor de instructies. Het is lastig een bepaald aantal meters in te schatten. Een instructie met 2 of 5 meter gaat nog wel, maar 77 meter is niet meer goed te testen. Hiervoor is het belangrijk dat als dit programma in de toekomst zou gaan werken de instructies iets uitgebreider worden zoals: "Loop 44 meter tot je voor de lift staat." of "Loop 53 meter tot lokaal HG00.068". Dit zou het voor de meesten al een stuk duidelijker maken. Iets anders wat natuurlijk heel erg zou helpen, maar waar we helaas geen tijd voor hadden is te zorgen dat hij niet alleen tekst instructies geeft, maar ook een kaart weergeeft waar de route op is aangegeven, dit zou voor de meesten al heel veel helpen.

Het vierde en mogelijk wat meer problematische punt is het punt van mobiliteit. We hebben het programma op twee verschillende telefoons getest, maar hebben het tot op heden nog niet aan de praat kunnen krijgen. Naar verder onderzoek gedaan te hebben blijkt dat er meer speciale code nodig is om een java bestand te kunnen draaien op telefoons. Mede omdat een .jar bestand niet geopend wordt als je er op klikt. Hiervoor zouden we bijvoorbeeld een java applet van het programma moeten maken. Hoewel het probleem dus waarschijnlijk vrij snel op te lossen is, werkt het programma op dit moment nog niet op mobieltjes wat dus betekent dat het punt

van mobiliteit een onvoldoende krijgt. Dit is iets wat we in de toekomst nog op zouden kunnen lossen en wat een zeer goede en benodigde verbetering voor dit prototype is, wat dus vooralsnog alleen op de computer of (toch mobiel) tablet pc werkt.

Als laatste hadden we besloten dat het belangrijk was dat de graafbesteden van een gebouw makkelijk gemaakt moeten kunnen worden. Het is denk ik wel duidelijk te zien dat deze tijden op het moment ver buiten de marge liggen. De reden hiervoor is, is dat vanaf de kaart alles duidelijk gemeten moet worden om per kaart een geschikt cordinaten systeem te maken en de afstanden te bepalen tussen alle punten. Ook moeten de geschikte "virtuele" punten gekozen worden. Om dit op te lossen is het vast mogelijk een programma te maken wat naar aanleiding van een ingescande kaart een .dat bestand met de graaf van deze kaart erop oplevert. Echter zou dit waarschijnlijk een heel nieuw project zijn en was dit veel te veel werk om nu nog voor dit prototype te maken. Met deze snelheid duurt het echter veel te lang om een heel gebouw in kaart te brengen. Philipp heeft onze testkaart al gemaakt wat waarschijnlijk verklaart waarom hij sneller klaar was dan Niklas en Anne, maar dan nog duurt het erg lang. Zo heeft het maken van de testkaart meer dan een uur gekost en dat was alleen nog maar de beneden verdieping van het Huygens gebouw.

Voor de toekomst zijn er dus nog dingen die gemaakt moeten worden voor dit programma in de praktijk gebruikt kan worden. De instructies zullen verbeterd moeten worden, waardoor er in de grafen van elk punt een naam moet worden opgeslagen, zodat mensen makkelijker kunnen herkennen waar ze heen worden gestuurd. Ten tweede is natuurlijk dat het programma geschikt moet worden gemaakt voor mobiele telefoons en vervolgens op internet klaargezet moet worden om te downloaden. Vermoedelijk op een website waar vervolgens ook mensen de graafbesteden voor gebouwen kunnen uploaden en downloaden. Als derde, hoewel dit niet echt nodig is, maar wel heel sterk aan te raden, is het maken van een programma wat de graafbesteden maakt aan de hand van een scan van een kaart. Anders is de kans klein dat mensen een bestand voor hun gebouw beschikbaar zullen stellen en zullen mensen alsnog niks aan hun programma hebben. Een vierde punt waar we zelf ontevreden over zijn is dat het niet meer gelukt is om binnen de tijd een mooie grafische outlook aan het programma te geven. Hoewel we waarschijnlijk wel pijlen kunnen geven zouden de grafen en kaarten ingewikkelder worden als ze bij iedere instructie de huidige locatie weer zouden moeten geven en de route die nog te lopen is. Dit bleek tijdens de implementatie toch een stuk ingewikkelder dan we gedacht hadden. Aan al deze punten moeten we misschien nog in de toekomst aan werken als we willen dat dit prototype echt gaat werken.

6 Conclusie

We kunnen concluderen dat het antwoord op onze vraag is dat op dit moment het programma nog niet in de praktijk gebruikt kan worden. Hoewel de aanpassingen hier en daar misschien niet zoveel tijd zouden kosten blijft het feit dat het op dit moment niet mogelijk is. Bestemming bereikt.

References

- [1] Kowoma GPS Systemen, <http://www.kowoma.de/en/gps/>, 2009
- [2] Kowoma GPS Achievable Accuracy, <http://www.kowoma.de/en/gps/accuracy.htm>, 2009
- [3] Waar werkt GPS niet?, <http://www.zowerkt.nl/auto-vervoer/overig/werkt-gps-overal/>
- [4] Hoe werkt GPS?, <http://www.zowerkt.nl/auto-vervoer/overig/hoe-werkt-gps/>
- [5] Analyse van algoritmen, <http://www.liacs.nl/rvvliet/hslava/dijkstra.pdf>, 2008.