

Handheld peephole displays voor een collaboratieve virtuele canvas

Thesis voorgedragen tot het behalen van de graad van licentiaat
in de Informatica, afstudeervariant Informatica-Multimedia

Kristof Verpoorten
Promotor: Prof. dr. Karin Coninx

Academiejaar 2004-2005

Voorwoord

Deze thesis wordt voorgedragen tot het behalen van de graad van licentiaat in de informatica, afstudeervariant Informatica-Multimedia. Graag zou ik enkele mensen willen bedanken zonder wiens hulp dit niet gelukt was.

Mijn promotor prof. dr. Karin Coninx en begeleider dr. Kris Luyten voor hun suggesties en tips zowel bij de literatuurstudie als bij de implementatie.

Maarten Cardinaels voor zijn hulp bij het vinden en oplossen van de fout in de V-scope drivers.

Mieke Haesen en mijn zus Kathleen voor hun hulp bij het opstellen en analyseren van de gebruikerstesten.

Alle medestudenten die gewillig dienst gedaan hebben als proefpersonen voor de gebruikerstesten.

Mijn vader Louis en mijn zus Kathleen voor het nalezen en verbeteren van deze tekst.

Mijn ouders voor hun steun en omdat ze mij de mogelijkheid gegeven hebben om deze studie te volgen.

Mijn vriendin Tamara voor haar steun en begrip telkens ik weer eens geen tijd had omdat ik aan deze thesis moest werken.

Kristof Verpoorten
20 juni 2005

Samenvatting

PDA's worden tegenwoordig voor steeds meer toepassingen gebruikt die vroeger enkel mogelijk waren met een gewone PC. Hierdoor is duidelijk geworden dat de PDA enkele belangrijke nadelen heeft ten opzichte van een PC. Het grootste nadeel is de beschikbare schermruimte. Vooral wanneer verschillende gebruikers samen aan een zelfde stuk data willen werken, wordt dit bijna onmogelijk gemaakt door het kleine scherm. In deze thesis wordt hiervoor een mogelijke oplossing beschreven, namelijk peephole displays op een collaboratieve virtuele canvas.

Peephole displays gebruiken hun positie in de omgeving om te navigeren door de virtuele ruimte. De gebruiker kan verschillende delen van het canvas bekijken door zijn PDA simpelweg te bewegen naar de positie van het deel van het virtuele canvas dat hij zichtbaar wil maken. Daarenboven is zoomen ook een belangrijke interactiemethode voor peephole displays. Door de PDA verticaal te bewegen, kan er in -of uitgezoomd worden op het zichtbare deel van het canvas. Om dit zoomen op een eenvoudige en consistente manier te implementeren, wordt gebruik gemaakt van de Piccolo toolkit. Deze toolkit is speciaal ontworpen om Zoomable User Interfaces te ontwikkelen.

Om dit alles mogelijk te maken, moeten de PDA's uiteraard hun positie in de omgeving kennen. Die positie wordt bekomen met behulp van de V-scope, een 3D tracking systeem dat draadloos de positie van 4 verschillende buttons kan achterhalen met behulp van ultrasone geluiden. Door zulk een button op elke PDA waarvan we de positie willen kennen te bevestigen, kunnen de posities van de PDA's bepaald worden. Deze posities worden vervolgens via een bluetooth verbinding naar de PDA's verzonden, zodat elke PDA zijn eigen positie in de omgeving kent.

Om het collaboratieve aspect mogelijk te maken, moeten de PDA's onderling communiceren. Ook hiervoor wordt er een bluetooth verbinding gebruikt. Er is een collaboratief tekenprogramma gemaakt waarbij 2 gebruikers kunnen tekenen op eenzelfde gedeelte canvas. Elke gebruiker kan zien wat de andere gebruiker tekent en dat eventueel ook aanpassen.

Inhoudsopgave

Voorwoord	i
Samenvatting	ii
Inhoudsopgave	v
Lijst van figuren	vii
1 Inleiding	1
1.1 Doel	1
1.2 Overzicht	1
2 Peephole Displays	3
2.1 Inleiding	3
2.2 Spatially Aware Displays	3
2.2.1 Spatially Aware Palmtop Computers	3
2.2.2 Applicaties	4
2.3 Peephole Metafoor voor Augmented Reality	7
2.3.1 Peephole Metafoor	7
2.3.2 Voordelen	8
2.4 Peephole Displays	8
2.4.1 Boom Chameleon	9
2.4.2 Pen Interactie op Peephole Displays	9
2.5 Two Handed Input	11
2.6 Conclusie	11
3 3D Tracking	13
3.1 Inleiding	13
3.2 Voorwaarden	13
3.3 3D Tracking Systemen	14
3.3.1 Elektromechanische Tracking	14
3.3.2 Elektromagnetische Tracking	16
3.3.3 Akoestische Tracking	18
3.3.4 Optische Tracking	19
3.4 V-scope	21
3.4.1 Werking	22
3.4.2 Voordelen	24

3.4.3	Nadelen	26
3.5	Conclusie	27
4	Zoomable User Interfaces	29
4.1	Inleiding	29
4.2	Toepassingen	29
4.2.1	PhotoMesa	30
4.2.2	Automatisch Zoomen	31
4.3	Toolkits	32
4.3.1	Monolitisch vs Polylitisch	33
4.3.2	Jazz	33
4.3.3	Piccolo	34
4.4	Zoomable User Interfaces en Overview+detail Interfaces	34
4.4.1	Overview+detail interfaces	35
4.4.2	Gebruikerstesten	35
4.5	Conclusie	36
5	Implementatie	37
5.1	Inleiding	37
5.2	Hardware	37
5.2.1	V-Scope	37
5.2.2	Bluetooth	38
5.3	Software	38
5.3.1	PDA Tracker	38
5.3.2	Peephole Library	39
5.3.3	Peephole Applicaties	42
5.3.4	MultiPeephole Library	44
5.3.5	Collaboratief Tekenprogramma	45
5.3.6	Beheer van Gedistribueerde Data	45
5.4	Toekomstig Werk	49
5.5	Conclusie	50
6	Gebruikerstesten	51
6.1	Inleiding	51
6.2	Testen	51
6.2.1	Peephole ImageViewer vs Pocket PhotoMesa	51
6.2.2	Peephole Tekenprogramma vs Windows CE Paint	52
6.2.3	Collaboratief Peephole Tekenprogramma	53
6.3	Vragenlijst	53
6.4	Resultaten	54
6.4.1	Peephole ImageViewer vs Pocket PhotoMesa	54
6.4.2	Peephole Tekenprogramma vs Windows CE Paint	55
6.4.3	Vragenlijst	55
6.5	Conclusie	57
7	Conclusie	58

Bibliografie

61

Lijst van figuren

2.1	Potentiële “hot spots” rond een fax machine	4
2.2	Chameleon prototype [13]	5
2.3	Computer-Augmented Bibliotheek [13]	6
2.4	Van thuis uit de inhoud van uw kantoor bekijken [13]	7
2.5	Peephole display over een grotere virtuele workspace	8
2.6	Navigeren met de Boom Chameleon [33]	9
2.7	Object kopiëren en plakken in het tekenprogramma	10
3.1	Een mechanische arm als 3D Tracking systeem	15
3.2	Een gyroscopisch tracking systeem [30]	16
3.3	Polhemus Fastrack Systeem	17
3.4	Schematische voorstelling van Time-of-Flight Tracking	18
3.5	Schematische voorstelling van Phase-coherent Tracking	19
3.6	Opto-electronic Ceiling Tracker [30]	20
3.7	Outside-in tracking	21
3.8	V-scope Systeem: Microprocessor, towers mount en vier buttons	22
3.9	De V-scope Towers Mount met de drie towers eraan	23
3.10	De vier V-scope buttons	24
3.11	De V-scope in werking met één button [24]	25
3.12	Maximale angle of sight van tower en button [23]	26
4.1	Screenshot van PhotoMesa met meer dan 500 afbeeldingen [2]	30
4.2	Screenshot van PhotoMesa, ingezoomd op een enkele map [2]	31
4.3	De objecten van de Jazz toolkit [4]	34
4.4	De objecten van de Piccolo toolkit [3]	35
5.1	PDA Tracker programma	39
5.2	PDA Tracker: Vscope Options	40
5.3	PDA Tracker: Bluetooth Options	40
5.4	PDA Tracker: PDA Tracking Options	41
5.5	Landkaart van Europa bekijken door de PDA te bewegen	42
5.6	De ImageViewer kan hetzelfde als PhotoMesa gebruikt worden	43
5.7	Het rode vierkant wordt verplaatst door de PDA te bewegen	43
5.8	De lange rode balk is op beide PDA’s zichtbaar	46
6.1	Test 1 met Pocket PhotoMesa	52
6.2	Test 1 met peephole ImageViewer	53

6.3	Resultaten Taak 1: ImageViewer vs Pocket PhotoMesa	54
6.4	Resultaten Taak 2: Tekenprogramma vs Windows CE Paint	55

Hoofdstuk 1

Inleiding

1.1 Doel

De schermruimte van een PDA is te klein om verschillende gebruikers collaboratief aan een zelfde stuk data te laten werken. Een mogelijke oplossing hiervoor is verschillende PDA's samen te laten werken, gebruik makend van peephole displays. Er bestaan verschillende implementaties van peephole displays, maar in essentie zijn ze gelijk. Er is een virtuele workspace die te groot is om in één keer weer te geven op het scherm. De gebruiker kan het scherm bewegen om verschillende delen van die workspace te bekijken en eventueel te manipuleren. Het doel van deze thesis is het onderzoeken en ontwikkelen van collaboratieve peephole displays, waarbij verschillende gebruikers over hun eigen peephole display beschikken en hiermee op een gedeelde workspace kunnen samenwerken.

De thesis bestaat uit twee delen. Enerzijds de literatuurstudie waarin onderzocht is wat peephole displays zijn en welke soft -en hardware nodig is om hiervan een implementatie te maken (hoofdstuk 2, 3 en 4). En anderzijds de implementatie van gewone en collaboratieve peephole displays, verschillende toepassingen hiervoor en een gebruikerstest van deze toepassingen (hoofdstuk 5 en 6).

1.2 Overzicht

Met de opkomst van handheld computers (PDA, GSM, ...) vormt zich een nieuw probleem waar een gewone PC meestal geen last van heeft. Namelijk een te klein scherm voor de toepassingen die de gebruiker erop wil uitvoeren. Dit vormt vooral een probleem wanneer het nodig is om grote hoeveelheden informatie te bekijken en/of te manipuleren. Er zijn reeds vele pogingen gedaan om dit euvel te verhelpen (scroll bars, scroll wieltje aan de zijkant van het display, touch-sensitive strips aan de zijkant van het display, ...), maar geen van de bestaande "oplossingen" zijn erg gebruiksvriendelijk, zeker niet wanneer er veel gewisseld moet worden tussen verschillende delen van de weergegeven data. Hiervoor bestaat sinds kort echter een oplossing die gebruik maakt van natuurlijke vaardigheden die elke gebruiker al kent vanuit het dagelijks leven. De meest logische en natuurlijke manier om andere delen van de data zichtbaar te maken, is om het display gewoon te bewegen naar de positie waar het gewenste deel van de data zich bevindt. Deze werkwijze wordt gebruikt bij peephole displays. In hoofdstuk 2 wordt uitgelegd wat peephole displays juist zijn en zullen ook enkele bestaande voorbeelden gegeven worden.

De oplossing om peephole displays te gebruiken voor het probleem van de te kleine schermen is voor de hand liggend. Er stelt zich echter één probleem: deze peephole displays moeten op een of andere manier weten wat hun positie in de omgeving is. Een oplossing is om het peephole display te laten tracken met behulp van 3D tracking hardware. Een 3D tracking systeem is in staat om de positie en/of oriëntatie van een bepaald object in de omgeving te achterhalen. Er bestaan verschillende soorten 3D tracking systemen waarvan de werking op gekende fysische fenomenen steunt. In hoofdstuk 3 zullen de tracking systemen en de voorwaarden waaraan een goed systeem moet voldoen uitgelegd worden. Er zal ook toegelicht worden welk systeem gekozen is voor deze thesis en waarom.

De user interface is één van de belangrijkste onderdelen van een applicatie. Deze bepaalt namelijk de manier waarop de gebruiker het programma waarneemt en dus ook de gebruiksvriendelijkheid. De programma's die gebruikt worden op de peephole displays moeten geschikt zijn om grote hoeveelheden data tegelijkertijd weer te geven, maar het moet ook mogelijk zijn om slechts een deel van deze data in detail te bekijken. De gebruiker mag niet gedesoriënteerd raken tijdens de overgang tussen verschillende delen of weergaven van de data. Een interactietechniek die dit alles mogelijk maakt, is zoomen. Wanneer het zicht is uitgezoomd, kan de gebruiker alle data tegelijkertijd overzien. Om een bepaald deel in detail te bekijken, moet enkel op dat deel ingezoomd worden. Grafische user interfaces die gebruik maken van zoomen, worden zoomable user interfaces genoemd. In hoofdstuk 4 worden verschillende reeds bestaande programma's voorgesteld die gebruik maken van zoomable user interfaces. Verder worden ook enkele toolkits voorgesteld die speciaal ontwikkeld zijn om zoomable user interfaces te programmeren.

In hoofdstuk 5 wordt een overzicht gegeven van de implementatie. Eerst zal de gebruikte hardware besproken worden en zal ook uitgelegd worden waarom er voor deze hardware gekozen is. Vervolgens zullen de verschillende programma's die ontwikkeld zijn voor gewone en collaboratieve peephole displays uitgelegd worden. Tenslotte worden nog enkele mogelijke uitbreidingen en aanpassingen voor de toekomst gegeven.

Buiten een goede implementatie van de software is het uiteraard belangrijk dat deze gebruiksvriendelijk is. Verder is het wenselijk dat de programma's enige meerwaarde bieden ten opzichte van reeds bestaande programma's. Het is echter niet aan de programmeur om hierover een oordeel te vellen omdat hij meestal bevooroordeeld is en de eigen software verkiest boven andere. Daarom worden hiervoor gebruikerstesten uitgevoerd. Hoofdstuk 6 beschrijft de gebruikerstesten die uitgevoerd zijn voor deze thesis. Er worden zowel objectieve als subjectieve data verzameld. De objectieve data volgen uit enkele testen die de gebruikers uitvoeren en de subjectieve data worden bekomen door de gebruikers na de testen een vragenlijst te laten invullen.

In hoofdstuk 7 wordt er een algemene conclusie van deze thesis gegeven.

Hoofdstuk 2

Peephole Displays

2.1 Inleiding

Er wordt meer gebruik gemaakt van draagbare computers met steeds kleinere schermen. Dit vormt een probleem wanneer men hiermee veel informatie tegelijk wil bekijken en/of manipuleren.

Er bestaan verschillende manieren om doorheen die informatie te navigeren. Veel handheld devices hebben knoppen om omhoog of omlaag te scrollen, een scroll-wieltje aan de zijkant van het scherm, etc. Om op deze manier te navigeren moet de gebruiker telkens de taken waarmee hij op dat moment bezig is, onderbreken.

In dit hoofdstuk worden peephole displays uitgelegd die een oplossing bieden voor dit probleem. Om duidelijk te maken wat peephole displays precies zijn, zal in sectie 2.2 uitgelegd worden wat spatially aware displays zijn. Vervolgens zal de peephole metafoor uit de doeken gedaan worden in sectie 2.3. In sectie 2.4 wordt verduidelijkt wat peephole displays zijn en hoe deze de genoemde problemen oplossen en daarenboven nog enkele voordelen bieden op de klassieke computers. Tenslotte wordt er in sectie 2.5 verklaard wat “two-handed input” te maken heeft met peephole displays.

2.2 Spatially Aware Displays

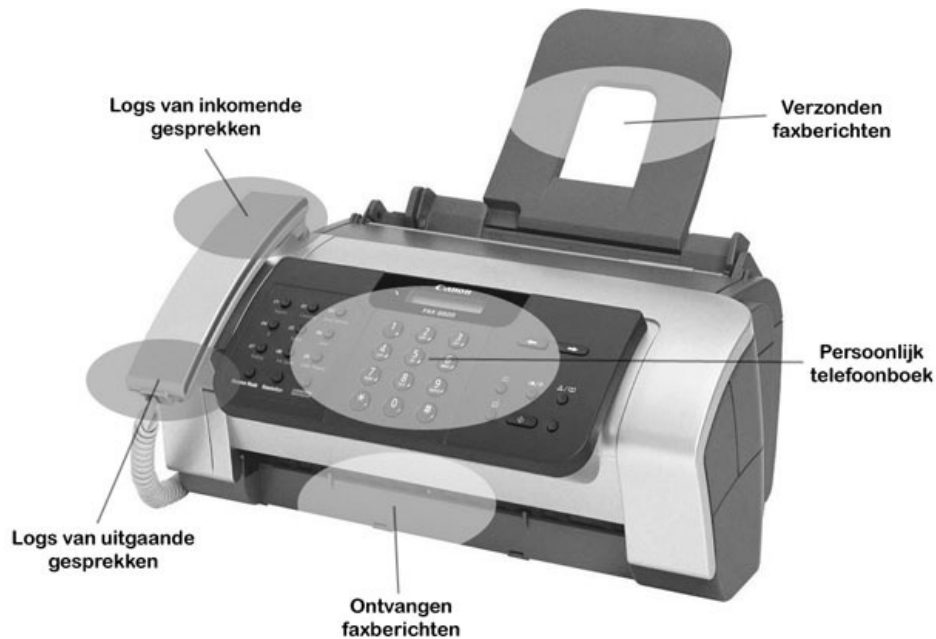
“Spatially aware displays” is een algemene term voor computers die informatie verkrijgen over hun fysieke omgeving. Deze informatie kan bekomen worden door sensoren aan de toestellen te bevestigen [16], of door het toestel te laten tracken met behulp van 3D tracking hardware en de informatie vervolgens naar het apparaat zelf door te sturen.

Meestal is het de positie en/of oriëntatie waarin we geïnteresseerd zijn. Soms is het enige wat men wil weten over het toestel of het zich dicht bij een ander interessant object in de kamer bevindt. Dan doet de positie of oriëntatie van het toestel zelf totaal niet ter zake.

In sectie 2.2.1 zal een voorbeeld gegeven worden van een spatially aware display en in sectie 2.2.2 zullen enkele toepassingen hiervan besproken worden.

2.2.1 Spatially Aware Palmtop Computers

Het idee is om elektronische informatie te associëren met fysieke objecten in de omgeving [13]. Als voorbeeld hiervan wordt een fax machine (figuur 2.1) gegeven, waaraan verschillende



Figuur 2.1: Potentiële “hot spots” rond een fax machine

informatiebronnen bevestigd zijn. De bedoeling is dat al deze informatie bekeken kan worden door een spatially aware palmtop computer rond de fax te bewegen. Deze informatie zou dan op dezelfde manier georganiseerd zijn als de lay-out van het fysieke toestel. Logs van inkomende gesprekken bij de luidspreker van de hoorn, logs van uitgaande gesprekken bij de microfoon van de hoorn, etc. De gebruiker kan door de informatie navigeren op een voor de hand liggende manier, omdat hij al gewend is om met het toestel te werken en dus weet welk soort informatie waar te vinden is.

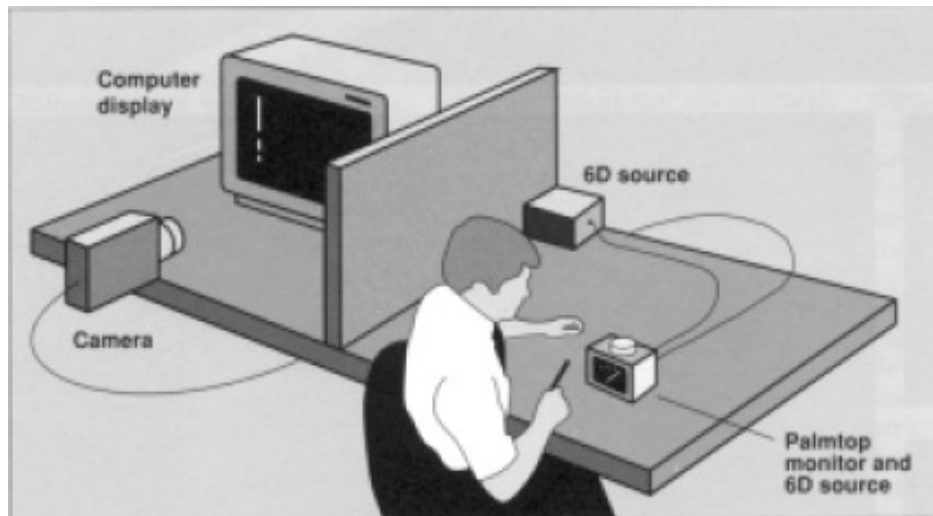
Om dit soort van interactie tussen fysieke objecten en een palmtop te bereiken, wordt het “ubiquitous computing” model gebruikt. Dit model stelt voor om zeer kleine apparaatjes onopvallend te integreren in de omgeving. Deze verborgen apparaatjes zenden regelmatig informatie uit. Wanneer een gebruiker in de buurt van een dergelijk apparaatje komt met een draagbare computer of palmtop, zal hij hierop de uitgezonden informatie te zien krijgen.

Hiervan is een prototype gemaakt dat “Chameleon” genoemd wordt (figuur 2.2). Omdat het prototype draagbaar en tegelijkertijd krachtig genoeg moet zijn om vlot te kunnen werken, wordt een combinatie van een Silicon Graphics 4D/310 Iris workstation en een palmtop gebruikt. De output van het workstation wordt met behulp van een videocamera overgebracht op het handheld display. Om de positie (x, y en z coördinaten), de oriëntatie (pitch, yaw en roll) en input van de gebruiker te detecteren is er een knop en een 6D input device aan het scherm bevestigd.

2.2.2 Applicaties

Er bestaan drie applicaties die gebruik maken van de technologie besproken in sectie 2.2.1 [13].

De eerste applicatie biedt de mogelijkheid tot interactie met actieve wegenkaarten en actief papier. Bij deze applicatie dient een papieren oppervlak of grote monitor als basis. Indien de



Figuur 2.2: Chameleon prototype [13]

gebruiker de palmtop dichterbij of verder van het scherm beweegt, of de oriëntatie van de palmtop verandert, worden er verschillende soorten informatie weergegeven.

Stel dat er een landkaart aan een muur hangt. Om te beginnen selecteert de gebruiker een regio door de palmtop bij de regio langs de kaart te houden. Vervolgens is het mogelijk bijkomende informatie over die regio op te vragen door de oriëntatie van de palmtop te veranderen.

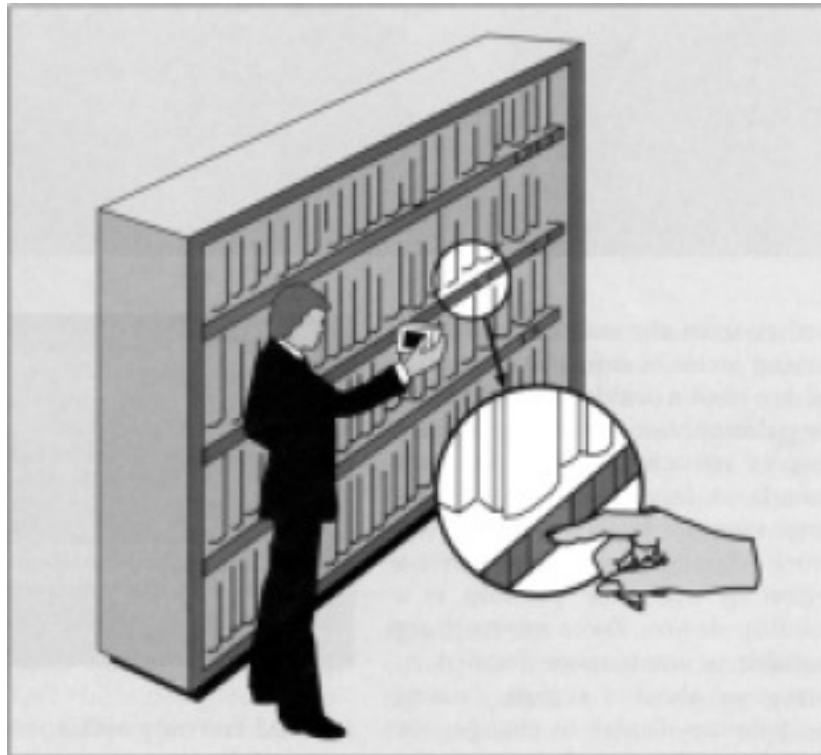
Een zelfde concept is mogelijk indien het om gewone papieren documenten gaat. Stel dat een papieren document gevoelige informatie bevat. Deze informatie zou dan niet in gewone inkt weergegeven worden, maar met behulp van elektronische inkt. Als de gebruiker met een palmtop over deze delen van het papier beweegt, zal hij die informatie op de palmtop te zien krijgen, indien hij hier toegang tot heeft.

De tweede applicatie is een “Computer-Augmented” bibliotheek. Dit soort bibliotheek is een grote verbetering ten opzichte van een gewone bibliotheek.

Stel dat we boeken en informatie over een specifiek onderwerp willen uitlenen. In een traditionele bibliotheek moeten daarvoor twee verschillende acties uitgevoerd worden:

1. Zoeken in een kaartenbak of een elektronische database naar boeken die aan onze wensen voldoen.
2. Achterhalen waar deze boeken zich bevinden in de bibliotheek en zoeken in de boekenrekken.

In een computer-augmented bibliotheek zijn deze acties gecombineerd in één procedure. De elektronische database van informatie is gesitueerd rond de boekenrekken. De boeken en de rekken zenden navigatie en semantische informatie uit, die de gebruiker naar de juiste plaats in de bibliotheek leidt. Onder elk boek in het rek is een touch sensitive LCD strip bevestigd (figuur 2.3), welk oplicht als het boek voldoet aan de gestelde eisen van de gebruiker. Om ons verder te helpen bij het zoeken, zal het aanraken van een boek, of de LCD strip eronder, ervoor zorgen dat een zoekopdracht uitgevoerd wordt naar boeken die gelijkaardige informatie



Figuur 2.3: Computer-Augmented Bibliotheek [13]

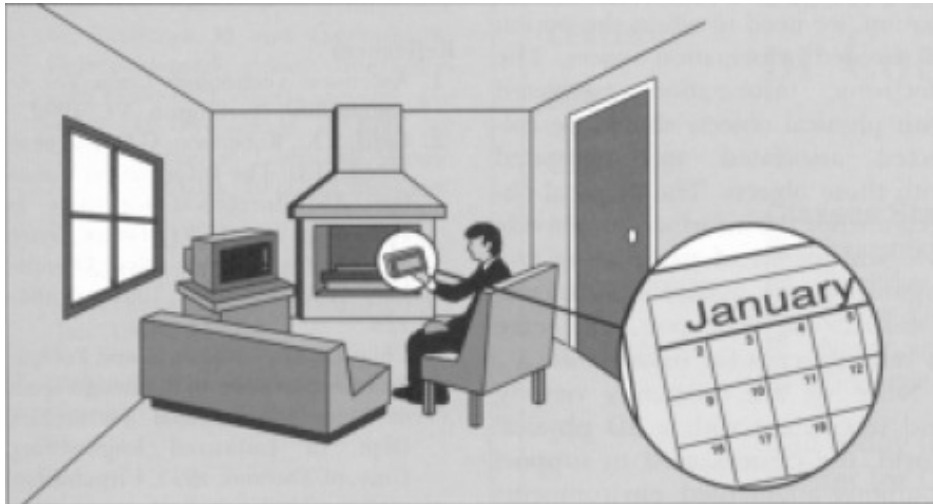
bevatten. De Chameleon zorgt ervoor dat de gebruikerservaring nog verbeterd wordt door informatie over geselecteerde boeken (inhoudsopgave, samenvatting, ...) weer te geven.

Tenslotte is er een applicatie waarbij de Chameleon gebruikt wordt om van op afstand toegang te geven tot een fysieke omgeving. Het voorgestelde ontwerp laat gebruikers toe elektronische notities te bevestigen aan eender welk object in de fysieke omgeving.

Om de fysieke omgeving te kunnen weergeven op de palmtop, wordt een 360° panoramisch beeld van het kantoor gemaakt. Dit wordt gedaan met behulp van een camera die bevestigd is in het midden van het plafond. Dit beeld wordt opgeslagen op de palmtop zodat de gebruiker de mogelijkheid heeft om door de inhoud van zijn kantooromgeving te navigeren terwijl hij niet fysiek aanwezig is in zijn kantoor.

Om toegang tot de videobeelden te vereenvoudigen, wordt gebruik gemaakt van "spatial mapping". Aangezien gebruikers over het algemeen vertrouwd zijn met hun kantooromgeving, krijgen ze toegang tot bepaalde delen van het videobeeld door de palmtop te richten op de bedoelde objecten in de fysieke omgeving.

Stel dat in een kantoor het bureau in het midden van de kamer staat, de boekenkast tegen de rechtermuur, een schetsbord aan de linkermuur en een kalender recht tegenover het bureau. Om de inhoud van de kalender te bekijken van op een andere plaats dan in het kantoor, moet de gebruiker zich voorstellen dat hij in de bureaustoel zit, en vervolgens de palmtop gewoon omhoog bewegen omdat de kalender recht tegenover hem hangt (figuur 2.4). Om de boekenkast te zien moet hij de palmtop 90° naar rechts draaien.



Figuur 2.4: Van thuis uit de inhoud van uw kantoor bekijken [13]

2.3 Peephole Metafoor voor Augmented Reality

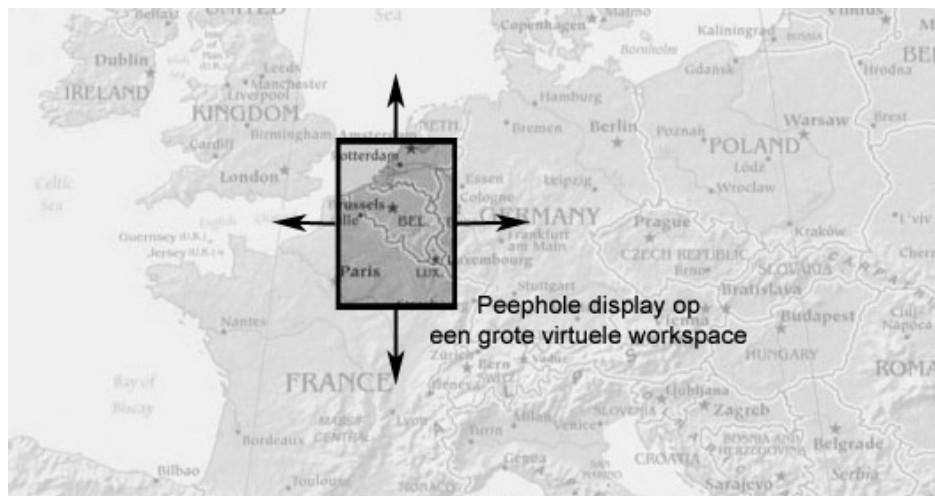
In deze sectie wordt een peephole metafoor uitgelegd voor “Augmented Reality” (AR) [9]. Dit zal de werking van peephole displays in deel 2.4 verduidelijken. Er wordt een “Instrumented Environment” gebruikt om de peephole metafoor uit te leggen en te testen. Een Instrumented Environment is een omgeving, in dit geval een kamer, waarin toestellen geïntegreerd zijn. Deze toestellen realiseren de verbinding tussen de fysieke wereld, waarin ze gesitueerd zijn, en de virtuele wereld, waarmee ze verbonden zijn. Input en output worden gerealiseerd met behulp van een projector en een camera die in het midden van het plafond bevestigd zijn en kunnen ronddraaien. Verder bevinden zich ook nog enkele touch-screens in de kamer.

2.3.1 Peephole Metafoor

Om vervorming van de geprojecteerde beelden te voorkomen bij de output van de projector, wordt een 3D model van de kamer gebruikt. De beelden worden texture-mapped op het 3D model en de virtuele camera krijgt dezelfde positie en oriëntatie in de virtuele wereld als de projector in de fysieke wereld. Wanneer de projector gericht wordt naar een bepaald deel van de kamer, zal deze de informatie projecteren die zich op dezelfde positie in het virtuele 3D model bevindt. De projector maakt de informatie die bij een bepaalde positie in de ruimte hoort dus zichtbaar wanneer hij ook daadwerkelijk op die positie gericht is. Informatie van andere posities bestaat ondertussen nog steeds in het virtuele model, maar is niet zichtbaar in de kamer. De projector verschaft dus een peephole in de virtuele ruimte.

Theoretisch is het mogelijk om de positie van de projector continu gelijk te stellen aan de plaats waar de gebruiker naar kijkt. Dit maakt alle informatie zichtbaar voor de gebruiker door er simpelweg naar te kijken. In de praktijk loopt dit echter niet zo vlot door de vertraging tussen het tracken van de bewegingen van de gebruiker en de reactie van de projector.

Op een gelijkaardige manier als bij output, wordt naar de input gekeken door een peephole. De projector die aan het plafond bevestigd is en voor de output zorgt, bevat ook een camera, die naar dezelfde plaats “kijkt”. Deze camera zorgt voor een continue low resolution video



Figuur 2.5: Peephole display over een grotere virtuele workspace

stream, of wanneer gewenst voor een high resolution foto. De camera kan gebruikt worden voor input omdat hiermee objecten herkend en gevolgd kunnen worden doorheen de ruimte. Logischerwijs is deze input enkel beschikbaar op de positie waar output is, en zorgt zo voor een interactief gebied in de kamer door tegelijk input en output op die plaats aan te bieden.

2.3.2 Voordelen

De peephole metafoor heeft enkele voordelen ten opzichte van andere benaderingen tot Augmented Reality [9]. Eén van de andere benaderingen is het aanbrengen van elektronisch behangpapier zodat de volledige informatie ruimte tegelijk getoond kan worden in plaats van enkel een peephole. Op het eerste zicht lijkt deze manier veel beter en vlotter te werken dan met peephole displays. Maar er zijn situaties waarin peephole displays toch de voorkeur genieten. Bijvoorbeeld in een snel veranderende informatie ruimte. In dat geval zou de gebruiker continu afgeleid worden door veranderingen die hij in zijn ooghoeken waarneemt, wat een zeer chaotische indruk geeft. Dit probleem hebben we niet bij het gebruik van peephole displays omdat de gebruiker enkel de veranderingen waarneemt in het gebied waar op dat moment zijn aandacht op gericht is.

2.4 Peephole Displays

Het verschil tussen een peephole display en een gewoon scherm is dat bij het peephole display de bewegingen van het scherm in de fysieke wereld gebruikt worden om te navigeren in de virtuele wereld. Het lijkt alsof de virtuele workspace achter of onder het peephole display vast ligt en het scherm fungeert als een venster (peephole) op die workspace (figuur 2.5). In de volgende secties (2.4.1 en 2.4.2) zullen enkele voorbeelden van reeds bestaande implementaties kort uitgelegd worden.



Figuur 2.6: Navigeren met de Boom Chameleon [33]

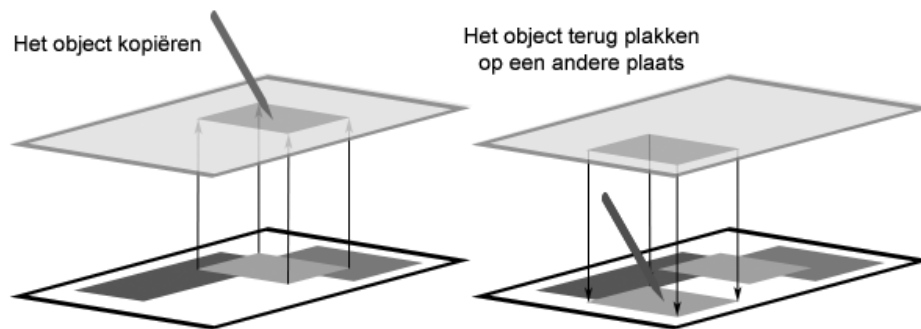
2.4.1 Boom Chameleon

De Boom Chameleon [33] is gebaseerd op het spatially aware display Chameleon dat in sectie 2.2.1 voorgesteld is. Waar de Chameleon nog een gewoon spatially aware display was, is de Boom Chameleon een echt peephole display. De Boom Chameleon is bedoeld voor ontwikkelaars van wagens. Een 3D model van een auto kan ermee bekeken worden vanuit verschillende posities, door het scherm rond het model te bewegen alsof de auto echt in de kamer staat (figuur 2.6). Het scherm is gemonteerd op een mechanische Boom. Door een tegengewicht te gebruiken kan het scherm zeer eenvoudig rond het model bewogen worden en zelfs “zweven” in de lucht. Een groot voordeel van de Boom boven andere tracking methoden is dat er zo goed als geen vertraging optreedt tussen het bewegen van het scherm en de reactie van het programma op deze beweging, en dat het tracken van de bewegingen zeer nauwkeurig is. Een nadeel is dat het scherm niet alle mogelijke posities kan aannemen. Het is bijvoorbeeld niet mogelijk om het model langs onder of van bovenaf te bekijken zonder het model zelf te roteren.

De Boom Chameleon is getest door een aantal gebruikers. Uit deze testen blijkt dat de peephole metafoor zeer snel aan te leren is en een eenvoudige techniek is om te navigeren in een 3D ruimte. Als nadeel wordt aangehaald dat sommige manieren van navigatie (bijvoorbeeld vliegen) niet mogelijk zijn en dat speciale hardware nodig is.

2.4.2 Pen Interactie op Peephole Displays

Het laatste voorbeeld dat hier gegeven wordt, komt van Ka-Ping Yee [37]. In deze paper worden gewone handheld displays (PocketPC) als peephole displays gebruikt. De reden om een handheld te tracken in de ruimte en als peephole display te gebruiken, is dat de schermruimte van een dergelijke handheld bijna altijd te klein is voor de informatie die ermee bekeken wordt (figuur 2.5). Er zijn verschillende applicaties ontwikkeld die gebruik maken van peephole dis-



Figuur 2.7: Object kopiëren en plakken in het tekenprogramma

plays, zowel 2D als 3D applicaties. Het grote verschil tussen beide is dat bij de 2D applicaties enkel horizontaal en verticaal gescrolled kan worden en dat de 3D applicaties semantisch zoomen ondersteunen. De 3D applicaties zullen hier in het kort toegelicht worden.

De eerste applicatie is de “Peephole Zoom Viewer”. Dit programma laat gebruikers toe om, door het scherm te bewegen, verschillende delen van een grote figuur te bekijken. Door het scherm op te heffen of te laten zakken, wordt er in -of uitgezoomd op de figuur.

Een tweede applicatie is de peephole kalender. Door het scherm op en neer te bewegen, wordt gewisseld tussen de verschillende weergaven van de kalender. Op de hoogste positie ziet de gebruiker een overzicht van alle maanden. Door het scherm horizontaal of verticaal te bewegen, kan naar de gewenste maand gezocht worden. Indien de gebruiker de gewenste maand gevonden heeft, kan hij deze maand meer in detail bekijken door het scherm omlaag te bewegen (semantisch zoomen). In deze stand ziet hij alle dagen van de maand en kan zo de afspraken die op een bepaalde dag gepland staan, bekijken of aanpassen. Het zoeken naar de gewenste dag in de huidige maand gebeurt door het scherm in de verticale of horizontale richting te bewegen.

Een laatste applicatie die gebruik maakt van de peephole metafoor is een tekenprogramma. Een gevolg van het gebruik van de peephole metafoor is dat de gebruiker tijdens het tekenen niet meer binnen de grenzen van het scherm moet blijven. Tijdens het tekenen kan het scherm rondbewogen worden om te tekenen op delen van het canvas die eerder niet zichtbaar waren. Het unieke aan dit programma is de manier waarop objecten gekopieerd en/of geplakt kunnen worden. Als het scherm omhoog beweegt, wordt het klembord van de applicatie zichtbaar. Men kan een object kopiëren door het vast te nemen met de pen en tegelijk het scherm omhoog te bewegen tot het klembord zichtbaar wordt. Om dit object vervolgens op het canvas te plakken, wordt het object opnieuw met de pen geselecteerd in het klembord en beweegt men tegelijk het scherm terug omlaag (figuur 2.7).

Handheld peephole displays kunnen ook gebruikt worden om een “Personal Information Space” [13, 34, 37] te creëren. In dit geval bestaat de information space uit verschillende programma’s die gesitueerd zijn rond (voor) de gebruiker. Bijvoorbeeld links van de gebruiker de kalender en rechts het tekenprogramma. Bepaalde gebeurtenissen in de kalender kunnen gekoppeld worden aan een tekening die gemaakt is met het tekenprogramma. Ka-Ping bekommt

zo een personal information space door de 3D tracking hardware in een rugzak te dragen. Op deze manier kent de peephole display ten allen tijde zijn positie ten opzichte van de gebruiker, hetgeen nodig is voor een personal information space.

Er zijn enkele gebruikerstesten uitgevoerd voor de 2D applicaties [37]. Daaruit blijkt dat de taken meestal sneller volbracht kunnen worden met peephole displays dan met een normale interface. Ook wordt duidelijk dat de meeste gebruikers liever met de peephole display werken dan met een gewone PocketPC.

2.5 Two Handed Input

Een belangrijk voordeel van handheld peephole displays [37] is dat de gebruiker beide handen gebruikt om ermee te werken. De niet-dominante hand wordt gebruikt om het handheld display te bewegen doorheen de informatie ruimte en met de dominante hand worden de weergegeven data gemanipuleerd.

Uit verschillende studies blijkt dat de niet-dominante hand zeer geschikt is voor het niet-precieze werk (scrollen of navigeren) terwijl de dominante hand het nauwkeurige werk uitvoert [10, 20, 21]. Als de taak die de gebruiker moet uitvoeren op deze manier opgedeeld kan worden tussen beide handen, leidt dit tot snellere en betere resultaten.

Als de gebruiker met twee handen aan een taak werkt, zal hij steeds weten waar zijn handen zich bevinden, zelfs zonder ernaar te kijken. De gebruiker is zich altijd bewust van de positie van beide handen ten opzichte van elkaar [1]. Dit zorgt ervoor dat het voor hem gemakkelijk en natuurlijk overkomt om beide handen te laten samenwerken aan één taak. Het is logisch dat de dominante hand gekozen wordt om de meer precieze en nauwkeurige taken uit te voeren. Uit onderzoek blijkt namelijk dat de dominante hand de positie van de niet-dominante hand altijd gebruikt als referentie [15]. Dus wanneer de gebruiker navigeert door het handheld display met de niet-dominante hand te bewegen, zal de dominante hand bijna automatisch volgen.

2.6 Conclusie

In dit hoofdstuk is uitgelegd wat peephole displays zijn en hoe deze werken.

Daarvoor werd om te beginnen uitgelegd wat spatially aware displays zijn. Dit zijn (draagbare) computers die informatie verzamelen over hun omgeving. Deze computers zijn er zich bijvoorbeeld van bewust als ze zich in de buurt van een belangrijk object bevinden en reageren hierop. Er werden enkele applicaties voorgesteld die hiervoor ontwikkeld zijn.

Vervolgens werd een peephole metafoor voor augmented reality uitgelegd omdat deze metafoor duidelijk aantoont wat er bedoeld wordt met “peepholes”. Peepholes laten een deel zien van een workspace die te groot is om volledig te bekijken. In dit geval werd dit niet gedaan omdat het onmogelijk is om de workspace volledig te bekijken (zoals bij PDA's), maar omdat deze manier van werken ook nog andere voordelen biedt, bijvoorbeeld meer privacy en een rustigere werkomgeving.

Tenslotte werden handheld peephole displays voorgesteld. Dit is het soort peephole displays dat gebruikt zal worden voor deze thesis. Deze handheld peephole displays zijn spatially aware displays die gebruik maken van de peephole metafoor.

Uit gebruikerstesten blijkt dat de peephole metafoor zeer snel aan te leren is en dat het de performantie van bepaalde taken gevoelig kan verhogen. De subjectieve tevredenheid van

de gebruiker is hoger dan bij dezelfde applicaties op normale computers. Het feit dat beide handen gebruikt worden op een natuurlijke manier heeft een positieve invloed op de resultaten en de gebruikerservaring.

Peephole displays bieden vele mogelijkheden en zullen waarschijnlijk de manier waarop we in de toekomst met handheld computers werken compleet veranderen en drastisch verbeteren.

Hoofdstuk 3

3D Tracking

3.1 Inleiding

Om peephole displays te kunnen implementeren moet het display uiteraard zijn positie kennen. De enige manier om deze informatie te bekomen is met behulp van een 3D tracking systeem.

Er bestaan veel verschillende soorten 3D tracking systemen die allen andere fysische principes gebruiken om de positie en/of oriëntatie van een object te bepalen. In sectie 3.2 zal uitgelegd worden aan welke vereisten een goed 3D tracking systeem moet voldoen. Sectie 3.3 zal vervolgens verschillende bestaande technieken en systemen voorstellen en kort verduidelijken. Tenslotte zal in sectie 3.4 de V-scope besproken worden, het systeem dat gebruikt wordt voor deze thesis.

3.2 Voorwaarden

In deze sectie worden de voorwaarden waaraan een goed 3D tracking systeem moet voldoen op een rijtje gezet. Geen enkel van de bestaande tracking systemen voldoet aan al deze voorwaarden, daarom moet er altijd een afweging gemaakt worden welk systeem het meest geschikt is voor de toepassing die ermee ontwikkeld wordt.

- **Nauwkeurigheid:** Geeft de fout aan tussen de positie en oriëntatie gemeten door het 3D tracking systeem en de echte positie en oriëntatie [5, 28]. Het is belangrijk dat deze fout zo klein mogelijk is (translatie $\leq 0,5$ cm en rotatie $\leq 1^\circ$ [14]).
- **Resolutie:** De kleinste hoeveelheid beweging die door het 3D tracking systeem gedetecteerd kan worden [5, 28]. Een beweging kleiner dan deze hoeveelheid zal niet weergegeven worden in de output van het systeem. Een kleine resolutie is nodig wanneer het systeem kleine bewegingen moet kunnen detecteren.
- **Update snelheid:** De hoeveelheid metingen per seconde [5, 28]. Wanneer de positie van de tracker uitgetekend zou worden, resulteert een lage update snelheid in schokkende bewegingen, ook al is de echte beweging vloeiend. Hoe hoger de update snelheid, hoe beter de resultaten.
- **Vertraging (Lag):** De vertraging tussen het moment waarop de beweging van de tracker plaatsvindt en het moment waarop het 3D tracking systeem deze beweging

doorgeeft aan de computer [5, 28]. Deze vertraging moet uiteraard zo klein mogelijk zijn (zeker ≤ 50 ms [14]).

- **Werkingsbereik:** Geeft het bereik aan waarin het systeem de vooropgestelde nauwkeurigheid behaalt [5, 28]. Buiten dit bereik zullen de trackers onnauwkeurig of zelfs helemaal niet getracked worden. Een groter bereik is uiteraard beter, maar ook hier hangt het van toepassing tot toepassing af of dit een belangrijk punt is.
- **Storing:** Bijna alle tracking systemen zijn op één of andere manier gevoelig voor storing door omgevingsfactoren (line-of-sight, fel licht, metalen voorwerpen, ...) [28]. Dit kan de nauwkeurigheid verminderen en zelfs voor grote fouten zorgen.
- **Absoluut vs Relatief:** Het systeem kan de absolute positie en oriëntatie melden of telkens de relatieve verandering ten opzichte van de vorige positie en oriëntatie [28]. De relatieve manier is gevoelig voor accumulatieve fouten en moet dus regelmatig juist gezet worden op een absolute positie.
- **Grootte/Gewicht:** De grootte en het gewicht van de tracker speelt ook een rol [28]. Een kleine, lichte tracker is eenvoudiger te bevestigen aan een object of een lichaamsdeel dan een grote, zware tracker.
- **Aantal Vrijheidsgraden:** Sommige tracking systemen meten enkel de positie of enkel de oriëntatie van de tracker in plaats van beide [28].
- **Veiligheid:** Houdt het gebruik van het systeem een gezondheidsrisico in omwille van bijvoorbeeld straling [28]?

Een goed 3D tracking systeem zou dus een hoge nauwkeurigheid, fijne resolutie en hoge update snelheid moeten hebben. De vertraging moet klein zijn en het bereik groot. In het beste geval zou het systeem in elke omgeving moeten kunnen werken. Bovendien zouden de trackers klein en licht moeten zijn zodat ze gemakkelijk aan iets of iemand bevestigd kunnen worden.

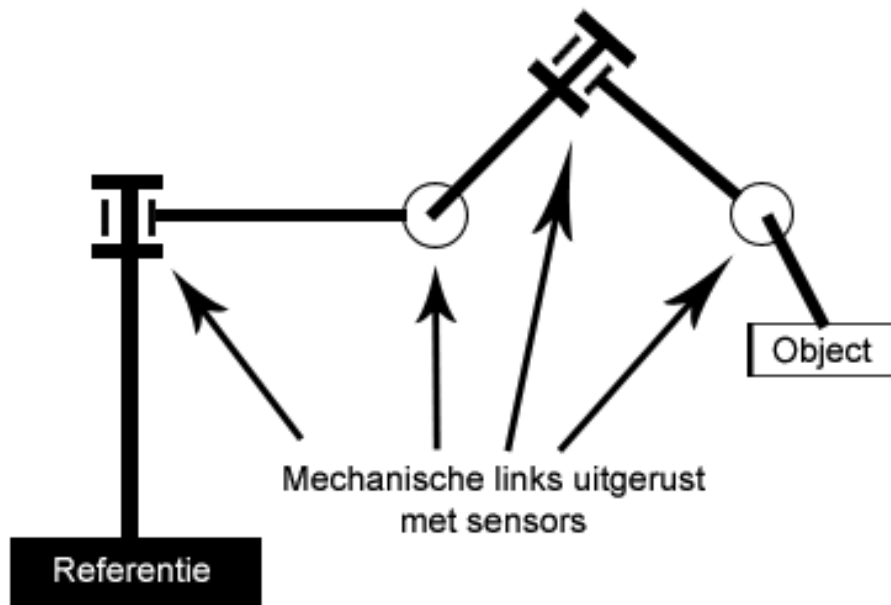
3.3 3D Tracking Systemen

De bestaande 3D tracking systemen kunnen onderverdeeld worden in 4 groepen. De onderverdeling gebeurt aan de hand van fysische principes die gebruikt worden voor het tracken. Deze 4 groepen zijn: Elektromechanische tracking (sectie 3.3.1), elektromagnetische tracking (sectie 3.3.2), optische tracking (sectie 3.3.4) en akoestische tracking (sectie 3.3.3) [5, 14, 28, 30, 32, 35]. Voor elk van deze groepen zal kort de werking uitgelegd worden. Verder zullen ook telkens de voor- en nadelen van die groep vermeld worden.

3.3.1 Elektromechanische Tracking

De meest voor de hand liggende groep tracking methoden is elektromechanische tracking.

Een eerste soort van elektromechanische tracking is met behulp van een mechanische arm [5, 14, 28, 30, 32, 35] (figuur 3.1). Hierbij wordt de positie van het uiteinde van de mechanische arm gemeten ten opzichte van een referentiepunt. De mechanische arm bestaat uit verschillende scharnieren waardoor de gebruiker in meerdere richtingen kan bewegen. De



Figuur 3.1: Een mechanische arm als 3D Tracking systeem

positie van het uiteinde van de arm kan bepaald worden omdat de lengte van de “links” tussen de scharnieren en het referentiepunt (startpunt van de arm) gekend zijn en door de hoek van elk scharnier te meten.

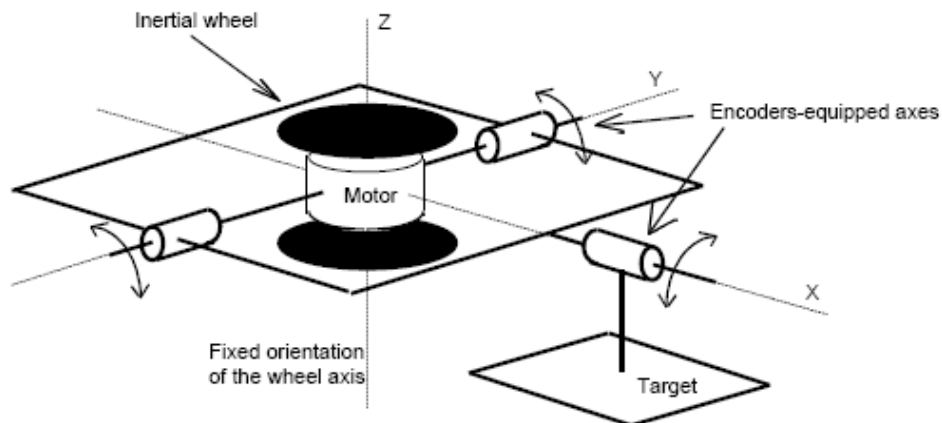
Een tweede soort van elektromechanische tracking is gyroscopische tracking [14, 30] (figuur 3.2). De oriëntatie van het object wordt bepaald door de draaiing van de gyroscopen te meten (minstens 2 gyroscopen zijn nodig om de oriëntatie te vinden). Een gyroscopisch tracking systeem is enkel geschikt om de oriëntatie van een object te meten, niet de positie.

De voordelen van elektromechanische tracking systemen zijn:

- Zeer nauwkeurige tracking
- Zeer weinig vertraging (≤ 5 ms)
- Geen storing door elementen uit de omgeving
- Kan gebruikt worden voor force feedback (enkel mechanische arm)
- Kan dienen als tegengewicht voor een beeldscherm (enkel mechanische arm)

De nadelen van elektromechanische tracking systemen zijn:

- Beperkte bewegingsvrijheid en een relatief klein bereik
- Moeilijk om meerdere objecten in de omgeving tegelijk te tracken
- Kan zwaar en moeilijk te bewegen zijn
- Problemen met drift, moet regelmatig opnieuw gekalibreerd worden (enkel bij gyroscopische tracking)



Figuur 3.2: Een gyroscopisch tracking systeem [30]

3.3.2 Elektromagnetische Tracking

De volgende, veel gebruikte, manier van tracking is elektromagnetische tracking (figuur 3.3). Deze groep is opgedeeld in 2 soorten: Alternating Current (AC) en Direct Current (DC) [5, 14, 30].

AC Elektromagnetische Tracking

AC trackers bestaan uit een zender en een sensor. De zender bestaat uit 3 magnetische spoelen en bevindt zich op een vaste plaats in de omgeving. De sensor bestaat eveneens uit 3 spoelen en is bevestigd aan het te tracken object.

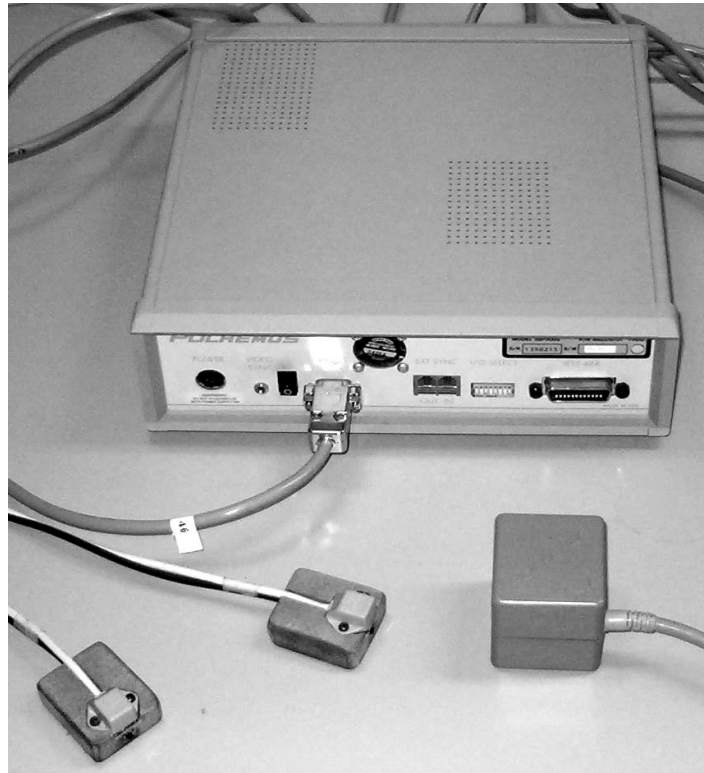
Vervolgens wordt er in de zender een wisselend elektromagnetisch veld opgewekt door er wisselstroom op aan te sluiten. Dit veld zorgt dat er een stroom ontstaat in de 3 spoelen van de sensor. Deze stroom hangt af van de positie en oriëntatie van de sensor ten opzichte van de zender. Door de stroom in de sensor te meten is het mogelijk de positie en oriëntatie van de sensor ten opzichte van de zender te bepalen.

In de nabijheid van metalen objecten hebben AC elektromagnetische trackers last van storingen doordat de metalen voorwerpen ook magnetische velden gaan ontwikkelen die de magnetische velden van de zender verstoren. Dit zorgt voor onjuiste tracking resultaten.

DC Elektromagnetische Tracking

De storingen bij AC tracking ontstaan doordat er wisselende magnetische velden opgewekt worden. Dit probleem wordt vermeden bij DC trackers door een constant magnetisch veld te gebruiken. De zender en sensor bestaan eveneens uit 3 spoelen waarin beurtelings een constant magnetisch veld wordt opgewekt door korte stroompulsen te geven. Dit magnetisch veld wordt vervolgens gemeten door de sensor en na aftrekken van het magnetische veld van de aarde gebruikt om de positie en oriëntatie van de sensor te berekenen.

DC trackers zijn niet gevoelig voor de storingen waar AC trackers last van hebben, maar zijn nog steeds gevoelig voor ferromagnetische materialen. Wanneer deze in de buurt zijn verstoren ze het magnetische veld door reflectie. Daardoor zijn de resultaten in de buurt van zulke materialen niet correct.



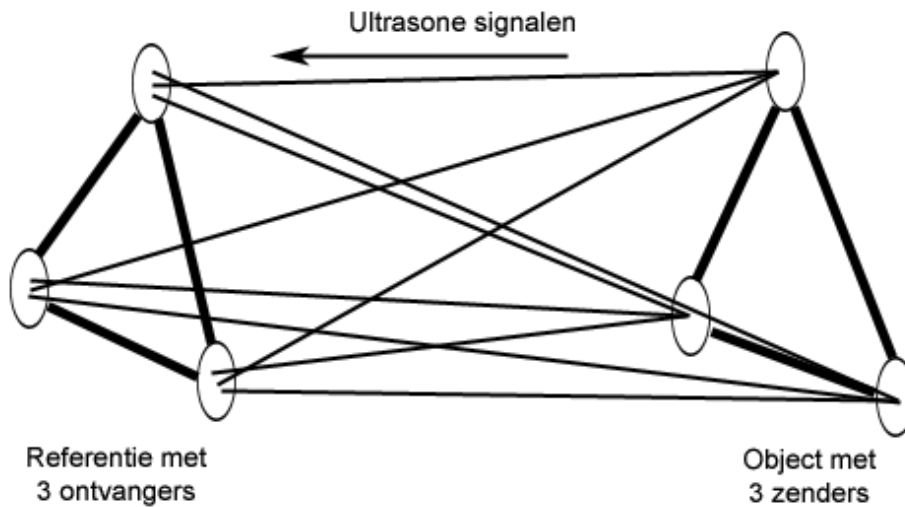
Figuur 3.3: Polhemus Fastrack Systeem

De voordelen van elektromagnetische tracking systemen zijn:

- Geen line-of-sight problemen. Er kan zelfs nauwkeurig getracked worden wanneer er zich een object tussen de zender en de sensor bevindt (indien dit geen metalen of ferromagnetisch object is)
- Positie en oriëntatie worden bepaald
- De tracking is gevoelig en redelijk nauwkeurig
- Kleine en lichte sensoren die eenvoudig aan objecten of personen bevestigd kunnen worden
- Er kunnen meerdere sensoren tegelijk getracked worden (de update snelheid per sensor daalt dan wel)

De nadelen van elektromagnetische tracking systemen zijn:

- Metalen en ferromagnetische objecten veroorzaken storingen
- De sensoren hangen aan een kabel
- Het bereik waarin de trackers goed werken kan niet te groot gemaakt worden omdat de magnetische straling dan gezondheidsrisico's voor de gebruikers kan inhouden.



Figuur 3.4: Schematische voorstelling van Time-of-Flight Tracking

3.3.3 Akoestische Tracking

Akoestische tracking maakt gebruik van ultrasone (> 20 kHz) geluidsgolven om de positie van een tracker te bepalen. Er bestaan 2 technieken die gebruik maken van ultrasoon geluid: time-of-flight tracking en phase-coherent tracking [5, 14, 28, 30, 32, 35]. Beide technieken worden kort uitgelegd.

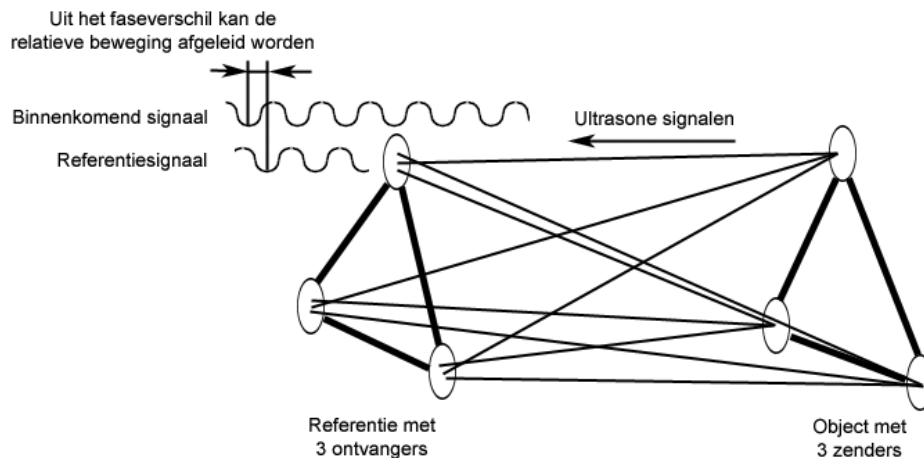
Time-of-Flight

Time-of-flight is een techniek die gebruik maakt van de snelheid van het geluid [5, 14, 28, 30, 32, 35]. Omdat geluidsgolven zich relatief traag (t.o.v. licht) voortbewegen en deze snelheid gekend is, kan uit de tijd die een geluidssignaal nodig heeft om zich van punt A naar punt B te verplaatsen, de afstand tussen deze 2 punten bepaald worden.

Het systeem bestaat uit 3 zenders en 3 ontvangers. De 3 zenders worden bevestigd aan het te tracken object en de 3 ontvangers worden op gekende posities in de omgeving geplaatst. Vervolgens sturen de 3 zenders om beurt een ultrasoon signaal uit. Elk van deze signalen wordt door de 3 ontvangers gedetecteerd (figuur 3.4). Omdat de tijd verstreken tussen het verzenden en het ontvangen van een signaal gekend is, kan de afstand tussen de zender en elk van de 3 ontvangers hieruit afgeleid worden. Vervolgens kan met behulp van triangulatie de exacte positie van de zender ten opzichte van de ontvangers berekend worden. Als op deze manier de posities van de 3 zenders gekend zijn, kan hieruit de positie en oriëntatie van het object afgeleid worden.

Phase-coherent

In tegenstelling tot de time-of-flight techniek wordt niet de snelheid van het geluid gebruikt, maar de fase van het ultrasone signaal [5, 14, 28, 30, 32, 35]. Er worden 3 zenders aan het te tracken object bevestigd en 3 ontvangers op vaste posities in de omgeving geplaatst. De 3 zenders sturen om beurt een ultrasoon signaal. Wanneer de ontvangers dit signaal detecteren, wordt de fase van dit signaal vergeleken met de fase van een referentiesignaal (figuur 3.5).



Figuur 3.5: Schematische voorstelling van Phase-coherent Tracking

Uit het verschil tussen beide fases kan de beweging van de zender afgeleid worden. Door deze beweging te voegen bij de vorige gekende positie wordt de nieuwe positie berekend. De initiële positie van de zenders moet door een externe bron berekend worden. Wanneer de posities van alle 3 de zenders gekend zijn, wordt hieruit de positie en oriëntatie van het object afgeleid.

De voordelen van akoestische tracking systemen zijn:

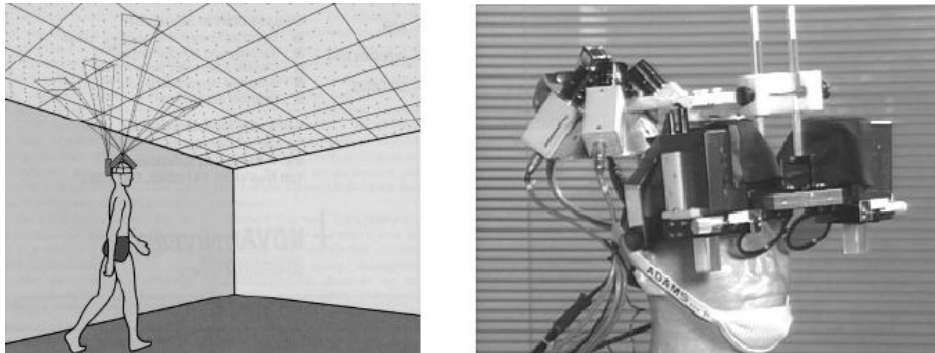
- De trackers zijn klein en licht. Daardoor zijn ze gemakkelijk te bevestigen aan objecten of personen die getracked moeten worden
- Kleine objecten in de line-of-sight geven geen problemen

De nadelen van akoestische tracking systemen zijn:

- Grote objecten in de line-of-sight zorgen voor foutieve resultaten
- Externe omgevingsfactoren kunnen voor storingen zorgen (ultrasonische geluiden, temperatuurschommelingen, ...)
- Niet draadloos
- Door de trage geluidssnelheid is de update rate beperkt (bij time-of-flight)
- Phase-coherent systemen zijn gevoelig voor cumulatieve fouten

3.3.4 Optische Tracking

Voor optische tracking wordt gebruik gemaakt van optische referentiepunten (LED's, lampjes, ...) die door sensoren (camera's, ...) gedetecteerd kunnen worden. De verschillende systemen worden opgedeeld in 2 groepen naargelang de plaatsing van de referentiepunten en de sensoren. Bij "inside-out" systemen bevinden de referentiepunten zich op vaste posities in de omgeving en de sensoren op het te tracken object. Bij "outside-in" systemen daarentegen



Figuur 3.6: Opto-electronic Ceiling Tracker [30]

bevinden de referentiepunten zich op het te tracken object en zijn de sensoren op vaste posities in de omgeving geplaatst [5, 14, 28, 30, 32, 35]. Beide technieken worden uitgelegd met behulp van een voorbeeld. Een derde speciale groep is “natural environment tracking”.

Inside-out Systemen

Als voorbeeld van een inside-out systeem wordt de “opto-electronic ceiling tracker” [5] gebruikt (figuur 3.6). Dit systeem is ontwikkeld aan de universiteit van North Carolina.

Als referentiepunten worden LED's gebruikt. Een grote hoeveelheid van deze LED's zijn op vaste posities aan het plafond van een kamer bevestigd. Op de helm staan 4 sensoren waarmee het plafond voortdurend “gescand” wordt. Zowel de LED's als de helm zijn verbonden met een computer.

De LED's in het gezichtsveld van de sensoren op de helm worden vervolgens in een bepaald patroon opgelicht door de computer. Doordat de sensoren dit patroon op het plafond kunnen herkennen, kan de positie en oriëntatie van de persoon (of het object waaraan de sensoren bevestigd zijn) bepaald worden.

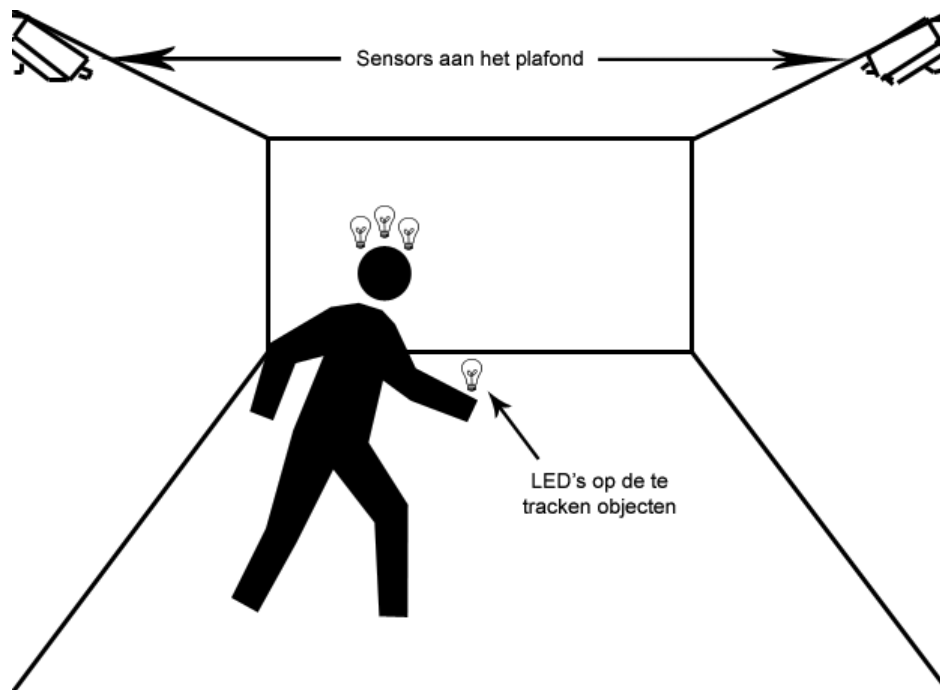
Outside-in Systemen

Het “Honeywell LED array” systeem [5] is een voorbeeld van een outside-in systeem. Er wordt gebruik gemaakt van 4 infrarode LED's die in een bepaald patroon op het te tracken object geplaatst zijn. De camera's zijn bevestigd op vaste plaatsen in de kamer en de LED's worden één voor één opgelicht (figuur 3.7). Vervolgens is het mogelijk om de positie en oriëntatie van dit object te detecteren door de camera's naar dit patroon te laten zoeken.

Natural Environment Trackers

Bij natural environment trackers worden geen referentie-bakens gebruikt [28]. Er worden sensoren (camera's) aan het te tracken object bevestigd en het systeem verkrijgt de nodige informatie door opeenvolgende beelden van de omgeving te vergelijken. Op deze manier kunnen de bewegingen van het object berekend worden.

De voordelen van optische tracking systemen zijn:



Figuur 3.7: Outside-in tracking

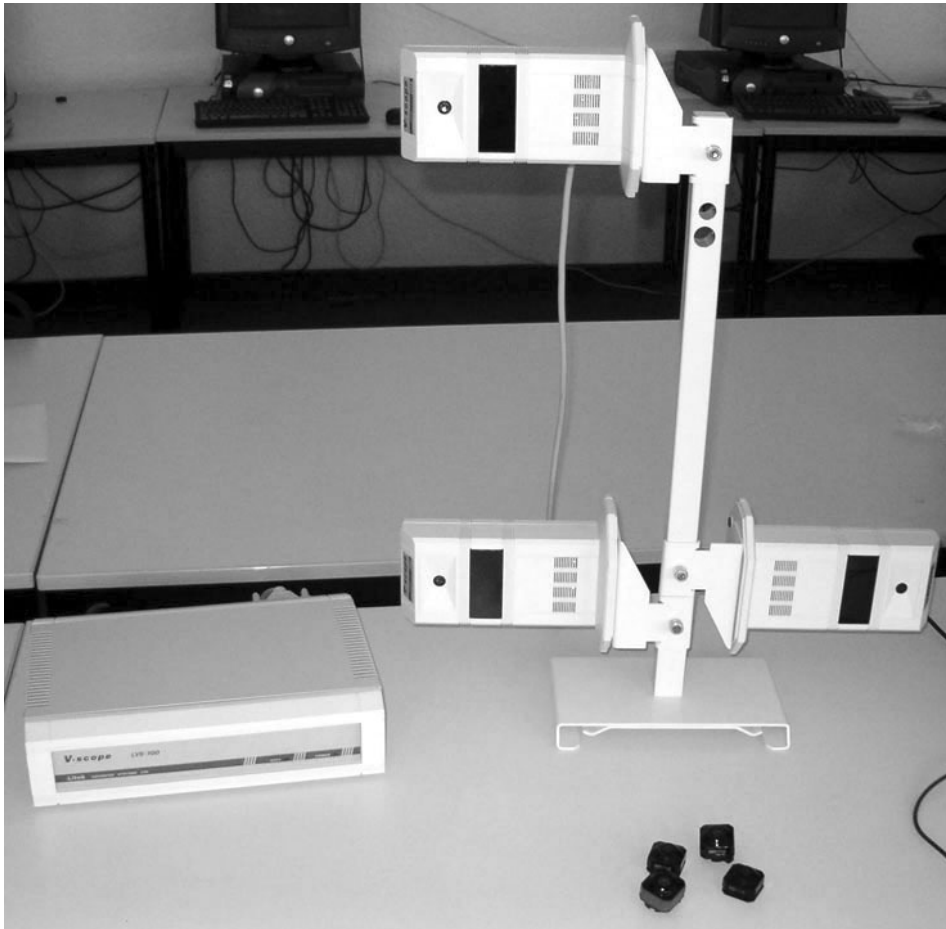
- Hoge update snelheid, enkel gelimiteerd door de snelheid van de processor gebruikt voor de berekeningen
- Het bereik is afhankelijk van het systeem, maar kan zeer groot zijn
- Heeft geen last van magnetische of akoestische signalen

De nadelen van optische tracking systemen zijn:

- Line-of-sight problemen
- Niet accuraat als de hoek tussen de sensor en de zender groot is
- Storing door andere lichtbronnen
- Moeilijk om meerdere objecten in dezelfde ruimte te tracken

3.4 V-scope

Het 3D tracking systeem dat gebruikt zal worden voor de implementatie is de V-scope van Lipman Electronic Engineering [23, 24]. Er zal kort uitgelegd worden hoe de V-scope werkt (sectie 3.4.1) en wat de voor- en nadelen ervan zijn (sectie 3.4.2 en 3.4.3).

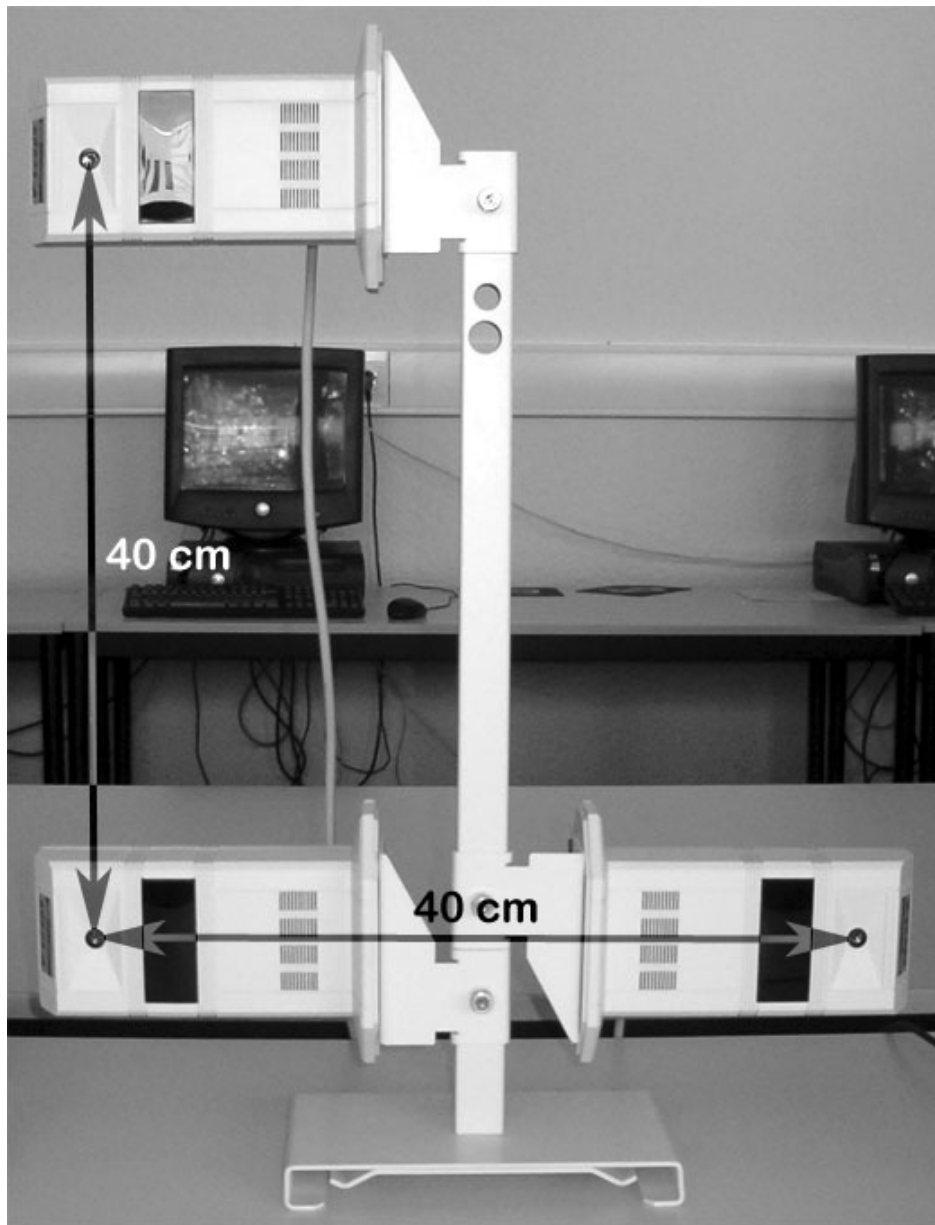


Figuur 3.8: V-scope System: Microprocessor, towers mount en vier buttons

3.4.1 Werking

De V-scope (figuur 3.8) is een akoestisch time-of-flight systeem (sectie 3.3.3). Het systeem bestaat uit een microprocessor, een “towers mount” (figuur 3.9) en 4 buttons (figuur 3.10). De towers mount kan gebruikt worden om de 3 towers aan te bevestigen zodat hun onderlinge afstand constant en gekend is. Elke tower kan optische infrarode signalen verzenden en akoestische ultrasone signalen ontvangen. De buttons bevatten een sensor voor infrarode signalen, een zender voor ultrasone geluiden en een batterijtje.

Om de positie van één van de buttons te tracken, zenden de 3 towers tegelijk een infrarood signaal uit. In dit infrarood signaal zit gecodeerd welke button getracked wordt zodat het V-scope systeem tot 4 verschillende buttons kan tracken (er bestaat ook een meer geavanceerde variant die acht buttons aankan). Indien meer dan één button gebruikt wordt, zal de V-scope alle gebruikte buttons sequentieel tracken. Het plastic waaruit de buttons vervaardigd zijn, laat infrarode signalen door zodat de sensoren binnen in de buttons bereikt kunnen worden. Zodra een button “zijn” infrarood signaal ontvangt, zendt deze button een ultrasoon geluidssignaal (40 kHz) terug. Op het moment dat de 3 towers dit ultrasone signaal ontvangen, kan berekend worden hoe ver de button van elke tower verwijderd is (figuur 3.11). Om deze afstand te berekenen, moet de V-scope wel weten hoe lang het ultrasone signaal er over gedaan



Figuur 3.9: De V-scope Towers Mount met de drie towers eraan



Figuur 3.10: De vier V-scope buttons

heeft om van de button tot de tower te geraken en hoe snel geluid zich voortbeweegt.

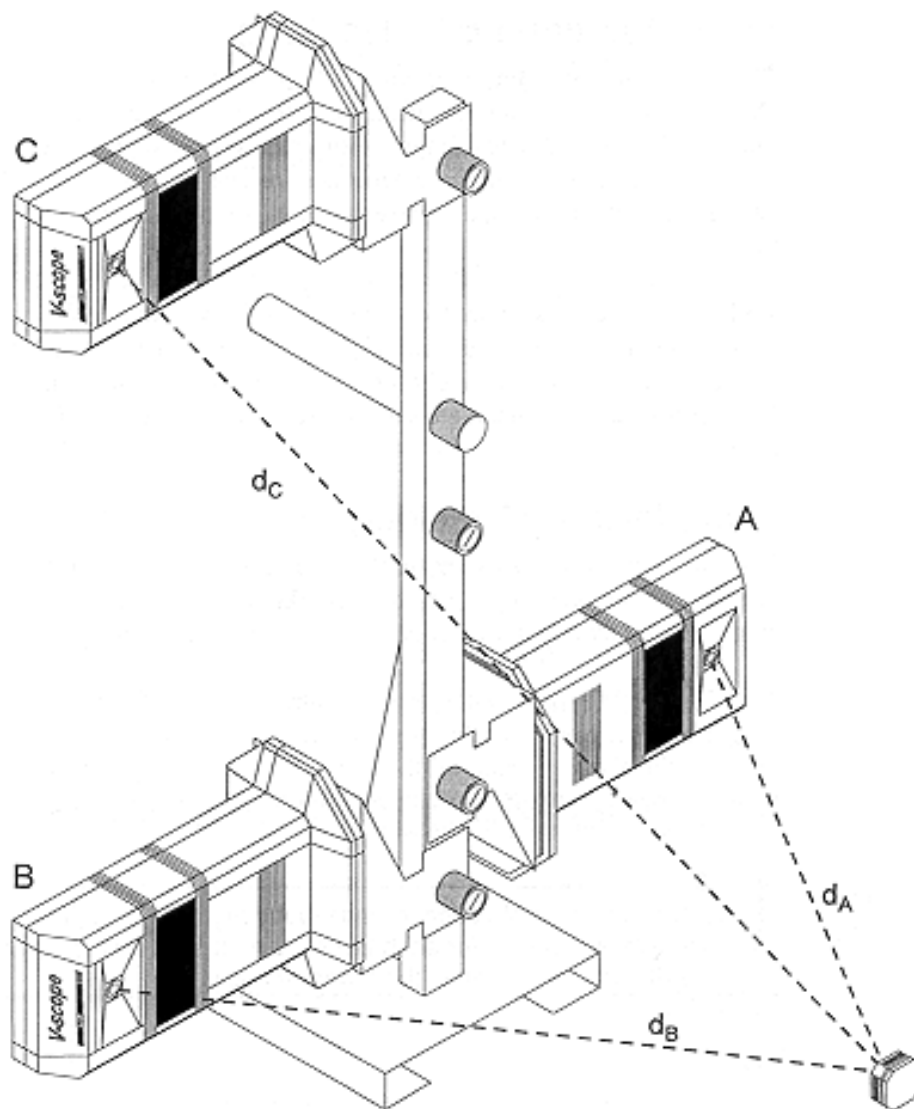
De tijd nodig om van de button tot de tower te geraken is gelijk aan de tijd die verstreken is tussen het uitsturen van het infrarode signaal en het ontvangen van het ultrasone signaal. Deze “benadering” kan gebruikt worden omdat de tijd die het infrarode licht nodig heeft om van de tower tot de button te geraken extreem kort en dus verwaarloosbaar is.

Ook de snelheid van het geluid is bekend, het enige probleem is dat deze snelheid lichtjes varieert met de temperatuur van de lucht waardoor het geluid zich voort beweegt. Om deze reden zijn de V-scope towers uitgerust met een thermometer zodat de exacte geluidssnelheid afgeleid kan worden.

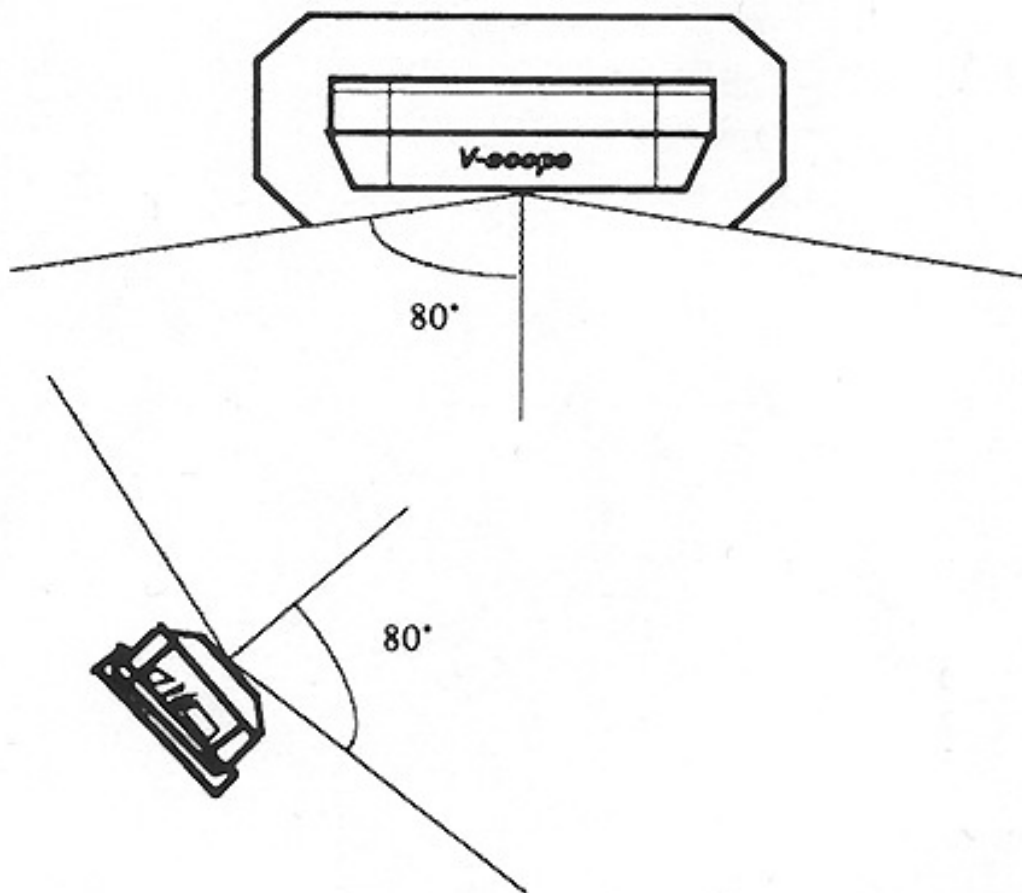
Als de 3 afstanden van de button tot elke tower bekend zijn (d_A , d_B en d_C in figuur 3.11), kan met behulp van triangulatie berekend worden wat de positie van de button is. Om deze techniek te kunnen gebruiken, moeten de 3 towers op een vaste afstand van elkaar staan. Dit is het geval als ze aan de towers mount bevestigd zijn. De onderlinge afstand tussen de sensoren is dan 40 cm (figuur 3.9)

3.4.2 Voordelen

- Een voordeel dat de V-scope heeft ten opzichte van de eerder besproken time-of-flight systemen (sectie 3.3.3) is het feit dat de V-scope draadloos werkt. Dit is mogelijk omdat er gebruik gemaakt wordt van infrarode signalen om de buttons te activeren.
- Er moet geen directe line-of-sight zijn tussen de 3 towers en de button. Omdat alle towers hetzelfde infrarode signaal tegelijk verzenden, is het voldoende als er een directe line-of-sight is tussen één van de towers en de button. Voor het ultrasone signaal is het voldoende dat er zich geen grote objecten tussen de button en de towers bevinden. Kleine objecten in de line-of-sight geven geen problemen.



Figuur 3.11: De V-scope in werking met één button [24]



Figuur 3.12: Maximale angle of sight van tower en button [23]

- Het bereik van de V-scope gaat van 10 centimeter tot 5 meter.
- De buttons wegen amper 12,5 gram en zijn klein. Het is dus eenvoudig om de buttons aan een object te bevestigen.
- Het tracken van de positie van de button kan zeer nauwkeurig gebeuren. Zeker indien er gebruik gemaakt wordt van de “Phase-lock” feature. In dit geval wordt ook nog rekening gehouden met de fase van het geluidssignaal. Helaas kan deze feature enkel gebruikt worden als de hoek tussen de as van de button en de as van de towers ongeveer gelijk blijft aan 0° of 90° . Deze feature is niet bruikbaar voor mijn implementatie omdat de hoek tussen de buttons en de towers dikwijls wijzigt (wanneer de PDA gekanteld wordt). Met phase lock kan een nauwkeurigheid van 0,1 mm behaald worden, zonder phase lock een nauwkeurigheid van ongeveer 2 mm.

3.4.3 Nadelen

- Enkel de positie van een button wordt getracked, niet de oriëntatie. Wanneer de oriëntatie van een object nodig is, moeten er 3 buttons aan dat object bevestigd worden. Doordat de positie van deze 3 buttons gekend is, kan hieruit de oriëntatie van het object

berekend worden.

- Indien meerdere buttons getracked moeten worden, zal de update rate dalen. Dit omdat de buttons niet allemaal tegelijk getracked kunnen worden en de V-scope elke button om beurt moet tracken. Daarom is de effectieve update rate per button gelijk aan de opgegeven update rate gedeeld door het aantal gebruikte buttons.
- Wanneer de update rate zeer snel ingesteld wordt (10 tot 15 ms), vermindert het bereik tot 2,5 à 3 meter. Dit omdat de ultrasone signalen ongeveer 10 ms nodig hebben om een afstand van 3 meter af te leggen.
- Het V-scope systeem kan storingen ondervinden indien er zich sterke ultrasone geluidsbronnen (elektrische apparaten) of sterke (infrarode) lichtbronnen in de omgeving bevinden (direct zonlicht).
- Elke button moet binnen een hoek van $\pm 80^\circ$ van de as van elke tower blijven. Elke tower moet zich eveneens binnen een hoek van $\pm 80^\circ$ van de as van elke button bevinden (figuur 3.12).
- Wanneer de buttons zich te kort bij de towers bevinden, treden er fouten op bij de triangulatie. Daarom is de minimale afstand tussen de towers mount en de buttons best rond de 30 centimeter.

3.5 Conclusie

Dit hoofdstuk ging over verschillende soorten 3D tracking systemen die bestaan.

Om te beginnen werd uitgelegd wat de voorwaarden zijn waaraan een goed tracking systeem moet voldoen. Er bestaat geen enkel systeem dat aan al die voorwaarden voldoet en daarom moet bij de keuze van het tracking systeem altijd rekening gehouden worden met de toepassing waarvoor het gebruikt zal worden. Afhankelijk van die toepassing kan beslist worden welk systeem het meest geschikt is.

Vervolgens werden de 4 grote groepen van tracking systemen voorgesteld: elektromechanische tracking, elektromagnetische tracking, optische tracking en akoestische tracking. Voor elke groep werd kort uitgelegd hoe deze systemen werken en wat hun voor- en nadelen zijn. Voor verschillende groepen zijn voorbeelden van bestaande tracking systemen gegeven.

Het 3D tracking systeem dat gebruikt wordt voor deze thesis moet aan enkele voorwaarden voldoen. Vooreerst is het belangrijk dat het systeem draadloos werkt. Indien dit niet het geval is, vermindert dit de gebruiksvriendelijkheid van de peephole displays. Het bereik van het systeem mag niet te klein zijn en het moet de mogelijkheid bieden om meerdere objecten tegelijk te tracken. Dit is belangrijk voor het collaboratieve deel van deze thesis. Het tracken van de positie van het display is het hoofddoel, maar het kan in de toekomst handig zijn als ook de oriëntatie van het display gekend is. Dit biedt mogelijkheden voor meer geavanceerde peephole applicaties. Tenslotte is het een voordeel als de trackers klein en licht zijn zodat ze gemakkelijk aan een PDA bevestigd kunnen worden.

De V-scope voldoet aan bijna elk van deze vereisten. De buttons werken draadloos, het bereik kan oplopen tot 5 meter, er kunnen tot 4 objecten tegelijk getracked worden en de trackers wegen amper 12,5 gram. Het enige probleem dat zich stelt met de V-scope is het feit dat de oriëntatie van de buttons niet getracked wordt, enkel de positie. De oplossing die

hiervoor gebruikt kan worden is om 3 buttons aan de PDA te bevestigen en vervolgens uit deze 3 posities zelf de oriëntatie van de PDA berekenen. Aangezien dit kleine probleem niet opweegt tegen de voordelen die het gebruik van de V-scope met zich meebrengt, is er gekozen om de V-scope te gebruiken als tracking systeem voor deze thesis.

Hoofdstuk 4

Zoomable User Interfaces

4.1 Inleiding

De user interface is één van de belangrijkste onderdelen van een applicatie. Deze bepaalt namelijk de manier waarop de gebruiker het programma waarneemt en dus ook de gebruiksvriendelijkheid van het programma. Er bestaan verschillende soorten user interfaces met elk hun eigen karakteristieken.

De eerste belangrijke karakteristiek van zoomable user interfaces is de manier waarop de informatie georganiseerd is in ruimte en schaal [17]. De meest voorkomende manier van zoomen is de geometrische zoom. In dit geval wijzigt enkel de grootte van het object wanneer er gezoomd wordt. Er kan echter ook een complexere relatie bestaan tussen de objecten en de schaal. Dit wordt semantisch zoomen genoemd, een techniek waarbij de manier waarop de informatie wordt weergegeven afhankelijk is van het huidige zoomlevel. Een derde manier van zoomen is “constant density zooming”, waarbij de manier waarop de objecten getoond worden aan de gebruiker afhangt van het aantal objecten dat zichtbaar is. Zo kan ervoor gezorgd worden dat er altijd een zelfde aantal objecten zichtbaar zijn voor de gebruiker.

Een tweede belangrijke karakteristiek van zoomable user interfaces is dat de informatieruimte direct zichtbaar en manipuleerbaar is met behulp van “panning” en “zooming” [17]. Panning verandert het gebied van de informatieruimte dat zichtbaar is en zooming wijzigt de schaal waarop de informatieruimte bekeken wordt.

In sectie 4.2 zullen kort enkele toepassingen besproken worden die werken met een zoomable user interface en in sectie 4.3 worden enkele toolkits besproken die gebruikt kunnen worden om zoomable user interfaces te ontwikkelen. Tenslotte wordt in sectie 4.4 de vergelijking gemaakt tussen zoomable user interfaces en overview+detail interfaces.

4.2 Toepassingen

Hoewel zoomable user interfaces (zui) nog niet veel gebruikt worden voor commerciële producten, zijn er reeds enkele applicaties te vinden waaruit blijkt dat zui’s handig zijn voor sommige toepassingen (sectie 4.2.1 en 4.2.2). Maar zoals in sectie 4.2.2 zal blijken zijn er echter ook toepassingen waarvoor zui’s minder geschikt zijn en traditionele grafische user interfaces (gui) nog steeds de voorkeur genieten.



Figuur 4.1: Screenshot van PhotoMesa met meer dan 500 afbeeldingen [2]

4.2.1 PhotoMesa

De eerste toepassing is PhotoMesa [2]. PhotoMesa is een programma, ontwikkeld om grote hoeveelheden afbeeldingen te doorzoeken en te bekijken op een gebruiksvriendelijke en eenvoudige manier. PhotoMesa is geprogrammeerd in Java met behulp van het Jazz Framework (sectie 4.3.2).

Om het programma te gebruiken worden één of meerdere mappen met afbeeldingen geopend. Deze afbeeldingen worden vervolgens allemaal naast elkaar weergegeven zodat de schermruimte optimaal benut wordt (figuur 4.1). Wanneer de gebruiker op een groep klikt, wordt ingezoomd op de geselecteerde map (figuur 4.2). Vervolgens kan de gebruiker een van de afbeeldingen in deze map op normale grootte bekijken door deze aan te klikken. De overgang tussen verschillende weergaven gebeurt telkens door middel van zoomen. De gebruiker kan steeds terugkeren naar het vorige zoomlevel door met de rechtermuisknop te klikken.

Het handige aan de traditionele imageviewer is dat er door de thumbnails gezocht kan worden door in één dimensie te scrollen (verticaal), terwijl bij PhotoMesa in twee dimensies genavigeerd moet worden. Dit is meestal moeilijker voor de gebruiker. PhotoMesa heeft als voordeel dat de gebruiker gemakkelijk een overzicht krijgt door uit te zoomen. Door deze manier van interactie kan de gebruiker zelf bepalen wat hij het belangrijkste vindt, het aantal getoonde afbeeldingen of de resolutie van de getoonde afbeeldingen. Dit verschil is een direct gevolg van het zoom aspect van PhotoMesa. Als een verticale grid van afbeeldingen (traditionele viewer) uitgezoomd zou worden, zou de schermruimte aan beide zijden van de lijst grotendeels ongebruikt blijven. Dus het is duidelijk dat zoomable user interfaces bij tweedimensionale weergave van data beter gebruik maken van de beschikbare schermruimte dan bij eendimensionale weergave van de data.



Figuur 4.2: Screenshot van PhotoMesa, ingezoomd op een enkele map [2]

Verder in deze thesis (sectie 5.3.3) wordt een vergelijking gemaakt tussen PhotoMesa en een zelf ontwikkelde peephole applicatie om afbeeldingen te bekijken.

4.2.2 Automatisch Zoomen

In deze sectie wordt het principe van automatisch zoomen [19] voorgesteld. Indien er door de gebruiker door de data gescrolled wordt, zal er automatisch uitgezoomd worden. Het zoomlevel wordt berekend met behulp van de scrollsnelheid. Hoe sneller er gescrolled wordt, hoe meer er uitgezoomd wordt. Dit zorgt ervoor dat de door de gebruiker waargenomen snelheid, waarmee de data over het scherm bewegen, constant blijft.

Dit principe is toegepast op verschillende soorten applicaties. Al deze applicaties hebben als doel dat de gebruiker grote hoeveelheden data kan bekijken of doorzoeken. Het verschil zit in het soort data.

Web Browser

Een logische toepassing om dit principe te testen is een web browser. Bestaande scrolling technieken werken niet goed wanneer lange documenten doorzocht moeten worden. Het gebruik van scroll bars is te onnauwkeurig en scrollen met het muiswielje gaat meestal te traag.

Automatisch zoomen tijdens het scrollen lost deze problemen op [19]. Het laat de gebruiker toe om zeer snel te scrollen doorheen het document zonder gedesoriënteerd te worden. Om ervoor te zorgen dat de gebruiker ten allen tijde weet in welk deel van het document hij zich bevindt, worden de belangrijke titels en afbeeldingen groter weergegeven naarmate er meer uitgezoomd wordt (semantisch zoomen).

Map Viewer

De tweede applicatie, die uitgebreid werd met automatisch zoomen, is een map viewer. Het grote verschil met de web browser is dat de gebruiker bij deze toepassing in twee dimensies kan navigeren. Ook hier wordt semantisch zoomen toegepast. Bij het uitzoomen verdwijnen de kleine, minder belangrijke objecten van de kaart en worden de grote snelwegen en belangrijke geografische herkenningspunten benadrukt. Op deze manier heeft de gebruiker zelfs wanneer de weergave volledig uitgezoomd is een goed idee van zijn positie.

Image Browser

Deze toepassing laat de gebruiker toe om een collectie afbeeldingen te bekijken en te doorzoeken. Alle afbeeldingen worden horizontaal naast elkaar geplaatst en wanneer de gebruiker door de lijst navigeert wordt er uitgezoomd. Bij deze applicatie is het automatisch zoomen minder effectief dan bij de vorige twee. Dit komt doordat semantisch zoomen hier onmogelijk is. Het programma weet niet welke delen van een afbeelding belangrijk zijn en dus benadrukt moeten worden. Daardoor wordt het voor de gebruiker moeilijk om te bepalen welke afbeeldingen getoond zijn wanneer er fel uitgezoomd is.

Woordenboek

Deze toepassing is een speciaal geval. Hier wordt geen grafische, maar symbolische informatie weergegeven. Het “zoomen” wordt in dit geval bekomen door de woordenlijst uit te dunnen naarmate sneller gescrolled wordt. De testresultaten van deze toepassing zijn niet veelbelovend. Het is verwarrend dat er steeds woorden verdwijnen en tevoorschijn komen en het vergt meer inspanning van de gebruiker om te bepalen of het gezochte woord zich alfabetisch tussen twee andere zichtbare woorden bevindt.

Voor de web browser en map viewer zijn gebruikerstesten uitgevoerd [19]. Hieruit blijkt dat de tijd nodig om een bepaalde opdracht te volbrengen ongeveer gelijk is voor de programma's met automatisch zoomen en zonder automatisch zoomen. Wel wordt vermeld dat de meeste gebruikers de methode met automatisch zoomen aangenamer vonden om mee te werken.

4.3 Toolkits

Er bestaan veel toolkits om grafische user interfaces te ontwikkelen en elk van deze toolkits bevat een uitgebreide verzameling aan widgets. Deze toolkits zijn handig wanneer de gebruiker een standaard applicatie ontwikkelt waarin gebruik gemaakt wordt van de standaard widgets (scroll bars, knoppen, menu's, ...). Maar in de meeste van deze toolkits ontbreekt de mogelijkheid om zoomable user interfaces te maken. Eén van de unieke eisen die zoomable user interfaces aan de toolkit stelt, is de mogelijkheid tot “semantisch zoomen”. Semantisch zoomen betekent dat de manier waarop de data weergegeven worden afhankelijk is van het zoomlevel.

Eerst zal in sectie 4.3.1 uitgelegd worden wat het verschil is tussen monolitische en polylitische toolkits. Vervolgens zal een polylitische (sectie 4.3.2) en een monolitische (sectie 4.3.3) toolkit voorgesteld worden die beiden geschikt zijn om zoomable user interfaces mee te ontwikkelen.

4.3.1 Monolitisch vs Polylitisch

Monolitische toolkits bestaan uit enkele zeer grote klassen die alle belangrijke functionaliteiten bevatten die door de applicaties gebruikt zullen worden [3].

Om hergebruik van code te bevorderen, plaatsen de toolkit ontwikkelaars dikwijls grote hoeveelheden algemeen bruikbare code in de belangrijkste klassen. Die code wordt dan overgeërfd door alle onderliggende klassen in de toolkit. Dit leidt tot het probleem van een grote, moeilijk te begrijpen, “top-level” klasse. Daarenboven zijn programmeurs verplicht de functionaliteiten van de top-level klasse te aanvaarden, ook al is deze niet nodig in hun eigen widget.

Polylitische toolkits zijn anders opgebouwd. Deze bestaan uit veel kleine klassen die allen een deel van de functionaliteiten bevatten [3].

Het voordeel van deze werkwijze is dat hergebruik van code en aanpassen van bestaande widgets eenvoudiger is. De functionaliteiten van een widget worden in dit geval bepaald “at runtime” door verschillende klassen te combineren in plaats van klassen af te leiden zoals bij monolitische toolkits.

Een nadeel van deze werkwijze is dat een groot programma zeer ingewikkeld en moeilijk te onderhouden wordt. Dit komt doordat het aantal klassen dat gecombineerd moet worden voor de nodige functionaliteiten groot is.

4.3.2 Jazz

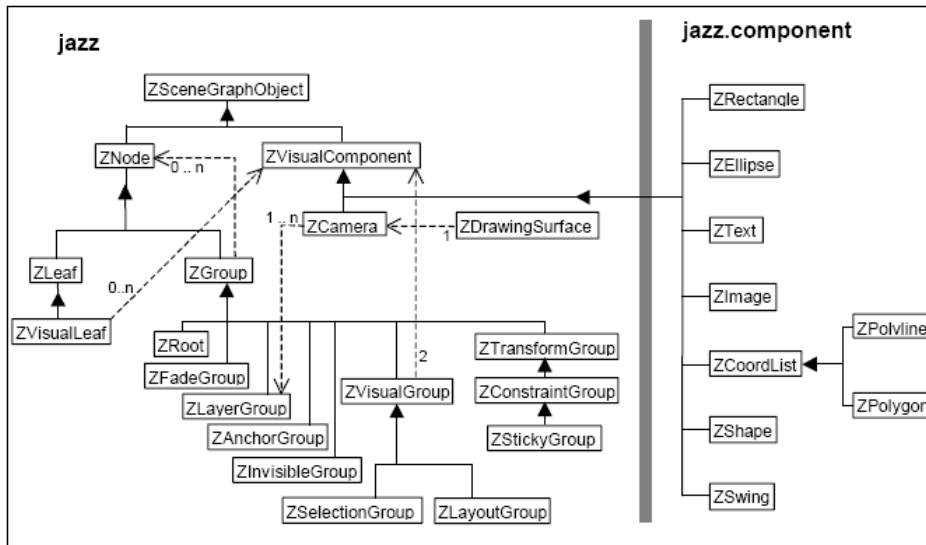
Jazz [3, 4] is een polylitische toolkit in Java om zoomable user interfaces te programmeren. Het unieke aan Jazz is de techniek die gebruikt wordt om de user interface op te bouwen. Hiervoor worden “scene-graphs” gebruikt.

Dit is een datastructuur die vooral gebruikt wordt bij 3D toepassingen. De scene-graph wordt opgebouwd uit drie soorten objecten: nodes, visuele componenten en camera’s. Door deze objecten op een bepaalde manier in een scene-graph te plaatsen kan de gebruiker de functionaliteiten en het uitzicht van de applicatie bepalen. De scene-graph bestaat uit een hiërarchie van nodes die relaties tussen objecten voorstellen. Deze nodes hebben geen visuele representatie op het scherm, daarvoor zijn er de visuele componenten die geometrie en kleuren informatie bevatten. De nodes bepalen waar iets zich bevindt in de scene-graph en de visuele componenten bepalen hoe het eruit ziet. Figuur 4.3 toont de verschillende objecten van Jazz.

Elke node kan slechts één ouder hebben en de functionaliteiten van een knoop in de boom worden telkens doorgegeven op al zijn kinderen. De nodes zorgen voor alle affine transformaties die toegepast kunnen worden op de visuele componenten (translatie, rotatie, schaleren, ...). De visuele componenten kunnen wel meerdere ouders hebben waardoor de grafische representatie gedeeld kan worden door verschillende delen van de scene-graph.

Het derde soort objecten dat aan een scene-graph toegevoegd kan worden zijn camera’s. Een camera is in feite ook een visuele component die een bepaald zicht op de scene-graph weergeeft. Er kunnen verschillende camera’s aan de graaf toegevoegd worden die allen hun eigen zicht op de graaf geven.

Het voordeel van een scene-graph is dat het voor de gebruiker veel gemakkelijker wordt om applicatiespecifieke widgets te ontwikkelen. Bij een traditionele toolkit moet de gebruiker hiervoor zijn widget afleiden van reeds bestaande widgets en de nodige functies overschrijven om zo de gewenste functionaliteiten toe te voegen. Bij een scene-graph daarentegen worden



Figuur 4.3: De objecten van de Jazz toolkit [4]

nieuwe functionaliteiten gecreëerd door objecten toe te voegen aan de graaf. Dit kan zelfs dynamisch gebeuren terwijl het programma draait.

4.3.3 Piccolo

Piccolo is gebaseerd op Jazz, maar is in tegenstelling tot Jazz een monolitische toolkit [3, 27]. Bij Piccolo is gekozen voor een monolithisch design omdat het onderhouden van en werken met de verschillende node types van Jazz een te grote last is voor de programmeur.

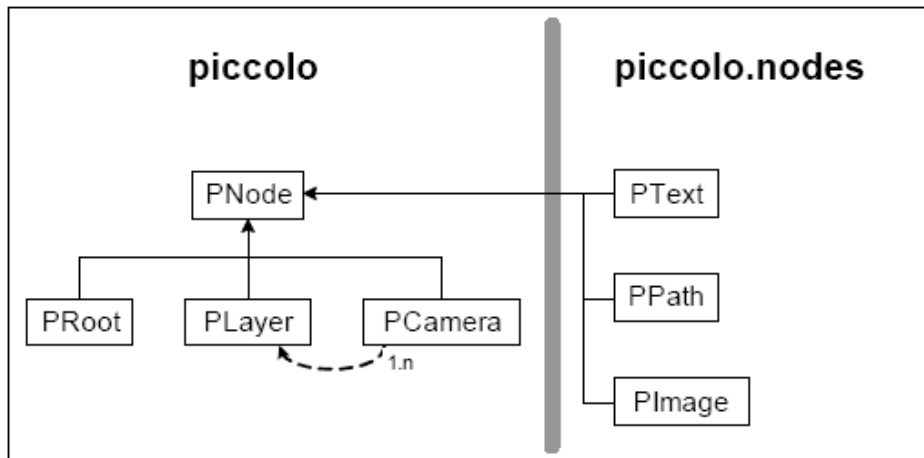
Doordat de belangrijkste functionaliteiten bij Piccolo in de basisklasse “PNode” zitten en elke node ook een visuele representatie kan hebben, is het aantal objecten ten opzichte van Jazz ongeveer gehalveerd (figuur 4.3 en figuur 4.4).

Net zoals bij Jazz wordt bij Piccolo gebruik gemaakt van een scene-graph om de interface op te bouwen. Qua functionaliteiten scoren beide toolkits even goed. Piccolo bevat net als Jazz ondersteuning voor semantisch zoomen, affiene transformaties, layers, camera’s, . . .

Er is een vergelijking gemaakt tussen Jazz en Piccolo qua snelheid, aantal lijnen code en geheugengebruik [3]. De conclusie is dat de Piccolo programma’s meestal korter en eenvoudiger zijn dan de Jazz programma’s. Voor snelheid en geheugengebruik doen de Piccolo varianten van de programma’s het even goed en meestal zelfs beter dan de Jazz programma’s.

4.4 Zoomable User Interfaces en Overview+detail Interfaces

In deze sectie wordt de vergelijking gemaakt tussen zoomable user interfaces met en zonder een extra overview window [17, 18]. In sectie 4.4.1 wordt uitgelegd wat overview+detail interfaces zijn en worden de voordelen die deze interfaces kunnen bieden op een rijtje gezet. In sectie 4.4.2 worden de resultaten van een vergelijkende studie tussen zuis en overview+detail interfaces getoond.



Figuur 4.4: De objecten van de Piccolo toolkit [3]

4.4.1 Overview+detail interfaces

Zoomable user interfaces met een overview window, ook wel overview+detail interfaces genoemd [17], zijn interfaces die meerdere (meestal 2) aanzichten op een informatieruimte presenteren aan de gebruiker. Eén van deze aanzichten geeft een deel van de informatie in detail weer (het detail window) en het andere aanzicht geeft een grof overzicht van de volledige informatieruimte weer (het overview window). Belangrijk aan zulke interfaces is dat het overview en detail window gekoppeld zijn. Een aanpassing of beweging in het detail window resulteert in dezelfde aanpassing in het overview window en omgekeerd. Deze interfaces hebben vier voordelen ten opzichte van gewone user interfaces zonder overview window.

1. De navigatie verloopt efficiënter omdat de gebruiker kan navigeren met behulp van het overview window in plaats van het detail window.
2. Het overview window helpt de gebruiker zijn huidige positie in de informatieruimte te bepalen.
3. Het overview window kan de gebruikers taakrelevante informatie geven.
4. Overview+detail interfaces geven de gebruiker een gevoel van controle.

Het grootste nadeel van overview+detail interfaces is dat het overview window kostbare schermruimte inneemt die anders gebruikt kan worden door het detail window. Dit maakt overview+detail interfaces minder geschikt voor toestellen met kleine schermen.

4.4.2 Gebruikerstesten

Uit testresultaten blijkt dat de subjectieve voorkeur van de meeste gebruikers naar overview+detail interfaces uitgaat [17, 18]. Dezelfde resultaten tonen echter ook aan dat alle taken sneller uitgevoerd worden met zoomable user interfaces. Er worden hiervoor drie verklaringen gegeven.

1. Het overview window leidt de gebruiker af van zijn huidige taak door continu de aandacht te trekken en daardoor de tijd nodig om de taak af te werken verhoogt.

2. Wisselen tussen het detail en overview window vraagt aandacht en tijd om de muis te bewegen.
3. Navigeren in het overview window kan moeilijk zijn, kleine bewegingen zijn bijna onmogelijk waardoor er nog extra navigatie in het detail window nodig is.

4.5 Conclusie

In dit hoofdstuk is uitgelegd wat zoomable user interfaces zijn en hoe deze ontwikkeld kunnen worden.

Uit reeds bestaande toepassingen [2, 19] blijkt dat zoomable user interfaces dikwijls een goed alternatief zijn voor gewone user interfaces. Dit vooral wanneer de gebruiker met het programma grote hoeveelheden data op een snelle en gebruiksvriendelijke manier moet kunnen bekijken en/of manipuleren. Door gebruikerstesten is het duidelijk dat ook de gebruiker het hiermee eens is en dikwijls liever met zoomable user interfaces werkt dan met gewone user interfaces [2, 19].

Vervolgens is aangetoond dat het voor de programmeur niet meer zo moeilijk is om zoomable user interfaces te ontwikkelen. Er bestaan immers verschillende toolkits speciaal ontworpen voor zui's [4, 3, 27] die de programmeur veel werk besparen.

Tenslotte werden gewone zui's vergeleken met overview+detail interfaces. Hieruit blijkt dat de meeste gebruikers overview+detail interfaces verkiezen boven gewone zui's. Aan de andere kant blijkt echter ook dat de taken met gewone zui's efficiënter en sneller uitgevoerd werden dan met overview+detail interfaces. Het is dus aan de programmeur om efficiëntie en subjectieve tevredenheid tegen elkaar af te wegen.

Voor deze thesis zijn zoomable user interfaces uiteraard zeer geschikt. Omdat een PDA een klein scherm heeft, is het belangrijk dat de gebruiker toch op een gebruiksvriendelijke manier door de grote hoeveelheden data kan navigeren. Het gebruik van overview+detail interfaces is niet mogelijk omdat het overview window teveel ruimte van het kleine PDA scherm zou innemen. Daarom zullen gewone zui's gebruikt worden. Het feit dat de Piccolo toolkit ook beschikbaar is voor het .NET Compact framework [25] zorgt ervoor dat de applicaties op de PDA's met behulp van Piccolo ontwikkeld kunnen worden.

Hoofdstuk 5

Implementatie

5.1 Inleiding

Naast de literatuurstudie is een implementatie van (collaboratieve) peephole displays een belangrijk deel van deze thesis. Voor deze implementatie worden verschillende soorten hardware gebruikt en zijn verschillende programma's gemaakt. Al deze hard -en software wordt in dit hoofdstuk voorgesteld.

Sectie 5.2 legt uit welke hardware voor de implementatie gebruikt werd en waarom. Vervolgens wordt in sectie 5.3 uitgelegd welke programma's gemaakt zijn en wat hun functie is. Een belangrijk aspect van de collaboratieve peephole displays is dat de applicaties volledig gedistribueerd zijn over de PDA's, er wordt dus geen aparte server gebruikt. Ook dit zal uitvoerig uitgelegd worden in sectie 5.3. Tenslotte bespreekt sectie 5.4 eventuele toekomstige toepassingen die mogelijk zijn met deze implementatie.

5.2 Hardware

5.2.1 V-Scope

De V-scope is het 3D tracking systeem dat gebruikt werd voor de implementatie. In sectie 3.4 is reeds uitgelegd hoe de V-scope werkt en zijn de voor -en nadelen ervan kort besproken.

Alvorens het mogelijk was om de V-scope te gebruiken, moest er wel nog een bug uit de software gehaald worden. Door deze bug werden de coördinaten die de V-scope naar de PC stuurde verkeerd "gelezen". Dit werd duidelijk wanneer de getrackte buttons uitgetekend werden op het scherm. Indien de buttons op bepaalde posities kwamen, versprong de uitgetekende waarde met meer dan een centimeter. Dit herhaalde zich voortdurend tijdens de beweging en telkens op dezelfde posities (wanneer de x, y of z coördinaten bepaalde waarden aannamen, telkens dezelfde waarden).

Het probleem bleek achteraf simpel op te lossen. Er werd namelijk een `char` gebruikt om de bytes die de V-scope doorstuurde in te lezen waardoor een overflow voorkwam wanneer de waarde van die byte hoger dan 128 was. Dan "versprong" de waarde in de `char` opeens naar -127, hetgeen het vreemde gedrag van de uitgetekende button verklaart (de ingelezen coördinaat wordt plots 255 kleiner). De oplossing was om simpelweg een `unsigned char` te gebruiken.

5.2.2 Bluetooth

Om de toepassingen gebruiksvriendelijk te houden is het belangrijk dat de bewegingsvrijheid van de gebruiker niet beperkt wordt door kabels. Daarom is gekozen voor draadloze verbindingen tussen de PDA's en de PC en tussen de PDA's onderling. Wanneer draadloze verbindingen tussen verschillende apparaten gelegd moeten worden, kunnen daarvoor verschillende technologieën gebruikt worden. Twee geschikte technologieën hiervoor zijn Wi-Fi [36] en bluetooth [7, 8].

De hoofdreden om voor bluetooth te kiezen was het feit dat de toepassing “overall” gebruikt moet kunnen worden. Voor wi-fi is een basisstation nodig en dit limiteert al dadelijk het aantal plaatsen waar de toepassing gebruikt zou kunnen worden. Voor bluetooth is geen basisstation nodig. Het is voldoende dat de apparaten die onderling willen verbinden, beschikken over een bluetooth radio.

De enige problemen met bluetooth zouden de bandbreedte en het bereik kunnen zijn. De bandbreedte ligt veel lager dan bij wi-fi. De bandbreedte van wi-fi varieert tussen de 11 en 54 Mbps (afhankelijk van het protocol) terwijl bluetooth “slechts” 723 kbps haalt. Voor de toepassingen die ermee gemaakt worden, is 723 kbps echter ruim voldoende. Het bereik van bluetooth is minder groot dan bij wi-fi. Bluetooth geraakt niet veel verder dan 10 meter, terwijl dit bij wi-fi normaal meer dan 30 meter is. Maar aangezien het maximale bereik van de V-scope slechts 5 meter is (onder optimale omstandigheden), zullen de apparaten nooit meer dan 5 meter van elkaar verwijderd zijn en is het bereik van bluetooth groot genoeg.

5.3 Software

Voor het implementatie gedeelte van deze thesis zijn verschillende programma's en libraries gemaakt.

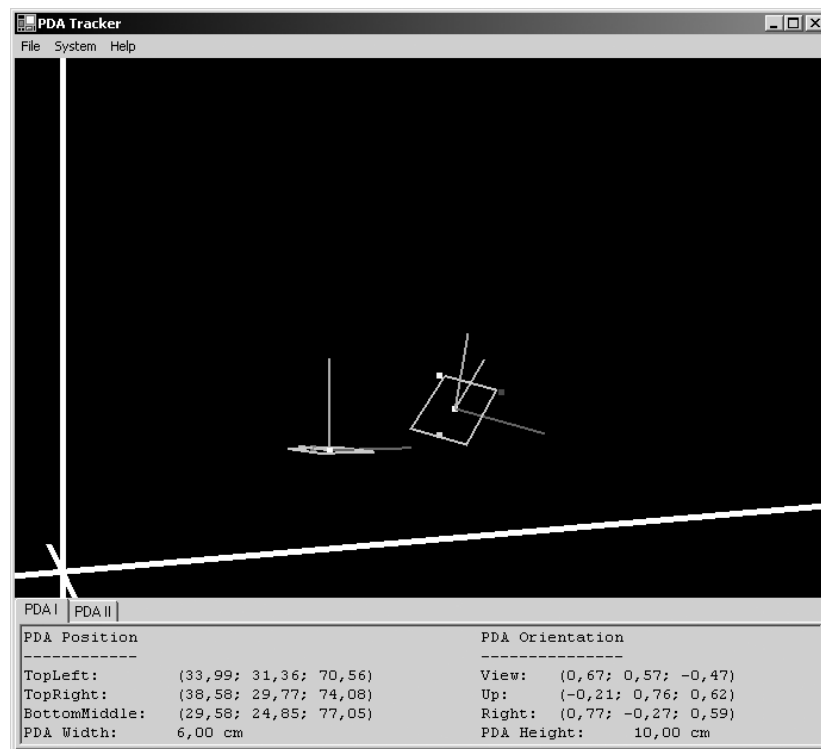
In sectie 5.3.1 wordt de toepassing besproken die ervoor zorgt dat de PDA's de coördinaten ontvangen die de V-scope gemeten heeft. Sectie 5.3.2 zal de peephole library bespreken die door alle gewone peephole applicaties als basis gebruikt wordt en in sectie 5.3.3 zullen enkele van deze peephole applicaties voorgesteld worden.

Vervolgens zal sectie 5.3.4 de multipeephole library bespreken. Deze library is een uitbreiding van de gewone peephole library en biedt de mogelijkheid om collaboratieve peephole applicaties te ontwikkelen. Een voorbeeld hiervan is het collaboratieve tekenprogramma dat in sectie 5.3.5 besproken wordt. Een belangrijk aspect van de collaboratieve programma's is het beheer van de gedistribueerde data. Dit zal uitvoerig uitgelegd worden in sectie 5.3.6.

5.3.1 PDA Tracker

Om peephole displays te kunnen implementeren, is het noodzakelijk dat het peephole display (de PocketPC in dit geval) zijn eigen positie in de omgeving kent. Deze positie wordt bepaald door het V-scope systeem, dat via de seriële poort met een PC verbonden is. Het probleem is dat de coördinaten van de PDA('s) gekend zijn door de PC, maar niet door de PDA('s) zelf.

De oplossing voor dit probleem is om die coördinaten via bluetooth van de PC naar de PDA('s) te zenden. Dit is de enige taak van het PDA Tracker programma. Bovendien wordt de info uitgetekend op het scherm van de PC (figuur 5.1) zodat de gebruiker ziet welke posities er verstuurd worden naar de PDA('s).



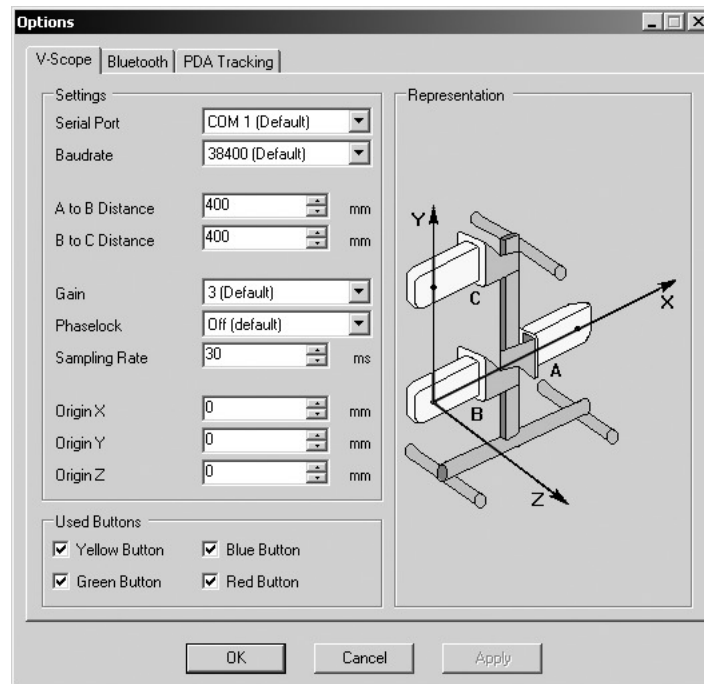
Figuur 5.1: PDA Tracker programma

Met dit programma kunnen verder nog de instellingen van de V-scope (figuur 5.2), de bluetooth connecties (figuur 5.3) en de tracking methode (figuur 5.4) aangepast worden. Hierdoor kan de gebruiker kiezen welke buttons gebruikt moeten worden, op welke manier deze buttons aan de PDA('s) bevestigd zijn, etc. Ook is het mogelijk om meerdere buttons aan één PDA te bevestigen en op deze manier de oriëntatie van de PDA gedeeltelijk (2 buttons) of volledig (3 buttons) te achterhalen (figuur 5.4). Wanneer hiervoor gekozen wordt, zal deze extra informatie eveneens via bluetooth naar de PDA verstuurd worden. Het programma dat op de PDA draait kan dan nog zelf beslissen of het deze informatie gebruikt of niet. Wanneer de oriëntatie echter niet gebruikt wordt, is het efficiënter om dit zo in te stellen bij het PDA Tracker programma. In dat geval is slechts één button per PDA nodig en kunnen de overige buttons uitgeschakeld worden. Dit zorgt voor een verhoging van de update rate per button en dat de peephole displays bijgevolg sneller “reageren” op een beweging van de gebruiker.

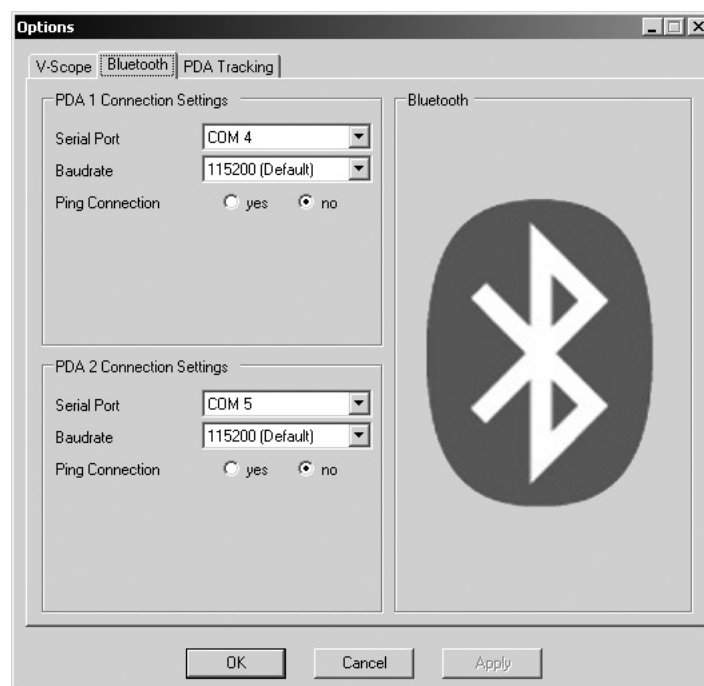
5.3.2 Peephole Library

Al de peephole programma's die ontwikkeld zijn, hebben het “peephole aspect” gemeenschappelijk. Daarom is ervoor gekozen een library te programmeren die de basisfunctionaliteiten van elk peephole programma bevat. De “Peephole Library” bevat de volgende functionaliteiten:

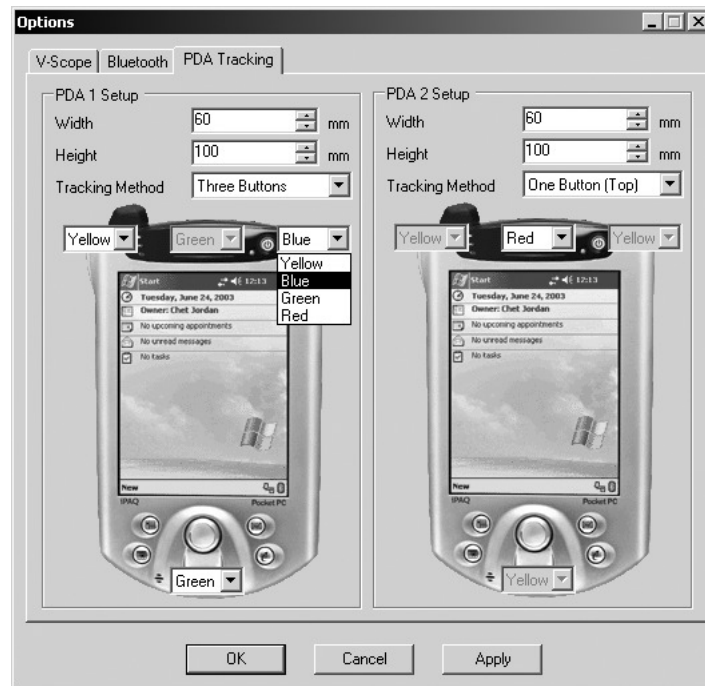
- **Piccolo PForm:** De hoofdklasse (`PeepholeForm`) van deze library is afgeleid van `PForm` (Piccolo toolkit, sectie 4.3.3). De klasse bevat dus reeds een canvas waar de programmeur naderhand enkel nog de inhoud aan moet toevoegen.



Figuur 5.2: PDA Tracker: Vscope Options



Figuur 5.3: PDA Tracker: Bluetooth Options



Figuur 5.4: PDA Tracker: PDA Tracking Options

- **Bluetooth Connectie:** De library biedt de gebruiker de mogelijkheid om een bluetooth verbinding te openen met de computer waarop de V-scope aangesloten is. Wanneer op deze computer het PDA Tracker programma draait, zal dit programma via deze connectie de coördinaten van de PDA doorsturen. De library bevat verder nog een optie dialoog die toelaat om de poort, die voor de bluetooth connectie gebruikt wordt, te wijzigen.
- **Pannen en Zoomen:** Wanneer de bluetooth connectie met de computer gemaakt is en de computer de coördinaten van de PDA doorstuurt, zal de library ervoor zorgen dat het programma op deze ontvangen coördinaten reageert zoals het hoort. De PDA horizontaal bewegen zorgt ervoor dat de inhoud van de Piccolo canvas in de tegengestelde richting beweegt. Verticaal bewegen van de PDA zorgt ervoor dat er op het canvas in -of uitgezoomd wordt. De zoomsnelheid en de zoomrichting kunnen ingesteld worden met behulp van een optie dialoog.
- **Feedback:** Aan het Piccolo canvas wordt reeds één element toegevoegd door de library zelf. Dit element kan een tekst bevatten waardoor de programmeur de mogelijkheid heeft een boodschap aan de gebruiker van het programma te tonen. Deze tekst zal onder aan het scherm weergegeven worden.

Om een peephole programma te implementeren moet de programmeur enkel zijn klasse afleiden van de `PeepholeForm` klasse en de gewenste inhoud aan het canvas van dit form toevoegen. Hierdoor wordt het relatief simpel om nieuwe peephole programma's te creëren en wordt de programmeur dubbel werk bespaard.

kan zijn PDA gewoon verticaal bewegen om in te zoomen en ondertussen naar de gewenste afbeelding bewegen. Figuur 5.6 laat zien hoe er ingezoomd wordt op een specifieke afbeelding uit de groep.

Tekenprogramma

Het tekenprogramma is gemaakt om de manipulatie van data met behulp van een peephole display te testen. Het is een eenvoudig tekenprogramma waarmee enkel lijnen en vierkanten getekend kunnen worden. De gebruiker kan de PDA rond bewegen om andere delen van het tekencanvas te bekijken en daar eventueel iets te tekenen. Het is mogelijk om reeds bestaande objecten “vast te nemen” en te slepen over het canvas. Dit slepen kan gebeuren door het object met de stylus te verplaatsen, of door het object met de stylus vast te houden en de PDA zelf te bewegen (figuur 5.7).

De hoofdklasse van dit programma is eveneens afgeleid van de `PeepholeForm` klasse (Peephole Library). Dus moest enkel nog de functionaliteit van een tekenprogramma toegevoegd worden. Het tekenmenu is een object dat zich op het canvas bevindt en kan dus door de gebruiker verplaatst worden. Het menu beweegt automatisch mee als de gebruiker zijn PDA beweegt, zodat het steeds in beeld blijft.

5.3.4 MultiPeephole Library

Zoals de Peephole Library bij de niet-collaboratieve programma's, is er ook een library gemaakt voor de collaboratieve programma's. Deze programma's moeten eveneens alle functionaliteiten van de niet-collaboratieve programma's bevatten. De MultiPeephole Library is dan ook afgeleid van de Peephole Library. Volgende extra functionaliteiten worden door de MultiPeephole Library toegevoegd:

- **Bluetooth connectie met de andere PDA:** De library biedt de mogelijkheid om een bluetooth connectie te maken met de andere PDA. Door deze connectie kunnen beide PDA's data naar elkaar verzenden om hun acties te synchroniseren.
- **Synchronisatie van het canvas:** Zodra de connectie gemaakt is, zullen beide PDA's ervoor zorgen dat ze een uniek ID hebben dat door de andere PDA gekend is. De PDA met het hoogste ID zal de positie van zijn canvas doorsturen naar de andere PDA. Deze moet dan de positie van zijn canvas synchroniseren met de doorgestuurde positie. Dit zodat beide PDA's hetzelfde zien wanneer ze zich op dezelfde positie bevinden.
- **Distributed Mutual Exclusion:** De gebruikte data zijn gedistribueerd over alle PDA's die samenwerken. Om het aanpassen van gedistribueerde data vlot te laten verlopen is er nood aan een distributed mutual exclusion algoritme. Dit algoritme gaat ervan uit dat voor elke PDA een uniek ID en een “Lamport Clock” [22] wordt bijgehouden. Daar zorgt de MultiPeephole Library ook voor.

Om een collaboratief peephole programma te maken moet de programmeur zijn programma afleiden van de `MultiPeepholeForm` klasse. Dit programma bevat dan alle functionaliteiten van zowel de Peephole als de MultiPeephole library. De programmeur moet vervolgens enkel nog de nodige inhoud toevoegen aan het canvas en de communicatie tussen de PDA's verzorgen. Voor deze communicatie worden verschillende functies ter beschikking gesteld,

inclusief functies voor distributed mutual exclusion, hetgeen in sectie (5.3.6) verder uitgelegd zal worden.

5.3.5 Collaboratief Tekenprogramma

Het collaboratieve tekenprogramma is een uitbreiding van het gewone tekenprogramma. De hoofdklasse is afgeleid van de MultiPeephole Library. Het grootste verschil met het gewone tekenprogramma is dat alle data gedistribueerd zijn over de 2 PDA's. Als een object toegevoegd of aangepast wordt, moet dit steeds doorgestuurd worden naar de andere PDA (figuur 5.8). Alle wijzigingen worden dadelijk doorgestuurd en niet pas als ze afgelopen zijn. De andere gebruiker kan deze aanpassingen dus "live" zien gebeuren. Dit brengt enkele problemen met zich mee.

Ten eerste moet de PDA weten welk object aangepast wordt wanneer hij bepaalde wijzigingen ontvangt. Dit wordt opgelost door elk object een uniek ID te geven. Bij het verzenden van de wijzigingen naar de andere PDA wordt dit ID meegestuurd om het object te identificeren.

De oplossing voor het tweede probleem is niet zo voor de hand liggend. Stel dat beide gebruikers op exact hetzelfde moment beslissen om een nieuw object aan te maken. In dat geval zullen deze nieuwe objecten hetzelfde ID hebben en voor conflicten zorgen. Een gelijkaardig probleem doet zich voor als beide gebruikers tegelijkertijd hetzelfde object willen aanpassen. Welke gebruiker krijgt voorrang?

Een goede oplossing hiervoor is om gebruik te maken van een "distributed mutual exclusion" algoritme. Dit zal verduidelijkt worden in de volgende sectie (5.3.6).

5.3.6 Beheer van Gedistribueerde Data

De enige reden dat een PC gebruikt wordt is omdat het niet mogelijk is om de 3D tracking hardware rechtstreeks met de PDA's te verbinden. De PC wordt dus **niet** gebruikt als server voor de collaboratieve toepassingen. Dit heeft als voordeel dat het in de toekomst (wanneer 3D tracking geen PC meer nodig heeft) mogelijk is om meerdere PDA's te laten samenwerken op een grote virtuele workspace zonder gebruik te maken van een PC. Omdat de gedeelde data niet opgeslagen worden op een server, moeten al deze data opgeslagen worden op de PDA's zelf. In zo'n geval is het uiteraard zeer belangrijk dat het beeld dat de PDA's van de gedeelde werkruimte hebben hetzelfde is voor alle PDA's. Dit is waar distributed mutual exclusion van pas komt.

Distributed mutual exclusion werkt op dezelfde manier als gewone mutual exclusion bij threads. Alleen werkt het in dit geval over een netwerk met data verspreid over verschillende computers. Om te garanderen dat de data die verspreid zijn over de verschillende PDA's consistent blijft, moet ervoor gezorgd worden dat slechts één PDA tegelijk een bepaald stuk data kan manipuleren. Alvorens de PDA een deel data kan aanpassen, zal deze PDA toegang moeten vragen tot de kritieke sectie waartoe deze data behoren. Slechts één PDA kan op een gegeven moment in een bepaalde kritieke sectie zijn. Het is mogelijk dat een PDA toegang heeft tot meerdere kritieke secties, maar het kan nooit dat meerdere PDA's tegelijk toegang hebben tot dezelfde kritieke sectie. Wanneer een andere PDA hetzelfde stuk data wil aanpassen (de kritieke sectie betreden) zal deze moeten wachten totdat de vorige PDA klaar is en zijn wijzigingen heeft doorgestuurd. Op deze manier zullen de gedeelde data altijd hetzelfde zijn op alle PDA's.

Wanneer een algoritme ontwikkeld wordt voor gewone of distributed mutual exclusion, moet met bepaalde zaken rekening gehouden worden [11, 29]:

- **Safety:** Er mag slechts één PDA tegelijk toegang hebben tot een bepaalde kritieke sectie.
- **Liveness:** Wanneer een PDA toegang tot een kritieke sectie vraagt zal hij deze toegang uiteindelijk ook krijgen.
- **Ordering:** Wanneer een PDA toegang tot een bepaalde kritieke sectie gevraagd heeft voor een andere PDA, zal de eerste PDA ook als eerste toegang krijgen. Dit zorgt ervoor dat er geen “starvation” kan optreden.

Het algoritme dat gebruikt is voor de implementatie voldoet aan deze 3 eisen. Het is ontwikkeld door Ricart en Agrawala [29] en maakt distributed mutual exclusion tussen N peer processen mogelijk met behulp van multicast en logische klokken [11, 22]. Wanneer multicast niet ondersteund wordt, kan dit opgelost worden door elk bericht naar elk proces apart te sturen.

Omdat de klokken van verschillende computers of PDA's nooit exact gelijk gezet kunnen worden, is een andere manier ontwikkeld om te achterhalen in welke volgorde bepaalde events voorkomen. Hiervoor worden logische klokken gebruikt. Ricart en Agrawala's algoritme [29] maakt hiervoor gebruik van “Lamport Logical Clocks” [22]. Logische klokken zijn gebaseerd op twee feiten:

- Als 2 events in hetzelfde proces gebeuren, dan gebeuren ze in de volgorde waarin dat proces ze waarneemt.
- Wanneer er een bericht gestuurd wordt tussen 2 processen is het event van het versturen van dit bericht gebeurd voor het event van het ontvangen van het bericht.

Lamport logical clocks bieden een manier om deze “happened-before” relatie numeriek voor te stellen. Een lamport logical clock is een softwarematige teller, waarvan de waarde niet afhangt van een echte klok. Elk proces bevat zijn eigen logische klok, welke het kan gebruiken om “timestamps” op een event te zetten. Om de happened-before relatie te behouden, worden de logische klokken als volgt verzonden en geüpdate door de processen:

- De logische klok L_i van proces P_i wordt verhoogd met de waarde 1 bij elk event dat voorkomt in P_i .
- 1. Wanneer een proces P_i een bericht m verzendt, wordt er met dit bericht de waarde van zijn logische klok L_i meegestuurd.
 2. Wanneer een proces P_j het bericht m ontvangt, wordt de waarde van zijn logische klok L_j gelijk gesteld aan het maximum van L_j en de meegestuurde klok bij m . Daarna wordt L_j nogmaals verhoogd om het event van het ontvangen van m te timestampen.

Als deze simpele regels gevolgd worden door alle processen zal wanneer event e voor event e' gebeurt, de waarde van de timestamp van e altijd kleiner zijn dan de waarde van de

timestamp van e' . Het omgekeerde is echter niet waar. Wanneer de timestamp van e kleiner is dan de timestamp van e' , kunnen we daaruit niet afleiden dat e voor e' gebeurd is.

Het idee achter Ricart en Agrawala's algoritme [29] is dat een proces (PDA) toegang tot een kritieke sectie moet vragen aan alle andere processen (PDA's) en die toegang pas krijgt wanneer hij hiervoor de toestemming gekregen heeft van alle andere processen. Elk proces moet een Lamport Clock en een uniek ID hebben. De condities waaronder een PDA een andere PDA toegang mag geven tot een kritieke sectie zijn zo gemaakt dat er aan de safety, liveness en ordering eisen voldaan wordt.

Ter vereenvoudiging wordt er bij deze uitleg vanuit gegaan dat er slechts 1 kritieke sectie is. In mijn implementatie kan met onbeperkt veel kritieke secties gewerkt worden, maar de werkwijze is voor elke sectie identiek.

Elke PDA houdt voor zichzelf een "state" bij die zijn toestand bevat. Die toestand is ofwel buiten de kritieke sectie (RELEASED), of hij wil toegang tot de kritieke sectie (WANTED) of hij heeft momenteel toegang tot de kritieke sectie (HELD).

Wanneer een proces toegang tot de kritieke sectie vraagt en de state van al de andere processen is RELEASED, zullen deze processen allemaal dadelijk antwoorden en zal de aanvrager dus toegang krijgen tot de kritieke sectie.

Wanneer één van de processen zijn state HELD is, zal dit proces niet antwoorden totdat hij klaar is met de kritieke sectie. Daardoor zal de aanvrager dus geen toegang krijgen totdat dit proces klaar is.

Als 2 of meer processen tegelijk toegang tot de kritieke sectie vragen zal het proces met de laagste timestamp als eerste alle nodige antwoorden verzamelen en dus toegang krijgen. Dit komt doordat een proces wanneer het zelf toegang wil tot de kritieke sectie, enkel zal antwoorden op een ander proces dat ook toegang wil wanneer de timestamp van dat proces lager is als zijn eigen timestamp. Wanneer de processen gelijke timestamps hebben zullen de requests geordend worden naargelang het unieke ID dat elk proces heeft.

Dit algoritme zorgt er dus voor dat de verschillende PDA's kunnen samenwerken op een gedeelde virtuele workspace zonder dat er een centrale server nodig is die de gedeelde data beheert. Er is een kleine aanpassing gemaakt aan het algoritme zodat het bruikbaar is voor mijn implementatie van collaboratieve peephole displays. In het originele algoritme kan het voorkomen dat een proces redelijk lang moet wachten alvorens het toegang krijgt tot een kritieke sectie indien een ander proces ermee bezig is. Om dit te voorkomen wordt bij mijn implementatie door alle PDA's bijgehouden welke kritieke sectie momenteel in gebruik is en welke niet. Wanneer de gebruiker vervolgens een object wil manipuleren dat momenteel in gebruik is door iemand anders, zal zijn applicatie niet blijven hangen omdat hij geen toegang krijgt, maar een melding laten zien aan de gebruiker dat dit object momenteel in gebruik is door iemand anders. De gebruiker kan vervolgens gewoon verder werken en later nog eens proberen om dit object te wijzigen. Het is theoretisch echter mogelijk dat 2 gebruikers op exact hetzelfde moment hetzelfde object willen manipuleren. In dat geval zullen beide PDA's toegang vragen tot de kritieke sectie en zal één van beide applicaties niet meer reageren totdat de persoon die eerst toegang kreeg klaar is. De kans dat dit gebeurt is echter zeer klein en het is dan ook nog niet voorgekomen bij het testen.

Om distributed mutual exclusion eenvoudig toegankelijk te maken voor de programmeur, worden er hiervoor in de MultiPeephole Library enkele functies voorzien:

- `void LockDistributedMutex(uint nMutexID)`: Deze functie moet aangeroepen worden door de programmeur alvorens hij een stuk van de gedeelde data gaat veranderen. Als argument wordt de ID van de kritieke sectie meegegeven. Hierdoor is het mogelijk zoveel kritieke secties te maken als nodig. Het is de verantwoordelijkheid van de programmeur om een ID met een bepaalde kritieke sectie te linken.
- `void ReleaseDistributedMutex(uint nMutexID)`: Deze functie wordt aangeroepen door de programmeur als hij klaar is met het aanpassen van data in de kritieke sectie en deze wijzigingen heeft doorgestuurd naar alle andere PDA's. Als argument moet wederom het ID van de kritieke sectie meegegeven worden.
- `bool IsLocked(uint nMutexID)`: Deze functie kan gebruikt worden om te controleren of een bepaalde kritieke sectie al gelocked is. Door dit altijd te checken alvorens zelf toegang tot die sectie te vragen, kan voorkomen worden dat het programma "hangt" als deze sectie al in gebruik is door een andere PDA.

Het collaboratieve tekenprogramma (sectie 5.3.5) gebruikt voor elk object een aparte kritieke sectie. Op deze manier kan het nooit voorkomen dat beide gebruikers tegelijkertijd hetzelfde object aanpassen. Als ID voor de kritieke sectie wordt het ID van het object gebruikt. De teller die gebruikt wordt om een uniek ID aan elk object toe te wijzen, is ook een kritieke sectie. De ID's van de objecten zullen vanaf 1 optellen zodat ID 0 overblijft voor de kritieke sectie van de teller.

5.4 Toekomstig Werk

Wanneer de tracking technologie draagbaar wordt, kan gebruik gemaakt worden van een "personal information space" [13, 34, 37]. Dit houdt in dat de informatieruimte rond de gebruiker gesitueerd is. Het zou dan mogelijk zijn om een bepaald programma altijd rechts van de gebruiker te plaatsen, een ander programma links van de gebruiker, etc. Het handige is dat deze information space de gebruiker altijd "volgt" omdat deze het 3D tracking systeem bij zich draagt. Wanneer de gebruiker een bepaald stuk data links voor zich plaatst, zal wanneer hij zich later op een andere plaats bevindt, dit stuk data nog altijd links terug te vinden zijn.

De huidige peephole applicaties gebruiken enkel de positie van de PDA. Het is echter mogelijk om ook de oriëntatie van de PDA te achterhalen. Deze informatie kan gebruikt worden om meer geavanceerde programma's met de peephole techniek te programmeren.

Er is nog een andere interessante toepassing met peephole displays mogelijk. Deze toepassing is gebaseerd op Toolglass Menu's en Magic Lenses [6, 31]. Toolglass Menu's en Magic Lenses maken gebruik van doorzichtige widgets (menu's) waardoor de gebruiker de onderliggende data op een andere manier kan bekijken en eventueel manipuleren. Het is mogelijk om een peephole display als een dergelijk menu te gebruiken als aanvulling op een virtual workbench. De PDA zou dan een ander uitzicht geven op de data dan de virtual workbench en ook de mogelijkheid bieden om deze data te manipuleren op dezelfde manier als met een toolglass menu.

Het collaboratieve aspect van de peephole displays werkt tot dusver slechts met een maximum van 2 PDA's. Aangezien gebruik gemaakt wordt van een distributed mutual exclusion

algoritme, dat bedoeld is om meerdere processen te synchroniseren, zou een uitbreiding naar meerdere PDA's niet veel problemen mogen opleveren. Het enige probleem is dat de V-scope maximum 4 PDA's tegelijk kan tracken (als we enkel geïnteresseerd zijn in de positie).

5.5 Conclusie

In dit hoofdstuk werd het implementatie gedeelte van deze thesis besproken.

Bij de implementatie werd gebruik gemaakt van verschillende soorten hardware, namelijk voor de 3D tracking en de draadloze communicatie tussen de PDA's. Voor de 3D tracking werd de V-scope gebruikt. Voor de draadloze communicatie werd gekozen voor bluetooth omdat deze technologie "overall" werkt en niet gebonden is aan een basisstation of server.

Vervolgens werd uitgelegd welke software ontwikkeld is. De eerste toepassing was het PDA Tracker programma. De V-scope stuurt de coördinaten van de PDA's naar dit programma, waarna het programma ze met behulp van bluetooth doorstuurt naar de PDA's. Het PDA Tracker programma dient niet als server voor de collaboratieve peephole toepassingen en wordt dus overbodig zodra de tracking hardware draagbaar wordt en de informatie hiervan rechtstreeks naar de PDA's gestuurd kan worden zonder tussenstop bij een PC.

Verder werden de peephole en multipeephole libraries voorgesteld. Deze 2 libraries bevatten alle basisfunctionaliteiten die elk peephole programma gebruikt. Gewone peephole programma's worden afgeleid van de peephole library, collaboratieve programma's worden afgeleid van de multipeephole library.

Een belangrijk deel van deze multipeephole library is distributed mutual exclusion. Het gebruikte algoritme hiervoor werd uitvoerig toegelicht. Het algoritme is bedoeld om de acties van meerdere processen te synchroniseren. Hoewel bij de huidige implementatie slechts 2 PDA's samenwerken, is het mogelijk om dit in de toekomst uit te breiden naar meerdere PDA's.

Hoofdstuk 6

Gebruikerstesten

6.1 Inleiding

Wanneer er nieuwe software ontwikkeld wordt, is het altijd belangrijk om na te gaan of deze software gebruiksvriendelijk en een verbetering is ten opzichte van reeds bestaande software. Het is niet aan de programmeur om hierover te oordelen omdat deze meestal bevooroordeeld is. Daarom worden hiervoor gebruikerstesten uitgevoerd.

Het voordeel van gebruikerstesten is dat er objectieve data verzameld kunnen worden in verband met de uitvoertijd van bepaalde taken of opdrachten en dat de gebruikers meestal ook andere en nieuwe ideeën aanbrengeen waaraan de programmeur nog niet gedacht had.

De gebruikerstesten van deze thesis bestaan uit twee grote onderdelen. Enerzijds de objectieve testen die de ontwikkelde programma's op bepaalde punten vergelijken met reeds bestaande programma's. Deze testen worden verder uitgelegd in sectie 6.2. Anderzijds mogen de gebruikers na de test een vragenlijst invullen waar naar hun subjectieve mening over de programma's gevraagd wordt. Er is eveneens de mogelijkheid om eventuele uitbreidingen of verbeteringen voor te stellen. Deze vragenlijst komt aan bod in sectie 6.3. Tenslotte worden de resultaten van de testen en de vragenlijst in sectie 6.4 toegelicht.

6.2 Testen

6.2.1 Peephole ImageViewer vs Pocket PhotoMesa

Bij de eerste test wordt de peephole ImageViewer (sectie 5.3.3) vergeleken met de PocketPC versie van PhotoMesa [2] (sectie 4.2.1). Alvorens de gebruikers de testen afleggen, mogen ze de programma's leren kennen. Ze krijgen even de tijd om te experimenteren met zowel de ImageViewer als PhotoMesa. Als de gebruiker denkt dat hij beide programma's voldoende onder de knie heeft, wordt met de test begonnen.

Voor deze test zijn 16 figuren gemaakt die allemaal een groot, groen cijfer op een zwarte achtergrond tonen. Er werd ook een programma gemaakt dat deze figuren random door elkaar zet, zodat de cijfers telkens in een andere volgorde getoond worden wanneer de map wordt ingeladen door PhotoMesa of de peephole ImageViewer.

Bij het starten van de test worden alle figuren naast elkaar getoond, ver genoeg uitgezoomd zodat ze allemaal zichtbaar zijn. Vervolgens is het aan de gebruiker om de cijfers zo snel mogelijk, in volgorde en volledig ingezoomd, in beeld te brengen. Wanneer de gebruiker bij



Figuur 6.1: Test 1 met Pocket PhotoMesa

het laatste cijfer (16) geraakt is, zal de tijd stopgezet worden. Elke gebruiker doet deze test 2 keer. Eén keer met PhotoMesa (figuur 6.1) en één keer met de peephole ImageViewer (figuur 6.2). De helft van de gebruikers zal de test eerst afleggen met PhotoMesa, de andere helft eerst met de peephole ImageViewer. De figuren worden na elke test door elkaar gezet zodat ze nooit in dezelfde volgorde voorkomen.

Nadat alle gebruikers deze testen afgelegd hebben, zal de tijd die nodig was voor het afleggen van de test met de peephole ImageViewer vergeleken worden met de tijd die nodig was voor PhotoMesa.

6.2.2 Peephole Tekenprogramma vs Windows CE Paint

Bij deze test wordt een vergelijking gemaakt tussen het peephole tekenprogramma (sectie 5.3.3) en een gewoon tekenprogramma voor de PocketPC [26]. Alvorens de gebruikers de testen afleggen, zullen ze de tijd krijgen om met beide programma's te leren werken. Zodra de gebruiker denkt dat hij klaar is, beginnen de testen.

Voor deze test zijn 4 afbeeldingen gemaakt waarvan één overgetekend moet worden met elk programma. De afbeeldingen zijn allen even groot en bestaan telkens uit dezelfde, gekleurde rechthoeken. Het enige verschil tussen de afbeeldingen is de positie van de rechthoeken. De afbeeldingen zijn zo gemaakt dat ze meer ruimte in beslag nemen dan er beschikbaar is op het scherm van de PDA (er moet dus gescrolled worden). De gebruiker moet een nummer trekken en aan de hand van dat nummer wordt de afbeelding, die hij moet overtekenen, gekozen. De test wordt afgelegd met het peephole tekenprogramma en met het gewone tekenprogramma. Voor elk programma moet een nieuwe afbeelding gekozen worden. De helft van de gebruikers zal de test eerst afleggen met het peephole tekenprogramma, de andere helft begint met het gewone tekenprogramma. De gebruiker mag zelf kiezen wanneer hij begint te tekenen en de tijd wordt stopgezet zodra de tekening klaar is.

Als alle gebruikers de testen afgelegd hebben wordt de tijd die nodig was om de opdracht tot een goed einde te brengen met het peephole tekenprogramma vergeleken met de tijd die nodig was om de tekening te kopiëren met het gewone tekenprogramma.



Figuur 6.2: Test 1 met peephole ImageViewer

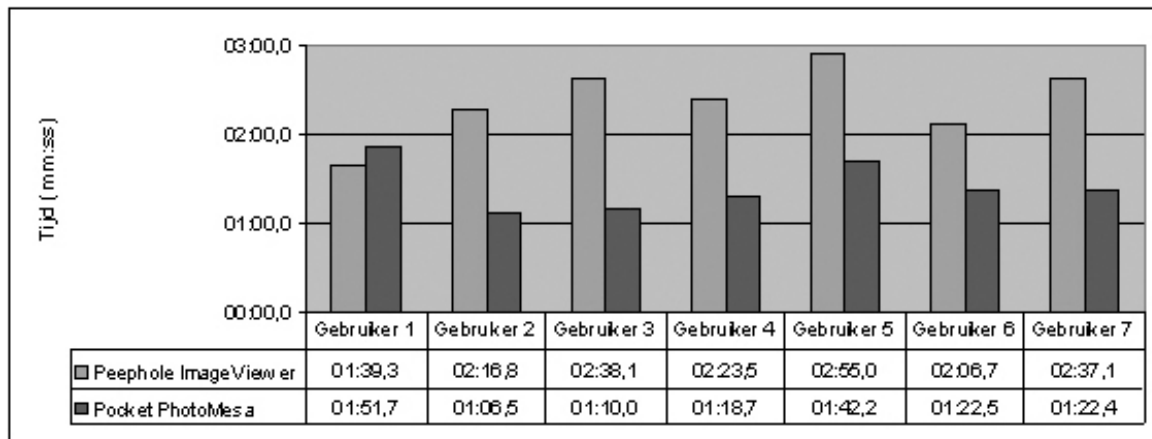
6.2.3 Collaboratief Peephole Tekenprogramma

Voor het collaboratieve tekenprogramma (sectie 5.3.5) zijn geen testen voorzien. De gebruikers mogen er wel even mee werken en naderhand bij het invullen van de vragenlijst (sectie 6.3) hun mening geven.

6.3 Vragenlijst

Na de testen van de ImageViewer en het tekenprogramma zal aan de gebruikers gevraagd worden om volgende vragenlijst in te vullen:

1. Peephole ImageViewer:
 - (a) Vond u de peephole ImageViewer of PhotoMesa het meest aangename programma om mee te werken? En waarom?
 - (b) Zijn er zaken die u stoorden bij het gebruik van de peephole ImageViewer? Zoja, welke? En hoe kan dit volgens u verbeterd worden?
2. Peephole Tekenprogramma (**NIET** collaboratief):
 - (a) Vond u het peephole tekenprogramma of het gewone tekenprogramma het meest aangenaam om mee te werken? En waarom?
 - (b) Zijn er zaken die u stoorden bij het gebruik van het peephole tekenprogramma? Zoja, welke? En hoe kan dit volgens u verbeterd worden?
3. Zijn er zaken die u stoorden bij het gebruik van het collaboratieve peephole tekenprogramma? Zoja, welke? En hoe kan dit volgens u verbeterd worden?



Figuur 6.3: Resultaten Taak 1: ImageViewer vs Pocket PhotoMesa

6.4 Resultaten

6.4.1 Peephole ImageViewer vs Pocket PhotoMesa

De resultaten van de eerste test worden getoond in figuur 6.3. De test werd uitgevoerd door 7 gebruikers en de resultaten die door de figuur getoond worden zijn de tijden die elke gebruiker nodig had om de test met beide programma's uit te voeren. Hierbij valt het op dat bijna alle gebruikers de test sneller uitgevoerd hebben met Pocket PhotoMesa. Het gemiddelde van de verschillen tussen de tijden van elke gebruiker (\bar{d}) bedraagt 57,5 seconden.

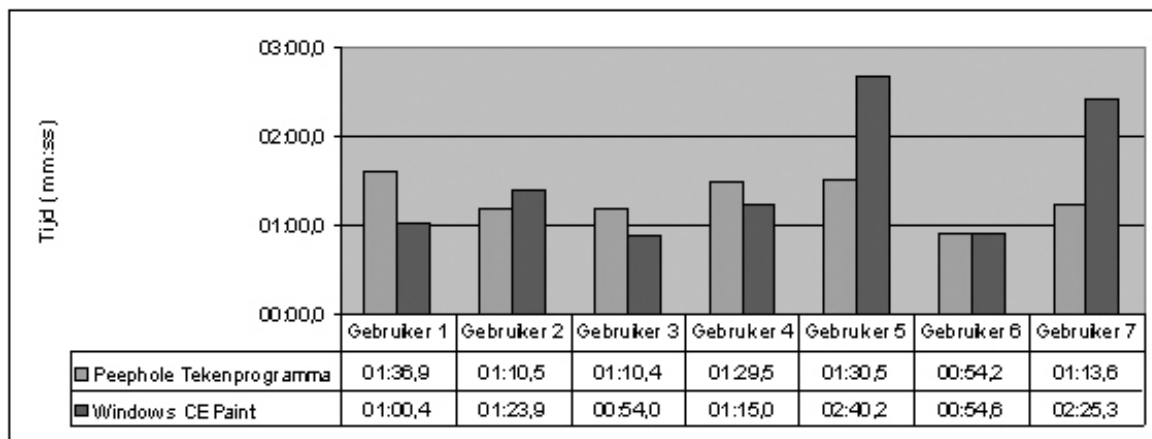
Om de significantie van deze gegevens te bepalen, wordt gebruik gemaakt van hypothesetoetsing voor kleine gekoppelde steekproeven [12]. De reden om voor deze methode te kiezen, is dat slechts 7 testgebruikers aan de testen deelgenomen hebben en omdat elke gebruiker de test met beide programma's uitgevoerd heeft (daarom een *gekoppelde* steekproef).

De nulhypothese stelt dat er geen significant verschil is tussen de uitvoeringstijden van beide programma's ($H_0 : \mu_d = 0$). De alternatieve hypothese stelt dat er wel een significant verschil is tussen de uitvoeringstijden van beide programma's ($H_1 : \mu_d \neq 0$). Indien de nulhypothese verworpen kan worden, mogen we concluderen dat gebruikers met Pocket PhotoMesa minder tijd nodig hebben om de eerste test uit te voeren dan met de peephole ImageViewer. Indien de nulhypothese niet verworpen kan worden, mogen we deze conclusie niet trekken.

We gebruiken de t-verdeling met 6 vrijheidsgraden (aantal gebruikers - 1) om de nulhypothese te toetsen. Met $\alpha = 0.05$ en 6 vrijheidsgraden ($t_{0,025} = 2,447$) kan H_0 verworpen worden als $t < -2,447$ of als $t > 2,447$. Bij deze test is $t = 4,537$ en kan H_0 dus duidelijk verworpen worden. Hieruit volgt dat Pocket PhotoMesa voor deze specifieke taak sneller is dan de peephole ImageViewer.

Een mogelijke verklaring is het feit dat de manieren waarop de taak met beide programma's uitgevoerd moest worden niet erg op elkaar leken. Bij Pocket PhotoMesa moest er slechts op een cijfer geklikt worden door de gebruiker om in te zoomen, terwijl hij bij de peephole ImageViewer ernaar moest scrollen en vervolgens ook nog eens handmatig moest inzoomen.

De vergelijking was beter geweest indien een ander programma dan Pocket PhotoMesa gebruikt werd waarvan de interactiemethode meer gelijkenissen vertoonde met deze van de



Figuur 6.4: Resultaten Taak 2: Tekenprogramma vs Windows CE Paint

peephole ImageViewer. Hiermee wordt bedoeld dat de gebruiker bijvoorbeeld moest scrollen met behulp van scroll bars en handmatig moest inzoomen met behulp van een slider.

6.4.2 Peephole Tekenprogramma vs Windows CE Paint

De resultaten van de tweede test met het peephole tekenprogramma en het bestaande tekenprogramma worden getoond in figuur 6.4. Voor deze test waren er eveneens 7 gebruikers en worden voor elke gebruiker de uitvoeringstijden van beide programma's getoond in de figuur. Bij deze test liggen de resultaten van de programma's dicht bij elkaar. Het gemiddelde van de verschillen tussen de tijden van elke gebruiker (\bar{d}) bedraagt hier -12,54 seconden en hierdoor lijkt het alsof het peephole tekenprogramma sneller is.

Ook bij deze test wordt er gebruik gemaakt van hypothesetoetsing voor kleine gekoppelde steekproeven [12] om te onderzoeken of het verschil tussen de tijden significant is. Net als bij de vorige test stelt de nulhypothese dat er geen significant verschil is tussen de uitvoeringstijden van beide programma's ($H_0 : \mu_d = 0$) en stelt de alternatieve hypothese dat er wel een significant verschil is tussen de uitvoeringstijden ($H_1 : \mu_d \neq 0$). Indien de nulhypothese verworpen kan worden, mogen we concluderen dat het peephole tekenprogramma sneller is bij het uitvoeren van de tweede test. Indien de nulhypothese niet verworpen kan worden, mogen we deze conclusie niet trekken.

We gebruiken opnieuw de t-verdeling met 6 vrijheidsgraden (aantal gebruikers - 1) om de nulhypothese te toetsen. Met $\alpha = 0.05$ en 6 vrijheidsgraden ($t_{0,025} = 2,447$) kan H_0 verworpen worden als $t < -2,447$ of als $t > 2,447$. Bij deze test is $t = -0,779$ en dus kan H_0 niet verworpen worden. Hieruit volgt dat de verschillen tussen de gemeten tijden niet significant zijn en dat deze specifieke taak met beide programma's even snel uitgevoerd kan worden.

6.4.3 Vragenlijst

Elke gebruiker heeft na het uitvoeren van beide testen de mogelijkheid gekregen om de vragenlijst uit sectie 6.3 in te vullen. De antwoorden hiervan zullen in het kort besproken worden:

1. *Peephole ImageViewer:*

- (a) *Vond u de peephole ImageViewer of PhotoMesa het meest aangename programma om mee te werken? En waarom?*

Bijna alle gebruikers verkozen PhotoMesa. De meest gegeven reden is dat het met slechts één muisklik mogelijk is om in te zoomen op het gewenste cijfer, terwijl hiervoor bij de peephole ImageViewer verschillende bewegingen nodig zijn. Ook het feit dat er een kleine vertraging is tussen de fysieke beweging van de PDA en de reactie hierop en dat het moeilijk is om het beeld perfect stil te houden bij de peephole ImageViewer, stoorde de meeste gebruikers.

Eén gebruiker verkoos de peephole ImageViewer en gaf hiervoor als reden dat het een nieuwe interactiemethode is.

- (b) *Zijn er zaken die u stoorden bij het gebruik van de peephole ImageViewer? Zoja, welke? En hoe kan dit volgens u verbeterd worden?*

De vertraging tussen de fysieke beweging van de PDA en de reactie van het programma hierop werd door de meeste gebruikers als storend ervaren (niet vloeiend genoeg). Ook het feit dat het onmogelijk is om het beeld perfect stil te houden door de trillingen van de hand en de jitter bij het tracken is een storend element voor de meeste gebruikers. Sommige gebruikers hadden ook last met het feit dat het canvas zich perfect plat voor hen uitstreckte en dat het schuin omhoog bewegen van de PDA dus niet enkel scrollen maar ook zoomen tot gevolg had.

Een voorgestelde verbetering is om de gevoeligheid voor horizontale bewegingen (scrollen) te verlagen in uitgezoomde toestand en te verhogen wanneer er ingezoomd is. Een ander voorstel was om de gebruiker de mogelijkheid te bieden om met behulp van een optie dialoog de bewegingssnelheid en gevoeligheid te verlagen of verhogen.

Een mogelijke oplossing voor het probleem met het canvas dat te plat ligt is om de oriëntatie van de PDA te tracken en het canvas altijd in hetzelfde vlak als de PDA te leggen.

Er werd bovendien een suggestie gegeven om ervoor te zorgen dat er niet zulke grote bewegingen gemaakt moeten worden om “veraf” liggende figuren te bekijken. Het idee is om een soort “bounding box” rond de gebruiker te maken. Wanneer de gebruiker zijn PDA dan over één van de randen van de bounding box beweegt, moet het beeld automatisch verder blijven scrollen in die richting totdat de gebruiker zijn PDA terug binnen de grenzen van de bounding box brengt.

2. *Peephole Tekenprogramma (NIET collaboratief):*

- (a) *Vond u het peephole tekenprogramma of het gewone tekenprogramma het meest aangenaam om mee te werken? En waarom?*

De meeste gebruikers verkozen het peephole tekenprogramma en de hoofdreden was dat er niet gescrolled moest worden met scroll bars om de figuur groter dan het scherm te tekenen. Dit scrollen werd door de meeste gebruikers, bij het gewone tekenprogramma, als desoriënterend ervaren. Ook het feit dat men door uit te zoomen een overzicht van de volledige figuur kon verkrijgen met het peephole tekenprogramma werd geapprecieerd.

Sommige gebruikers verkozen echter het gewone tekenprogramma en de hoofdreden was dat het beeld perfect stil stond tijdens het tekenen, hetgeen niet het geval

was bij het peephole tekenprogramma. Hierdoor is het niet zo gemakkelijk om nauwkeurig te tekenen met het peephole programma.

- (b) *Zijn er zaken die u stoorden bij het gebruik van het peephole tekenprogramma? Zoja, welke? En hoe kan dit volgens u verbeterd worden?*

Ook bij deze applicatie werd de vertraging en het constante “bibberen” van het beeld als storend ervaren. Eén gebruiker vermeldt ook dat hij soms moeilijkheden had om de tekening terug te vinden en stelt voor om een overview window toe te voegen waarop de tekening en de huidige positie van de gebruiker zichtbaar zijn.

3. *Zijn er zaken die u stoorden bij het gebruik van het collaboratieve peephole tekenprogramma? Zoja, welke? En hoe kan dit volgens u verbeterd worden?*

De meeste gebruikers hadden niets aan te merken op deze applicatie en vonden het zelfs leuk om ermee te werken. Enkele gebruikers vermeldden wel dat het moeilijk is om te tekenen op de positie waar de andere gebruiker op dat moment bezig is omdat beide PDA's zich dan boven elkaar bevinden.

6.5 Conclusie

In dit hoofdstuk zijn de gebruikerstesten van de implementatie van deze thesis aan bod gekomen.

Allereerst zijn twee testen voorgesteld die door de testgebruikers uitgevoerd moesten worden. Het doel van deze testen was om objectieve data te verzamelen over de tijd die nodig is om bepaalde taken uit te voeren met beide programma's. De eerste test vergelijkt het reeds bestaande programma Pocket PhotoMesa met de Peephole ImageViewer. De tweede test vergelijkt Windows CE Paint met het peephole tekenprogramma. Na het uitvoeren van deze testen is aan de gebruikers gevraagd om een vragenlijst in te vullen waarin naar hun subjectieve mening over de programma's en de testen gevraagd werd.

Vervolgens zijn de resultaten van deze testen toegelicht. Hieruit is gebleken dat Pocket PhotoMesa sneller was dan de Peephole ImageViewer voor die specifieke taak. De reden voor het grote verschil is dat Pocket PhotoMesa niet zo'n goede keuze was om te vergelijken met de Peephole ImageViewer omdat de interactietechnieken voor beide programma's teveel verschillen bij het uitvoeren van deze taak. Het peephole tekenprogramma en Windows CE Paint zijn daarentegen wel aan elkaar gewaagd als de uitvoeringstijden bekeken worden. De gemiddelde uitvoeringstijd was iets korter voor het peephole tekenprogramma, maar dit verschil was niet significant.

Uit de resultaten van de vragenlijst is gebleken dat de meeste gebruikers zich storen aan de vertraging tussen de fysieke beweging van de PDA en de reactie hierop van het programma. Ook het feit dat het niet mogelijk is om het beeld perfect stil te houden door het bibberen van de hand en de jitter bij het tracken, is een bron van irritatie. Voor de tweede test verkozen de meeste gebruikers het peephole tekenprogramma omdat de mogelijkheid tot zoomen een grote hulp was om een goed overzicht te behouden. Op het collaboratieve tekenprogramma was er niets aan te merken en de testgebruikers vonden het zelfs leuk om hiermee te werken.

Hoofdstuk 7

Conclusie

Peephole displays bieden een oplossing voor het probleem waar alle handheld displays last van hebben: te veel data voor een te klein scherm. De peephole techniek is gebruiksvriendelijk en eenvoudig aan te leren en de taken die ermee uitgevoerd worden, gebeuren meestal sneller en vlotter dan met gewone handheld displays. Dit komt doordat het een natuurlijke wijze van werken is die iedereen kent vanuit het dagelijks leven. Wanneer de gebruiker bepaalde stukken data zichtbaar wil maken is het logisch en intuïtief om het display gewoon naar de positie te bewegen waar deze data zich bevinden. Ook de manipulatie van data gebeurt op een natuurlijke manier, namelijk door beide handen te laten samenwerken. De niet-dominante hand zorgt voor de navigatie terwijl de dominante hand de manipulatie van de data verzorgt. Dit voelt zeer natuurlijk aan voor de gebruiker omdat de dominante hand de niet-dominante hand altijd als referentie gebruikt en dus bijna automatisch volgt wanneer er met de niet-dominante hand genavigeerd wordt.

Voor de grafische user interfaces op de handheld computers werd gekozen om met zoomable user interfaces te werken. De functionaliteit van deze zui's is ideaal om te gebruiken voor peephole displays. Zoomen en pannen zijn de belangrijkste interactietechnieken bij peephole displays en deze technieken worden uitstekend ondersteund door de zui toolkits. Uit onderzoek blijkt dat programma's die zui's gebruiken dikwijls sneller en vlotter werken dan de gewone variant. Het is natuurlijk mooi meegenomen dat een zui toolkit bestaat voor het .NET Compact framework, waarmee applicaties voor de PocketPC ontwikkeld kunnen worden.

Peephole displays moeten uiteraard hun positie in de omgeving kennen om bruikbaar te zijn en hiervoor is een goed 3D tracking systeem nodig. Het is belangrijk dat dit tracking systeem aan enkele eisen voldoet. De belangrijkste eisen zijn dat het tracken draadloos gebeurt en dat meerdere objecten tegelijk getracked kunnen worden. Het V-scope systeem dat voor deze thesis gebruikt is, voldoet aan deze twee eisen en heeft nog enkele andere nuttige eigenschappen. Het bereik is redelijk groot en de buttons zijn klein en licht. Het enige nadeel van de V-scope is dat de oriëntatie van de buttons niet getracked kan worden. Dit is echter op te lossen door drie buttons aan de PDA te bevestigen en met deze drie posities zelf de oriëntatie te bepalen.

Het doel van deze thesis was het onderzoeken en implementeren van collaboratieve peephole displays. Er zijn twee applicaties voor gewone peephole displays en één applicatie voor collaboratieve peephole displays ontwikkeld. Hiermee is aangetoond dat het mogelijk is om peephole displays collaboratief te laten samenwerken op een gedeelde virtuele canvas en is het doel van deze thesis bereikt.

Bibliografie

- [1] Ravin Balakrishnan and Ken Hinckley. The role of kinesthetic reference frames in two-handed input performance. In *ACM Symposium on User Interface Software and Technology*, pages 171–178, 1999.
- [2] Benjamin B. Bederson. Photomesa: A zoomable image browser using quantum treemaps and bubblemaps. In *ACM Conference on User Interface and Software Technology (UIST 2001)*, pages 71–80, 2001.
- [3] Benjamin B. Bederson, Jesse Grosjean, and Jon Meyer. Toolkit design for interactive structured graphics, 2003.
- [4] Benjamin B. Bederson, Jon Meyer, and Lance Good. Jazz: an extensible zoomable user interface graphics toolkit in java. In *UIST*, pages 171–180, 2000.
- [5] Devesh K Bhatnagar. Position trackers for head mounted display systems: A survey. Technical report, University of North Carolina, Chapel Hill, NC, USA, 1993.
- [6] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: The see-through interface. *Computer Graphics*, 27(Annual Conference Series):73–80, 1993.
- [7] Bluetooth specifications. World Wide Web. <http://www.thewirelessdirectory.com/Bluetooth-Overview/Bluetooth-Specification.htm>.
- [8] Official bluetooth site. World Wide Web. <http://www.bluetooth.org>.
- [9] Andreas Butz and Antonio Krüger. A generalized peephole metaphor for augmented reality and instrumented environments, 2003.
- [10] W. Buxton and B. Myers. A study in two-handed input. In *CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 321–326. ACM Press, 1986.
- [11] George Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed systems (3rd ed.): concepts and design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [12] Dennis J. Sweeney David R. Anderson and Thomas A. Williams. *Statistiek voor economie en bedrijfskunde*. Academic Service, Schoonhoven, Nederland, 2000.
- [13] George W. Fitzmaurice. Situated information spaces and spatially aware palmtop computers. *Commun. ACM*, 36(7):39–49, 1993.

-
- [14] Eddy Flerackers. Cursus virtuele realiteit, 2003-2004.
- [15] Yves Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *The Journal of Motor Behavior*, 19(4):486–517, 1987.
- [16] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. Sensing techniques for mobile interaction. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 91–100. ACM Press, 2000.
- [17] Kasper Hornbæk, Benjamin B. Bederson, and Catherine Plaisant. Navigation patterns and usability of overview + detail and zoomable user interfaces for maps. Technical report, University of Maryland, Human-Computer Interaction Lab, College Park, MD, USA, nov 2001.
- [18] Kasper Hornbæk, Benjamin B. Bederson, and Catherine Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM Trans. Comput.-Hum. Interact.*, 9(4):362–389, 2002.
- [19] Takeo Igarashi and Ken Hinckley. Speed-dependent automatic zooming for browsing large documents. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 139–148. ACM Press, 2000.
- [20] Paul Kabbash, William Buxton, and Abigail Sellen. Two-handed input in a compound task. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 417–423. ACM Press, 1994.
- [21] Paul Kabbash, I. Scott MacKenzie, and William Buxton. Human performance using computer input devices in the preferred and non-preferred hands. In *CHI '93: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 474–481. ACM Press, 1993.
- [22] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.
- [23] Lipman Electronic Engineering Ltd. V-scope vs-100 owner's guide, rev. 1.3, 1990.
- [24] Lipman Electronic Engineering Ltd. V-scope system user's manual, 1995.
- [25] Microsoft .net. World Wide Web. <http://www.microsoft.com/net/>.
- [26] Windows ce paint. World Wide Web. <http://www.abisoft.spb.ru/products/cepaint.html>.
- [27] Piccolo toolkit. World Wide Web. <http://www.cs.umd.edu/hcil/piccolo/learn/>.
- [28] A. Lastra R. Holloway. Virtual environments: A survey of the technology. In *SIGGRAPH'95 Course*, pages A.1–A.40, 1995.
- [29] Glenn Ricart and Ashok K. Agrawala. An optimal algorithm for mutual exclusion in computer networks. *Commun. ACM*, 24(1):9–17, 1981.

-
- [30] Baillot Y. Rolland, P.J. and Goon A. A survey of tracking technology for virtual environments. Technical report, University of Central Florida, Center for research and education in optics lasers (CREOL), Orlando FL 32816, 1999.
- [31] Maureen C. Stone, Ken Fishkin, and Eric A. Bier. The movable filter as a user interface tool. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 306–312. ACM Press, 1994.
- [32] Robert J. Stone. Position and orientation sensing in virtual environments. *Sensor Review*, 16(1):40–46, 1996.
- [33] Michael Tsang, George W. Fitzmaurice, Gordon Kurtenbach, Azam Khan, and Bill Buxton. Boom chameleon: simultaneous capture of 3d viewpoint, voice and gesture annotations on a spatially-aware display. In *UIST '02: Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 111–120. ACM Press, 2002.
- [34] J.A. Waterworth. Personal spaces: 3d spatial worlds for information exploration, organisation and communication.
- [35] Greg Welch and Eric Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22(6):24–38, 2002.
- [36] Wi-fi alliance. World Wide Web. <http://www.wi-fi.com/>.
- [37] Ka-Ping Yee. Peephole displays: pen interaction on spatially aware handheld computers. In *CHI '03: Proceedings of the conference on Human factors in computing systems*, pages 1–8. ACM Press, 2003.