

## Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen, de gevraagde informatie in te vullen (en de overeenkomst te ondertekenen en af te geven).

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling met

Titel: Interaction In-The-Box

Richting: master in de informatica - Human Computer Interaction

Jaar: 2008

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Ik ga akkoord,

SWINNEN, Bert

Datum: 5.11.2008

# ***Interaction In-The-Box***

**Bert Swinnen**

promotor :  
Prof. dr. Karin CONINX

co-promotor :  
Prof. dr. Chris RAYMAEKERS

# Abstract

Er is de laatste tijd een grote toename merkbaar in het onderzoek naar multi-touch technologie. Dit onderzoek spitst zich voor een groot deel toe op het interageren met tweedimensionale omgevingen. Nochtans liggen er ook mogelijkheden bij driedimensionale omgevingen. Denk maar aan het simuleren van een operatie bij het opleiden van chirurgen of het geven van een virtuele toer op basis van de plattegrond van een nog te bouwen huis. Om dit soort toepassingen te kunnen realiseren, is er nood aan gebruiksvriendelijke interactietechnieken. Vandaar hebben we onderzocht welke bestaande interactietechnieken in aanmerking komen. Hieruit zijn een aantal interactietechnieken voortgevloeid die werden uitgetest in het 'Interior Designer' concept zodat we kunnen spreken over 'Interaction in the box'.

'Interior Designer' is een applicatie met als doel de gebruiker toe te laten om een omgeving in te richten met meubilair en dergelijke. In tegenstelling tot desktop-applicaties, vertrekt het idee van manuele invoer via de tabletop. De binnenhuisarchitect kan meubilair toevoegen aan de scene en vervolgens manipuleren via natuurlijke interactietechnieken. Eens zijn klanten arriveren kan hij het voorstel tonen en een virtuele toer afleggen in de scene. De klant kan vervolgens samen met de binnenhuisarchitect wijzigingen aanbrengen via de interface zodat een anders incrementeel proces wordt herleid tot een collaboratieve en interactieve ervaring.

# Voorwoord

Deze thesis is het resultaat van een jaar lang hard werken met vallen en opstaan en aan het eind van de rit moet ik toegeven dat het een erg leerrijke ervaring was om een dergelijk werk te kunnen volbrengen. Deze ervaring heeft me gesterkt en zal me zeker goed van pas komen in mijn latere carrière.

Echter de realisatie van deze thesis is niet alleen mijn verdienste, in tegendeel. Mijn begeleiders Maarten Cardinaels en Davy Vanacken waren van goudwaarde voor mij. Zij hebben me immers meermaals van de nodige goede raad voorzien. Ook mijn promotoren Prof. Dr. Karin Coninx en Prof. Dr. Chris Raymaekers mag ik zeker niet vergeten, voor het vertrouwen dat ze in mij hadden en om mij de kans te geven deze thesis tot een goed einde te brengen. Verder wil ik nog de mensen van het EDM bedanken voor het gebruiken van de apparatuur. In het bijzonder wil ik Tom Cuypers bedanken voor de praktische hulp die ik van hem kreeg bij het instellen van de apparatuur.

Uiteraard moet ik ook mijn ouders eeuwig dankbaar zijn voor de geweldige kans die ik van hen heb gekregen om verder te studeren. Zij bleven altijd onvoorwaardelijk in me geloven, ook op momenten dat ik het zelf even niet zag zitten. Ik heb veel aan hen te danken, zonder hen had ik immers nooit boven me zelf kunnen uitstijgen en dat was nodig om deze studies tot een goed einde te brengen. Dan is er nog die rots in de branding, mijn vriendin Goele die altijd voor me klaar stond en me op sleeptouw nam als het wat minder ging met me.

Tot slot wil ik nog alle mensen bedanken die hier niet vernoemd werden maar toch hebben bijgedragen tot het voltooien van dit werk.

Bedankt!

# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>8</b>
<b>2</b>	<b>Multi-touch technologie</b>	<b>10</b>
2.1	Camera-gebaseerde systemen . . . . .	10
2.2	Sensor-gebaseerde systemen . . . . .	12
2.3	Samenvatting . . . . .	13
<b>3</b>	<b>Tabletop interactie</b>	<b>14</b>
3.1	Inleiding . . . . .	14
3.2	Translatie- en rotatietechnieken . . . . .	15
3.2.1	Verplaatsen van objecten . . . . .	15
3.2.2	Oriëntatie van objecten . . . . .	18
3.3	Precisie . . . . .	24
3.3.1	Manipulatie van de cursor . . . . .	24
3.3.2	Gestures . . . . .	26
3.4	Lange afstanden . . . . .	32
3.4.1	Manipulatie vanop een afstand . . . . .	32
3.4.2	Manipulatie van de omgeving . . . . .	32
3.4.3	Fysieke verplaatsingen . . . . .	36
3.5	Tekstuele invoer . . . . .	37
3.5.1	Virtuele keyboards . . . . .	38
3.5.2	Gesture-based keyboards . . . . .	39
3.6	Collaboratieve systemen . . . . .	40
3.6.1	Territorialiteit . . . . .	40
3.6.2	Policies . . . . .	41
3.7	Samenvatting . . . . .	43
<b>4</b>	<b>Tabletops en de derde dimensie</b>	<b>44</b>
4.1	Inleiding . . . . .	44
4.2	Een stand van zaken . . . . .	45
4.3	Diepteperceptie . . . . .	47
4.4	3D-Interactie . . . . .	48
4.4.1	Manipulatie met behulp van constraints . . . . .	48

<i>INHOUDSOPGAVE</i>	4
4.4.2 Voodoo dolls . . . . .	49
4.4.3 3D-tabletopinteractie: Shallow depth 3D . . . . .	49
4.5 Occlusie-management . . . . .	52
4.6 Camera-management . . . . .	55
4.7 Samenvatting . . . . .	57
<b>5 Interior Designer</b>	<b>58</b>
5.1 Inleiding . . . . .	58
5.2 Menu-interface . . . . .	59
5.2.1 Structuur . . . . .	59
5.3 Interactietechnieken . . . . .	61
5.3.1 Transleren . . . . .	61
5.3.2 Roteren . . . . .	64
5.3.3 Werken op verschillende schaal . . . . .	70
5.3.4 Lange afstanden . . . . .	73
5.4 Occlusie-management . . . . .	73
5.5 Camera-management . . . . .	77
5.6 Physics . . . . .	78
5.7 Gebruikte technologie . . . . .	78
5.7.1 Hardware . . . . .	78
5.7.2 Software . . . . .	78
5.8 Samenvatting . . . . .	79
<b>6 Gebruikersevaluatie</b>	<b>80</b>
6.1 Inleiding . . . . .	80
6.2 Testopstelling . . . . .	80
6.2.1 Omgeving . . . . .	80
6.2.2 Testgebruikers . . . . .	80
6.2.3 Sessieverloop . . . . .	81
6.3 Resultaten . . . . .	82
6.3.1 Menu . . . . .	82
6.3.2 Roteren van objecten . . . . .	82
6.3.3 Verplaatsen in de hoogte . . . . .	83
6.3.4 Camerabeheer . . . . .	84
6.3.5 Werpen van objecten . . . . .	84
6.4 Samenvatting . . . . .	84
<b>7 Conclusie</b>	<b>86</b>

# Lijst van figuren

2.1	Tonen van extra informatie over mobiele telefoons via Microsoft Surfaces. . .	11
2.2	Frustrated total internal reflection. [16] . . . . .	12
2.3	Het gebruik van tangible objects op de reacTable. . . . .	13
3.1	On screen widget. [18]. . . . .	15
3.2	Dragging. T is het initiële contactpunt en T' is het eindpunt van de interactie [18]. . . . .	16
3.3	Sweeping gesture. [47] . . . . .	17
3.4	Sweeping gesture met twee handen. [47] . . . . .	17
3.5	Editing plane via het 'two-cornered hands' gesture. [47] . . . . .	18
3.6	Het onderhandelen van een groepsoriëntatie is een sociaal proces. [22] . . .	19
3.7	Continue automatische rotatie. [18] . . . . .	20
3.8	Gebruik van 'handles' op de hoeken om rotatie af te dwingen [18]. . . . .	21
3.9	'Two point rotation' [18]. . . . .	21
3.10	'Flat hand' rotatie. . . . .	22
3.11	RNT, balans tussen stroom en beweging. [23] . . . . .	23
3.12	RNT, onbalans met rotatie als gevolg. [23] . . . . .	23
3.13	Cursor-grootte 'Area-cursor'. (a) cursorgrootte te omvangrijk. (b) De cursor-grootte te klein [13]. . . . .	25
3.14	De Bubble cursor. [13]. . . . .	26
3.15	Dual finger stretch. Links voor het inzoomen, rechts na inzoomen [4]. . . .	27
3.16	Virtual keys met handles op de crosshair. [1]. . . . .	28
3.17	a. Cross lever b. Precision handle [1]. . . . .	29
3.18	Dual finger X-menu. [4]. . . . .	29
3.19	Dual finger Slider. [4]. . . . .	30
3.20	Dual finger midpoint. [4]. . . . .	31
3.21	TractorBeam. Links, hardware setup en overview. Rechts, het TractorBeam-concept [30]. . . . .	33
3.22	Pick and pop en het gebruik van rubber bands [3]. . . . .	33
3.23	De radar bevindt zich centraal onderaan en bevindt zich binnen het bereik van de gebruiker. [33] . . . . .	34
3.24	Flick and catch. [47] . . . . .	35
3.25	Object werd niet gevangen met het catch gesture. [47] . . . . .	35

3.26	Superflick. [33]	36
3.27	Pick and drop. Aantikken object (A), Opnemen object (B), Stylus is weg van het canvas [34].	37
3.28	Soft keyboards. A) Traditionele QWERTY. B) Metropolis layout [19].	38
3.29	BubbleQuerty magnificatie functie. [19].	39
3.30	Gesture-based keyboard (Cirrin). [19].	39
3.31	Evolutie territoria. Links, beginsituatie met één gebruiker. Midden en rechts resp. twee en drie gebruikers [38].	40
3.32	Release (A), Relocate (B), Reorient (C), Resize (D) [37].	42
4.1	mismatch in vrijheidsgraden invoer (links) en uitvoer (rechts).	44
4.2	Taxonomy 3D op de tabletop [14].	46
4.3	De invloed van schaduw op diepte.	48
4.4	De kasten aan de muur werden via de dual constraint gegroepeerd [42].	49
4.5	Voodoo dolls, referentiekader links en te manipuleren object rechts [32].	50
4.6	3D interactie via één contactpunt [17].	51
4.7	3D interactie via twee contactpunten [17].	52
4.8	3D interactie via drie contactpunten [17].	52
4.9	Occlusie scenario's [11].	53
4.10	Het gebruik van meerdere aanzichten lost enkele occlusieproblemen op [45].	53
4.11	Het gebruik van digital X-ray om doelen in een scene terug te vinden [2].	54
4.12	Een voorbeeld van het exploder concept toegepast op de anatomie van een voet [43].	55
4.13	Click to focus, camera animeert na klikken van punt a naar punt b [48].	55
4.14	Camera-management op basis van gestures [48].	56
5.1	Menu legende.	60
5.2	Kleurcodes menu-items.	61
5.3	De menu interface werd gekopieerd zodat gebruikers het aan twee zijden kunnen bedienen.	62
5.4	Links bovenaan vinden we het 'iLevitate' widget terug dat door P2 wordt gecontroleerd. Bovendien is de meetlat zichtbaar zodat een referentiekader voor de hoogte aanwezig is.	63
5.5	Bovenaan in beeld bevindt zich de hoogtemeter die dient als referentie voor de hoogte van een object. Depth Coloring is bovendien ingeschakeld, links wordt de stoel niet aangeraakt, rechts wel waardoor de stoel een andere kleur krijgt toegewezen en de kleuren van bed en stoel overeen komen.	64
5.6	Zowel in het bovenaanzicht als in de viewport is de xz-locator duidelijk zichtbaar.	65
5.7	De richting waarin de vinger beweegt, bepaalt de rotatierichting.	66
5.8	De tweepunts Y-rotatie in actie. De rode lijn tussen P1 en P2 geeft de indruk een hendel te zijn.	66
5.9	Links rotatie rond de Z-as, rechts rotatie rond de X-as.	67



5.10	De rotatie-as wordt bepaald door P1 en P2 en wordt in het groen weergegeven. De verplaatsing van P3 zal leiden tot een rotatie rond deze as. . . .	68
5.11	Rotaties rond de Z-as gebeuren door over de X-as te transleren (links) en vice versa (rechts). . . . .	69
5.12	Y-rotatie gebeurt door P2 rond P1 te bewegen, terwijl rotatie rond de overige assen wordt gecontroleerd door P3. P3 wordt in dit geval naar links bewogen waardoor de Z-as door het object verschijnt. . . . .	69
5.13	De afstand tussen P1 en P2 bepalen de grootte van het menu. Het verplaatsen van P2 zal het menu roteren. . . . .	70
5.14	De offset tussen vinger en cursor (crosshair) zorgt ervoor dat de asbak niet onder de vinger belandt. . . . .	71
5.15	De richting van de offset wordt bepaald door het bijbehorende icoon (36) in het menu. De vier mogelijkheden even op een rijtje. . . . .	72
5.16	Links bovenaan vinden we de context terug (computerscherm). Rechts ervan vinden we het relatieve object (asbak) terug. . . . .	73
5.17	De viewport werd los van het menu geroteerd en gescaleerd in één vloeiende beweging. . . . .	74
5.18	Het bed bovenaan wordt in wireframe getoond. . . . .	75
5.19	Bovenaan vind je de situatie voor het inschakelen van de 'interactive exploder' terug. Onderaan vind je de situatie na het inschakelen van de 'interactive exploder' terug. . . . .	76
5.20	De camera zweeft rondom zijn eigen focus. Bovenaan wordt telkens de bijbehorende viewport getoond. De focus wordt gevisualiseerd door de witte cirkel. . . . .	78
5.21	Tabletop setup in het expertisecentrum voor digitale media. [6] . . . . .	79
6.1	Een gebruiker test Interior Designer op de tabletop in het EDM. . . . .	81

# Hoofdstuk 1

## Inleiding

Dat er een markt is weggelegd voor multi-touch systemen is duidelijk te merken aan de inspanningen die bedrijven de laatste jaren hebben geleverd om kant en klare producten op de markt te brengen. Hetzelfde geldt voor tabletop systemen die zo stilaan hun weg beginnen te vinden in casino's, winkels en musea. Hier is al een hele evolutie aan vooraf gegaan om tot deze producten te komen. Binnen de research wereld is het 'Multi-touch' gebeuren dan ook een erg populair topic geworden en dat is ook logisch want tabletops hebben ons veel te bieden. Tabletops brengen invoer en uitvoer immers samen op één interactievlak en laten gebruikers toe via natuurlijke bewegingen, handgebaren te interageren met digitale data. De technologie verdwijnt hierdoor naar de achtergrond waardoor de gebruiker zich volledig kan focussen op zijn taak. De drempel om met tabletop technologie te werken zal bijgevolg ook lager zijn dan bij traditionele desktop systemen.

Ook het collaboratieve aspect is een belangrijk voordeel van tabletops. Tabletops werken samenwerking tussen gebruikers meer in de hand dan desktops. Gebruikers kunnen plaatsnemen rond de tafel en zowel individueel werken in hun nabije omgeving als in groep werken in de gemeenschappelijke ruimte tussen de gebruikers. Uiteraard vereist dit wel de nodige communicatie tussen de gebruikers maar positief is dat zij daarin niet worden gehinderd door de technologie zelf. Iets wat minder evident is bij desktops.

Tabletops zijn ook niet perfect en hebben te kampen met andere problemen dan desktops. Zo is het niet vanzelfsprekend om voor alle gebruikers de correcte oriëntatie te voorzien. Elke gebruiker beschikt immers over een ander perspectief op de data. Ook precisie is niet evident wanneer we werken met onze vingers. Kleine doelwitten zijn dan moeilijk te bereiken vanwege de dikte van onze vingers. Ook de omvang van een tabletop kan voor problemen zorgen. We zijn uiteindelijk fysiek beperkt tot die regio's op de tabletop die zich binnen armlengte bevinden. In deze thesis wordt dieper ingegaan op elk van deze problemen.

De huidige research beperkt zich voor een groot stuk tot tweedimensionale omgevingen. De integratie van driedimensionale omgevingen op tabletops is voorlopig nog voor een groot

stuk braakliggend terrein. Nochtans zijn ook hier meerdere toepassingen mogelijk. Denk maar aan het simuleren van operaties voor chirurgen of het inrichten van een woning voor binnenhuisarchitecten. Echter de integratie van drie dimensies op een tweedimensionale interface brengt ook bepaalde moeilijkheden met zich mee. Een voordehandliggend probleem stelt zich bij verplaatsingen in de hoogte omdat je als gebruiker slechts beschikt over twee invoerdimensies. Ook rotaties in drie dimensies zijn niet evident via een tabletop interface. Daarbij komen nog de standaard problemen die opduiken in elke driedimensionale wereld zoals oclusie, diepteperceptie en camera-management.

Hier ligt dan ook het zwaartepunt van deze thesis. In de literatuurstudie kom ik op elk van deze problemen terug om een totaalbeeld te krijgen over de materie. Als implementatie heb ik gekozen voor een prototype dat kan worden aangewend om aan binnenhuisinrichting te doen. Typisch is het inrichten van een huis een incrementeel proces, waarbij de binnenhuisarchitect een voorstel doet naar de klant toe. De klant komt vervolgens dit voorstel bespreken en zal zelf wijzigingen voorstellen. 'Interior Designer' is een 'proof of concept' dat er op doelt om dit proces om te vormen naar een interactief collaboratief proces dat gebruik maakt van een tabletop zodat de klant samen met de binnenhuis-architect zijn huis kan inrichten op een gebruiksvriendelijke interface.

'Interior Designer' tracht via een aantal bewust gekozen interactietechnieken al deze problemen aan te pakken zodat de gebruikers zich optimaal kunnen richten op de taak die zij moeten uitvoeren.

# Hoofdstuk 2

## Multi-touch technologie

Op dit ogenblik is het multi-touch gebeuren levendiger dan ooit. Logisch want deze technologie heeft ons veel te bieden. Zo leent multi-touch technologie zich uitstekend tot gebruik in collaboratieve tabletop omgevingen. Desktop computers zijn goed om collaboratie tussen mensen te faciliteren wanneer zij zich op verschillende locaties bevinden. Echter wanneer deze mensen zich op dezelfde locatie bevinden, zijn traditionele computers niet altijd even efficiënt. Digitale tafels zijn het ideale medium om te interageren in kleine groepjes. Dit soort interactie laat de gebruikers toe het oogcontact te behouden terwijl zij met de data bezig zijn. Toch is de ontwikkeling van dergelijke technologie nog een vrij nieuw gegeven en pas de laatste jaren heeft het zich ontpopt tot een populaire onderzoekstopic. Vandaar even een stand van zaken.

### 2.1 Camera-gebaseerde systemen

#### Microsoft Surface

Surface is een multi-touch tafel die werd ontwikkeld door Microsoft [25] om meerdere gebruikers de mogelijkheid te bieden via natuurlijke bewegingen op een directe manier te interageren met de digitale data. Bovendien kan Microsoft surface ook fysieke objecten herkennen zodat ze kunnen geïntegreerd worden in de omgeving.

Figuur 2.1 illustreert hoe Microsoft Surface in een winkel kan worden aangewend om aan klanten informatie te tonen over mobieltjes. Dit is echter maar één van de vele mogelijkheden van het product. Om deze functionaliteit te realiseren, gebruikt dit systeem een projector die het beeld projecteert op een acryl plaat. De beweging boven de tafel wordt vervolgens geregistreerd door een reeks camera's die zich onder de tafel bevinden.



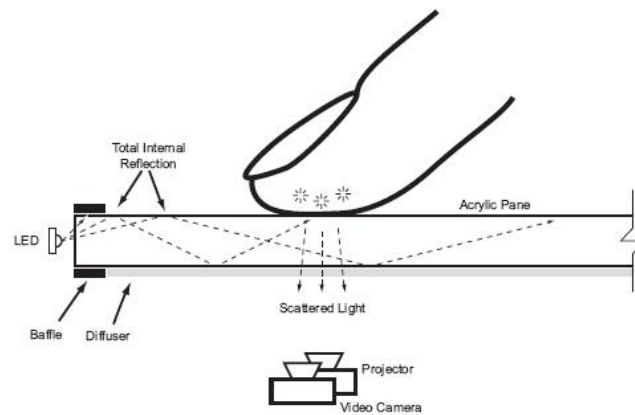
Figuur 2.1: Tonen van extra informatie over mobiele telefoons via Microsoft Surfaces.

## FTIR

Ook het 'frustrated total internal reflection' van Jeff Han et al. [16] is een veelbesproken techniek. De techniek is vergelijkbaar met deze van Microsoft [25] en gebruikt één of meerdere camera's om de bewegingen bovenop de tafel te registreren. Het systeem kan eveneens om met multi-touch en meerdere gebruikers. Eén van de voordelen van Han's systeem is dat het vrij goedkoop is om te bouwen en ondanks dat toch kan concurreren met zijn commerciële tegenhangers.

In het prototype van Jeff Han [16] wordt er gewerkt met een acrylplaat. Leds zorgen voor het licht in de acrylplaat zodat totale interne reflectie ontstaat. Een camera met bypass-filter wordt vervolgens loodrecht geplaatst tegenover de acrylplaat. Wanneer de gebruiker nu een vinger plaatst op de acrylplaat wordt het licht gefrustreerd en ontsnapt het licht in de richting van de camera. Deze camera zal het beeld registreren en aanbieden ter verwerking. Op figuur 2.2 is te zien hoe het plaatsen van een vinger ervoor zorgt dat het licht wordt gefrustreerd en ontsnapt via de onderzijde van de plaat.

Eventueel kan een plastic film worden toegevoegd aan de voorzijde van het interactievlak. Als gevolg van het vervormen van de plastic tijdens het drukken, zal een contactpunt meerdere pixels innemen. Dit kan worden geïnterpreteerd als het uitoefenen van meer of minder druk. Een nadeel is dat deze film na loslaten enige tijd nodig heeft om de initiële staat opnieuw te bereiken, waardoor het systeem het punt nog steeds als actief zal beschouwen.



Figuur 2.2: Frustrated total internal reflection. [16]

## Reactable

De Reactable [21] is eveneens een collaboratieve omgeving die gebruik maakt van multi-touch en 'tangible objects'. De Reactable is eigenlijk een elektronisch muziekinstrument dat via multi-touch en manipulatie van 'tangible objects' muziek produceert. Op figuur 2.3 zie je hoe men via het manipuleren van 'tangible objects', steeds andere melodieën componeert. Ook dit systeem maakt gebruik van camera's om de activiteit op de tafel te registreren. Deze 'tangible objects' zijn allen voorzien van een uniek patroon waardoor het systeem hen eenduidig kan registreren.

## 2.2 Sensor-gebaseerde systemen

### DiamondTouch

Een ander interessant product is de DiamondTouch [8] van Mitsubishi [26]. Net als Microsoft Surface [25] biedt de DiamondTouch de mogelijkheid om met meerdere gebruikers op hetzelfde ogenblik te werken met de data. Interactie gebeurt ook hier via natuurlijke bewegingen en gestures. Toch onderscheidt de DiamondTouch zich van het Microsoft Surface systeem omdat de DiamondTouch ook onderscheid maakt tussen de gebruikers onderling. De tafel weet als het ware wie, wat op welk moment aanraakt. Hierdoor kan de tafel inspelen op de verschillende noden van de gebruikers. Om dit te realiseren gebruikt de DiamondTouch antennes die zich in het interactievlak bevinden. Deze antennes zijn verbonden met de stoelen van de gebruikers. Als de gebruiker vervolgens de tafel aanraakt, ontstaat een gesloten circuit zodat kan de gebruiker kan worden geïdentificeerd door de tafel.



Figuur 2.3: Het gebruik van tangible objects op de reacTable.

### SmartSkin

De SmartSkin multi-touch technologie van Rekimoto et al. [35] gebruikt net als de DiamondTouch [8] elektrische sensoren om interactie te identificeren. Net als de eerder beschreven technologieën kan de SmartSkin om met meerdere gebruikers. De sterkte van de SmartSkin is dat ze in staat is de relatieve afstand tussen de vingertoppen en de tafel kan bepalen. De gebruiker is dus niet verplicht om de tafel aan te raken. Hierdoor kan het SmartSkin systeem 'hover'-technieken ondersteunen. In tegenstelling tot cameragebaseerde systemen kunnen alle sensorische elementen bovendien in de tafel worden geïntegreerd. De projectie van de data gebeurt hier overigens van bovenaf.

## 2.3 Samenvatting

De voorgaande sectie geeft een kort overzicht over reeds bestaande technologieën die hun pluimen al verdiend hebben. De laatste jaren was er een enorme groei in multi-touch gerelateerde research maar ondanks dat is er nog steeds een hoop werk te verrichten om alle mogelijkheden van deze technologie te doorgronden. Toch maken de eerste commerciële toepassingen stilaan hun opwachting in het dagelijkse leven, maar dit is slechts het begin van een wereld die zal doordrongen zijn van interactieve oppervlakten.

# Hoofdstuk 3

## Tabletop interactie

### 3.1 Inleiding

Multi-touch interactie is een manier om met digitale data te interageren. Het is voor de gebruiker een unieke ervaring om via een hands-on benadering met digitale data om te gaan. Echter multi-touch interactie is niet te vergelijken met klassieke pointersystemen. Het is dus verstandig om multi-touch interactie eens van naderbij te bekijken. Eerst bespreek ik enkele bestaande mechanismen om data te positioneren op de tabletop in de sectie 'translatie- en rotatietechnieken'.

Vervolgens stel ik enkele problemen aan de kaak zoals het gebrek aan precisie bij touch-interfaces. In de sectie 'Precisie' ga ik hier verder op in. Bovendien hebben tabletops typisch een behoorlijke omvang, en is het niet altijd evident om het gehele interactievlak te bereiken. Hiervoor zijn reeds enkele oplossingen ontwikkeld en deze komen aan bod in de sectie 'Lange afstanden'. In de laatste sectie 'Tekstuele invoer' bekijk ik nog even kort op welke manier men tekst kan invoeren zonder het traditionele toetsenbord te gebruiken. Hierna komt het collaboratieve aspect van tabletops nog aan bod.



## 3.2 Translatie- en rotatietechnieken

Om productief te kunnen werken met de data hebben we nood aan technieken die ons toelaten objecten te verplaatsen op de tabletop. Bovendien blijken de positie en oriëntatie van een object implicaties te hebben voor de communicatie en collaboratie rond de tabletop [18]. De positie en oriëntatie van een object geven immers aan wie de eigenaar van het object is en of dat object beschikbaar is voor de groep. Zelfs kleine veranderingen van positie en oriëntatie kunnen al erg betekenisvol zijn om de status van een object te wijzigen. Het is dus van kapitaal belang dat het transleren en roteren van objecten eenvoudig en snel kan gebeuren. De vraag is op welke manier dit best kan worden aangepakt. In de komende sectie beschrijf ik enkele mogelijkheden om te transleren en te roteren.

### 3.2.1 Verplaatsen van objecten

#### Explicit specification

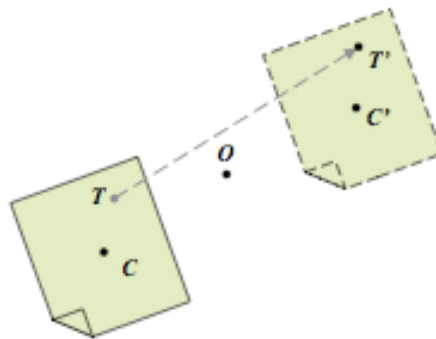
De meest eenvoudige aanpak is dat men uit gaat van expliciete specificatie. Dit zou betekenen dat het object na selectie, nieuwe coördinaten krijgt toebedeeld via een widget. Dit widget kan tevoorschijn worden gehaald door er een gesture aan toe te kennen. Dit lijkt in eerste instantie niet echt waardevol vanwege het verlies aan voeling met het te manipuleren object, doch niets is minder waar. Een dergelijke werkwijze leidt immers tot zeer precieze resultaten. In CAD-toepassingen en dergelijke kan dit een handige aanvulling zijn op andere invoermethodes.



Figuur 3.1: On screen widget. [18].

### Dragging

Een andere benadering die men kan hanteren is het slepen van objecten. Bij deze techniek sleept de gebruiker het object naar de gewenste locatie door het als het ware vast te nemen met één of meerdere vingers en het te verslepen naar een nieuwe locatie. Hancock et al. [18] stellen voor om de tabletop in te delen in translatie- en rotatiezones. In een translatiezone zal de gebruiker enkel kunnen transleren. Hierdoor worden ongewenste rotaties vermeden. Het initiële raakpunt ( $T$ ) wordt geregistreerd en het object beweegt mee gedurende de beweging tot het punt van loslaten ( $T'$ ), zoals te zien is op figuur 3.2.

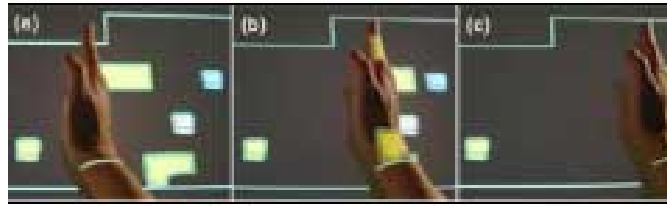


Figuur 3.2: Dragging.  $T$  is het intiële contactpunt en  $T'$  is het eindpunt van de interactie [18].

### Sweeping

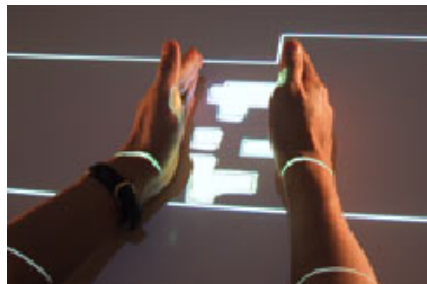
'Sweeping' is een techniek die de gebruiker toelaat één of meerdere objecten over het interactievlak te verplaatsen. Dit is een interactietechniek waarbij de gebruiker zijn hand verticaal op het canvas plaatst en vervolgens een veegbeweging uitvoert. Alle objecten die zich op het pad van deze beweging bevinden zullen vooruit worden geveegd. Merk ook de overeenkomst op met het leegmaken van een fysieke werktafel door even met de hand of de arm over het oppervlak te schuiven. Hierdoor zal de eindgebruiker wellicht weinig problemen ondervinden om zich de techniek eigen te maken [47]. Op figuur 3.3 is duidelijk te zien hoe de hand van de gebruiker de objecten voor zich uit duwt.

Het is dikwijls zo dat dergelijke metaforen een extra betekenis mee krijgen en dat is ook hier het geval. Wanneer de gebruiker over gaat tot 'sweeping' en de objecten op die manier verplaatst naar zijn eigen persoonlijke ruimte, gaat het systeem over tot het kopiëren van de objecten naar de persoonlijke ruimte van de gebruiker. De orginele objecten blijven dan gevrijwaard.



Figuur 3.3: Sweeping gesture. [47]

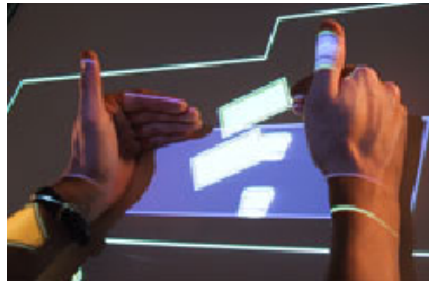
Sweeping kan overigens ook met twee handen gebeuren. Het naar mekaar toe bewegen van beide handen resulteert in dat geval in het bij mekaar brengen van de objecten zoals te zien is op figuur 3.4. De omgekeerde beweging vertaalt zich in het rekken van de afstanden tussen de objecten.



Figuur 3.4: Sweeping gesture met twee handen. [47]

### Editing planes

Een 'editing plane' [47] gebruikt men om een aantal objecten te kopiëren naar een transparante laag die zich boven de scene bevindt. Dit komt van pas wanneer de gebruiker meerdere objecten wil bewerken en tegelijk de ruimtelijke relaties tussen de objecten wil behouden. Dit kan door via een gesture een 'bounding box' af te bakenen zoals te zien is op figuur 3.5. De objecten binnen de 'bounding box' kunnen dan verplaatst worden als geheel. Eventueel kan deze techniek ook gebruikt worden om kopieën te maken van de scene.



Figuur 3.5: Editing plane via het 'two-cornered hands' gesture. [47]

### 3.2.2 Oriëntatie van objecten

#### De rol van oriëntatie

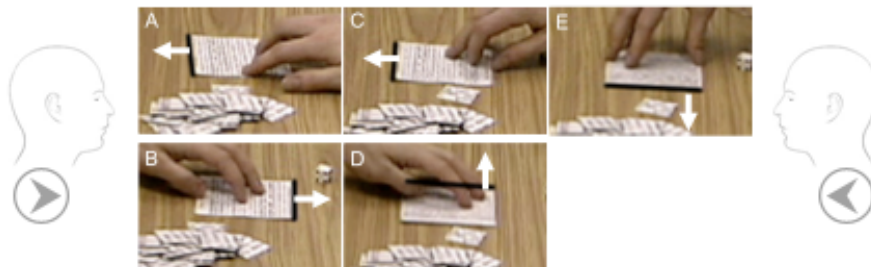
Oriëntatie speelt een grote rol in tabletop interactie. In tegenstelling tot desktopsystemen is het moeilijk te bepalen wanneer een object nu de juiste oriëntatie heeft. Gebruikers bevinden zich immers op verschillende plaatsen rondom de tabletop en hebben elk een eigen 'point of view'. Wat voor de ene gebruiker correct georiënteerd is, is dat helemaal niet voor de andere. Krüger et al. [22] ontdekten dat oriëntatie een kritieke rol speelt in hoe de gebruikers informatie interpreteren. Het blijkt overigens dat de manier van samenwerken tussen de gebruikers beïnvloed wordt door de oriëntatie van de objecten en de mogelijkheid deze aan te passen. Oriëntatie beïnvloedt de manier waarop gebruikers de werkruimte indelen en afbakenen en de wijze waarop zij aangeven eigenaar te zijn van een object. Naar communicatie toe is oriëntatie een handig hulpmiddel om een conversatie te initiëren en informatie te delen met de andere gebruikers. Uiteindelijk is men gekomen tot de vaststelling dat oriëntatie drie rollen vervult namelijk *comprehensie*, *coördinatie* en *communicatie*.

De meest voornamelijk functie van oriëntatie is **comprehensie**. Om bepaalde objecten te kunnen begrijpen moet de gebruiker ze roteren, denk maar aan het lezen van tekst die zich ondersteboven bevindt. Vandaar worden objecten typisch georiënteerd naar de zijde waaraan de gebruiker zich bevindt. Echter dit blijkt niet per sé altijd zo te zijn. Zij stellen vast dat de gebruiker de objecten zo roteert zodat deze naar hem zijn gericht. Wanneer de gebruiker dus zijdelings aan de tafel zit, zal dit een invloed hebben op de oriëntatie van de data.

Oriëntatie speelt ook een belangrijke rol in de **coördinatie** van activiteiten. Gebruikers delen de tabletop typisch in drie soorten ruimten: persoonlijke ruimten, opslagruimten en groepsruimten. Toch zijn er geen expliciete grenzen afgesproken tussen deze ruimten onderling en verloopt dit proces opmerkelijk vlot. Men zal typisch objecten die zich dicht bij de gebruiker bevinden zo oriënteren zodat zij voor de gebruiker over de ideale oriëntatie beschikken [22]. Het geheel van deze optimaal georiënteerde objecten vormt dan ook im-

pliciet de persoonlijke ruimte van de gebruiker. De groepsruimte ontstaat typisch ergens centraal gelegen en beschikt over een oriëntatie die het resultaat is van een compromis tussen de gebruikers.

Ook naar **communicatie** toe, speelt de oriëntatie van objecten een duidelijke rol. Het gericht oriënteren van een object naar een andere gebruiker toe, is immers een indicator dat de gebruiker zijn collega wil inlichten over de data. Vanaf dat moment zal de gebruiker zich ook typisch verbaal richten tot zijn collega. Ook eventuele gebaren en lichaamstaal zijn op dat moment geadresseerd aan die persoon. Anderzijds nodigt een object dat zich voor beide gebruikers onder een aanvaardbare hoek bevindt tot samenwerking uit.



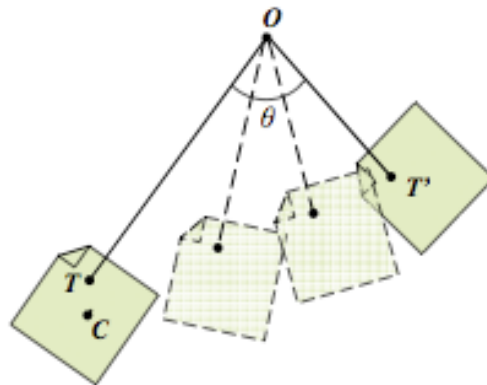
Figuur 3.6: Het onderhandelen van een groepsoriëntatie is een sociaal proces. [22]

Om er voor te zorgen dat oriëntatie ook daadwerkelijk zijn rol kan spelen in zake comprehensie, coördinatie en communicatie moeten een aantal zaken nauw in het oog gehouden worden. Het moet voor de gebruiker steeds mogelijk zijn om objecten vrij te roteren. Deze rotatietechnieken moeten bovendien eenvoudig en snel toegankelijk zijn voor de gebruikers.

### Automatic rotation

Bij automatische rotatie zal het systeem instaan voor de oriëntatie van de objecten. Het systeem zal proberen om een geschikte oriëntatie te bekomen door rekening te houden met de locatie van de andere objecten op de tafel enerzijds en de locatie van de gebruiker anderzijds. Men gaat er in dat geval van uit dat de locatie van de gebruiker overeenkomt met één van de zijden van de tabletop. Het systeem zal het object dan zo oriënteren dat de onderkant altijd gealigneerd is met de rand van de tafel en maakt hiervoor gebruik van poolcoördinaten. Deze aanpak levert dan ook een vrij begrijpbaar systeem op met een voorspelbaar gedrag. Volgens Hancock et al. [18] is dit ook meteen het zwakke punt van de techniek omdat slechts enkele positie- of rotatiecombinaties mogelijk zijn, namelijk deze die overeen komen met poolcoördinaten. Het systeem kan zowel discreet als continu geïmplementeerd worden. In het discrete geval zal rotatie enkel voorkomen wanneer een

grens, typisch een hoek, wordt overschreden. In het continue geval zal rotatie doorlopend gebeuren. Op figuur 3.7 wordt geïllustreerd hoe een object wordt geheroriënteerd door het systeem op basis van de hoek tussen het initiële raakpunt  $T$ , het centrum van de tafel en het eindpunt  $T'$ .



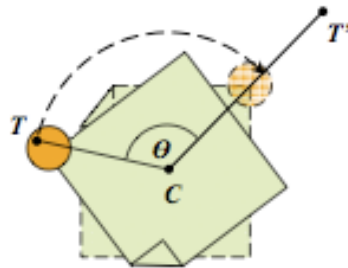
Figuur 3.7: Continue automatische rotatie. [18]

### Corner to rotate

'Corner to rotate' [18, 23] is een rotatietechniek die gebruik maakt van vooraf gedefiniëerde zones waarin de gebruiker kan overgaan tot rotatie. Typisch bevinden deze zones zich aan de hoeken van het object zodat deze hoeken fungeren als 'handles' maar dit is zeker geen vereiste. Rotatie gebeurt rondom het centrum van het object ( $C$ ) vervolgens door een fysieke translatie van  $T$  naar  $T'$  uit te voeren. Het object zal vervolgens roteren rond het centrum van het object zelve in de richting van de translatie. Op figuur 3.8 wordt dit concept geïllustreerd.

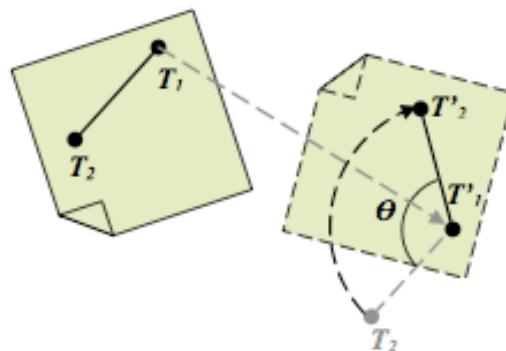
### Two-point rotation

Bij 'Two-point rotation' [18, 47] wordt gebruik gemaakt van twee contactpunten. Het eerst contactpunt ( $T1$ ) fungeert in dit geval als spil van de rotatie. Het tweede contactpunt ( $T2$ ) is nodig om de rotatiehoek te bepalen. Deze hoek wordt bepaald door het eindpunt van de translatie ( $T'1$ ) en de begin- ( $T2$ ) en eindpositie ( $T'2$ ) van het tweede contactpunt zoals te zien is op figuur 3.9. In de oplossing van Hancock et al. [18] moeten beide vingers het contact met het oppervlak behouden. In de oplossing van Wu et al [47] daarentegen, moet



Figuur 3.8: Gebruik van 'handles' op de hoeken om rotatie af te dwingen [18].

de spilvinger niet op het canvas blijven waardoor de bewegingsvrijheid van de gebruiker wordt vergroot.



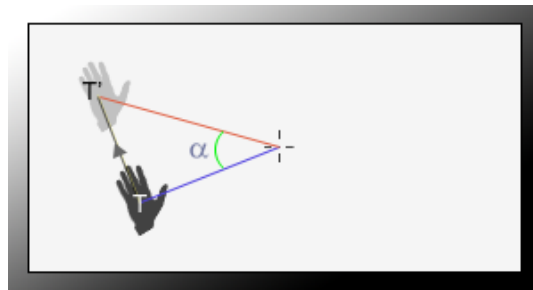
Figuur 3.9: 'Two point rotation' [18].

Verder leent 'two-point rotation' zich goed tot de integratie van scatering. De afstand tussen beide vingers kan immers gebruikt worden als factor voor de grootte van het object [18]. Een dergelijke techniek vereist een goede coördinatie van de gebruiker omdat hij een constante afstand moet bewaren tussen beide vingers [18]. Wu et al [47] merken bovendien op dat sommige objecten te klein zijn om via deze techniek gerooteerd te worden. Zij stellen daarom voor om in dat geval een 'bounding box' te creëren rond het object, zodat deze 'bounding box' vervolgens kan worden gerooteerd via 'Two-point rotation'.

### Flat hand

'Flat hand' rotatie [47] is een techniek waarbij de gebruiker zijn hand plat op het canvas legt en deze vervolgens verplaatst in om het even welke richting. Het gevolg van deze actie is dat de scene als geheel zal roteren. De grootte van de rotatie wordt bepaald door de

hoek alpha die wordt gevormd tussen het initiële raakpunt T, het centrum van de scene en het uiteindelijke punt van loslaten T' waarbij T en T' overeenkomen met het centrum van de hand. Dit concept wordt geïllustreerd op figuur 3.10 waarbij de zwarte pijl de translatie weergeeft van de hand. Het plaatsen van de hand dient enkel om de juiste rotatiemodus te selecteren, hierdoor moet de gebruiker zijn hand niet plat op het canvas houden terwijl hij de scene roteert. Het is voldoende dat hij contact behoudt met het oppervlak, dit zorgt voor een meer ergonomische houding tijdens de interactie volgens Wu et al. [47]. Een andere mogelijkheid zou erin bestaan dat de oriëntatie van de hand zou worden getraceerd zodat de scene op basis daarvan zou roteren. Echter dit mapt niet goed naar de fysieke situatie aldus Wu et al. [47].



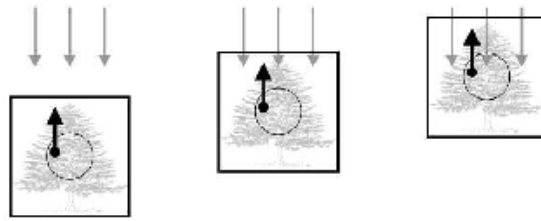
Figuur 3.10: 'Flat hand' rotatie.

### Rotate 'N Translate

Een methode die rotatie en translatie integreert is het 'Rotate 'N Translate' (RNT) van Krüger et al. [23, 18]. Dit is een interactievorm die de gebruiker toelaat om via één enkel raakpunt tegelijk te transleren en te roteren. Kinestetische studies tonen nochtans aan dat beide operaties onlosmakelijk verbonden zijn met elkaar. Desondanks verplicht de traditionele aanpak de gebruiker om translatie en rotatie typisch in twee sequentiële acties uit te voeren. Bij RNT daarentegen kunnen beiden wel simultaan gebeuren. Hiervoor baseerde de techniek zich op de fysica en maakt men gebruik van frictie om een object al dan niet te laten roteren. Deze frictie ontstaat als gevolg van een denkbeeldige stroom die inwerkt op het object. De richting van deze stroom is steeds omgekeerd aan de richting van de beweging.

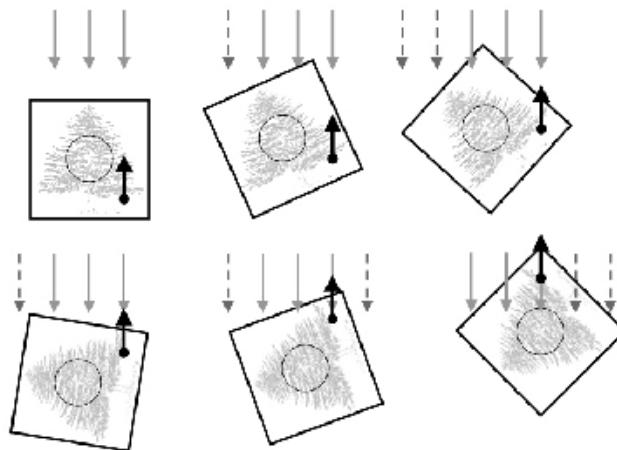
Concreet betekent dit dat wanneer de richting van bewegen wijzigt, de richting van de stroom mee zal wijzigen. Het is niet eenvoudig voor de gebruiker om deze balans te vinden, daarom werd een translatiezone voorzien op de objecten. Deze zone wordt aangeduid door de grijze cirkel in het midden van het object zoals te zien is op figuur 3.11.





Figuur 3.11: RNT, balans tussen stroom en beweging. [23]

Bij een onbalans zal het object beginnen te roteren. Op figuur 3.12 wordt een opwaartse beweging geïnitieerd in de rechterbenedenhoek. Hierdoor ontstaat een grote onbalans en zal de stroom inwerken op de bovenkant van het object waardoor een contrakloksgewijze rotatie optreedt. Des te groter deze onbalans des te meer het object zal roteren. De rotatie stopt echter wanneer er terug een balans ontstaat tussen de bewegingsvector en de stroom.



Figuur 3.12: RNT, onbalans met rotatie als gevolg. [23]

Krüger et al. [23] hebben ontdekt dat RNT superieur is aan het traditionele 'Corner to rotate'. RNT blijkt immers minder 'touches' nodig te hebben om tot goede resultaten te komen. Bovendien is RNT tijdsefficiënter dan 'Corner to rotate'. Gebruikers hebben bovendien een lichte voorkeur voor RNT te meer omdat ze vinden dat de techniek makkelijk is aan te leren en vlot in gebruik is. Eén van de mogelijke problemen die opduiken bij RNT is dat de uit te voeren taak dikwijls vereist om ook ter plaatse kleine aanpassingen te doen aan de oriëntatie van het object, helaas behoort onafhankelijke rotatie niet tot de mogelijkheden in RNT [23].

### 3.3 Precisie

In desktopsystemen is precisie tot op één pixel mogelijk via de cursor dankzij invoerapparaten als de muis en de trackball. Ook in tabletopsystemen moet men streven naar een dergelijke precisie. Een naïeve benadering zou kunnen zijn deze invoerapparaten te integreren om diezelfde nauwkeurigheid te realiseren. Wil men echter de sterkte van de tabletop ten volle uitbuiten, is dit geen verstandige keuze. De tabletop interface brengt immers invoer en uitvoer samen op één oppervlak waardoor de gebruikers dit soort interfaces als zeer intuïtief ervaren. Helaas is de resolutie van onze vingers te beperkt om doelwitten die een stuk kleiner zijn dan onze vingertop te selecteren [1]. Gelukkig zijn er in de laatste jaren reeds een aantal technieken ontwikkeld om het probleem aan te pakken. Deze sectie bekijk ik een aantal van deze technieken van naderbij.

#### 3.3.1 Manipulatie van de cursor

Een invalshoek om dit probleem op te lossen is om te proberen het gebrek aan precisie te omzeilen. Dit kan door de cursor aan te passen op zo een manier dat hij zich anders zal gedragen dan de traditionele desktopcursor.

##### Object pointing

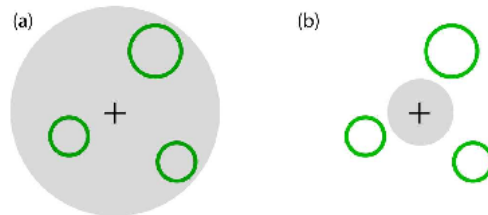
Een benadering die dit gebrek aan nauwkeurigheid in feite omzeilt is 'Object pointing'. Doordat de cursor bij 'Object pointing' zelf objecten zal opzoeken is de gebruiker niet meer zelf verantwoordelijk om dit te doen waardoor het gebrek aan nauwkeurigheid niet meer aan de orde is. Het is een techniek die werd ontwikkeld door Guiard et al. [15] die er vanuit gaat dat de ruimte tussen objecten enkel voor vertraging zorgt. Wanneer twee objecten ver van mekaar verwijderd liggen, moet de gebruiker immers een grote afstand overbruggen. Vandaar heeft men de 'Tim'-cursor geïmplementeerd. Dit is een cursor die deze afstand negeert en van object naar object zal springen.

'Object pointing' doet het goed in situaties waarin de objecten ver van mekaar verwijderd liggen en in situaties waarbij de doelwitten erg klein zijn. Echter de mate waarin 'Object pointing' het beter doet dan de traditionele manier van werken is sterk afhankelijk van de objectdichtheid. Als deze sterk toeneemt blijkt immers dat 'Object pointing' en de traditionele manier van werken mekaar weinig ontlopen [15, 13].

##### Area cursor

In tegenstelling tot de traditionele cursor die uitgaat van één enkel activatiepunt, gebruiken 'Area cursors' een groter activatiegebied. Hierdoor wordt het makkelijker om kleine objecten te selecteren [46]. Ook deze aanpak krijgt helaas problemen wanneer de objectdichtheid toeneemt. De 'Area cursor' zal immers onvoorspelbaar worden in het geval er zich meerdere objecten onder het activatiegebied bevinden. Het is daarom belangrijk te

bepalen welke grootte de 'Area cursor' moet aannemen in functie van de situatie waarin hij wordt gebruikt. Op figuur 3.13 is duidelijk te zien wat het resultaat is van het hanteren van een verkeerde cursorgrootte.

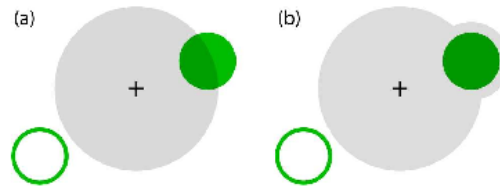


Figuur 3.13: Cursor-grootte 'Area-cursor'. (a) cursorgrootte te omvangrijk. (b) De cursorgrootte te klein [13].

Om tegemoet te komen aan deze tekortkoming is de 'Area cursor' voorzien van een crosshair die zich centraal in het cirkelvormige activatiegebied bevindt. Deze zal fungeren als een traditionele cursor wanneer er zich meerdere objecten onder de 'Area cursor' bevinden. Helaas is deze oplossing in dit geval niet toereikend in tabletop-interfaces daar de gebruiker terug wordt geconfronteerd met een gebrek aan nauwkeurigheid. Een bijkomend probleem dat opduikt is dat de cursor een deel van de scene bedekt waardoor enkele objecten uit het zicht kunnen verdwijnen. Vandaar heeft men de 'Area cursor' grotendeels transparant gemaakt waardoor de performantie niet in het gedrang komt [46].

### Bubble cursor

De 'Bubble cursor' die werd ontwikkeld door Grossman et al. [13] is eveneens een soort van 'Area cursor' [46] maar is meer gesofisticeerd in die zin dat de 'Bubble cursor' zijn activatiegebied dynamisch zal aanpassen aan de hand van de dichtheid van de objecten. Op die manier is de 'Bubble cursor' minder kwetsbaar voor het voorkomen van meerdere objecten onder het activatiegebied. Net zoals de 'Area cursor' is de 'Bubble cursor' cirkelvormig, transparant en bevat deze een crosshair in het midden om de huidige locatie van de standaard cursor aan te geven. Wanneer er zich meerdere objecten onder het activatiegebied bevinden, worden de intersecties berekend met deze objecten. Aan de hand van deze intersecties zal de 'Bubble cursor' de straal van het activatiegebied aanpassen zodat slechts één object hetzij volledig of hetzij deels onder het activatiegebied terechtkomt.



Figuur 3.14: De Bubble cursor. [13].

### 3.3.2 Gestures

Nadat ik het in de vorige sectie het gehad heb over manipulatie van de cursor, ga ik in deze sectie verder met gesture gebaseerde technieken. Deze technieken kunnen zowel single- als multitouch zijn maar hebben allen gemeen dat zij een oplossing trachten te bieden voor het gebrek aan precisie. Eerst bespreek ik enkele technieken die gebruik maken van zooming, vervolgens komen enkele technieken aan bod die gebruik maken van visuele hulpmiddelen en als laatste bespreek ik nog enkele technieken die gebruik maken van de ruimtelijke relaties tussen de vingers van de gebruikers.

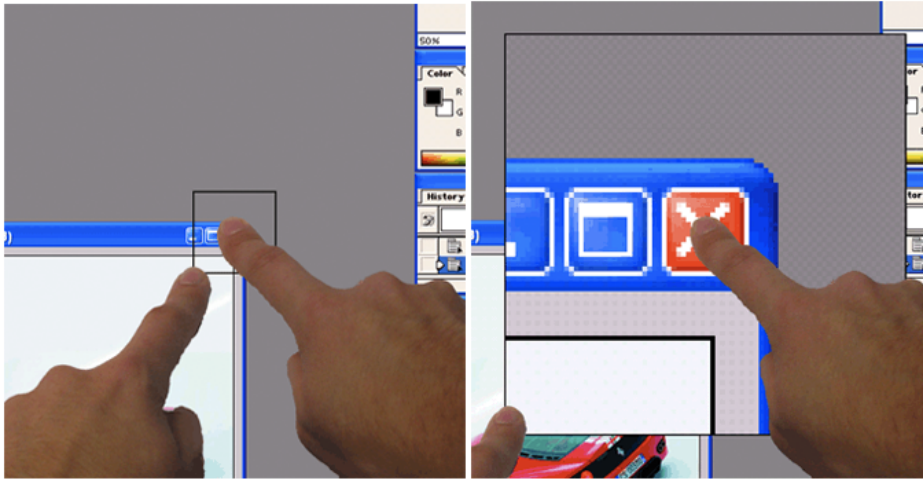
#### Bounding box zoom

Een voordehandliggende oplossing is om in te zoomen op de te selecteren objecten zodat deze makkelijker kunnen geselecteerd worden. De 'Bounding box zoom' die wordt besproken door Albinsson et al. [1], werkt volgens dit principe. Deze techniek laat de gebruiker toe om met zijn wijsvinger een diagonaal te slepen over het te selecteren object zodat een 'bounding box' tot stand komt. De gebruiker geeft op die manier aan op welk zoomniveau hij het object wil bekijken. Het systeem zal vervolgens inzoomen tot op het gewenste niveau. Ondanks dit een zeer handig mechanisme blijkt, is het probleem dat de context waarin het object zich bevindt, totaal verloren gaat. Dit kan worden aangepakt door een extra overzichtswaergave toe te voegen maar het wisselen tussen waergaven blijkt eerder cognitief belastend te zijn voor de gebruikers.

#### Dual finger stretch

Het 'Dual finger stretch' [4] concept is nagenoeg identiek aan de 'Bounding Box Zoom' [1]. Het enige verschil is hier dat de gebruiker kan inzoomen door eerst met zijn primaire vinger het doelgebied aan te geven en vervolgens zijn secundaire vinger erlangs te plaatsen en weg te bewegen van de primaire vinger. Daar beide technieken vergelijkbaar zijn,

hebben ze ook min of meer dezelfde sterktes en zwaktes. Echter in tegenstelling tot de techniek 'Bounding box zoom' gebeuren selectie en zooming hier wel gelijktijdig omdat ze zijn geïntegreerd in één actie.



Figuur 3.15: Dual finger stretch. Links voor het inzoomen, rechts na inzoomen [4].

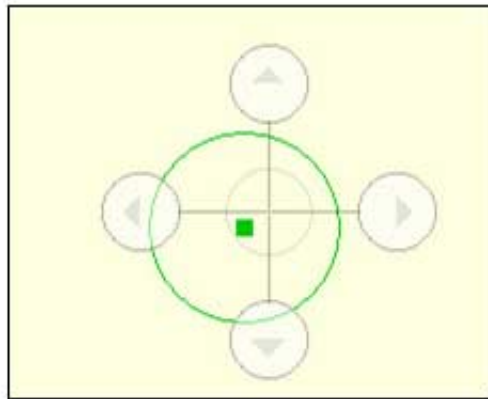
### Cross lever

Een andere aanpak die bemoedigende resultaten levert naar extra precisie toe is de 'Cross lever' [1]. Het concept is eenvoudig, de gebruiker beschikt over twee gekruiste lijnen die worden geactiveerd bij het aanraken van het scherm. De cursor bevindt zich op de intersectie van de lijnen en kan worden bestuurd door de handles van de lijnen te manipuleren zoals te zien is op figuur 3.17a. Extra precisie wordt verkregen door de lengte van de lijnen te vergroten, op die manier verkrijgt men een hefboom-effect dat zorgt voor extra controle. De effectieve selectie gebeurt vervolgens door in een 'halo' te klikken die zich rondom de intersectie bevindt. Deze techniek heeft helaas een grote tekortkoming, ze kan namelijk niet gebruikt worden in de hoeken van het interactievlak. Shen et al. [41] stellen als oplossing voor dit probleem voor dat het beeld kleiner wordt geprojecteerd dan de grootte van de tafel. Op die manier blijft er immers nog ruimte over om in de hoeken te kunnen werken.

### Virtual keys

De 'Virtual keys' techniek die wordt voorgesteld door Albinsson et al. [1], is gebaseerd op het gebruik van een crosshair die kan worden bestuurd aan de hand van een widget dat zich ergens aan de rand van het scherm bevindt. Helaas blijkt dit voor de nodige 'eye gaze' te zorgen omdat de gebruiker zijn ogen steeds moet afwenden van het doelwit om de crosshair te kunnen besturen [1]. Een kleine aanpassing bestaat erin om het widget te laten

samen vallen met de crosshair zodat de gebruiker het object rechtstreeks kan manipuleren. Deze techniek blijkt effectief te zijn om erg kleine objecten te selecteren met een relatief kleine foutenmarge omdat de gebruiker de kans heeft om de cursor in discrete stappen te besturen.



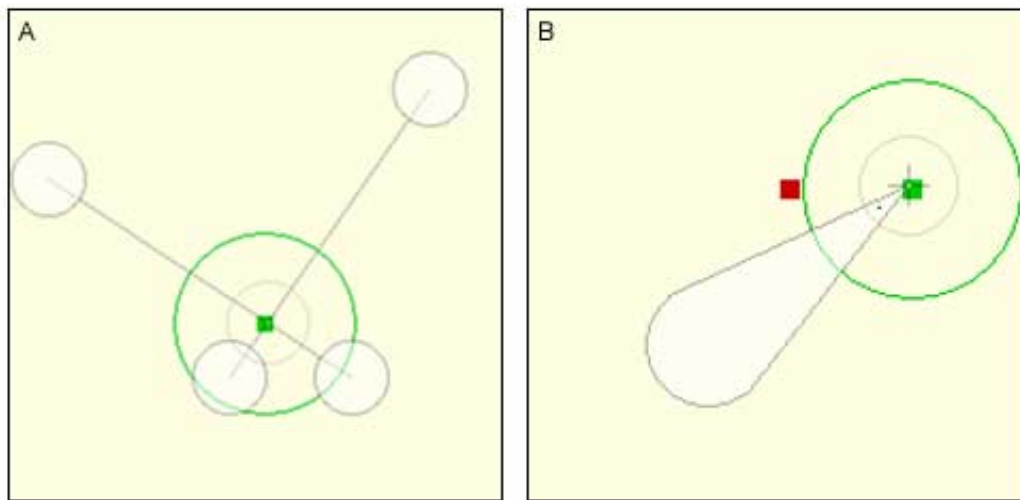
Figuur 3.16: Virtual keys met handles op de crosshair. [1].

### Precision handle

Deze techniek maakt gebruik van een hendel die scharniert rondom een draaipunt. Op de punt van de hendel is een crosshair geplaatst die fungeert als punt-cursor. De 'Precision handle' [1] wordt geactiveerd door in de omgeving van het te selecteren object te klikken. Hierna zal de hendel verschijnen en kan de gebruiker de punt-cursor besturen door de hendel te manipuleren. Vanwege het hefboom-effect dat optreedt door de hendel rond het draaipunt te roteren, kan de gebruiker vrij precies werken. De lengte van de hendel kan overigens worden aangepast door de tip van de vinger korter bij of verder af te bewegen van het doelobject.

### Dual finger X-menu

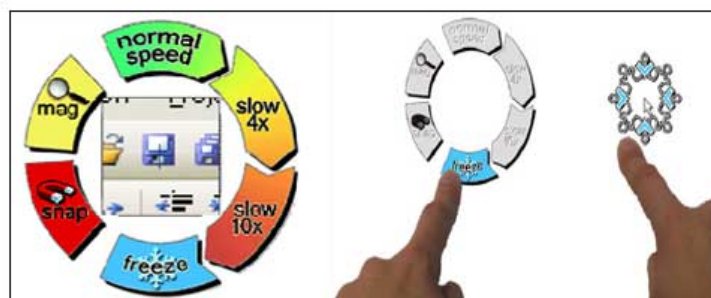
Een buitenbeentje tussen deze 'multi-finger' technieken is het 'dual-finger X-menu' [4]. Deze techniek is eveneens gebaseerd op het gebruik van een crosshair, gekoppeld aan de primaire vinger van de gebruiker. Men introduceert echter een menu dat extra functionaliteit en ondersteuning moet bieden aan de gebruiker. Dit menu wordt geactiveerd van zodra de gebruiker zijn secundaire vinger op het canvas plaatst. Het 'Dual finger X-menu' wordt vervolgens rond de secundaire vinger geprojecteerd en de aangeboden functies kunnen worden geactiveerd door de secundaire vinger te verplaatsen over één van de zones. Naast de mogelijkheid om in te zoomen met een factor twee, bestaat ook de mogelijkheid om de snelheid van de cursor aan te passen. Dit kan in drie discrete stappen, waardoor



Figuur 3.17: a. Cross lever b. Precision handle [1].

de gebruiker beschikt over extra controle over de cursor. Op die manier zal hij preciezer kunnen werken.

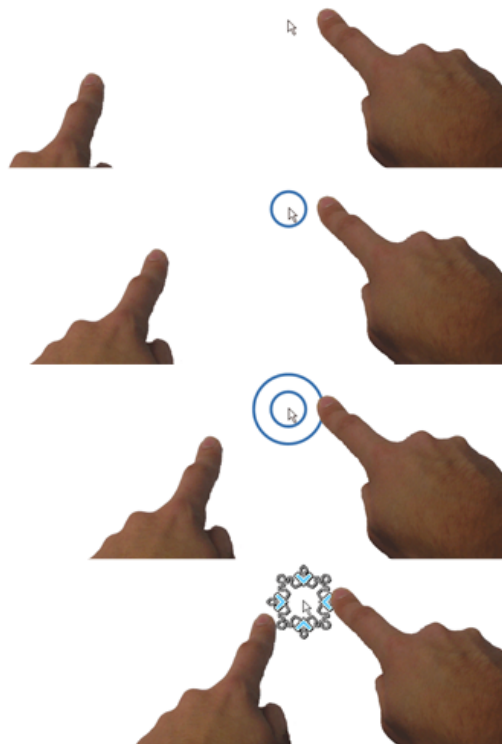
De snelheid waarmee de cursor beweegt wordt steeds aangegeven aan de hand van een projectie van één of meerdere ringetjes rond de cursor. Het is immers primordiaal de gebruiker ten alle tijde te informeren over de staat van het systeem [4, 41]. Indien dit nog niet tegemoet komt aan de eisen van de gebruiker, heeft deze nog steeds de optie om de cursor te bevriezen. Na activatie van deze 'freeze'-functie verschijnt een soort van crosshair die uitgerust is met pijltjes zoals te zien is op figuur 3.18. Met deze pijltjes kan men vervolgens de cursor pixel per pixel verplaatsen waardoor selectie tot op één pixel tot de mogelijkheden behoort.



Figuur 3.18: Dual finger X-menu. [4].

### Dual finger slider

Een variant op deze manier van werken, laat dit menu achterwege maar behoudt de functionaliteit ervan. In plaats van expliciet de gewenste functie aan te duiden in het 'X-menu', kiest men ervoor om de functionaliteit ervan te koppelen aan de afstand tussen beide vingers. Dit gebeurt opnieuw in discrete niveaus. Naarmate de afstand tussen beide vingers kleiner wordt, vertraagt de cursor met als eindfase het bevroren van de cursor en vice versa (zie figuur 3.19). Merk op dat deze methode erop vertrouwt dat de gebruiker steeds de afstand tussen beide vingers kan inschatten. Vandaar wordt het huidige niveau nog eens extra aangegeven via de concentrische cirkels die zich rond de cursor bevinden [4].



Figuur 3.19: Dual finger Slider. [4].

### Dual finger offset

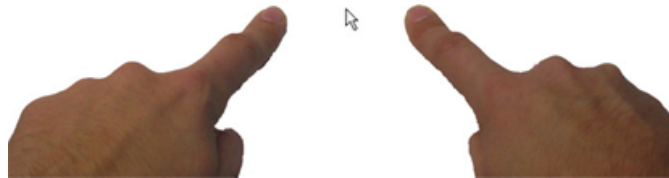
Dit is een eerder rudimentaire aanpak die uitgaat van een cursor die zich bevindt onder de wijsvinger. Als de gebruiker echter een tweede vinger op de tabletop plaatst, zal de cursor zich verplaatsen naast de vingertop van de primaire wijsvinger. De afstand tussen wijsvinger en cursor is vooraf bepaald en zal niet wijzigen. Concreet betekent dit dat als de gebruiker linkshandig is en hij zijn rechterwijsvinger op het canvas plaatst, de cursor zich



met een vaste 'offset' verplaatst naar rechts en vice versa. Het voordeel van deze techniek is dat de cursor zich niet meer onder de vinger bevindt waardoor de gebruiker de cursor effectief kan zien en zo een beter zicht heeft op de positie van de cursor [4]. Hierdoor kan hij preciezer tewerk gaan. Bovendien werkt deze techniek ook behoorlijk in de hoeken van de tabletop interface omdat de gebruiker altijd kan omschakelen tussen rechter- en linkerwijsvinger [4].

### Dual finger midpoint

Een tweede, meer gesofisticeerde benadering gaat uit van hetzelfde opzet, met dat verschil dat de cursor zich nu tussen beide vingers bevindt zoals te zien is op figuur 3.20. De toegevoegde waarde van deze techniek zit hem in de extra controle die de gebruiker krijgt over de cursor, maar eveneens in de extra functionaliteit die aan 'Dual finger midpoint' [4] verbonden is. Zo is het mogelijk om de snelheid waarmee de cursor beweegt in discrete stappen te regelen. Het concept is zeer eenvoudig. Zolang beide vingers bewegen in éénzelfde ritme, verplaatst de cursor zich op volle snelheid. Wanneer de gebruiker extra controle wenst, kan hij over gaan tot het verplaatsen van slechts één vinger. Dit zorgt ervoor dat de snelheid van de cursor halveert. Ondanks dat blijkt de techniek erg vlot aan te leren en kan hij op veel bijval rekenen van de gebruikers [4], voldeed hij niet op het kleinste niveau.



Figuur 3.20: Dual finger midpoint. [4].

## 3.4 Lange afstanden

Traditionele invoerapparaten als de muis en de trackball zijn niet geschikt voor 'large screen displays' [7] want het is best vermoeiend om er grote afstanden mee af te leggen. Er is dus nood aan een andere aanpak. De gebruikte apparatuur en bijbehorende technieken moeten schaalbaar zijn. Dit zal immers bepalend zijn voor de performantie. In de komende sectie komen een aantal mogelijke oplossingen aan bod.

### 3.4.1 Manipulatie vanop een afstand

#### Laserpointing

Laserpointing [7] is een techniek die de gebruiker toelaat om data op het scherm rechtstreeks te kunnen manipuleren via de laserpointer. Het concept is gebaseerd op de laserpointer die wordt gebruikt tijdens presentaties om iets extra onder de aandacht te brengen. Zo kan de presentator zones die buiten zijn bereik liggen, makkelijk bereiken. Davis et al. [7] hebben op basis hiervan een systeem ontwikkeld dat de gebruiker toelaat om via de laserpointer te interageren met grote schermen. Hun systeem gebruikt een reeks camera's die zich bevinden aan de achterzijde van het scherm. Deze camera's registreren de bewegingen van de laserpointer. Het systeem maakt voor de invoer van data verder gebruik van gestures via de 'Unistroke'-toolkit van Goldberg et al. [12]. Data invoer kan anderzijds ook gebeuren via het toevoegen van extra knoppen zoals Cavens et al. [5] voorstellen.

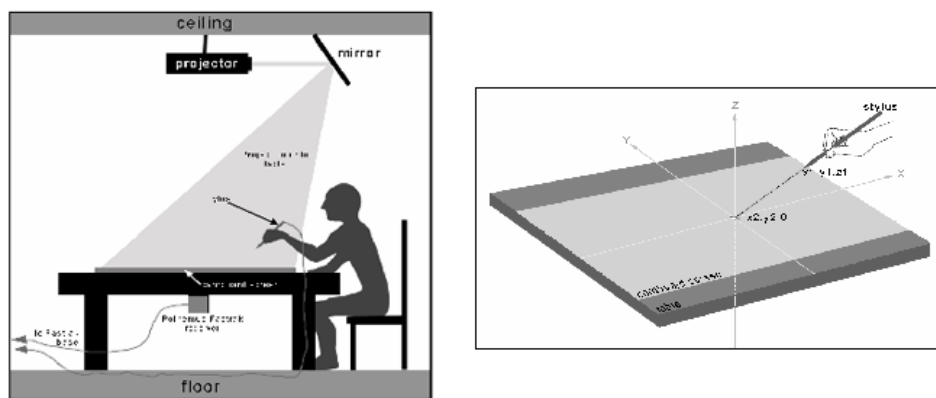
#### TractorBeam

Parker et al. [30, 31] stellen een oplossing voor die gebruik maakt van een stylus om selecties uit te voeren. De 'TractorBeam' is eigenlijk een combinatie van 'touch' en 'remote pointing' die probeert om de kwaliteiten van beide interactiemodi uit te buiten en ze te integreren in één hybride techniek. Op die manier kan de gebruiker dichtbij objecten selecteren via de stylus maar ook verder afgelegen objecten selecteren door ernaar te wijzen met de stylus. De gebruiker moet zich dan niet nodeloos verplaatsen. Hij kan immers vanop zijn huidige locatie een virtuele straal uitsturen naar zijn doelwit. Op de tabletop verschijnt vervolgens een cursor op de plaats waar de virtuele straal en de tabletop interacteren zoals te zien is op figuur 3.21. Registratie van de bewegingen van de pen gebeurt via het 'fastrak' systeem van Polhemus<sup>TM</sup> dat toelaat om bewegingen in real time te tracken. Nadeel hiervan is wel dat de styli zijn verbonden met kabels, zodat de bewegingsvrijheid wordt beperkt door de lengte van de kabel.

### 3.4.2 Manipulatie van de omgeving

#### Drag, pick and pop

Net als Pick and drop [34] is drag and pop [3] een pengebaseerde techniek die ervoor zorgt dat gebruikers ook vanop afstand kunnen interageren met iconen die anders buiten



Figuur 3.21: TractorBeam. Links, hardware setup en overview. Rechts, het TractorBeam-concept [30].

het bereik van de gebruiker zouden liggen. Net als bij het traditionele drag and drop zet de gebruiker zijn actie in door zijn icoon te slepen in de richting van het einddoel. Bij drag and pop, verplaatsen de potentiële doelen zich tijdelijk naar de directe omgeving van het object. Op die manier moet de gebruiker geen grote afstanden overbruggen. Deze pseudoiconen worden via een rubberbandmetafoor verbonden met de originele iconen zoals te zien is op figuur 3.22.



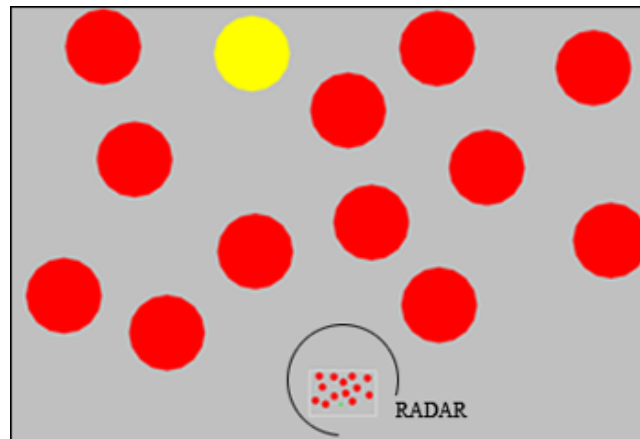
Figuur 3.22: Pick and pop en het gebruik van rubber bands [3].

Bij Pick and pop hanteert men hetzelfde concept om selecties uit te voeren vanop een afstand. In plaats van iconen te slepen, volstaat het nu om de cursor in de richting van

de interesseregio te verplaatsen. De iconen uit deze regio zullen vervolgens verschijnen in de directe omgeving van de cursor. Het voordeel van de technieken van Baudisch is dat ze geen fysieke verplaatsing vereisen in tegenstelling tot het 'Pick and drop' van Rekimoto en het 'Taking' van Geißler [24]. Deze komen nog verder aan bod in deze sectie.

### Radar

Een vrij eenvoudige maar erg efficiënte en accurate techniek is de 'Radar' techniek die wordt beschreven door Reetz et al. [33]. Deze proxy-techniek biedt de gebruiker een miniaturweergave van de volledige scene binnen handbereik. Binnen deze miniaturweergave of radar kan de gebruiker eveneens verplaatsingen doorvoeren, zodat de gebruiker op een eenvoudige manier objecten kan bedienen die anders buiten zijn bereik liggen. Deze techniek is ook vrij snel maar wordt uiteraard trager naar mate het de dichtheid in de scene toeneemt. Hierdoor wordt het immers moeilijker om de objecten in het radar widget te selecteren. Een ander nadeel van de 'Radar' is dat objecten nog moeilijk van elkaar te onderscheiden zijn omdat ze te sterk zijn verkleind.

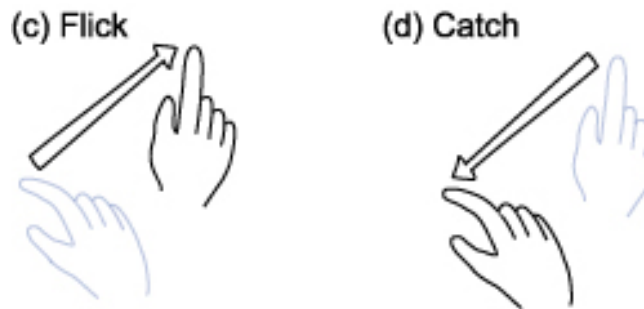


Figuur 3.23: De radar bevindt zich centraal onderaan en bevindt zich binnen het bereik van de gebruiker. [33]

### Flick and catch

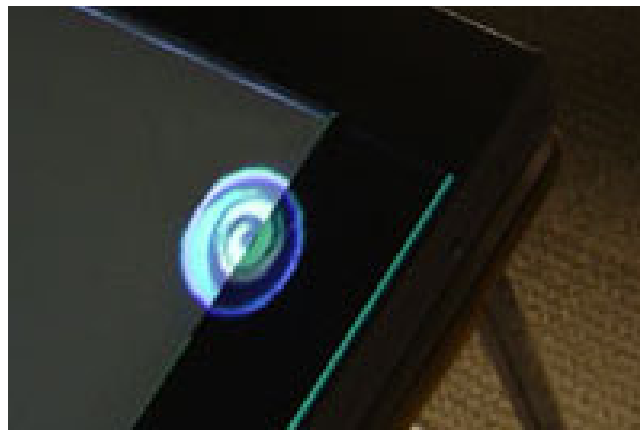
Het werpen van documenten in de fysieke wereld is afhankelijk van verschillende parameters zoals het gewicht van het object, de kracht waarmee het werd beroerd, de richting van de kracht en de wrijving tussen object en oppervlak. Deze factoren bepalen samen de snelheid en richting van het object.

De flicking techniek van Wu et al. [47] probeert dit gedrag te simuleren. Op die manier ontstaat een techniek die de gebruiker toelaat virtuele objecten te werpen over grote



Figuur 3.24: Flick and catch. [47]

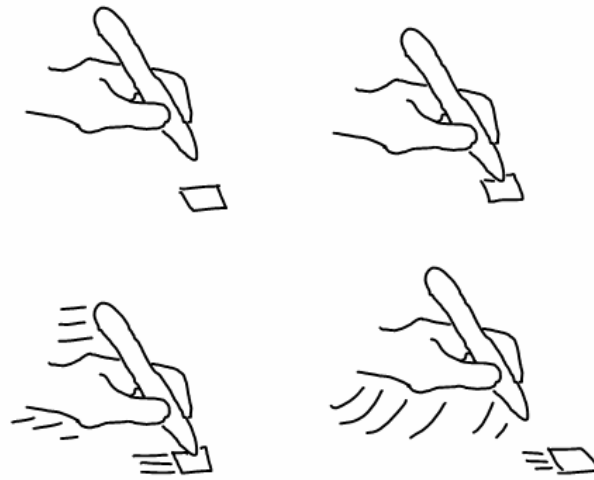
afstanden. Wanneer de gebruiker nu de 'flick' gesture inzet door een object een bepaalde snelheid mee te geven, zal het object zich blijven verder bewegen tot het via het 'catch' gesture wordt gevangen of strandt op één van de randen zoals te zien is op figuur 3.25. Dit concept wordt duidelijk geïllustreerd op figuur 3.24.



Figuur 3.25: Object werd niet gevangen met het catch gesture. [47]

### Superflick

Superflick is een uitbreiding op 'Flick and catch' van Wu et al. [47] waarbij de gebruiker na het flicken het object nog kan bijsturen via remote 'drag and drop'. Het gebeurt immers vaker dat gebruikers merken dat het object zich niet helemaal in de juiste richting verplaatst. Via de 'superflick' techniek van Reetz et al. [33] kan de gebruiker het object nog manipuleren door tijdens het reizen van het object, de pen terug op het interactievlak te plaatsen zoals te zien is op figuur 3.26. Vanaf dan kan het object worden bijgestuurd door de pen in de gewenste richting te bewegen.



Figuur 3.26: Superflick. [33]

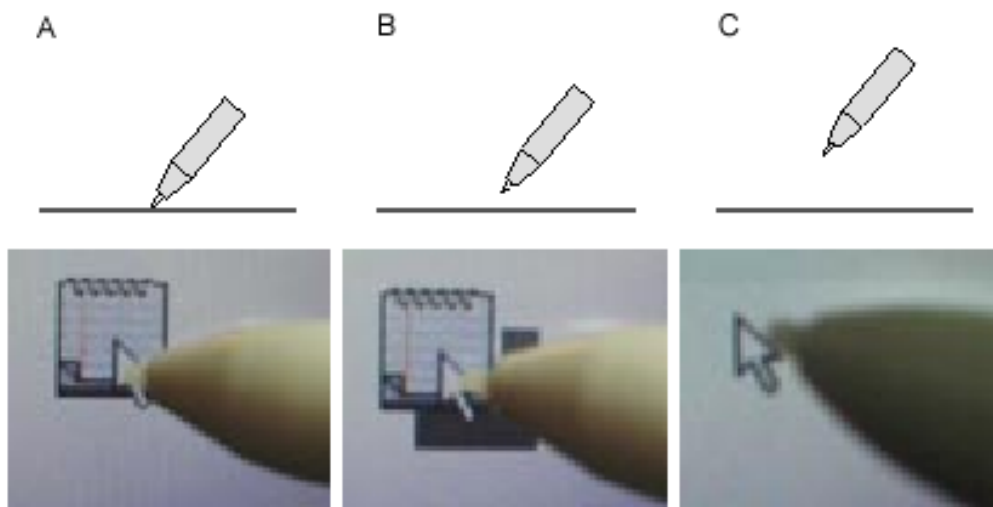
### 3.4.3 Fysieke verplaatsingen

#### Pick and drop

De vorige twee technieken waren gebaseerd op het gebruik van pointing technieken. 'Pick and drop' daarentegen, werd door Rekimoto [34] ontwikkeld en vertrekt vanuit een andere invalshoek. Hij presenteert een op de stylus gebaseerde interactietechniek die de gebruiker toelaat om objecten van het scherm te nemen en deze verderop terug te plaatsen. Dit doet hij door het object als het ware vast te nemen met de stylus. Het opnemen van een object gebeurt door het object aan te tikken met de stylus waar na het object verdwijnt van het canvas. Echter wanneer de gebruiker de stylus terug in de buurt van het canvas brengt, verschijnt de schaduw van het object op de tafel. Wanneer de gebruiker zijn pen terug op het canvas plaatst zal het object op de nieuwe locatie verschijnen. In tegenstelling tot de 'remote' pointing technieken uit de vorige secties vereist deze techniek wel dat de gebruiker zich effectief verplaatst om objecten over langere afstanden te verplaatsen. Voordeel is wel dat deze aanpak duidelijk zichtbaar is voor de andere gebruikers en dus geen extra communicatie vereist om duidelijk te maken dat een bepaald object in gebruik is.

#### Taking

Een variant op 'Pick and drop' [34] is het 'Taking' van Geißler [24]. Het concept is vergelijkbaar met dat van Rekimoto, maar Geißler maakt geen gebruik van de stylus om een object vast te nemen. Hij laat de stylus achterwege en geeft de gebruiker de mogelijkheid om een object van het canvas te nemen door zijn hand een halve seconde lang plat op één van de handles van het object te plaatsen. Hierna zal het object krimpen zodat het



Figuur 3.27: Pick and drop. Aantikken object (A), Opnemen object (B), Stylus is weg van het canvas [34].

verdwijnt onder de hand van de gebruiker. Vanaf dat moment kan men het object mee nemen naar een andere locatie op het scherm. Wanneer de gebruiker even later zijn hand terug op het canvas plaatst zal het object opnieuw verschijnen. Op die manier laat Geißler zien dat deze techniek eveneens bruikbaar is in een omgeving zonder styli.

### 3.5 Tekstuele invoer

Ook in tabletop systemen speelt tekstuele invoer nog steeds een belangrijke rol. In deze sectie beschrijf ik enkele technieken om het toetsenbord te vervangen in tabletop omgevingen. Het invoeren van tekst op tabletop interfaces is bovendien noodzakelijk om ervoor te zorgen dat tabletops hun weg vinden in het dagelijks leven. Het invoeren van tekst is immers een basisvereiste om met veel van onze huidige software te kunnen werken. In desktopsystemen grijpen we naar ons toetsenbord om tekst in te geven of objecten te labelen. Echter voor tabletop systemen is dit geen goede oplossing daar het onduidelijk is waar een toetsenbord dan juist thuis hoort. Bovendien is het niet duidelijk hoe een dergelijk toetsenbord wordt gedeeld tussen de gebruikers. Overigens past het integreren van een toetsenbord niet in de filosofie van 'direct touch'-interactie. De illusie dat men niet meer met een computer te maken heeft, gaat dan verloren. In deze sectie beschrijf ik enkele mogelijkheden inzake projectie toetsenborden. Een grote tekortkoming van dit soort projectietoetsenborden is dat er een gemis is aan tactiele feedback [1], waardoor ze gevoeliger zijn aan fouten.

### 3.5.1 Virtuele keyboards

#### Soft keyboards

'Soft keyboards' zijn de virtuele tegenhangers van onze fysieke toetsenborden. Een dergelijk toetsenbord wordt geprojecteerd en kan worden gebruikt door achtereenvolgens de nodige letters aan te raken. Dit is voor de gebruikers eenvoudig te begrijpen daar dit perfect mapt naar het fysieke toetsenbord. Uiteraard biedt een 'soft' toetsenbord ook meer vrijheid dan zijn fysieke tegenhanger. Men kan immers vrij snel de layout van de toetsen aanpassen. Het is bewezen dat er layouts bestaan die efficiënter zijn dan de Qwerty standaard, maar Hinrichs et al.[19] wijzen erop dat deze alternatieve layouts niet zijn ingeburgerd bij het gros van de gebruikers. Alvorens een dergelijke alternatieve layout zal renderen, zal de gebruiker een inspanning moeten leveren om zich de nieuwe layout eigen te maken en dat kost tijd. Op figuur 3.28 is overigens goed te zien dat ook de vorm van het toetsenbord evenals de vorm van de toetsen vrij kunnen worden gekozen. Bovendien neemt een 'soft keyboard' ook plaats in op het interactievlak waardoor deze ruimte niet kan worden aangewend om te interageren met de data.

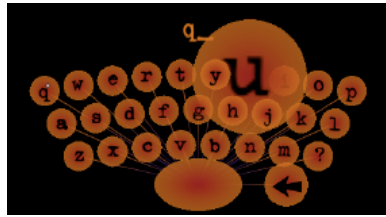


Figuur 3.28: Soft keyboards. A) Traditionele QWERTY. B) Metropolis layout [19].

#### BubbleQuery

'BubbleQuery' is een meer geavanceerde versie van het 'soft keyboard' die probeert enkele van de tekortkomingen van het 'soft keyboard' aan te pakken. Vooreerst integreerde men [19] een mechanisme voor het openen en sluiten van het toetsenbord. Op die manier neemt het toetsenbord al geen kostbare ruimte in die gebruikt kan worden door de data. Daarenboven implementeerde Hinrichs et al. [19] een algoritme dat op basis van statistische gegevens berekent hoe waarschijnlijk het is dat een bepaald karakter zal voorkomen op de volgende positie van het woord dat door de gebruiker op dat moment wordt gevormd. Op basis van deze berekeningen worden de potentiële karakters vervolgens groter voorgesteld in de 'keyboard' lay-out.



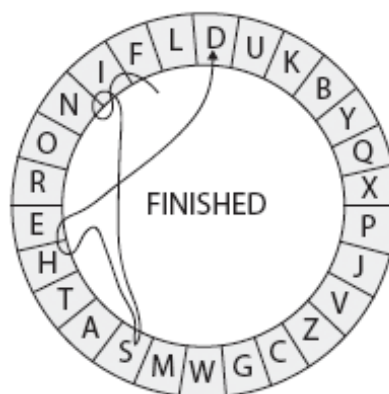


Figuur 3.29: BubbleQuery magnificatie functie. [19].

Op figuur 3.29 is te zien hoe eerst de letter 'Q' werd ingegeven met als gevolg dat de letter 'u' sterk wordt vergroot daar deze naar alle waarschijnlijkheid het volgende karakter zal worden in het te vormen woord. Merk verder op dat de toetsen van het dit toetsenbord transparant zijn, zodat ze er niet voor zorgen dat andere toetsen uit het zicht verdwijnen na magnificatie.

### 3.5.2 Gesture-based keyboards

In tegenstelling tot de eerder traditionele 'Soft keyboards' spreken Hinrichs et al. [19] ook nog over 'keyboards' die zijn gebaseerd op 'strokes'. De gebruiker kan woorden vormen door de stylus in één vloeiende beweging van letter naar letter te bewegen. Op figuur 3.30 wordt dit concept aangetoond. Het nadeel van een dergelijke layout is dat de letters zich eerder ver van mekaar bevinden. Er is immers een trade-off aanwezig tussen de selectiegebieden rond de letters en de afstand tussen de letters (straal). Een bijkomend nadeel van deze strategie is dat een deel van de letters kunnen verdwijnen onder de handpalm en/of arm van de gebruiker.



Figuur 3.30: Gesture-based keyboard (Cirrin). [19].

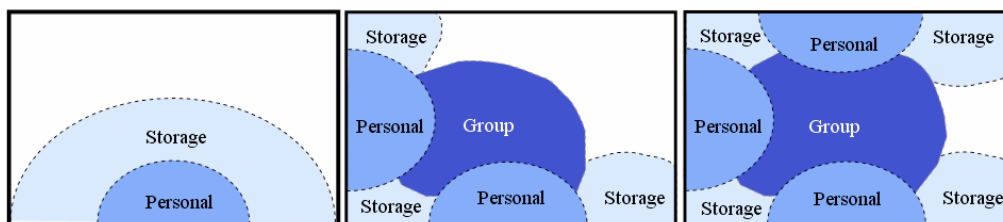
## 3.6 Collaboratieve systemen

In deze sectie bespreek ik op welke manier gebruikers samenwerken in collaboratieve tabletop omgevingen. Telkens wanneer een object moet worden gedeeld tussen de gebruikers ontstaan er immers conflicten. Het afhandelen van deze conflicten kan via de sociale etiquette van de gebruikers gebeuren. Deze sociale protocollen kunnen wel eens tekort schieten, waardoor er een nood ontstaat naar een technische oplossing voor het probleem. Eerst bespreek ik in de sectie 'Territorialiteit' op welke manier gebruikers omgaan met de beschikbare ruimte op de tabletop.

Het is niet altijd even duidelijk in een dergelijke omgeving welke ruimte beschikbaar is voor gebruik. Het kan best dat bepaalde gebruikers zich een deel van het interactieoppervlak toe eigenen. Er moet ook worden samengewerkt in de groepsruimte en dat zorgt hoe dan ook voor conflicten tussen de gebruikers. Logisch, want zij hebben dikwijls verschillende belangen te verdedigen. Om hier een antwoord op te bieden, moeten afspraken gemaakt worden en indien nodig technieken worden aangereikt die dergelijke afspraken kunnen afdwingen. In de sectie 'Policies' stel ik kort voor op welke manier hierop kan ingespeeld worden.

### 3.6.1 Territorialiteit

Een belangrijk aspect in het werken met tabletop interfaces is de territorialiteit tussen de gebruikers. Als gevolg van deze territorialiteit verdelen de eindgebruikers de tabletop interface in verschillende zones. Deze zones helpen de groepsinteractie in goede banen te leiden. Men is erachter gekomen dat men typisch gebruik maakt van drie soorten ruimten [38, 40, 39, 22], zijnde 'Personal space', 'Group space' en 'Storage'. Deze ruimten worden schijnbaar moeiteloos ingedeeld als gevolg van de fysieke beperkingen van de eindgebruiker en de sociale vaardigheden die zij hebben opgebouwd in het dagelijkse leven.



Figuur 3.31: Evolutie territoria. Links, beginsituatie met één gebruiker. Midden en rechts resp. twee en drie gebruikers [38].

### Personal space

Blijkbaar hebben gebruikers eerst en vooral nood aan persoonlijke ruimte. Dit is de ruimte die zich meestal vlakbij de gebruiker bevindt en wordt gebruikt voor het afwerken van persoonlijke (sub)taken. Meestal wordt een object eerst naar deze persoonlijke ruimte gehaald om het object in kwestie te kunnen bewerken. Even later kan het gemanipuleerde object dan opnieuw worden geïntegreerd in de groepsruimte. Het verkrijgen van een dergelijke persoonlijke ruimte kan op twee manieren gebeuren. Enerzijds expliciet zoals Rekimoto et al. [36] voorstellen via afgebakende zones. Anderzijds kan die indeling op een impliciete manier gebeuren op basis van sociale principes zoals Scott et al. [38, 40] voorstellen. Uit hun studie blijkt dat de grenzen van de persoonlijke ruimten impliciet worden afgebakend als gevolg van sociale principes die door de jaren werden opgebouwd bij elk van de gebruikers. Op figuur 3.31 wordt geïllustreerd hoe dit proces verloopt.

### Group space

Buiten deze persoonlijke ruimtes, duikt er ook een gezamenlijke groepsruimte op waarin alle gebruikers kunnen interageren met de data. In deze ruimte hebben alle gebruikers in theorie evenveel rechten. In de praktijk blijkt dat niet altijd zo evident te zijn. De groepsruimte gebruikt men vooral om anderen te helpen en om materiaal door te geven. De grootte van deze groepsruimte is afhankelijk van het aantal persoonlijke ruimtes en de plaats die zij in beslag nemen zoals te zien is op figuur 3.31. Een bijkomend verschijnsel is het ontstaan van subgroepsruimten. Dit treedt op wanneer twee aangrenzende gebruikers samenwerken en de ruimte tussen beiden aangrijpen om de subtaak in kwestie tot een goed einde te kunnen brengen.

### Storage space

Opslagruimte gebruikt men om tijdelijk objecten, tools en dergelijke op te slaan. Later kunnen we deze instrumenten dan opnieuw aanwenden. In tegenstelling tot de twee voorgaande ruimten ligt deze ruimte verspreid over het tafeloppervlak. Stapels materiaal kunnen zich zowel in group space bevinden als erlangs in de afzonderlijke persoonlijke ruimten. Opvallend hierbij is dat hoe meer de uit te voeren opdracht zijn einddoel nadert, de opslagruimten zich naar de randen verplaatsen [40].

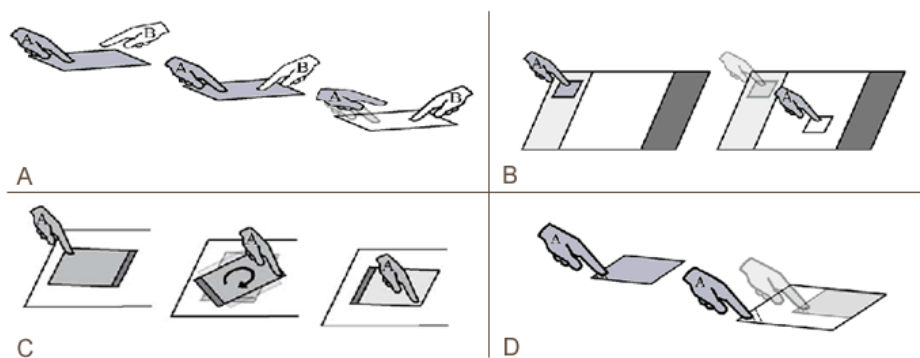
## 3.6.2 Policies

Dommel et al. [9] hebben onderzoek gedaan naar de coördinatie van activiteiten in collaboratieve omgevingen. Ook zij onderschrijven dat samenwerken rond een interactieve tafel conflicten kan opleveren. Om dergelijke conflicten te voorkomen hanteren zij het principe van 'mutual exclusion', dat ook wordt gebruikt in threaded software. Wanneer een gebruiker werkt met een bepaald object krijgt hij hier tijdelijk de rechten voor, zodat een concurrerende gebruiker hem het object niet kan ontnemen. Het verwerven van deze rechten gebeurt via een aantal principes, 'policies' genaamd. Dankzij deze 'policies' kunnen

gebruikers op een natuurlijke manier de rechten over een object verwerven (request) en later weer afstaan (release) aan een andere gebruiker. Deze 'policies' proberen de situatie te verbeteren door meer structuur te brengen in de interactie, en tegelijk zo flexibel mogelijk te blijven. Enkele interessante policies zijn:

- *No selections, no touches, no holding documents*, enkel wanneer alle drie deze voorwaarden zijn voldaan, kan de gebruiker de rechten over een object verkrijgen.
- *Rank*, een gebruiker kan de rechten over een object overnemen wanneer hij over een hogere rang beschikt.
- *Object sharing*, een gebruiker geeft via een actie de rechten van het object door aan een andere gebruiker.

Een erg krachtig mechanisme om conflicten te vermijden is om een mechanisme te voorzien dat de gebruikers toelaat objecten te delen. Morris et al. [37] ontwikkelden daarom een aantal eenvoudige technieken om het delen van objecten te vergemakkelijken. We illustreren ze op figuur 3.32.



Figuur 3.32: Release (A), Relocate (B), Reorient (C), Resize (D) [37].

- *Release*, objecten die worden doorgegeven zijn beschikbaar voor de groep, de andere objecten zijn dat niet.
- *Relocate*, objecten die zich in de centrale zone bevinden zijn beschikbaar voor de groep, de andere objecten zijn dat niet.
- *Reorient*, objecten die zijn georiënteerd naar het midden van de tafel zijn beschikbaar voor de groep, de andere objecten zijn dat niet.
- *Resize*, objecten die groter zijn dan een bepaalde limiet zijn beschikbaar voor de groep, de andere objecten zijn dat niet.

### **3.7 Samenvatting**

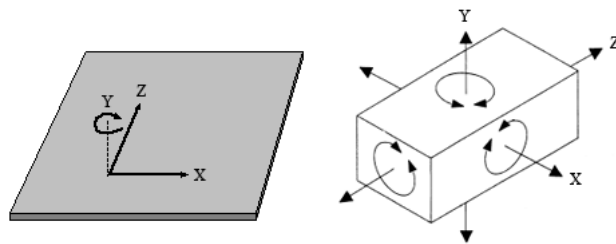
In dit hoofdstuk hebben we een aantal technieken bekeken die een antwoord bieden op enkele typische tabletopgerelateerde problemen. Elk van deze technieken heeft zo zijn sterke en minder sterke punten en het is zaak 'the right technique for the job' uit te kiezen. Het is belangrijk de doelstellingen van de toepassing te onderzoeken om de juiste keuze te maken. We kunnen de kanttekening maken dat deze technieken werden ontwikkeld voor tweedimensionale interactie. We moeten dus verder bekijken in hoeverre deze technieken kunnen gebruikt worden bij driedimensionale interactie. Ook het collaboratieve aspect mogen we niet over het hoofd zien want het speelt een belangrijke rol in hoe gebruikers met de tabletop omspringen.

# Hoofdstuk 4

## Tabletops en de derde dimensie

### 4.1 Inleiding

Dat tabletops een hot topic zijn binnen de HCI wereld is duidelijk. Er wordt dan ook uitvoerig onderzoek gedaan naar het werken met tabletops. Een groot aandeel van dit onderzoek gaat uit naar het ontwikkelen van 2D-toepassingen. Grossman et al. [14] merken op dat de activiteit op fysieke tafels eerder 3D getint is. Zij vinden het dan ook niet meer dan logisch dat er wordt nagedacht over manieren om ook op digitale tafels in drie dimensies te kunnen werken. De toepassingen van dergelijke technologie zijn legio, denk maar aan simulaties van chirurgische ingrepen of het modelleren en annoteren van nieuwe producten alvorens een fysiek prototype te ontwikkelen.



Figuur 4.1: mismatch in vrijheidsgraden invoer (links) en uitvoer (rechts).

Om dergelijke interactie mogelijk te maken is het belangrijk dat men rekening houdt met de eigenschappen van dergelijke driedimensionale werelden. Interageren met driedimensionale objecten via een tweedimensionaal invoerapparaat als een tabletop is immers niet evident omdat dergelijke invoerapparaten slechts beschikken over twee vrijheidsgraden, namelijk een  $x$ -positie en een  $y$ -positie. Driedimensionale objecten bezitten daarentegen zes vrijheidsgraden. Zowel  $x$ ,  $y$  als  $z$  positie als de rotaties rond de bijbehorende assen kunnen worden gemanipuleerd. Het probleem bestaat er in dat de tweedimensionale invoer moet worden gemapt naar een driedimensionale wereld zoals te zien is op figuur 4.1.

Binnen een topprojectie zijn translaties over  $X$  en  $Z$ -as vrij eenvoudig te implementeren daar deze zich net als hun tweedimensionale tegenhangers gedragen. De derde dimensie daarentegen, in dit geval de  $Y$ -as, is veel minder voordehandliggend. Het bepalen van diepte is immers moeilijk te realiseren. In 3D-modelleromgevingen als Maya en 3D studio Max wordt typisch gebruik gemaakt van meerdere viewports met complementaire projecties waarin het 3D object verder kan worden gemanipuleerd. Dit zorgt voor precieze interactie maar het is niet makkelijk om al deze viewports tegelijk te volgen. Dit neemt niet weg dat het zinvol is om extra standpunten weer te geven. Voor rotaties stelt zich hetzelfde probleem. Rotaties rond de  $Y$ -as zijn wederom vrij eenvoudig te realiseren via technieken als het 'two-point rotation' van Hancock et al. [18] maar rotaties rond de andere assen  $X$  en  $Z$  zijn zeer moeilijk te definiëren. Een bijkomend probleem is bovendien dat sommige objecten ook onzichtbaar kunnen zijn voor de gebruiker omdat zij door andere objecten worden bedekt.

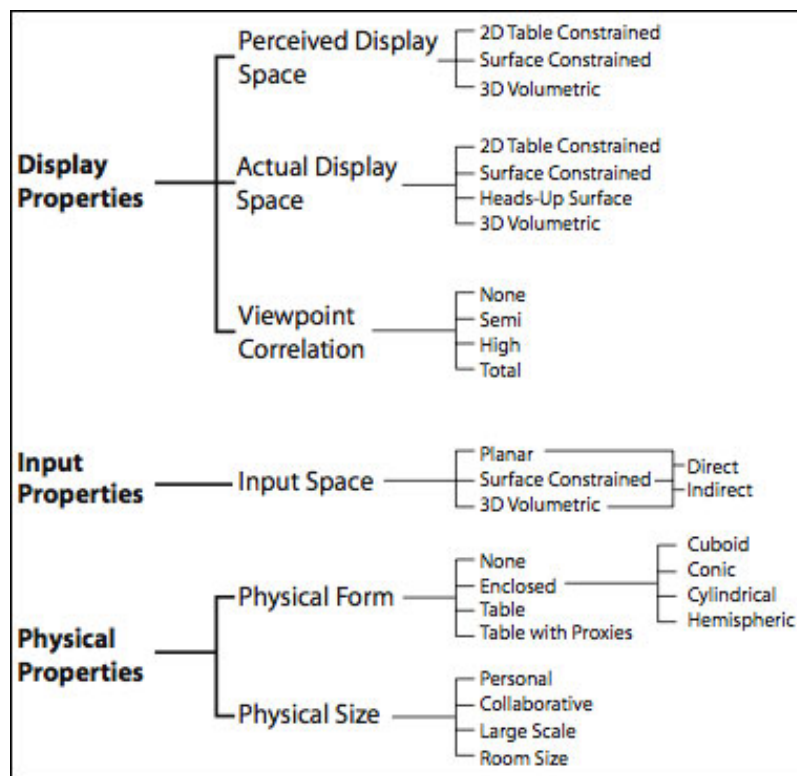
## 4.2 Een stand van zaken

In de inleiding van dit hoofdstuk werd al aangehaald dat tafels zich in de echte wereld meer manifesteren als een medium waarop geïnterageerd wordt met driedimensionale objecten. Toch blijkt het transfereren van dergelijke taken naar de digitale tabletop geen sinecure te zijn. Dit kan één van de redenen zijn waarom interactie op tabletops in drie dimensies nog grotendeels braakliggend terrein is in de onderzoekswereld. Toch zijn er reeds enkele interessante initiatieven genomen om de mogelijkheden van deze technologie uit te testen en te evalueren.

Grossman et al. [14] hebben daarom een taxonomie opgemaakt om de reeds bestaande technieken in kaart te brengen. Zij identificeerden drie dimensies om deze technieken in te delen zoals te zien is op figuur 4.2.

### Weergave

Men maakt hierbij het onderscheid tussen de gepercipieerde ruimte en de eigenlijke ruimte waarin de objecten zich bevinden. De gepercipieerde ruimte wordt gedefinieerd als de ruimte waarin de objecten worden getoond al dan niet via het gebruik van stereoscopie.



Figuur 4.2: Taxonomy 3D op de tabletop [14].

De eigenlijke ruimte geeft aan waar de objecten effectief worden geprojecteerd. Een goed voorbeeld om het verschil tussen beiden te illustreren is het gebruik van een tabletop in combinatie met shutterglases. Hier zal de eigenlijke ruimte nog steeds slechts twee dimensies bevatten terwijl de gebruiker de objecten via de shutterglases in drie dimensies zal percipieren. Verder merken Grossman et al. [14] op dat er ook een verschil bestaat tussen technologieën die het beeld in de viewport van de gebruiker wijzigen aan de hand van de kijkrichting en de locatie van de gebruiker en technieken die dit niet doen. Tot nog toe is volledige driedimensionale weergave enkel mogelijk via supplementaire hardware. Echter het is net deze hardware die ervoor zorgt dat de onderdompeling niet compleet is [14]. Bovendien is dergelijke apparatuur dikwijls oncomfortabel.

### Invoer

Een tweede dimensie maken zij op basis van de invoerruimte. Volgens Grossman et al. [14] zijn er hierin drie mogelijkheden. De meest voorkomende ruimte is de tweedimensionale invoerruimte waarbij de objecten kunnen worden gemanipuleerd door de tabletop aan te raken. Wanneer er gebruik wordt gemaakt van fysieke objecten boven op de tabletop spreekt men van oppervlaktegebonden interactie. Hierbij kan de gebruiker interageren met echte objecten in drie dimensies. De invoerruimte blijft steeds beperkt tot de nabije



omgeving van het tabletop oppervlak. De invoer kan ook gebeuren in drie dimensies wanneer gebruik wordt gemaakt van invoerapparaten als een digitale handschoen. Dit laat de gebruiker toe om effectief in de scene te reiken om zo met de objecten te interageren.

Het introduceren van een 3D-invoerruimte laat gebruikers toe om de objecten ook effectief in 3D te manipuleren. Helaas betaalt men hiervoor een prijs. Het is immers niet meer mogelijk om typische 2D-triggers te gebruiken. Er moeten dus interactietechnieken worden ontwikkeld om dit op te vangen. Dit kan door middel van tracking van gestures of door middel van invoerapparatuur als de digitale handschoen. Tactiele feedback is niet aanwezig wanneer er in de ijle ruimte wordt gewerkt. Dat kan wel deels worden opgevangen door gebruik te maken van fysieke objecten, 'haptic devices' en bimanuele invoer [14].

### Fysieke eigenschappen

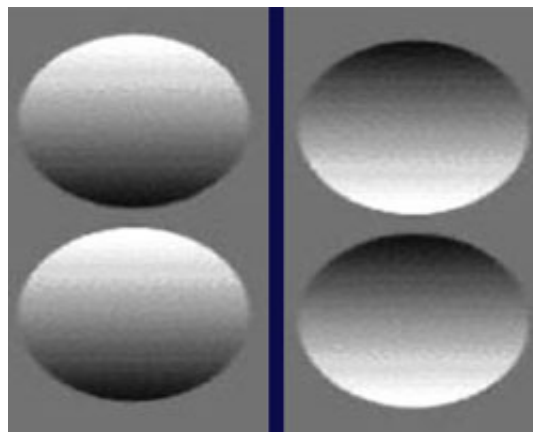
De derde dimensie in hun overzicht gaat over de fysieke eigenschappen van de technologie in kwestie. Dit vormt volgens Grossman et al. [14] een erg bepalende factor voor de manier waarop gebruikers zullen interageren met de technologie. Hierin beschouwen zij voornamelijk twee eigenschappen, enerzijds de vorm en anderzijds de omvang van de interactie ruimte. De vorm van deze ruimte kan gaan van de omvang van de tabletop tot het ontbreken van enige vorm bij het dragen van een head mounted display waarbij de hele omgeving virtueel is. Wat betreft de omvang is het spectrum afgebakend tussen enerzijds een persoonlijke interface die gericht is op persoonlijk gebruik, en anderzijds collaboratieve omgevingen. Dergelijke collaboratieve omgevingen kunnen opgebouwd zijn rond grote schermen. Ze kunnen ook een volledige ruimte bestrijken.

## 4.3 Diepteperceptie

Diepteperceptie is de mogelijkheid om de wereld in drie dimensies waar te nemen. Een goede diepteperceptie is nodig om jezelf makkelijk te kunnen bewegen en om op een juiste manier te kunnen reageren op onze omgeving op basis van de ruimtelijke relaties van de objecten die er zich in bevinden. Om diepte te kunnen ervaren hanteren mensen een complex systeem van dieptehints. Erg belangrijk hierin is het stereoscopisch zicht waarover wij allemaal beschikken. Echter om dit te integreren in een tabletop omgeving moet er gebruik gemaakt worden van shutter glasses. Wanneer we enkel over een tabletop beschikken is dit niet realiseerbaar, maar er zijn wel andere mogelijkheden om tot een goede diepteperceptie te komen.

- **Perspectief** draagt sterk bij tot een goede diepteperceptie.
- **Associatie** maakt gebruik van de kennis over de grootte van bekende objecten in de omgeving. De relatieve grootteverschillen van objecten kunnen op die manier dienen om de afstand tussen de objecten in te schatten.

- **Oclusie** is een soortgelijke dieptehint die optreedt wanneer een object een ander object deels uit het zicht onttrekt. De interactie die op dat moment optreedt tussen de objecten geeft de observator informatie over de relatieve afstand tussen beide.
- **Schaduw** speelt een belangrijke rol in het inschatten van diepte in een scene. Ons visueel systeem neemt immers aan dat de meest belichte oppervlakken zich typisch bovenaan bevinden en vice versa. Dit concept wordt geïllustreerd op figuur 4.3.



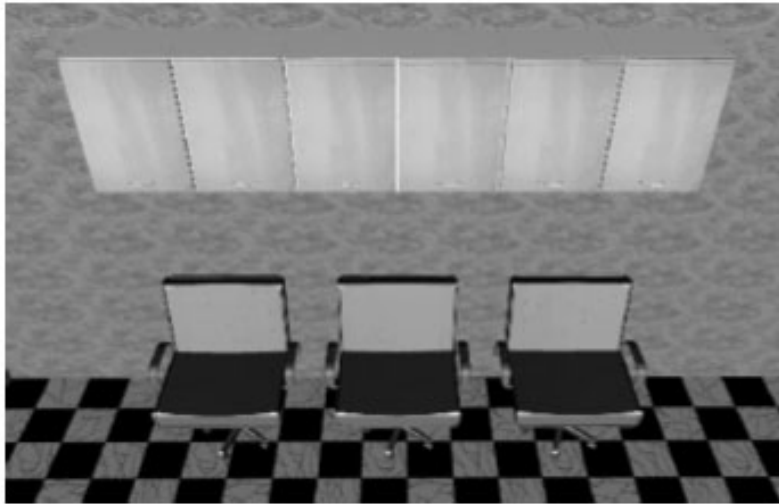
Figuur 4.3: De invloed van schaduw op diepte.

## 4.4 3D-Interactie

### 4.4.1 Manipulatie met behulp van constraints

3D modelling applicaties zijn typisch erg complex omdat ze bestaan uit een doolhof van menu's, widgets en modes. Het is niet eenvoudig om wegwijs te geraken in de vele mogelijkheden die dergelijke applicaties aanbieden. Bovendien is het controleren van 3D objecten sowieso een moeilijke oefening. Dit vereist immers de coördinatie van zes vrijheidsgraden tegelijk.

Smith et al. [42] merken terecht op dat fysieke objecten zijn gebonden aan de wetten van de fysica. Wij mensen maken bewust gebruik van de fysieke eigenschappen van een object om ze te manipuleren in de ruimte. Het beperken van de bewegingsvrijheid van objecten in een virtuele wereld kan ons dus sterk helpen om 3D-scenes te creëren.



Figuur 4.4: De kasten aan de muur werden via de dual constraint gegroepeerd [42].

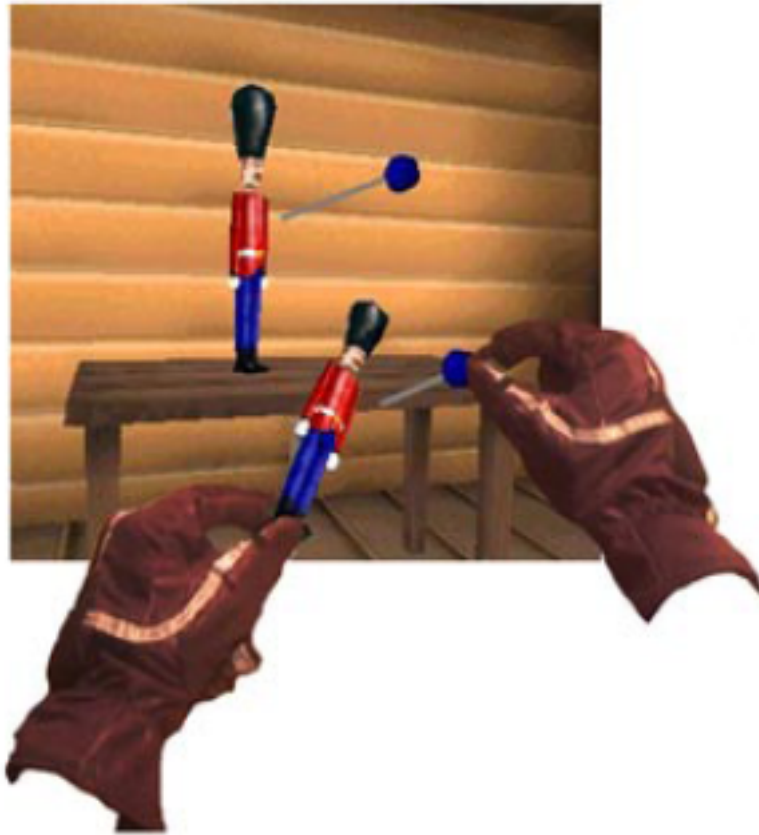
Zij stellen daarom voor om objecten te voorzien van 'constraints'. Deze constraints geven aan welke delen van een object geschikt zijn voor koppeling met andere objecten. De objecten zelf bevinden zich overigens in een boomstructuur waarbij de ouderobjecten steeds koppelingen aanbieden aan de kindobjecten. Deze beschikken op hun beurt over delen die beschikbaar zijn voor binding met de ouderobjecten. Ook koppelingen tussen objecten op het zelfde niveau zijn mogelijk via de dual constraint. Deze constraint laat toe om een aantal objecten te groeperen, zoals te zien is op figuur 4.4.

#### 4.4.2 Voodoo dolls

Voodoo dolls [32] is een tweehandige techniek die wordt gebruikt om de ruimtelijke relaties tussen twee objecten aan te passen. De techniek gebruikt hier voor tijdelijke kopieën van de objecten in de scene. Deze kopieën worden verkregen door pinch bewegingen uit te voeren over de objecten. Afhankelijk van de hand waarin de kopie zich bevindt, wijzigt de betekenis ervan. Het object in de linkse hand fungeert als referentiekader en bepaalt de schaal waarop gewerkt wordt. Het object in de rechtse hand is het object dat ten opzichte van het referentiekader wordt verplaatst. Op figuur 4.5 werd de soldaat links als referentie genomen en wordt dit concept gedemonstreerd.

#### 4.4.3 3D-tabletopinteractie: Shallow depth 3D

Wanneer we vergaderen rondom een fysieke tafel, wordt er dikwijls gebruik gemaakt van stapels materiaal. Hancock et al. [17] merken op dat de derde dimensie zich op dat moment beperkt tot de omgeving net boven het interactievlak. Om ook op digitale tafels gebruik te kunnen maken van de derde dimensie stellen zij enkele technieken voor, die uitgaan

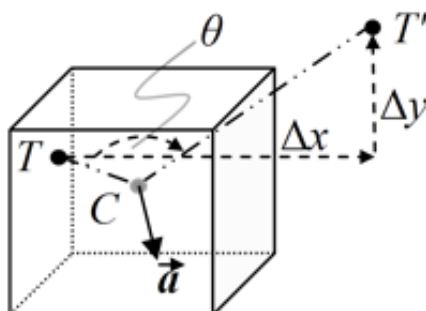


Figuur 4.5: Voodoo dolls, referentiekader links en te manipuleren object rechts [32].

van een beperkte diepte in de 3D-omgeving. Deze technieken laten de gebruiker toe om driedimensionale objecten te kunnen manipuleren in zes vrijheidsgraden door translatie en rotatie simultaan te laten verlopen. Het voorzien van zoveel mogelijk vrijheidsgraden is volgens hen cruciaal voor de communicatie rondom de tafel.

### One-touch input

Via deze techniek kan de gebruiker een beweging uitvoeren in vijf vrijheidsgraden door slechts één punt te manipuleren. In plaats van te roteren rond één van de drie vaste assen, wordt er geroteerd rond een as die bepaald wordt door het crossproduct van de vector vanuit het initiële raakpunt naar het centrum van het te roteren object en de vector vanuit het centrum van het te manipuleren object naar het uiteindelijke eindpunt. Op figuur 4.6 wordt dit concept verder geïllustreerd.



Figuur 4.6: 3D interactie via één contactpunt [17].

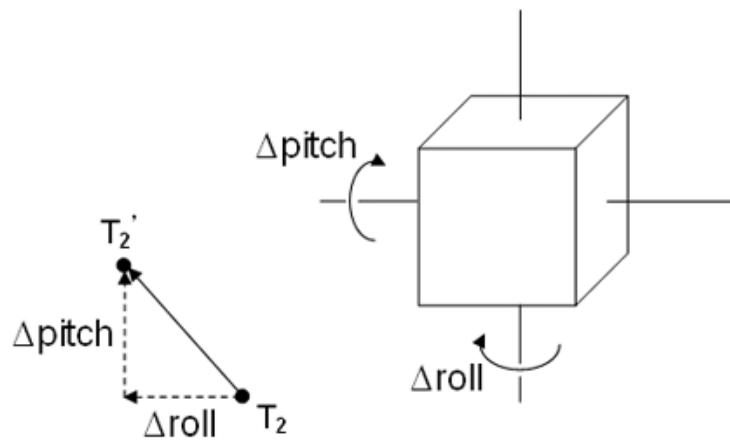
Ondanks dat de invoer slechts uit twee vrijheidsgraden bestaat, blijkt deze methode best natuurlijk over te komen aldus Hancock et al. [17]. Gebruikers willen typisch de mogelijkheid hebben om translaties en rotaties afzonderlijk van mekaar uit te voeren. Daarom werden toegewijde zones op de objecten voorzien. Zo werd het centrum van een object gedefinieerd als translatiezone. De regio daarbuiten doet daarentegen dienst als rotatiezone en maakt hiervoor gebruik van de 'RNT-techniek'.

### Two-touch input

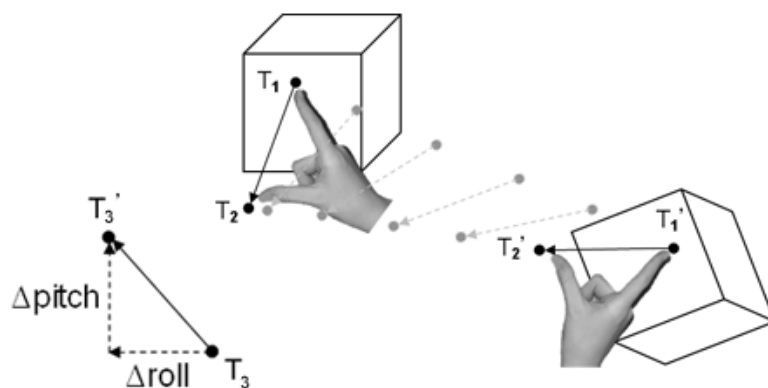
Vervolgens hebben zij geëxperimenteerd met invoer via twee punten. Hierdoor is de mismatch tussen invoer- en uitvoervrijheidsgraden een stuk kleiner. Het eerste invoerpunt controleert de x- en z-positie in het vlak, en gebruikt het eerder besproken RNT algoritme om rotaties rondom de Y-as te faciliteren. Het tweede invoerpunt wordt vervolgens gebruikt om de rotatie rondom de X- en de Z-as te controleren. De rotatie rondom deze twee assen wordt bepaald aan de hand van de translatie van het tweede invoerpunt.

### Three-touch input

Een derde variant van 'shallow-depth 3D interaction' maakt gebruik van drie invoerpunten. Hier wordt het eerste contactpunt gebruikt om translaties in x en z uit te voeren. Het tweede contactpunt wordt vervolgens gebruikt om de rotatie rondom de Y-as te faciliteren en het derde contactpunt bepaalt de rotatie rondom de overige assen X en Z. In theorie is deze techniek vrij eenvoudig in gebruik. Echter Hancock et al. [17] waarschuwen ervoor dat het introduceren van deze extra vrijheid verwarrend kan zijn voor de gebruiker.



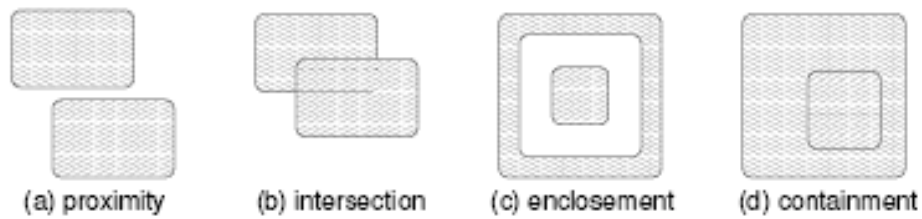
Figuur 4.7: 3D interactie via twee contactpunten [17].



Figuur 4.8: 3D interactie via drie contactpunten [17].

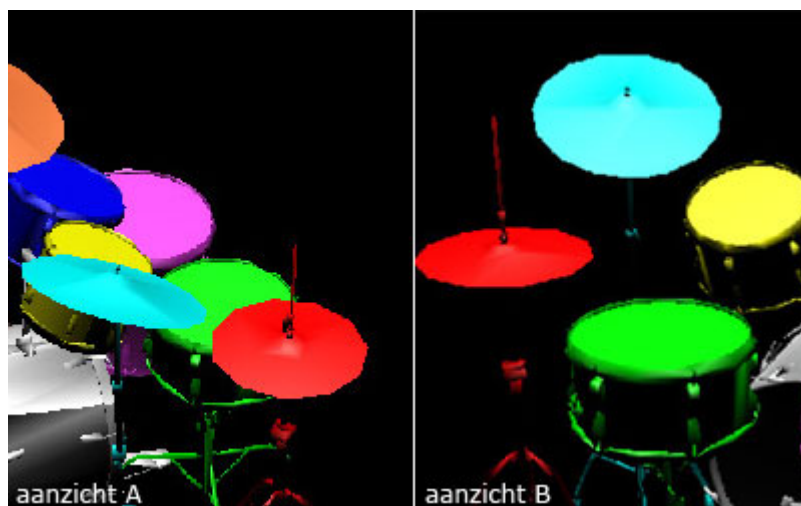
## 4.5 Occlusie-management

De mens gebruikt en interpreteert alle beschikbare informatie uit de omgeving om diepte in te schatten. Ook occlusie speelt hierin zoals eerder gezegd een belangrijke rol. Occlusie kan ook een vervelend iets zijn in 3D-omgevingen. Het kan in een dergelijke 3D-omgeving knap lastig zijn om een bepaald object terug te vinden. Het is moeilijk om te interageren met objecten die deels of volledig uit zicht verdwijnen. Objecten die volledig onzichtbaar zijn zullen logischerwijze nooit in aanmerking komen voor manipulatie omdat gebruikers ze nu eenmaal niet zullen terug vinden.



Figuur 4.9: Occlusie scenario's [11].

Bij objecten die deels uit het zicht onttrokken zijn, is de mate waarin een object zichtbaar is afhankelijk van de grootte van het zichtbare deel en de manier waarop de objecten met elkaar interageren. Dit is te zien op figuur 4.9. Dikwijls kan een object wel gevonden worden, maar blijft het zeer moeilijk het object correct te percipiëren. Ook de objectdichtheid speelt hierin een belangrijke rol. Naar mate er meer objecten voorkomen in de omgeving zal de kans op occlusies toenemen.

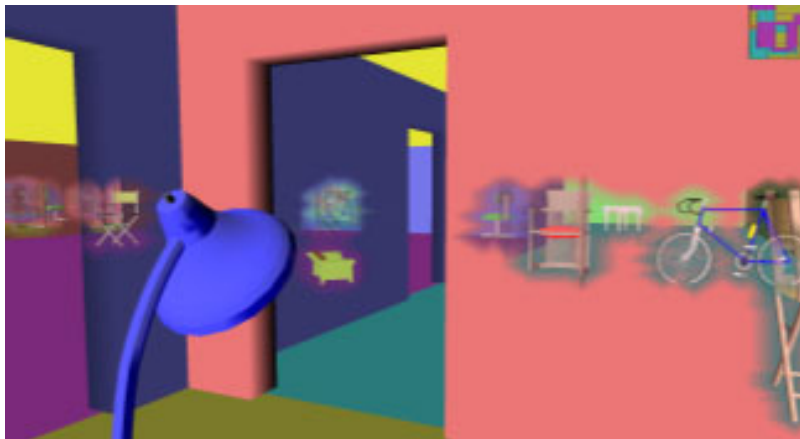


Figuur 4.10: Het gebruik van meerdere aanzichten lost enkele occlusieproblemen op [45].

Verder heeft ook de complexiteit van het object een behoorlijke impact op het voorkomen van occlusie. Complexe objecten bevatten immers meer informatie dan primitieve objecten en zijn daarom extra gevoelig aan occlusie. Het niet volledig kunnen percipiëren van het object kan in zo'n geval nog dramatischer zijn volgens Elmqvist et al. [11].

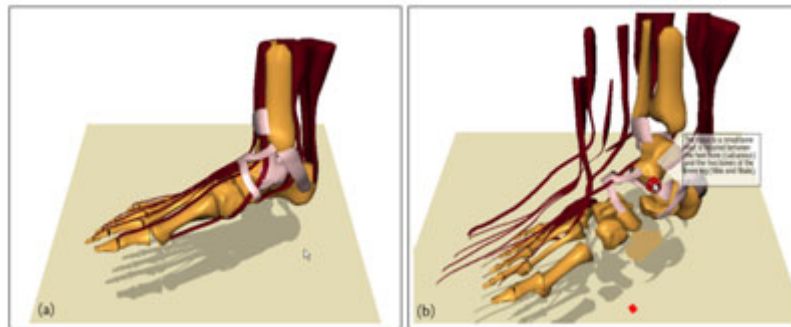
Om te anticiperen op het occlusieprobleem werden een aantal uiteenlopende strategieën uitgedacht.

- **Multiple viewports** komen tegemoet aan occlusie door meerdere (camera) standpunten in de scene te voorzien en deze te tonen in meerdere viewports. Op figuur 4.10 wordt dit geïllustreerd.
- **Virtual X-ray** gebruikt transparantie om verborgen objecten terug te vinden. Op figuur 4.11 zijn bepaalde delen van de kamer transparant geworden zodat de gebruiker de doelen in de scene eenvoudig kan terugvinden.
- **Interactive exploder** wijzigt de ruimtelijke relaties tussen de objecten zodat ze allen zichtbaar worden zoals te zien is op figuur 4.12 [43].
- **Tourplanner** probeert occlusie aan te pakken door een pad doorheen de scene af te leggen.



Figuur 4.11: Het gebruik van digital X-ray om doelen in een scene terug te vinden [2].

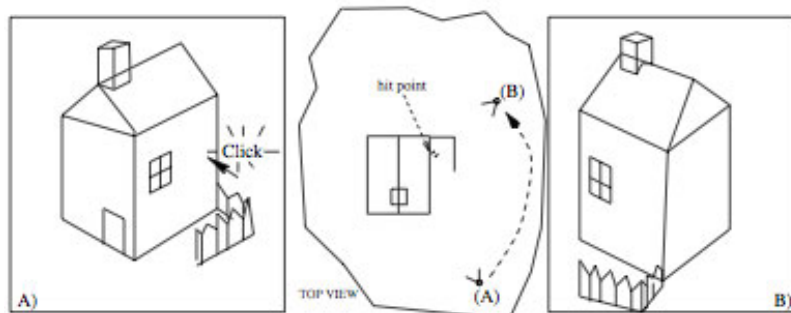




Figuur 4.12: Een voorbeeld van het exploder concept toegepast op de anatomie van een voet [43].

## 4.6 Camera-management

In 3D-omgevingen spreekt men dikwijls over het begrip 'camera' om het kijkpunt te omschrijven dat in de viewport wordt getoond. De functie van een dergelijke camera kan volgens Drucker et al. [10] grotendeels worden terug gebracht tot een viertal operaties: oriëntatie, navigatie, exploratie en presentatie. Het kunnen manipuleren van de camera is essentieel om deze operaties op een vlotte manier te kunnen uitvoeren. Zowel Drucker et al. [10] als Zeleznik et al. [48] zijn het er over eens dat het manipuleren van een dergelijke camera in zes vrijheidsgraden best moeilijk is. Typisch wordt dit opgelost door voor elke subset van deze zes vrijheidsgraden een afzonderlijke techniek te implementeren. Via modi kunnen dan de benodigde operaties met de camera worden uitgevoerd.

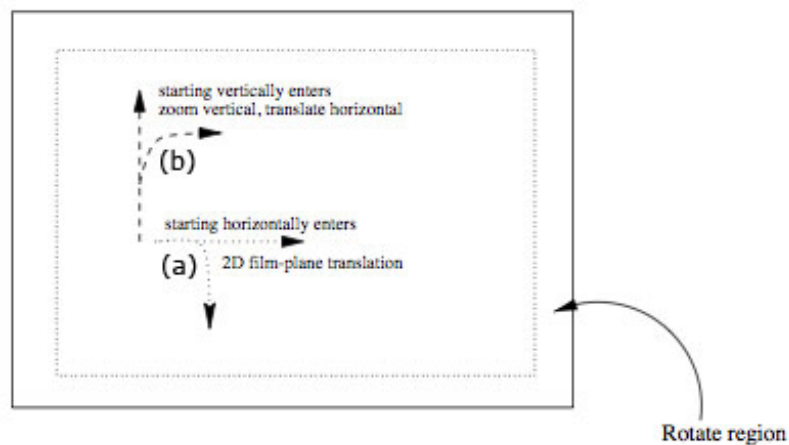


Figuur 4.13: Click to focus, camera animeert na klikken van punt a naar punt b [48].

Helaas is het wisselen tussen modi verwarrend voor de gebruikers. De gebruiker zal uiteindelijk zijn focus moeten verleggen naar het bedienen van de camera interface [10]. Om te

kunnen wisselen tussen de verschillende modi, moet de gebruiker de juiste coördinatie zien te vinden tussen zijn linker- en rechterhand. Zeleznik et al. [48] ontwikkelden daarom een camera-managementtechniek op basis van gestures. Via de gestures op figuur 4.14 kunnen deze bewegingen worden uitgevoerd. Een korte horizontale beweging gevolgd door een verticale beweging zorgt voor 'film plane translation'. Een korte verticale beweging gevolgd door een horizontale beweging initieert 'dollying'. 'Orbiting' kan door in de rotatieregio te bewegen.

- **Film plane translation** is het transleren over de X- en de Y-as.
- **Dollying** is het transleren over de X- en de Z-as.
- **Oribiting** is het bewegen van de camera in een baan rond het doelwit.
- **Click to focus** is het focussen op een object zoals te zien is op figuur 4.13.



Figuur 4.14: Camera-management op basis van gestures [48].

## 4.7 Samenvatting

In dit hoofdstuk heb ik eerst een huidige stand van zaken gegeven wat betreft de integratie van 3D-omgevingen op tabletops. Interactie met 3D-objecten via een tabletop is zoals eerder gezegd niet evident en brengt een aantal specifieke uitdagingen met zich mee. Eén van deze uitdagingen bestaat erin om te kunnen roteren en transleren in drie dimensies, terwijl tabletops slechts over tweedimensionale invoer beschikken. Helaas zijn er in de literatuur nog maar weinig concrete studies verschenen omtrent dit onderwerp. In de sectie 'Shallow depth 3D' heb ik een mogelijke oplossing voor dit probleem besproken. Dit is slechts één voorstel en andere oplossingen zijn best mogelijk. Verder zijn ook enkele typische problemen waar driedimensionale omgevingen mee te kampen hebben aan bod gekomen in dit hoofdstuk.

# Hoofdstuk 5

## Interior Designer

### 5.1 Inleiding

Bij het inrichten van een huis wordt er typisch gewerkt op een incrementele manier. De klant legt uit bij de binnenhuisarchitect wat hij voor ogen heeft, waarna de binnenhuisarchitect op zijn beurt een voorstel zal doen naar de klant toe. De klant kan hier vervolgens weer input op geven zodat het idee van de binnenhuisarchitect stelselmatig evolueert naar het eindproduct. Dit is dus een incrementeel proces dat behoorlijk wat tijd kost. Interior Designer is een proof of concept dat erop doelt om van dat incrementeel proces een interactief collaboratief proces te maken waarbij de klant en de designer samenwerken rondom een tabletop.

Interior Designer is een applicatie met als doel de gebruiker toe te laten om een omgeving in te richten met meubilair en dergelijke. In tegenstelling tot desktop applicaties, vertrekt het idee van manuele invoer via de tabletop. De binnenhuisarchitect kan objecten toevoegen aan de scene door gebruik te maken van een contextueel menu en kan deze objecten vervolgens manipuleren en plaatsen in de scene via natuurlijke interactietechnieken. Eens zijn klanten arriveren kan hij hen het voorstel tonen en een virtuele toer afleggen in de scene. De klant kan vervolgens samen met de binnenhuisarchitect wijzigingen aanbrengen via de interface zodat een anders incrementeel proces wordt herleid naar een collaboratieve en interactieve onderneming. Om dit collaboratief proces te coördineren rekent Interior Designer op de sociale vaardigheden van de gebruikers.

Een dergelijke applicatie heeft specifieke noden. Denk maar aan de rol die oriëntatie speelt op dergelijke interfaces. Bovendien moeten er dikwijls grotere afstanden worden afgelegd. Daar komt nog bij dat er zich meerdere gebruikers rond de tafel bevinden en de interface moet hierop voorzien zijn. Verder is Interior Designer een 3D-applicatie, dat zo zijn eigen problemen introduceert zoals het definiëren en percipiëren van diepte, oclusie-management en camerabeheer. Ook de applicatie zelf vereist specifieke maatregelen om te kunnen werken op verschillende schaalfactoren.

Bij het ontwikkelen van Interior Designer is getracht rekening te houden met elk van deze factoren. In de volgende secties wordt de applicatie besproken in functie van deze specifieke problemen.

## 5.2 Menu-interface

### 5.2.1 Structuur

Op figuur 5.1 vinden we een algemene legende terug van de functies die in het menu zijn opgenomen. In deze legende is elke functie geannoteerd met een cijfer. Deze cijfers zullen in de komende secties regelmatig worden gebruikt als referentie en worden tussen ronde haakjes weergegeven. Het menu kan als geheel worden verplaatst door je vinger op de menu-placeholder (30) te plaatsen. Het menu zal deze vinger volgen over het canvas.

#### Algemeen menu

Het menu is opgebouwd uit drie submenu's. Onderaan bevindt zich het submenu met algemene functies. Door het icoon met het oog (31) aan te tikken worden alle submenu's verborgen. Op die manier kan de gebruiker het menu als geheel doen verdwijnen zodat er geen kostbare schermruimte wordt ingenomen. Enkel wanneer de gebruiker het menu nodig heeft, kan hij het te voorschijn toveren door snel terug het icoon met het oog aan te duiden.

Vlak naast het icoon met het oog vinden we een icoon terug met de woorden 'depth color' (32). Dit zal er voor zorgen dat de objecten in de scene een kleur krijgen toegekend op basis van de hoogte waarop ze zich bevinden. Hierover later meer in de sectie 'Interactietechnieken'. Langs dit icoon vinden we een icoon van een huisje terug (33). Via deze functie kan de gebruiker op een snelle manier de scene terug centreren rondom het centrum van de wereld. Als laatste vinden we een icoon terug met daarop twee documenten (34). Via deze functie kan de gebruiker een kopie maken van het menu.

#### Objectmenu

Het tweede deelmenu is het objectmenu. Dit is een menu dat als doel heeft nieuwe objecten toe te voegen aan de scene. Dit gebeurt door met de vinger het gewenste object aan te duiden. Een voorbeeld van een dergelijk menu-item vind je terug in de legende (40).

#### Contextgevoelig tools-menu

Het laatste deelmenu wordt ingevuld afhankelijk van het type object. Dit menu wordt aangegeven door de placeholder met de hamer en sleutel (1) op. Er zijn in dit prototype twee types objecten voorzien. Het eerste type zijn de objecten die worden gebruikt om de omgeving in te richten (meubilair, elektronische apparaten, planten, etc...). Wanneer

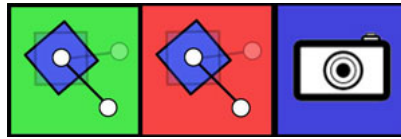
een dergelijk fysiek object wordt aangeduid, configureert het menu zich om interactie met dit object toe te laten. Binnen dit menu zijn functies voorzien om translaties, rotaties en dergelijke uit te kunnen voeren. Deze komen later uitgebreid aan bod in het vervolg van dit hoofdstuk.



Figuur 5.1: Menu legende.

Het tweede type object is de camera. De cameraobjecten zijn geen fysieke maar logische objecten die een andere set functies met zich mee brengen. De set tools past zich daarom automatisch aan om zo de juiste functionaliteit aan te bieden. Dit deulmenu wordt aangegeven door de placeholder met de camera (20).

Om te schakelen tussen de verschillende opties, volstaat het om de gewenste optie even aan te duiden met je vinger. Ingeschakelde opties worden steeds met een groene achtergrond weergegeven. Uitgeschakelde opties worden telkens met rode achtergrond weergegeven. Op figuur 5.2 vinden we hier een voorbeeld van terug. De blauwe menu-items geven telkens aan dat het gaat om een actie zoals bijvoorbeeld het nemen van een screenshot (8) zoals uiterst rechts te zien is op figuur 5.2.



Figuur 5.2: Kleurcodes menu-items.

Deze interactietechnieken komen allen verder uitgebreid aan bod in de sectie 'Interactietechnieken'.

### Viewport

Verder is elk menu voorzien van een viewport. Deze viewport bevindt zich steeds rechts van het menu en is net als het menu, contextgebonden. Wanneer je een fysiek object selecteert, zal de viewport het zijaanzicht voor dat object tonen. Dit is te zien op figuur 5.12. Wanneer je een camera selecteert, geeft de viewport het camerastandpunt weer.

### Kopiëren van menu's

De applicatie werd ontworpen met als doel een designer toe te laten om op een interactieve manier samen met zijn klanten een omgeving in te richten. Zij bevinden zich verspreid rond de tafel. Er is dus naast heroriëntatie ook nood aan een manier om alle gebruikers te voorzien van een menu. Dit kan door via het icoon met de twee documenten (34) een nieuw menu te creëren en het vervolgens naar de overzijde van de tafel te werpen. Op figuur 5.3 is het resultaat van deze actie te zien. Alle menu's zijn gesynchroniseerd zodat zij steeds dezelfde informatie tonen aan alle gebruikers. Helaas laat de hardware-configuratie geen identificatie van individuele gebruikers toe. In dat geval hadden menu's kunnen voorzien worden op maat van de individuele gebruiker.

## 5.3 Interactietechnieken

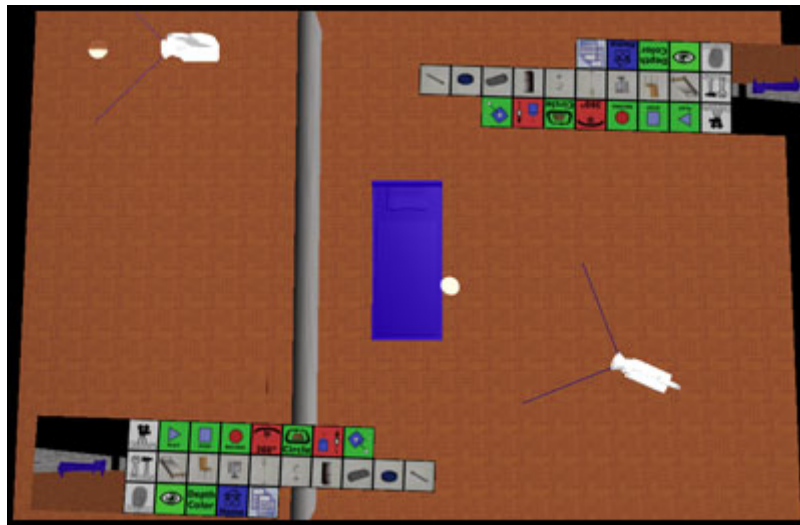
### 5.3.1 Transleren

#### Plane translation

Om objecten te transleren in de scene volstaat het om met je vinger het object aan te duiden en het vervolgens te verslepen naar de nieuwe locatie. Hetzelfde geldt voor de menu's in de scene.

#### iLevitate

De 'iLevitate'-techniek is een aanvulling op de standaard translatie en laat toe om objecten te verplaatsen in de hoogte. Deze techniek maakt hiervoor gebruik van twee invoer-



Figuur 5.3: De menu interface werd gekopieerd zodat gebruikers het aan twee zijden kunnen bedienen.

punten. Eerst moet het object geselecteerd worden en de 'iLevitate' modus (3) worden ingeschakeld. Daarna kan de gebruiker zijn tweede vinger op het canvas plaatsen. Op dat moment zal een widget verschijnen dat gebruikt kan worden om het object in de hoogte te verplaatsen. Dit widget is te zien op figuur 5.4 en fungeert als schuifregelaar. De schuifregelaar omhoog verplaatsen, heeft als effect dat het object dezelfde beweging zal maken. De schuifregelaar omlaag bewegen, zorgt er op zijn beurt voor dat het object omlaag beweegt.

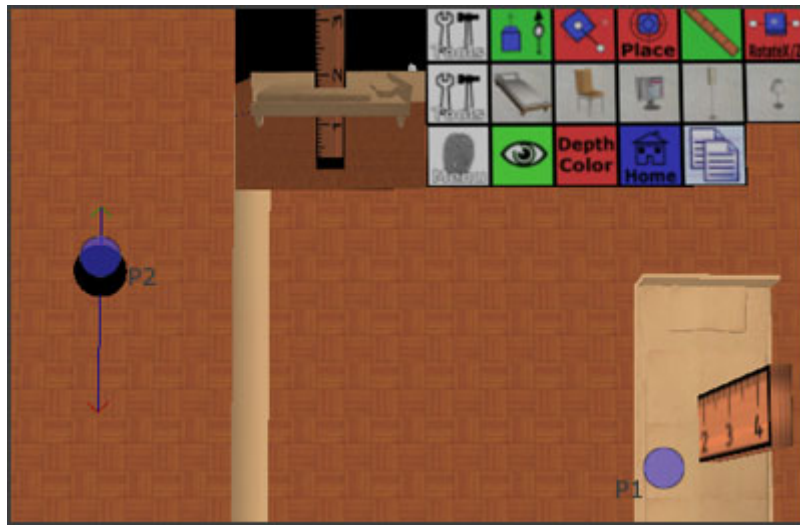
### **iPlace**

Opmerkelijk is dat het niet altijd nodig is om de exacte hoogte van een object te controleren. Dikwijls volstaat het om aan te geven waarop je een object wil plaatsen. Deze techniek biedt de gebruiker deze mogelijkheid aan. Eerst duidt de gebruiker het te verplaatsen object aan en activeert hij de 'iPlace' modus (4) in het menu. Vervolgens volstaat het een tweede vinger op een ander object te plaatsen. Hierdoor verschijnt het eerste object bovenop het tweede object. Dit bespaart ons de inspanning een object naar de juiste locatie te verplaatsen.

### **Meetlat**

Het is niet evident voor een gebruiker om te bepalen op welke hoogte hij eigenlijk aan het werken is. Vandaar werd een meetlatmetafoor toegevoegd aan het menu. Deze kan worden geactiveerd door het icoon met de meetlat (5) aan te duiden. De meetlat verschijnt vervolgens in het centrum van het te verplaatsen object zoals wordt geïllustreerd op figuur 5.4. In de viewport aan het menu is te zien op welke hoogte een object zich juist bevindt. De meetlat fungeert hier immers als referentie.





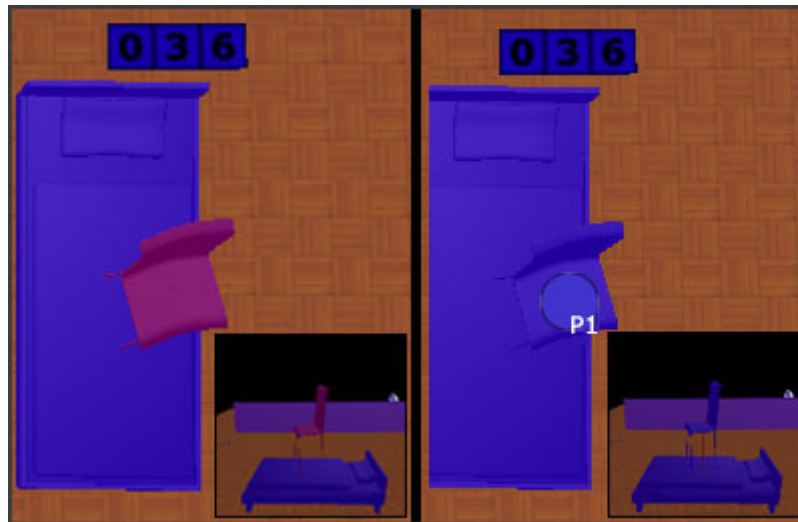
Figuur 5.4: Links bovenaan vinden we het 'iLevitate' widget terug dat door P2 wordt gecontroleerd. Bovendien is de meetlat zichtbaar zodat een referentiekader voor de hoogte aanwezig is.

### Hoogtemeter

Het menu geeft ook steeds aan op welke hoogte de bovenzijde van een object zich bevindt. Deze hoogtemeter is te zien op figuur 5.5. Merk op dat deze ook van een variabele kleur voorzien is. Deze kleur is afhankelijk van de hoogte waarop de bovenzijde van het object zich bevindt. Het is immers niet altijd nodig om exact te weten op welke hoogte een object zich bevindt. Het is dikwijls voldoende om te weten wat het relatieve hoogteverschil is tussen twee objecten. Vandaar werd een extra mode toegevoegd: 'Depth Coloring'.

### Depth Coloring

'Depth Coloring' zorgt naast de schaduwen in de scene voor een extra dieptehint. Deze modus wordt geactiveerd door in het menu te kiezen voor het icoon met de tekst 'Depth Color' (32). Eens deze modus geactiveerd is, krijgen alle objecten in de scene een kleur toegekend op basis van de hoogte van de bovenzijde. Op uitzondering van het object dat je op dat moment selecteert. Dit object krijgt een kleur toegekend op basis van de hoogte van zijn onderzijde. Dit is handig wanneer je dit object op een ander object wil plaatsen. Je kan je dan baseren op de kleuren. Wanneer de kleuren overeenkomen, komt ook de onderzijde van het object overeen met de hoogte van het object waar het op geplaatst wordt. Rechts op figuur 5.5 is te zien hoe de kleuren van stoel en bed overeenkomen, wat erop wijst dat de stoel zich op het niveau van het bed bevindt.



Figuur 5.5: Bovenaan in beeld bevindt zich de hoogtemeter die dient als referentie voor de hoogte van een object. Depth Coloring is bovendien ingeschakeld, links wordt de stoel niet aangeraakt, rechts wel waardoor de stoel een andere kleur krijgt toegewezen en de kleuren van bed en stoel overeen komen.

### XZ-locator

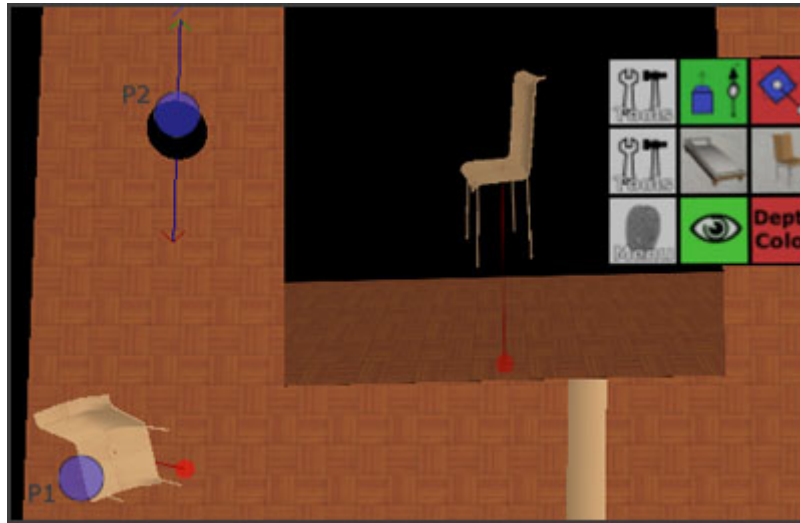
Een bijkomende hulp voor het positioneren van objecten zijn de XZ-locators. Dit zijn eenvoudige markers die aangeven waar dat een object zich bevindt ten opzichte van het X/Z-vlak. Door het perspectief binnen de scene kan deze positie immers sterk vertekenen. Deze XZ-locators worden voorgesteld door een lijn met aan het einde een transparante rode bol zoals te zien is op figuur 5.6.

### 5.3.2 Roteren

Rotaties zijn zeer belangrijk in deze toepassing. In eerste instantie voor de oriëntatie van de menu's. Deze moeten immers op een eenvoudige manier gedraaid kunnen worden zodat je het menu niet omgekeerd moet bedienen. Rotaties zijn eveneens van groot belang om de fysieke objecten in de scene naar believen te kunnen heroriënteren en dat in drie dimensies. In deze toepassing werden drie rotatietechnieken geïntegreerd, tweepunts Y-rotatie, driepunts X/Z rotatie en driepunts Shallow-Depth rotatie.

#### Tweepunts Y-rotatie

Deze rotatietechniek gebruik je om rotaties uit te voeren rond de Y-as zoals te zien is op figuur 5.7. Je plaatst hiervoor één vinger op het te roteren object. Vervolgens schakel je het menu in de juiste modus door het icoon (2) van de Y-rotatie aan te duiden. Hierna kan je overgaan tot Y-rotatie door je tweede vinger op het canvas te plaatsen. Vanaf dat



Figuur 5.6: Zowel in het bovenaanzicht als in de viewport is de xz-locator duidelijk zichtbaar.

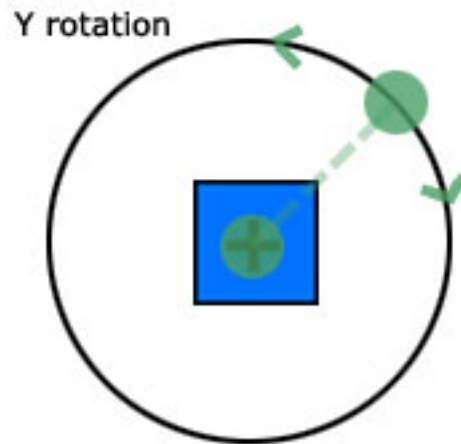
moment verschijnt er een lijn tussen beide vingers ter ondersteuning van de actie (zie figuur 5.8). Je krijgt op die manier de indruk dat de tweede vinger via een hendel verbonden is met het te roteren object.

Hetzelfde geldt overigens voor de scene. Ook deze kan worden geroteerd via deze techniek. Dit is nuttig om er voor te zorgen dat een gebruiker, die zich aan de overzijde van de tafel bevindt, op een snelle manier toegang krijgt tot objecten die buiten zijn bereik liggen. Even de scene omdraaien, lost in dit geval al veel problemen op.

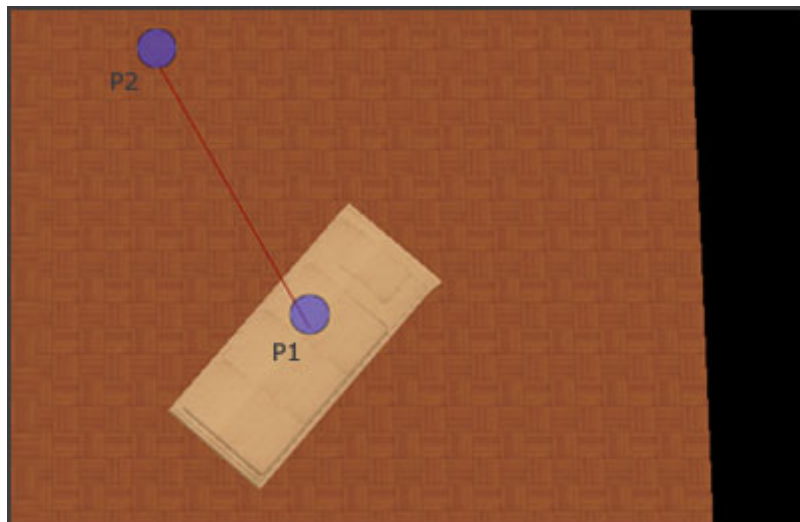
Helaas is deze techniek niet geschikt voor rotaties rond de overige twee assen. Vandaar zijn twee bijkomende rotatietechnieken voorzien om hieraan tegemoet te komen. Ook voor het menu passen we deze techniek toe. In dat geval plaatst de gebruiker wederom één vinger op de placeholder (30), waarna hij vervolgens het menu roteert op dezelfde wijze.

### Driepunts X/Z rotatie

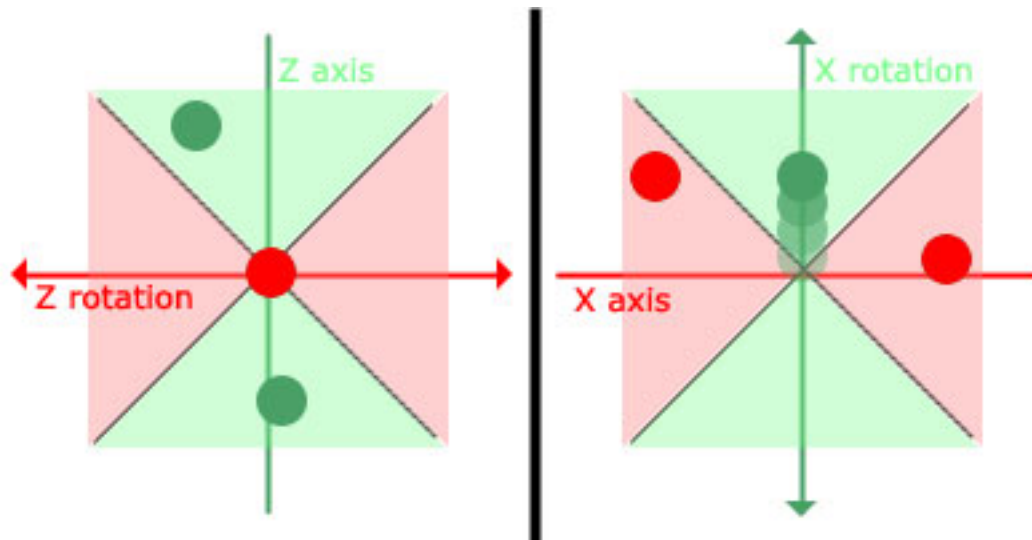
Hierbij duidt je eerst het te roteren object aan, en kies je voor de juiste modus (10) in het menu. Vervolgens plaats je twee extra vingers op het canvas. Op figuur 5.9 worden deze twee vingers links in het groen en rechts in het rood aangegeven. Het is op basis van deze twee vingers dat de toepassing bepaalt rond welke as er wordt geroteerd. In het geval links, geven de twee punten de Z-as aan. Dit komt omdat de punten zich in de groene quadranten rondom de Z-as bevinden. Op dat moment wordt de rotatie-as getoond zoals te zien is op figuur 5.10.



Figuur 5.7: De richting waarin de vinger beweegt, bepaalt de rotatierichting.

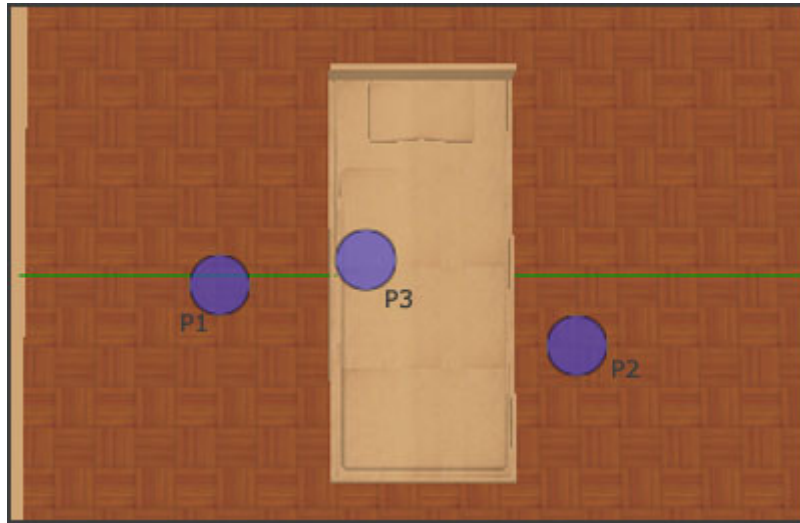


Figuur 5.8: De tweepunts Y-rotatie in actie. De rode lijn tussen P1 en P2 geeft de indruk een hendel te zijn.



Figuur 5.9: Links rotatie rond de Z-as, rechts rotatie rond de X-as.

De vinger die zich nog steeds op het object zelf bevindt, bepaalt de richting van de rotatie. Een kleine verplaatsing in de richting van de pijl zal leiden tot een kleine rotatie, terwijl verder weg bewegen van het object een grotere rotatie met zich mee zal brengen. Ondertussen kan je steeds je vinger opnieuw verplaatsen naar het object zodat je kan blijven verder roteren. Het voordeel van deze techniek is dat je op een eenvoudige manier kan aangeven rond welke as je wil roteren. Toch is een basiskennis van assenstelsels vereist om het concept te vatten, en dit is zeker niet bij iedereen even goed ingeburgerd. Bovendien is het niet mogelijk om tegelijk te roteren rond om de Y-as via de voorgaande techniek. Dit is dan weer wel mogelijk via Shallow-Depth rotatie.

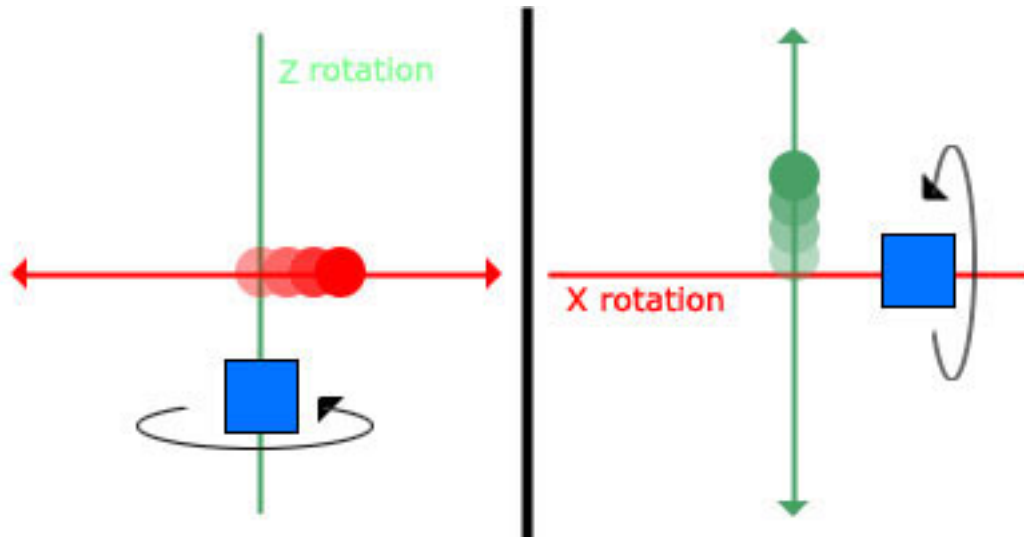


Figuur 5.10: De rotatie-as wordt bepaald door P1 en P2 en wordt in het groen weergegeven. De verplaatsing van P3 zal leiden tot een rotatie rond deze as.

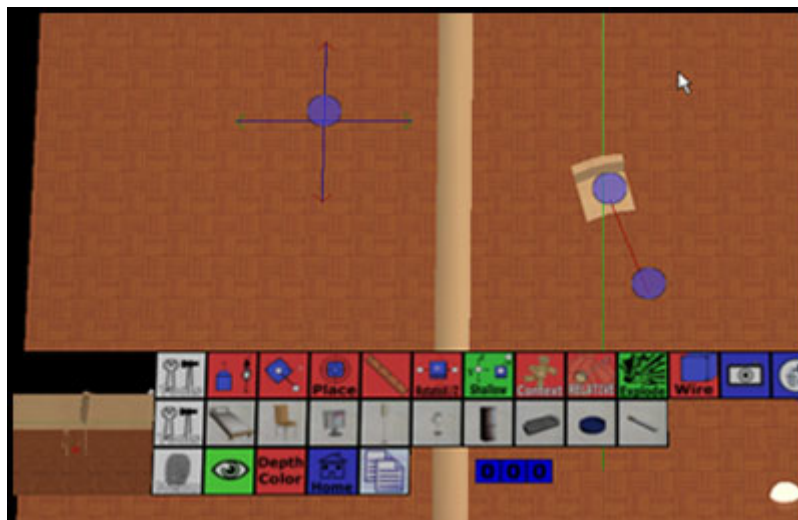
### Driepunts Shallow-Depth rotatie

Driepunts Shallow-Depth rotatie [17] heeft het voordeel dat er in één beweging over drie assen kan worden geroteerd. Rotaties rond de Y-as gebeuren via de eerder besproken tweepunts Y-rotatie. Om over te gaan tot deze rotatiemodus moet eerst in het menu gekozen worden voor de 'Shallow depth' modus (12). Je kan vanaf dan via een derde vinger de rotatie rond de overige assen controleren. Translaties met deze derde vinger over de X-as vertalen zich naar rotaties rond de Z-as zoals links op figuur 5.11 wordt geïllustreerd. Andersom vertalen translaties van deze derde vinger over de Z-as (scherm Y-as) zich naar rotaties rond de X-as zoals rechts op figuur 5.11 wordt geïllustreerd.

Ook hier wordt je ondersteund. Tijdens de translatie van de derde vinger geeft de toepassing immers de translatie-as aan. Op figuur 5.12 is deze duidelijk zichtbaar. Deze techniek geeft je veel vrijheid. Je kan in één beweging zowel transleren in twee dimensies als roteren in drie dimensies. Keerzijde van de medaille is dat het niet eenvoudig is om zoveel vrijheidsgraden ineens te controleren, aldus Hancock et al. [17].



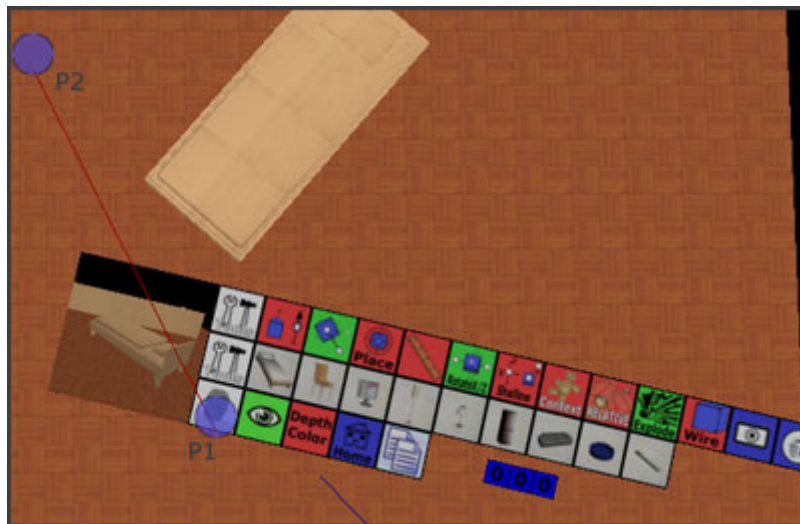
Figuur 5.11: Rotaties rond de Z-as gebeuren door over de X-as te transleren (links) en vice versa (rechts).



Figuur 5.12: Y-rotatie gebeurt door P2 rond P1 te bewegen, terwijl rotatie rond de overige assen wordt gecontroleerd door P3. P3 wordt in dit geval naar links bewogen waardoor de Z-as door het object verschijnt.

### 5.3.3 Werken op verschillende schaal

Binnen deze applicatie wordt er gewerkt met fysieke objecten van verschillende maten en gewichten. Zo kan je net zo goed een koelkast als een asbak toevoegen aan de scene. Om beide objecten te kunnen manipuleren op een gebruiksvriendelijke manier moet er een mechanisme aanwezig zijn dat de gebruiker helpt om dit te verwezenlijken. De asbak zal al snel verdwijnen onder de vingertop van de gebruiker. Vandaar werden enkele technieken geïmplementeerd om dit probleem te verhelpen. Een eerste techniek voert een offset in om ervoor te zorgen dat kleine objecten niet onder de vinger verdwijnen. Een tweede techniek gebruikt zooming om op een aangepaste schaal te kunnen werken. Een derde techniek is gebaseerd op de voodoo dolls van Pierce et al. [32].



Figuur 5.13: De afstand tussen P1 en P2 bepalen de grootte van het menu. Het verplaatsen van P2 zal het menu roteren.

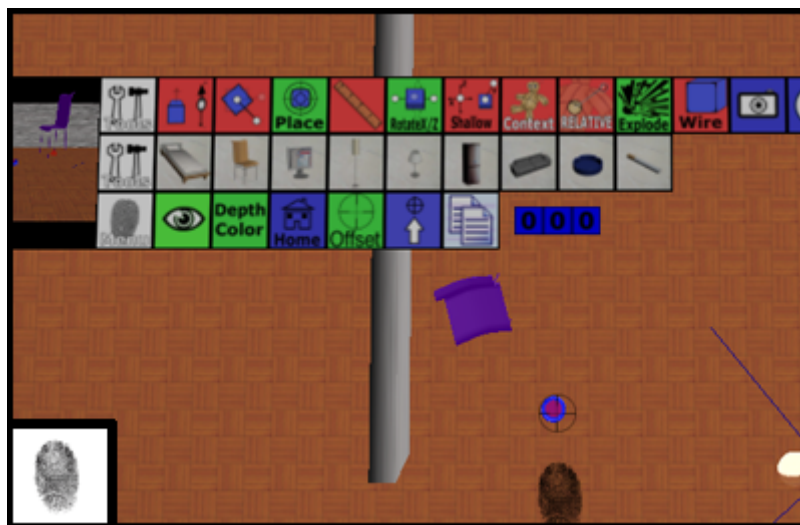
#### Scaleren van menu's

Eerder werd al kort aangehaald dat het best handig kan zijn om menu's te scaleren om kostbare schermruimte vrij te houden. Scaleren van de menu's kan door de menu-placeholder (30) te selecteren en vervolgens een tweede vinger op het canvas te plaatsen. Deze tweede vinger dicht bij de menu-placeholder brengen zorgt ervoor dat het menu verkleint. Door de vinger verder weg te bewegen, zal het menu vergroten. Dit kan in sommige gevallen ook handig zijn wanneer de gebruiker dikkere vingers heeft, of slechtziend is. Het scaleren van menu's verloopt tegelijk met het roteren van het menu via tweepunts rotatie zoals op figuur 5.13 te zien is.



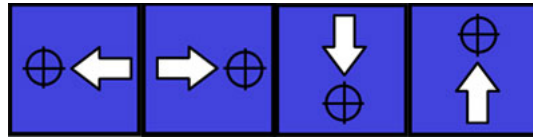
### Finger offset

Kleine objecten kunnen deels of volledig worden bedekt door de vinger van de gebruiker. Via een eenvoudige ingreep kan dit euvel verholpen worden. Door het introduceren van een offset tussen vinger en 'cursor' heb je toch de mogelijkheid om kleine objecten te blijven zien tijdens het selecteren. Bovendien zorgt dit er voor dat je ook iets preciezer te werk kan gaan [4]. Het nadeel is wel dat je als gebruiker de objecten niet meer rechtstreeks kan aanraken, zodat het gevoel van directe manipulatie toch voor een deel verdwijnt. Op figuur 5.14 wordt de positie van de vinger getoond via een vingerafdruk, en is de offset duidelijk vast te stellen.



Figuur 5.14: De offset tussen vinger en cursor (crosshair) zorgt ervoor dat de asbak niet onder de vinger belandt.

Deze modus activeer je door in het menu het icoon met de crosshair (35) aan te toetsen. Vervolgens verspringt de cursor op basis van de richting van de offset. Deze richting wordt aangegeven door het icoon met pijl en crosshair (36) in het menu. Afhankelijk van je positie, kan je hier de richting van de offset aanpassen. De mogelijkheden zijn zichtbaar op figuur 5.15. Indien de tafel in staat zou zijn de locatie van de gebruiker vast te stellen, zou dit laatste overbodig worden. Helaas is dit in deze hardware-configuratie niet mogelijk.



Figuur 5.15: De richting van de offset wordt bepaald door het bijbehorende icoon (36) in het menu. De vier mogelijkheden even op een rijtje.

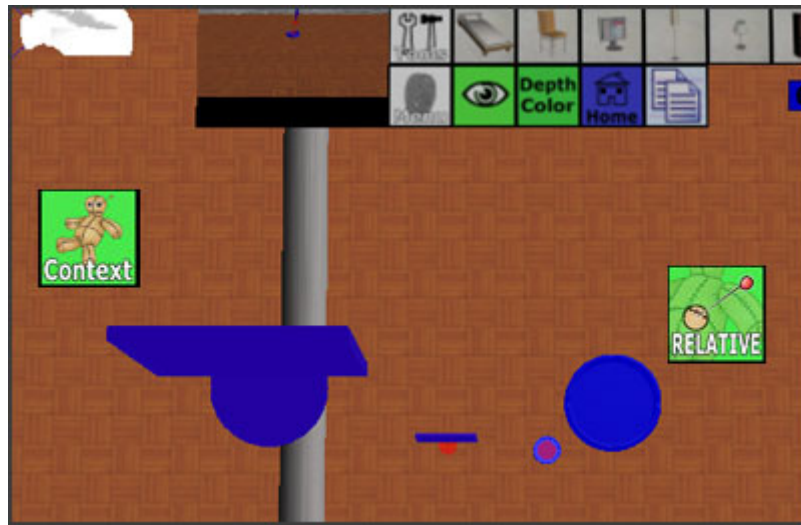
### Zooming

De zoomingstechniek roep je op door twee vingers te plaatsen in de scene en deze dichterbij mekaar te brengen (inzoomen) of verder van mekaar weg te bewegen (uitzoomen). Deze techniek komt overeen met 'dual finger stretch'.

### iContext

'iContext' is een alternatief voor zooming en is geïnspireerd op de 'Voodoo dolls' techniek. Hierbij kan een object worden gekozen dat fungeert als context voor de interactie. Dit door even in het menu op het contexticoon te tikken (6). Op dat moment verschijnt een kopie van dit object met een vast gedefiniëerde grootte. De context is statisch, kan niet gemanipuleerd worden en dient enkel als referentiekader voor een tweede object. Dit tweede object kan worden gekozen door het aan te duiden en in het menu te kiezen voor 'relative' (7). Vanaf dat moment verschijnt een tweede kopie centraal in de wereld, zoals te zien is op figuur 5.16. De grootte van dit object is relatief aan de grootte van het referentieobject. Voor de afstand tussen dit tweede object en het referentiekader geldt hetzelfde. Deze tweede kopie kan vervolgens worden gemanipuleerd via dezelfde interactietechnieken die anders beschikbaar zijn.

Deze techniek zorgt ervoor dat men een asbak op eenzelfde schaal als een koelkast kan manipuleren mits de context goed gekozen is. Het is echter niet altijd even eenvoudig om de juiste context te kiezen. Bovendien kan het zijn dat de afstand tussen context en manipulatieobject zo groot is dat dit buiten het scherm komt te liggen. Dat kan verwarrend zijn voor de gebruiker. Deze techniek is niet zo voordehandliggend voor de gebruiker en vereist wellicht de nodige training.



Figuur 5.16: Links bovenaan vinden we de context terug (computerscherm). Rechts ervan vinden we het relatieve object (asbak) terug.

### 5.3.4 Lange afstanden

#### Flicking

Lange afstanden zijn een typisch probleem van tabletops. Stel dat je een object wenst door te geven aan één van je collega's aan de overzijde van de tabletop. Je moet dan rechtstaan en over de tafel reiken om het object door te geven. Dit werkt niet makkelijk en om ergonomische redenen is het eenvoudiger om dit object te werpen naar de overzijde. Vandaar werd 'Flicking' geïntegreerd tijdens het transleren. Wanneer je een object met een bepaalde snelheid verplaatst, zal dit object na loslaten nog even verder bewegen. De snelheid waarmee dit gebeurt, zal exponentieel afnemen en uiteindelijk nul worden.

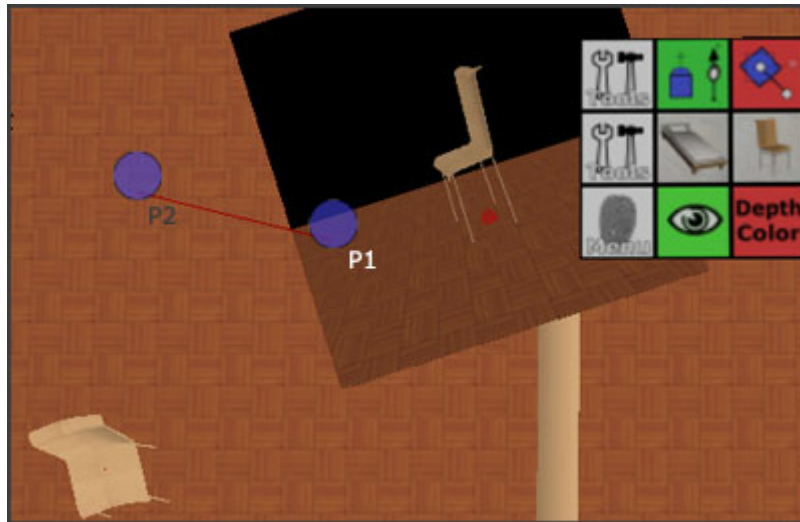
## 5.4 Occlusie-management

Ook occlusies zijn een hekel punt in 3D-omgevingen, dikwijls zijn objecten deels of volledig verborgen achter andere objecten. Om dit tegen te gaan, werden een aantal technieken toegevoegd aan de toepassing.

#### Multiple viewports

Werken met meerdere viewports kan veel problemen oplossen althans wat betreft object-discovery. In de scene zijn meerdere camera's voorzien die zo kunnen geplaatst worden dat zij uit alle hoeken een situatie weten weer te geven in de viewport van het menu. Elk object heeft ook zijn eigen virtuele camera, die de gebruiker het zijaanzicht toont. Deze viewports kunnen ook naar wens worden groteerd en gescalaerd via zooming en tweepunts

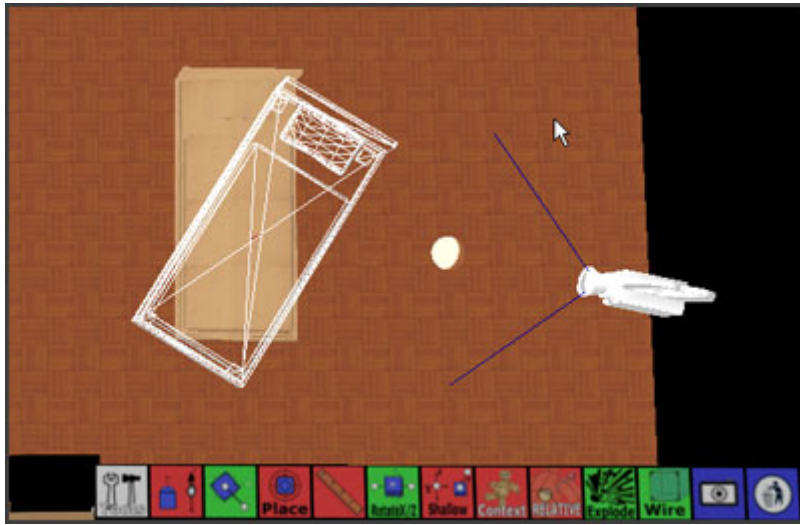
Y-rotatie, zoals op figuur 5.17 wordt getoond. Het camera-management komt nog verder aan bod in de gelijknamige sectie later in dit hoofdstuk.



Figuur 5.17: De viewport werd los van het menu geroteerd en gescaleerd in één vloeiende beweging.

### Wireframe

Een eerder naïeve maar doeltreffende benadering bestaat erin het object te tonen in 'wireframe' modus zodat achterliggende objecten kunnen ontdekt worden. Dit is in zekere mate equivalent aan het transparant [11] maken van het object en ook 'wireframes' kampen met een verlies aan diepteperceptie. Deze modus kan worden opgeroepen door het object aan te duiden en in het menu te kiezen voor het 'Wireframe' icoon (13). Het object zal vervolgens als 'wireframe' worden getoond zoals op figuur 5.18 gedemonstreerd wordt. Helaas is deze methode ontoereikend wanneer er zich meer dan één object in de 'line of sight' bevindt. Om die reden werd de 'interactive exploder'-techniek toegevoegd.



Figuur 5.18: Het bed bovenaan wordt in wireframe getoond.

### Interactive exploder

De 'interactive exploder' kan worden opgeroepen door een object te kiezen en vervolgens het 'explode' icoon (14) aan te duiden in het menu. Dit zal er voor zorgen dat alle objecten die zich achter het geselecteerde object bevinden, zich trapsgewijs zullen verplaatsen naar buiten, zodat zij zichtbaar worden voor de gebruiker. De selectie kan ook worden teruggeplaatst door het nieuw verschenen 'implode' icoon aan te duiden. Een voorbeeld van een 'exploded view' vind je onderaan terug op figuur 5.19. Het voordeel van deze techniek is dat alle objecten die zich onder het geselecteerde object bevinden, zichtbaar worden. Helaas gaat dit ten koste van de ruimtelijke relaties tussen de objecten [43].



## 5.5 Camera-management

### Directe manipulatie

Buiten het manipuleren van het bovenaanzicht, heb je als gebruiker ook de mogelijkheid om camera's te manipuleren. Op die manier kan je de scene bekijken vanuit meerdere standpunten door vrij eenvoudig de camera met de vinger te verplaatsen. Het gezichtsveld van de camera wordt aangegeven door de blauwe lijnen zoals te zien op figuur 5.18. Het beeld van de camera wordt vervolgens weergegeven in de viewport van het cameramenu.

Het gezichtsveld van de camera kan worden verlegd door de camera te roteren rond de Y-as via de eerder besproken tweepunts Y-rotatie. De cameraobjecten werden verder uitgebreid met een focuspunt dat ook kan worden aangewend om de camera te draaien. Dit kan eenvoudig door de focus van locatie te verplaatsen. Deze focus wordt gerepresenteerd door de witte cirkel op figuur 5.20. Ook de hoogte waarop de camera zich bevindt kan verder aangepast worden via 'iLevitate'. Dezelfde technieken werden bewust gebruikt om de applicatie niet nodeloos complexer te maken.

### iTour

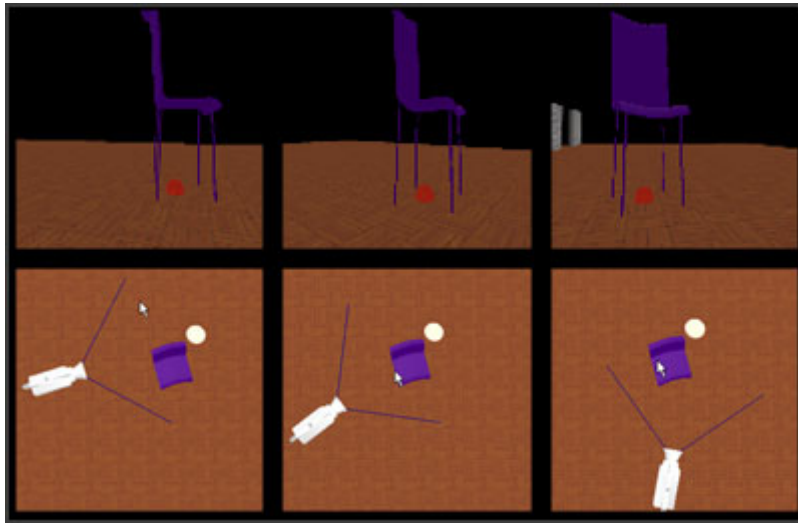
Verder is het mogelijk via 'iTour' een pad op te nemen naar analogie van het 'Tour Planner' concept. Je kiest hier voor de 'record'-knop (24). Vanaf dat moment switcht de toepassing naar recording mode en worden alle bewegingen van de camera opgeslagen. Vervolgens kan er gekozen worden om het gedefinieerde pad opnieuw te laten bewandelen door de camera. Dit door even de 'play'-knop (21) aan te toetsen. Vanaf dat moment zal de camera het pad opnieuw afleggen, zodat de je je optimaal kan concentreren op het verkennen van de omgeving. Het pad kan ook tijdelijk onderbroken worden door even de 'pause'-knop (22) aan te toetsen, of worden gereset door de stop knop (23) aan te toetsen.

### Panorama

De camera kan je ook zo instellen dat hij continu rond zijn eigen as draait. Op die manier verkrijg je een 360 graden panoramisch zicht van de omgeving. Dit is handig om de omgeving vanuit een centraal standpunt te verkennen. Het blijft in deze modus overigens mogelijk om de camera te manipuleren.

### Orbit

Het kan ook zijn dat je een bepaald punt in de scene wil bekijken van buitenaf. Dit kan via de 'circle'-functie (25) in het menu. Wanneer je hiervoor kiest zal de camera rondom zijn eigen focuspunt roteren. De camera zweeft dan als het ware als een satelliet rondom zijn eigen focuspunt zoals te zien is op figuur 5.20. Ook tijdens deze modus kan het focuspunt nog worden verlegd.



Figuur 5.20: De camera zweeft rondom zijn eigen focus. Bovenaan wordt telkens de bijbehorende viewport getoond. De focus wordt gevisualiseerd door de witte cirkel.

## 5.6 Physics

'Interior Designer' maakt ook gebruik van een physics engine om een meer realistisch en meer voorspelbaar karakter te geven aan de fysieke objecten in de scene. Mensen beroepen zich immers op de kennis die zij hebben opgebouwd in het dagelijkse leven en hebben dus ook een bepaald beeld over hoe een fysiek object hoort te bewegen. De physics in de scene zorgen er bovendien voor dat fysieke objecten op een realistische manier op mekaar kunnen geplaatst worden. De bewegingsvrijheid van de objecten wordt dus beperkt via fysische constraints. Het is dankzij deze constraints dat het bijvoorbeeld onmogelijk is twee objecten in mekaar te plaatsen. Iets wat erg verwarrend overkomt bij de gebruiker.

## 5.7 Gebruikte technologie

### 5.7.1 Hardware

De implementatie van 'Interior Designer' werd gerealiseerd op een tabletop die gebaseerd is op de technologie van Jeff Han. Deze opstelling wordt getoond op figuur 5.21 en slaagt erin via een high-definition projector een beeld te realiseren van 1920x1200 pixels.

### 5.7.2 Software

De implementatie is geschreven in C++ en gebruikt ter ondersteuning vooral open source technologieën. Een kort overzicht:





Figuur 5.21: Tabletop setup in het expertisecentrum voor digitale media. [6]

- **OpenGL v2.1** of 'Open graphics library' [29] is een open source API voor het genereren van 2D en 3D graphics.
- **OGRE v1.4.9** of 'Object-Oriented Graphics Rendering Engine' [28] is een open source, scene-georiënteerde 3D-engine.
- **ODE v0.9** of 'Open Dynamics Engine' [27] is een open source physics engine die kan worden gebruikt om simulaties uit te voeren.
- **QT v4.3.4** van Trolltech [44] is een applicatie framework dat vrij te gebruiken is onder bepaalde voorwaarden.

## 5.8 Samenvatting

In dit hoofdstuk werd 'Interior Designer' besproken, een collaboratieve applicatie die gebruik maakt van een tabletop ter ondersteuning van binnenhuisarchitecten. 'Interior Designer' probeert een aantal specifieke problemen aan te pakken die voorkomen wanneer een 3D-omgeving wordt geïntegreerd op een tabletop. De problemen zijn legio, denk maar aan heroriëntatie van data, rotaties in drie dimensies, schaalverschillen, het afleggen van afstanden, het voorkomen van occlusies en dergelijke. Via een uitgebreid gamma aan interactietechnieken biedt 'Interior Designer' een aantal oplossingen op elk van deze problemen. Een gebruikersevaluatie is nodig om vast te stellen of de aangereikte technieken ook daadwerkelijk leiden tot een gebruiksvriendelijke en werkbare toepassing.

# Hoofdstuk 6

## Gebruikersevaluatie

### 6.1 Inleiding

Om na te gaan of deze technieken ook werkelijk een oplossing bieden op de problemen die tabletop interactie in een 3D omgeving met zich mee brengt, werd een informele gebruikersevaluatie uitgevoerd. Verder werd ook even nagegaan in hoeverre de gebruikers met minimale training de gebruikte technieken meester waren. De evaluatie gebeurde via de analyse van videobeelden en het afnemen van een vragenlijst.

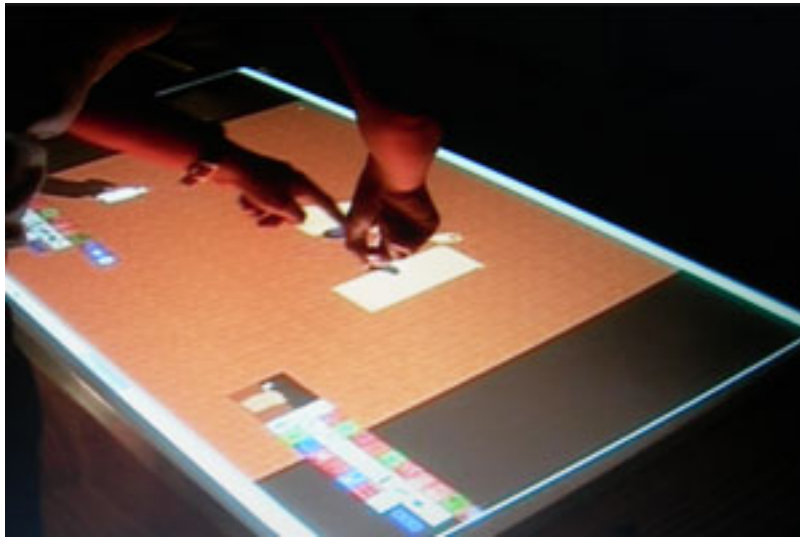
### 6.2 Testopstelling

#### 6.2.1 Omgeving

De testen werden uitgevoerd in het Expertisecentrum voor Digitale Media (EDM) te Diepenbeek. Alle testen werden uitgevoerd op een digitale tafel die gebruik maakt van de FTIR-technologie van Jeff Han. De testsessies werden telkens opgenomen voor latere analyse met een camera. Op figuur 6.1 zien we aan gebruiker aan het werk.

#### 6.2.2 Testgebruikers

De test werd uitgevoerd door vijf test-gebruikers tussen 18 en 25 jaar oud waaronder twee vrouwen en drie mannen. De testers hadden ook elk een uiteenlopende achtergrond en uiteenlopende ervaringen met computers. Alle testgebruikers waren het er unaniem over eens dat het een leuke ervaring was om met een tabletop te kunnen werken en iedereen gaf aan dat zij graag in de toekomst met een tabletop willen werken. Nochtans had geen van hen al eerder met een tabletop gewerkt.



Figuur 6.1: Een gebruiker test Interior Designer op de tabletop in het EDM.

### 6.2.3 Sessieverloop

De interactieve sessie nam telkens 30 minuten in beslag. De testgebruikers kregen hierbij elk een bundel toebedeeld met daarin een briefing, het sessieverloop en een aansluitende vragenlijst. Deze bundel vindt men terug in de bijlagen van deze thesis. Na een korte introductie, werd er telkens een interactieve sessie afgewerkt waarbij een scene werd opgebouwd aan de hand van een afgebakend scenario. Hierbij werd er steeds een korte demonstratie gegeven van de gebruikte technieken, waarna de testgebruiker het heft in eigen handen mocht nemen. Elke sessie bestond uit drie delen:

- **Roteren van objecten**, de testgebruikers kregen de opdracht om enkele objecten aan te maken en te roteren met de verschillende rotatietechnieken.
- **Verplaatsen in de hoogte**, de testgebruikers kregen de opdracht om enkele objecten te stapelen via de geïmplementeerde technieken.
- **Camerabeheer**, de testgebruikers kregen de kans een vooraf opgebouwde scene te verkennen via de camera.

Na de interactieve sessie kreeg elke gebruiker rustig de tijd om even de bijbehorende vragenlijst in te vullen.

## 6.3 Resultaten

### 6.3.1 Menu

Eén iemand vond het menu te groot in omvang en verkleinde daarop prompt het menu. Iedereen was het er wel over eens dat het menu makkelijk te bedienen was. Verder was er nog iemand die het menu probeerde te verslepen door ergens lukraak het menu aan te raken met als gevolg dat een object werd geïntroduceerd in de scene.

### 6.3.2 Roteren van objecten

Het eerste deel van de sessie bestond uit het opbouwen van een scene waarbij de geïmplementeerde rotatietechnieken aan bod kwamen. Alle subjecten zijn erin geslaagd om alle rotaties tot een goed eind te brengen maar dit gebeurde niet bij iedereen even vlot. We overlopen even de resultaten per techniek.

#### Tweepunts Y-rotatie

De testgebruikers hadden opvallend weinig tot geen problemen bij het gebruiken van de Y-rotatie. Eén van de testgebruikers vergat wel even om het object aan te duiden en probeerde het object te roteren door één vinger rond het object te bewegen zoals in het two-point rotation door Wu et al. [47] reeds werd toegepast. Opvallend was verder dat alle gebruikers de techniek uitvoerden met één hand via duim en wijsvinger zodat zij steeds de tweede hand konden gebruiken om het menu te bedienen.

#### Driepunts X/Z rotatie

Ook rotaties over X- en Z-as verliepen via deze techniek zonder al te veel problemen. Eén subject vergat bij een eerste poging het te roteren object aan te raken, zij gaf de rotatie-as correct aan maar bewoog tevergeefs de derde vinger in de open ruimte. Echter zij vond zelf al snel de reden voor het falen van de actie en kon de interactie tot een goed einde brengen. Verder liet iemand weten problemen te hebben met assenstelsels, maar desondanks slaagde zij er toch in om het object correct te roteren. Uit de vragenlijst blijkt overigens dat alle testgebruikers deze techniek begrepen.

#### Driepunts Shallow-Depth rotatie

In tegenstelling tot 'X/Z rotatie' verliepen de rotaties via 'Shallow-Depth rotatie' heel wat moeizamer bij drie van de vijf testgebruikers. Eén subject begreep de techniek zelfs helemaal niet. Dit lijkt twee voorname redenen te hebben, enerzijds blijkt uit de video-analyse dat zij erg veel problemen hadden om de drie rotatierichtingen tegelijk te controleren. Het coördineren van de drie rotaties blijkt dus voor een aantal gebruikers te moeilijk te zijn.

Anderzijds is de ruimtelijke relatie tussen de vingers belangrijk. Het is steeds de vinger die zicht het kortst bij het object bevindt die de rotatie rond de Y-as beheerst en de vinger die zich het verst van het object bevindt die de rotatie over X- en Z-as bepaalt. Tijdens de interactie viel het wel eens voor dat beide vingers van positie werden gewisseld waardoor ook de as(sen) die ze beheersen werden gewisseld. Dit zorgde bij twee van de vijf testgebruikers voor verwarring. Dit wijst erop dat de visualisatie van deze techniek wellicht niet voldoende is. Eventueel zou de techniek ook zo kunnen geïmplementeerd worden zodat de betekenis van beide vingers niet wijzigt tijdens de interactie ongeacht de positie van beide vingers.

Bij twee testers verliep het roteren dan weer als een fluitje van een cent. Opvallend is hier dat beide subjecten actief zijn in de IT-wereld en ervaring hebben met 3D omgevingen in tegenstelling tot de andere testgebruikers.

### 6.3.3 Verplaatsen in de hoogte

Ook het verplaatsen van objecten in de hoogte verliep zonder al te veel problemen. Eén iemand haalde aan dat zij problemen had om de hoogte van een object correct in te schatten wat erop wijst dat er zich nog steeds te weinig dieptehints in de omgeving bevinden. Nochtans werd geprobeerd om via een kleur codering (Depth Coloring) een extra dieptehint te integreren in de omgeving, maar dat was evenmin duidelijk voor deze gebruiker. Dit in tegenstelling tot de andere gebruikers, die duidelijk hebben aangegeven in de vragenlijst dat deze extra dieptehint handig van pas komt om de hoogte correct te kunnen inschatten.

#### **iLevitate**

Het verplaatsen in de hoogte via 'iLevitate' leverde bij niemand problemen op. Het widget was nochtans zo gebouwd dat men telkens slechts een beperkte hoogte per keer kon overbruggen. Dit werd gedaan om te voorkomen dat gebruikers per ongeluk het object kwijt zouden raken als gevolg van een ongecontroleerde beweging. Om dus een groter hoogteverschil te overbruggen moest men dus telkens de tweede vinger even van het canvas lichten om verder te gaan. Dit gebeurde opvallend genoeg op spontane wijze en alle gebruikers slaagden erin om de nodige objecten op mekaar te plaatsen.

#### **iPlace**

Ondanks dat de 'iLevitate' techniek door iedereen vlot werd gebruikt, blijkt de voorkeur van de gebruikers toch uit te gaan naar de 'iPlace'-techniek. Dit komt wellicht door de eenvoud van de techniek. Echter één iemand probeerde om een object al slepend op een ander object te plaatsen in plaats van het object dat als basis dient, simpelweg aan te duiden. Zij ging er wellicht vanuit dat dit object de contouren van de omgeving zou volgen. Dit lijkt op het eerste zicht een valabel alternatief te meer omdat dit eenvoudig

kan worden geïntegreerd in de basistranslatie van de toepassing. Bovendien heb je hier als gebruiker slechts één vinger voor nodig, waardoor je de andere vingers vrij hebt voor andere doeleinden.

#### 6.3.4 Camerabeheer

Het gebruik van de camera's verliep vlot omdat men direct de associatie legde tussen de virtuele camera's in de scene en camera's in de fysieke wereld. Bij de 'orbit' modus was er wel een testgebruiker die de focus van de camera in een object plaatste en die terecht opmerkte dat het nu onmogelijk was geworden om de focus opnieuw te verleggen naar een nieuwe positie. Hij loste dit wel snel op door de camera zelf even te verplaatsen waardoor het euvel werd verholpen. Toch zou dit niet mogen gebeuren.

Het opnemen van een traject in de scene werd door iedereen tot een goed einde gebracht en verliep zonder problemen. Eén van de testgebruikers vroeg tijdens de test of het ook mogelijk was om de camera op zijn positie te laten en vervolgens de omgeving aan te passen en deze 'film' als het ware achteraf af te spelen. Dit werd echter niet geïmplementeerd in de toepassing niet echt een meerwaarde biedt binnen het concept 'Interior Designer'.

#### 6.3.5 Werpen van objecten

De gebruikers werd ook gevraagd of zij het makkelijk vonden om een object te werpen. De meningen lopen hierover nogal uiteen. Niet iedereen slaagde erin om een object tot aan de overzijde te werpen. Het is niet duidelijk waaraan dit te wijten is, wellicht vereist dit enige oefening. Echter een doorgedreven test is nodig om hier effectief uitsluitel over te geven.

### 6.4 Samenvatting

Uit de testen blijkt dat 'Interior Designer' op een paar tekortkomingen na is geslaagd in zijn opzet. Alle gebruikers zijn er immers in geslaagd de testsessies tot een goed einde te brengen. Toch is er nog veel ruimte voor verbetering. Meer dan eens vergat één van de gebruikers het object te selecteren en dit wijst er op dat zij niet altijd op de hoogte was, welk object op dat moment daadwerkelijk geselecteerd was. Dit geeft aan dat het zinvol is om een extra aanwijzing te voorzien.

Bovendien moet worden opgemerkt dat het niet voor alle gebruikers altijd even duidelijk was op welke diepte de objecten zich nu eigenlijk bevonden. Er moet dus worden bekeken op welke manier extra dieptehints kunnen worden geïntegreerd in de omgeving. Ook de verbuiging van het perspectief zorgde nogal eens voor verwarring.

Wat betreft de rotatietechnieken blijkt XZ rotatie het te halen van Shallow-Depth rotatie en dit wellicht vanwege de eenvoud van de techniek. Toch haalde één van de gebruikers aan dat hij wellicht toch voor Shallow-Depth zou kiezen bij veelvuldig gebruik, om deze techniek volgens hem sneller werkte.

Een algemene opmerking die dikwijls terugkeerde is dat de gebruikers niet altijd wisten hoeveel vingers zij eigenlijk nodig hadden om een bepaalde techniek uit te voeren. Dit is logisch, omdat deze 'affordances' niet duidelijk kunnen worden afgeleid uit de toepassing. Eén van de subjecten suggereerde zelfs dat een dynamische helpfunctie niet zou misstaan om de gebruiker te ondersteunen. Dit is zeker een werkpunt naar een eventuele volgende versie. Ook qua visualisatie van de technieken is er zeker nog ruimte voor verbetering aanwezig.

# Hoofdstuk 7

## Conclusie

In de laatste 25 jaar is er een grote evolutie geweest in de manier waarop mensen interageren met computers. Eerst werden computers bestuurd via commandotalen waardoor ze uitsluitend domein waren voor specialisten, maar hier kwam al snel verandering in via de introductie van grafische user interfaces. Vanaf dat moment werden computers in één klap toegankelijk voor het grote publiek. De computermarkt explodeerde met als gevolg dat computers de dag van vandaag overal aanwezig zijn. De onderzoekswereld is daarom steeds actief op zoek naar nieuwe manieren om mensen met computers te laten communiceren.

De tabletop vormt zo een nieuw medium en is een stap in de goede richting. Onze handen zijn fantastische instrumenten om te interageren met onze omgeving en dat is net de sterkte van deze technologie. Het laat ons toe te werken met onze handen via een medium dat invoer en uitvoer samenbrengt. Het is dan ook geen toeval dat dergelijke producten het zo goed doen bij het grote publiek.

Tabletop-interactie is niet zonder gebreken. Gebruikers kunnen om het even waar plaats nemen aan de tafel, waardoor het bepalen van een optimale oriëntatie niet evident is. Bovendien heeft niet elke gebruiker dezelfde fysieke eigenschappen. Data die voor sommige gebruikers nog binnen handbereik liggen, kunnen voor andere gebruikers onbereikbaar zijn. Ook de dikte van onze vingers speelt een rol bij het selecteren van kleine objecten. Onze vingers nemen uiteindelijk meer ruimte in dan een muiscursor waardoor precies werken niet altijd even gemakkelijk is.

We zijn tot de conclusie gekomen dat er reeds een groot aantal interactietechnieken zijn ontworpen om elk van deze specifieke problemen aan te pakken. Helaas beperkt het toepassingsgebied van deze technieken zich voor een groot deel tot tweedimensionale omgevingen. Het toevoegen van de derde dimensie zorgt voor een mismatch tussen invoer en uitvoer waardoor deze technieken dikwijls ontoereikend zijn. We zijn daarom verder gegaan met het analyseren van dit probleem en hebben vastgesteld dat we enkel door slim gebruik te maken van het multi-touch gebeuren een oplossing kunnen vinden voor rotaties in drie dimensies en het verplaatsen in de hoogte.



Vanuit deze wetenschap hebben we vervolgens een aantal oplossingen voorgesteld die werden gebruikt binnen de 'Interior Designer' toepassing. Deze toepassing heeft als doelstelling om mensen te ondersteunen bij het inrichten van hun woning en maakt hiervoor gebruik van een tabletop. Deze oplossingen werden vervolgens geëvalueerd aan de hand van een informele gebruikerstest.

Uit deze gebruikerstest blijkt dat onze voorstellen wel degelijk een oplossing bieden op deze problemen. Toch bleek het niet altijd duidelijk te zijn voor de gebruikers hoe zij nu een bepaalde techniek moeten gebruiken. Je kan als gebruiker uiteindelijk moeilijk zien of een techniek nu één, twee of meerdere vingers nodig heeft om te functioneren. Via de nodige visualisaties werd geprobeerd dit te verduidelijken, maar dit bleek niet voor iedereen voldoende te zijn. Een dynamische helpfunctie zou hier zeker voor verbetering kunnen zorgen.

Het 'Interior Designer' concept gaat uit van een collaboratieve omgeving waarin gebruikers samenwerken om tot gewenste resultaten te komen. Vandaar dat er werd gekozen voor menu's die eenvoudig te kopiëren zijn zodat elke gebruiker over zijn eigen set tools beschikt. Helaas laat de hardware niet toe om de acties op de tafel te koppelen aan specifieke gebruikers waardoor het dikwijls moeilijk is om tegelijk op de tafel actief te zijn. Dit is zeker een interessant topic om verder te onderzoeken.

Ondanks een aantal tekortkomingen, mogen we toch besluiten dat er zeker mogelijkheden zijn om via tabletops te interageren met driedimensionale omgevingen. Dat blijkt te meer uit het enthousiasme van de gebruikers, na het testen van onze applicatie, 'Interior Designer'. Het loont dan ook zeker de moeite om dit concept verder uit te diepen.

# Bibliografie

- [1] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 105–112, New York, NY, USA, 2003.
- [2] Ulf Assarsson, Niklas Elmqvist, and Philippos Tsigas. Employing dynamic transparency for 3d occlusion management. In *Interact '07: Proceedings of the 11th IFIP TC13 International Conference on Human-Computer Interaction*, 2007.
- [3] Patrick Baudisch, Edward Cutrell, Dan Robbins, Mary Czerwinski, Peter Tandler, Benjamin Bederson, and Alex Zierlinger. Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In *Interact '03: Proceedings of the 9th IFIP TC13 International Conference on Human-Computer Interaction*, pages 57–64, Zurich, Switzerland, 2003.
- [4] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 1263–1272, New York, NY, USA, 2006.
- [5] Duncan Cavens, Florian Vogt, Sidney Fels, and Michael Meitner. Interacting with the big screen: pointers to ponder. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 678–679, New York, NY, USA, 2002.
- [6] Tom Cuypers, Jan Schneider, Johannes Taelman, Kris Luyten, and Philippe Bekaert. Eunomia: Toward a framework for multi-touch information displays in public spaces. 2008.
- [7] James Davis and Xing Chen. Lumipoint: Multi-user laser-based interaction on large tiled displays. *Displays*, 23(5), 2002.
- [8] Paul Dietz and Darren Leigh. Diamondtouch: a multi-user touch technology. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 219–226, New York, NY, USA, 2001. ACM.
- [9] Hans-Peter Dommel and J. J. Garcia-Luna-Aceves. Floor control for multimedia conferencing and collaboration. *Multimedia Syst.*, 5(1):23–38, 1997.

- [10] Steven M. Drucker and David Zeltzer. Intelligent camera control in a virtual environment. In *Proceedings of Graphics Interface '94*, pages 190–199, Banff, Alberta, Canada, 1994.
- [11] Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3d occlusion management techniques. In *Proceedings of the IEEE Conference on Virtual Reality 2007*, pages 51–58, 2007.
- [12] David Goldberg and Cate Richardson. Touch-typing with a stylus. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 80–87, New York, NY, USA, 1993.
- [13] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–290, New York, NY, USA, 2005.
- [14] Tovi Grossman and Daniel Wigdor. Going deeper: a taxonomy of 3d on the tabletop. In *Tabletop*, pages 137–144. IEEE Computer Society, 2007.
- [15] Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. Object pointing: a complement to bitmap pointing in guis. In *GI '04: Proceedings of Graphics Interface 2004*, pages 9–16, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.
- [16] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, New York, NY, USA, 2005. ACM.
- [17] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1147–1156, New York, NY, USA, 2007.
- [18] Mark S. Hancock, Sheelagh Carpendale, Frederic D. Vernier, Daniel Wigdor, and Chia Shen. Rotation and translation mechanisms for tabletop interaction. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 79–88, Washington, DC, USA, 2006. IEEE Computer Society.
- [19] Uta Hinrichs, Holly Schmidt, Tobias Isenberg, Mark Hancock, and Sheelagh Carpendale. Technical report.
- [20] Apple inc. <http://www.apple.com>.

- [21] Martin Kaltenbrunner, Sergi Jorda, Gunter Geiger, and Marcos Alonso. The reactable\*: A collaborative musical instrument. In *WETICE '06: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 406–411, Washington, DC, USA, 2006. IEEE Computer Society.
- [22] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Saul Greenberg. How people use orientation on tables: comprehension, coordination and communication. In *GROUP '03: Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 369–378, New York, NY, USA, 2003.
- [23] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. Fluid integration of rotation and translation. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 601–610, New York, NY, USA, 2005.
- [24] Jörg Geißler. Shuffle, throw or take it! working efficiently with an interactive wall. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 265–266, New York, NY, USA, 1998.
- [25] Microsoft. <http://www.microsoft.com>.
- [26] Mitsubishi. <http://www.mitsubishi.com>.
- [27] Ode.org. <http://www.ode.org>.
- [28] Ogre3D.org. <http://www.ogre3d.org>.
- [29] OpenGL. <http://www.opengl.org>.
- [30] J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. Tractorbeam: seamless integration of local and remote pointing for tabletop displays. In *GI '05: Proceedings of Graphics Interface 2005*, pages 33–40, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.
- [31] Karen Parker, Mandryk Regan, and Michael Nunes. Tractorbeam selection aids: improving target acquisition for pointing input on tabletop displays. In *Interact '05: Proceedings of the 10th IFIP TC13 International Conference on Human-Computer Interaction*, pages 80–93, Rome, Italy, 2005.
- [32] Jeffrey S. Pierce, Brian C. Stearns, and Randy Pausch. Voodoo dolls: seamless interaction at multiple scales in virtual environments. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 141–145, New York, NY, USA, 1999. ACM.

- [33] Adrian Reetz, Carl Gutwin, Tadeusz Stach, Miguel Nacenta, and Sriram Subramanian. Superflick: a natural and efficient technique for long-distance object placement on digital tables. In *GI '06: Proceedings of Graphics Interface 2006*, pages 163–170, Toronto, Ont., Canada, Canada, 2006. Canadian Information Processing Society.
- [34] Jun Rekimoto. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *UIST '97: Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 31–39, New York, NY, USA, 1997.
- [35] Jun Rekimoto. Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120, New York, NY, USA, 2002. ACM.
- [36] Jun Rekimoto and Masanori Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 378–385, New York, NY, USA, 1999.
- [37] Meredith Ringel, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. Release, relocate, reorient, resize: fluid techniques for document sharing on multi-user interactive tables. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1441–1444, New York, NY, USA, 2004.
- [38] Stacey D. Scott, Karen D. Grant, and Regan L. Mandryk. System guidelines for co-located, collaborative work on a tabletop display. In *ECSCW'03: Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*, pages 159–178, Norwell, MA, USA, 2003. Kluwer Academic Publishers.
- [39] Stacey D. Scott, M. Sheelagh, T. Carpendale, and Kori Inkpen. Technical report.
- [40] Stacey D. Scott, M. Sheelagh, T. Carpendale, and Kori M. Inkpen. Territoriality in collaborative tabletop workspaces. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 294–303, New York, NY, USA, 2004.
- [41] Chia Shen, Kathy Ryall, Clifton Forlines, Alan Esenther, Frederic D. Vernier, Katherine Everitt, Mike Wu, Daniel Wigdor, Meredith Ringel Morris, Mark Hancock, and Edward Tse. Informing the design of direct-touch tabletops. *IEEE Comput. Graph. Appl.*, 26(5):36–46, 2006.
- [42] Graham Smith, Tim Salzman, and Wolfgang Stuerzlinger. 3d scene manipulation with 2d devices and constraints. In *GRIN'01: No description on Graphics interface 2001*, pages 135–142, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.

- [43] Henry Sonnet, Sheelagh Carpendale, and Thomas Strothotte. Integrating expanding annotations with a 3d explosion probe. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 63–70, New York, NY, USA, 2004. ACM.
- [44] Trolltech. <http://www.trolltech.com>.
- [45] Virtual-Drums.com. <http://www.virtual-drums.com>.
- [46] Aileen Worden, Nef Walker, Krishna Bharat, and Scott Hudson. Making computers easier for older adults to use: area cursors and sticky icons. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 266–271, New York, NY, USA, 1997.
- [47] Mike Wu and Ravin Balakrishnan. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 193–202, New York, NY, USA, 2003.
- [48] Robert Zeleznik and Andrew Forsberg. Unicam—2d gestural camera controls for 3d environments. In *I3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 169–173, New York, NY, USA, 1999. ACM.

# Bijlagen

- Bijlage 1: Briefing gebruikerstest
- Bijlage 2: Resultaten gebruikerstest

# Bijlage 1

## *Briefing*

Dat er een markt is weggelegd voor multitouch systemen is duidelijk te merken aan de inspanningen die bedrijven de laatste jaren hebben geleverd om kant en klare producten op de markt te brengen. Denk maar aan de populaire iPhone. Hetzelfde geldt voor tabletop systemen die zo stilaan hun weg beginnen te vinden in casino's, winkels en musea. Microsoft bekend van windows is dan ook vol in de weer om Microsoft surface op de markt te brengen.

Dat dergelijke systemen populair aan het worden zijn is geen toeval. Tabletops brengen immers invoer en uitvoer samen op één interactievlak en laten gebruikers toe om via natuurlijke bewegingen en handgebaren te interageren met digitale data. De technologie verdwijnt hierdoor wat naar de achtergrond waardoor u zich als gebruiker volledig kan focussen op uw taak.

'Interior designer' is een proof of concept dat van deze technologie gebruik maakt om te komen tot een interactieve oplossing voor binnenhuisinrichting. Via directe invoer is het de bedoeling dat u als gebruiker samen met de binnenhuisarchitect aan het werk kan gaan om uw huis in te richten.

Helaas zijn er ook een aantal problemen die opduiken bij het gebruik van deze technologie. In deze proof of concept, wordt getracht een aantal van deze problemen op te lossen via enkele interactie-technieken. Echter om na te gaan of deze interactie-technieken ook werkelijk bijdragen tot het oplossen van deze problemen, is het nodig een evaluatie uit te voeren.

Deze evaluatie gebeurt aan de hand van een informeel onderzoek te samen met testgebruikers zoals u. U neemt straks deel aan een interactieve sessie waarbij u samen met een begeleider aan de tafel plaatsneemt. Deze sessie zal worden opgenomen zodat deze later kan ge-analyseerd worden. Uw gegevens blijven echter strikt confidencieel.

De begeleider zal steeds de actie demonstreren, waarna u deze actie op uw beurt dient te herhalen.

Merk op, dat niet u maar het product getest wordt en dat uw bijdrage erg belangrijk is in de evaluatie van dit product. Wij moedigen u dan ook aan om luidop na te denken tijdens het uitvoeren van de test en te vertellen wat er in u opkomt. Als iets mislukt is dat te wijten aan het product en niet aan u. Bovendien mag u op elk moment de test staken.

Aan het eind van de test, krijgt u nog een korte enquête om in te vullen die zal gebruikt worden als kwalitatieve indicator. Als u nog vragen heeft, mag u deze uiteraard altijd stellen voor, tijdens of na de sessie.



## Scenario

<p><b>Rotaties</b></p> <ul style="list-style-type: none"><li>● Object creatie (bed)</li><li>● Bed verplaatsen (translate)</li><li>● Bed rotatie (Y-rotatie)</li> <li>● Object creatie (stoel)</li><li>● Stoel verplaatsen (translate)</li><li>● Stoel rotatie (Y-rotatie)</li> <li>● Object creatie (kast/frigo)</li><li>● Kast verplaatsen</li><li>● Kast rotatie (X/Z-rotatie)</li><li>● Kast rotatie (Y-rotatie)</li><li>● Object creatie (stoel)</li><li>● Stoel verplaatsen (translate)</li><li>● Stoel rotatie (X/Z-rotatie)</li> <li>● Object creatie (kast/frigo)</li><li>● Kast verplaatsen</li><li>● Kast rotatie (Shallow-rotatie)</li><li>● Object creatie (stoel)</li><li>● Stoel verplaatsen (translate)</li><li>● Stoel rotatie (Shallow-rotatie)</li></ul>	<p><b>Verplaatsen in de hoogte</b></p> <ul style="list-style-type: none"><li>● Object creatie (bed)</li><li>● Bed verplaatsen (translate)</li><li>● Bed rotatie (Y-rotatie)</li> <li>● Object creatie (bed)</li><li>● Bed verplaatsen</li><li>● Bed in de hoogte verplaatsen (levitate)</li><li>● Bed op vorige plaatsen</li> <li>● Colorcoding aan</li><li>● Object creatie (kast)</li><li>● Kast verplaatsen</li> <li>● Object creatie (kast)</li><li>● kast rotatie (Y-rotatie)</li><li>● kast rotatie (X/Z-rotatie)</li><li>● Kast verplaatsen</li><li>● Kast in de hoogte verplaatsen (levitate)</li><li>● Kast op vorige plaatsen</li> <li>● Object creatie (scherm)</li><li>● Scherm via place op bovenste kast plaatsen</li> <li>● Colorcoding uit</li></ul>
<p><b>Camera's en interactive exploder</b></p> <p><i>3 kasten op mekaar plaatsen bed, stoel en lamp plaatsen</i></p> <ul style="list-style-type: none"><li>● Camera direct manipuleren</li><li>● Viewport groter maken (scale)</li><li>● Camera centraal plaatsen (360°)</li><li>● Camera direct manipuleren</li><li>● Camera rond de scene laten orbit (circle)</li><li>● Camera pad door scene opnemen</li><li>● Camera pad afspelen</li></ul>	<p><b>Werpen object</b></p> <ul style="list-style-type: none"><li>● werpen van een object. (throw)</li></ul>

## Vragenlijst

- ++: sterk mee eens
- +: mee eens
- 0: geen mening
- : niet mee eens
- : helemaal niet mee eens

### Algemeen

- Ik heb ... jaar ervaring met computers.
- Ik heb al eerder met een tabletop gewerkt?
  - Ja
  - Nee
  
- Ik voelde me op mijn gemak tijdens de test.

++	+	0	-	--

- Ik heb kennis van assenstelsels.

++	+	0	-	--

- Ik vond het een leuke ervaring om met een tabletop te werken.

++	+	0	-	--

- Ik zou in de toekomst graag werken met tabletops.

++	+	0	-	--

- Ik zou een tabletop systeem aanschaffen indien het op de markt komt tegen een betaalbare prijs.

++	+	0	-	--

- Ik doe beroep op een binnenhuis-architect wanneer ik mijn huis ga inrichten.

++	+	0	-	--

- Ik had het gevoel dat ik de objecten kon aanraken.

++	+	0	-	--

## Interior Designer

- Het 'Interior Designer' concept zou kunnen van pas komen bij het inrichten van uw huis.

++	+	0	-	--

## Selectie

- Ik kon de objecten makkelijk selecteren.

++	+	0	-	--

## Menu

- Het menu is makkelijk te verplaatsen.

++	+	0	-	--

- Het menu is makkelijk te draaien.

++	+	0	-	--

- Het menu is makkelijk te bedienen.

++	+	0	-	--

- Het menu neemt te veel plaats in.

++	+	0	-	--

## Rotatietechnieken

- Ik begrijp hoe de XZ-rotatie werkt.

++	+	0	-	--

- Ik begrijp hoe de Shallow-Depth rotatie werkt.

++	+	0	-	--

- XZ-rotatie reageert zoals ik wil.

++	+	0	-	--

- Shallow-Depth rotatie reageert zoals ik wil.

++	+	0	-	--

- XZ-rotatie laat me toe precies te werken.

++	+	0	-	--

- Shallow-Depth rotatie laat me toe precies te werken.

++	+	0	-	--

- Ik verkies XZ-rotatie boven Shallow-Depth rotatie.

++	+	0	-	--

## Verplaatsen in de hoogte

- Ik begrijp hoe het verplaatsen in de hoogte werkt.

++	+	0	-	--

- Het gebruik van kleuren is handig om in te schatten op welke hoogte ik zit te werken.

++	+	0	-	--

- Ik vind het makkelijker om een meubel rechtstreeks op een ander meubel te plaatsen.

++	+	0	-	--

## Camera management

- Ik kan de camera's makkelijk besturen

++	+	0	-	--

- Ik begrijp hoe ik een traject met de camera kan opnemen.

++	+	0	-	--

- Ik begrijp hoe ik de camera rond een punt kan laten bewegen.

++	+	0	-	--

- Ik vind het eenvoudig om de viewport (beeld) van grootte te wijzigen.

++	+	0	-	--

## Werpen

- Ik kon makkelijk een meubel werpen

++	+	0	-	--

Bedankt voor uw deelname!

## Bijlage 2

Nr	Vraag	S1	S2	S3	S4	S5
1	Ik voelde me op mijn gemak tijdens de test.	++	+	++	++	++
2	Ik heb kennis van assenstelsels.	+	0	-	++	+
3	Ik vond het een leuke ervaring om met een tabletop te werken.	++	++	++	++	++
4	Ik zou in de toekomst graag werken met tabletops.	++	++	++	+	++
5	Ik zou een tabletop systeem aanschaffen indien het op de markt komt tegen een betaalbare prijs.	-	++	0	+	+
6	Ik doe beroep op een binnenhuis-architect wanneer ik mijn huis ga inrichten.	+	0	-	-	--
7	Ik had het gevoel dat ik de objecten kon aanraken.	++	+	+	0	+
8	Het 'Interior Designer' concept zou kunnen van pas komen bij het inrichten van uw huis.	+	++	++	++	+
9	Ik kon de objecten makkelijk selecteren.	++	++	++	+	+
10	Het menu is makkelijk te verplaatsen.	++	++	++	+	++
11	Het menu is makkelijk te draaien.	++	++	++	+	++
12	Het menu is makkelijk te bedienen.	+	++	++	+	++
13	Het menu neemt te veel plaats in.	-	-	+	-	--
14	Ik begrijp hoe de XZ-rotatie werkt.	++	++	++	++	++
15	Ik begrijp hoe de Shallow-Depth rotatie werkt.	+	++	-	++	++
16	XZ-rotatie reageert zoals ik wil.	++	++	+	+	+
17	Shallow-Depth rotatie reageert zoals ik wil.	+	+	--	+	+
18	XZ-rotatie laat me toe precies te werken.	++	++	++	++	0
19	Shallow-Depth rotatie laat me toe precies te werken.	+	0	--	+	+
20	Ik verkies XZ-rotatie boven Shallow-Depth rotatie.	++	++	++	-	0
21	Ik begrijp hoe het verplaatsen in de hoogte werkt.	++	++	++	++	++
22	Het gebruik van kleuren is handig om in te schatten op welke hoogte ik zit te werken.	++	++	--	+	++
23	Ik vind het makkelijker om een meubel rechtstreeks op een ander meubel te plaatsen.	+	0	--	+	+
24	Ik kan de camera's makkelijk besturen	++	++	++	++	++
25	Ik begrijp hoe ik een traject met de camera kan opnemen.	++	++	++	++	++
26	Ik begrijp hoe ik de camera rond een punt kan laten bewegen.	+	++	++	++	++
27	Ik vind het eenvoudig om de viewport (beeld) van grootte te wijzigen.	++	+	+	++	++
28	Ik kon makkelijk een meubel werpen	+	-	0	+	++