



Fakulteten för ekonomi, kommunikation och IT
Avdelningen för Datavetenskap

Mathias Andersson och Henrik Bäck

Flerpunktspeksskärm med användargränssnitt för multipla operationer

Multi-touch screen with interface
for multiple operations

Examensarbete 30 poäng
Civilingenjörsprogrammet IT

Datum/Termin: XX-XX-XX
Handledare: Katarina Asplund
Examinator: Donald. F. Ross
Ev. löpnummer: X-XX XX XX

Denna uppsats är skriven som en del av det arbete som krävs för att erhålla en civilingenjörsexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Mathias Andersson

Henrik Bäck

Godkänd, Date of defense

Opponent: NN

Handledare: Katarina Asplund

Examinator: Donald F. Ross

Sammanfattning

Svenskt abstrakt här

Multi-touch screen with interface for multiple operations

Put the text of your english abstract here

Tack

Ett stort tack till Magnus Andersson för tillverkningen av det chassi som all hårdvara monterats i.

Ett stort tack till Katarina Asplund vid Karlstads universitet för all den tid hon har lagt ned på detta arbete.

Tack till Peter Rönnlund vid Karlstads universitet som bjöd på fika i en trött stund.

Innehåll

1	Inledning	1
1.1	Problem	1
1.2	Motivering	1
1.3	Mål	2
1.4	Genomförande	2
1.5	Primära resultat	2
1.6	Denna uppsats disposition	2
1.7	Sammanfattning	3
2	Bakgrund	5
2.1	Introduktion till flerpunktspeksskärmar	5
2.1.1	Simultan interaktion	5
2.1.2	Mjukvaruanpassning	6
2.1.3	Tillämpningar	6
2.2	Historia	7
2.2.1	En-punktsinmatning	7
2.2.2	Fler-punktsinmatning	9
2.3	Peksskärmar och flerpunktspeksskärmar idag	12
2.4	Sammanfattning	13

3	Potentiella lösningar	15
3.1	Hårdvara	15
3.1.1	Bildåtergivning	15
3.1.2	Pekpunktsregistrering	16
3.1.3	Chassi	21
3.2	Mjukvara	21
3.2.1	Pekpunktsinformation	21
3.2.2	Klient	22
3.2.3	Spårare och klientkommunikation	24
3.3	Begränsningar och krav	25
3.3.1	Hårdvara	25
3.3.2	Mjukvara	25
4	Hårdvaruprototyp	27
4.1	Pekpunktsregistrering	27
4.1.1	Val av lösning	27
4.1.2	Kamera	27
4.1.3	Beräkningar	30
4.1.4	Bildinhämtning	34
4.2	Pekning	34
4.2.1	Val av lösning	34
4.2.2	Problem	35
4.2.3	Infraröd ljuspenna	36
4.3	Bildåtergivning	37
4.3.1	Val av lösning	37
4.4	Chassi	40
4.4.1	Mått	40
4.4.2	Intern uppbyggnad	41

4.4.3	Anslutningar	42
4.4.4	Start och avstängning	43
4.5	Sammanfattning	43
5	Mjukvaruprototyp	45
5.1	Spårare	45
5.1.1	Funktionalitet	45
5.2	Klient	46
5.2.1	Val av programspråk och plattform	46
5.2.2	Programdesign	47
5.2.3	Användargränssnitt	51
5.2.4	Gester	51
5.3	Spårare och klientkommunikation	51
5.3.1	Om TUIO	51
5.3.2	TUIO i mjukvaruprototyp	52
5.4	ÖVRIGT	54
6	Utvärdering av prototyp	55
7	Slutsats	57
7.1	Översikt	57
7.2	Detta projekt	57
7.3	Framtida vidareutveckling	57
7.3.1	Hårdvara	57
7.3.2	Mjukvara	57
	Litteraturförteckning	59
A	Figurreferenser	63

B	Ritningar för chassi	65
C	Programkod för klient	67

Figurer

2.1	Historisk tidslinje för inmatningssystem	7
2.2	The Rand Tablet	8
2.3	PLATO 1978	10
2.4	reactTable*	12
2.5	iPhone 3G	13
2.6	Microsoft Surface	14
3.1	Utspridd upplysning	18
3.2	Frustrerad total intern reflektion	19
3.3	Laserplansupplysning	20
3.4	Lysdiodspenna	20
4.1	Hårdvaran	28
4.2	XBOX Live Vision	29
4.3	Kamerans synfält	31
4.4	Kamerans synfält utökas genom en spegel	33
4.5	Utspridd belysning uppmonterad	35
4.6	Pekpennor	37
4.7	Plattskärmens lager	38
4.8	Plattskärmens bildpanel	38
4.9	Lager för att bland annat ge vit färg	39

4.10	Plattskärm monterad i chassi. På bilden syns undersidan av skärmen.	40
4.11	Hårdvarans uppbyggnad internt	41
4.12	Kameran är monterad ovan plattskärmens kretskort	42
4.13	T.v intag till chassit och t.h. intern koppling	43
5.1	Quartz uppbyggnad	48
5.2	TUIO, Meddelandetyper	52
5.3	TUIO, Profiler	53
6.1	Bildåtergivning ger godkänt resultat	55

Tabeller

3.1	Olika spårare som implementerar TUIO	22
3.2	Meddelandetyper, internt i klient	23
4.1	Kameraspecifikationer	28
4.2	Mätvärden för kamerans synfält	31
4.3	Komponenter för konstruktion av infraröda ljuspennor	36
4.4	Plattskärmsspecifikationer	40
5.1	TUIO, Meddelandeargument	53

Kapitel 1

Inledning

Den här uppsatsen kommer att visa hur man kan lösa problemet med att flera användare simultant kan interagera med ett system via en och samma in-dataenhet. Som in-dataenhet kommer en flerpunktspeksskärm¹ att användas. Flerpunktspeksskärmen kommer att konstrueras från standardkomponenter. En applikation kommer att utvecklas för att demonstrera delar av den funktionalitet som flerpunktspeksskärmen tillhandahåller.

I det här kapitlet ...

1.1 Problem

Klassisk mus-interaktion ger enbart en pekpunkt åt gången. Pekaren kan flyttas mellan olika objekt och klicka på dessa.

I vissa situationer är det önskvärt att kunna ha flera simultana punkter att arbeta med.

1.2 Motivering

Om ett system kan detektera flera simultana pekpunkter öppnas nya möjligheter för vilka operationer som kan utföras på enheten [25]. Dessutom öppnar det upp

¹Engelska: Multi-Touch Screen, ej officiell översättning men korrekt enligt Svenska Datatermsgruppen [4].

möjlighet för flera personer att samtidigt, beroende eller oberoende av varandra, använda systemet.

1.3 Mål

Målet med uppsatsen är att lösa problemet med att skapa ett komplett system för hantering av flera simultana pekpunkter.

Målet består av två delar, att skapa en hårdvaruenhet samt utveckla en mjukvaruenhet. Hårdvaruenheten skall kunna återge en skärmavbild i realtid samt kunna detektera flera pekpunkter simultant. Mjukvaran måste kunna ta hand om flera simultana pekpunkter och utföra operationer beroende på olika pekpunkters rörelse individuellt eller i förhållande till varandra.

1.4 Genomförande

ÖVERSIKTSBILD ÖVER SYSTEMET

DATA IN - BERARBETNING - DATA UT

1.5 Primära resultat

1.6 Denna uppsats disposition

Den här uppsatsen består av sju kapitel. Det första kapitlet är enbart för introduktion av ämnet och av denna uppsats.

I kapitel 2 introduceras begreppet flerpunktspeksskärmar och en kort historik kring datainmatning av ges. Kapitel 3 ger en överblick av vilka delar som måste utredas samt vilka olika lösningar som är potentiellt möjliga. Några av de lösningar som finns här utreds vidare och implementeras i kapitel 4 och 5.

Prototypkonstruktionen har delats upp i två kapitel för att göra arbetet med överskådligt. Kapitel 4 handlar enbart om den hårdvara som konstruerats och hur denna fungerar. Samma sak gäller för mjukvaran, som finns i kapitel 5.

I kapitel 6 presenteras resultat från två olika utvärderingar. En utvärdering är efter att grundläggande funktionalitet har implementerats enligt kapitel 3. Den andra utvärderingen behandlar utökad funktionalitet. Slutsatsen för uppsatsen återfinns i kapitel 7.

Utöver detta finns ett antal bilagor innehållandes ritningar och programkod.

1.7 Sammanfattning

Kapitel 2

Bakgrund

Flerpunktspekaskärmar är ingen ny teknik men det är först nu som de har kommit till användning i olika applikationer [31]. Mobiltelefoner [10], datorer med mera utrustas idag med flerpunktspekaskärmar av olika slag och användningsområdet varierar. I det här kapitlet kommer en introduktion till begreppet flerpunktspekaskärmar att ges. Utöver detta kommer en kort historik om olika inmatningsmetoder, relaterade till pekaskärmar eller flerpunktspekaskärmar, att ges.

2.1 Introduktion till flerpunktspekaskärmar

Till skillnad från pekaskärmar har flerpunktspekaskärmar, som tidigare nämnt, möjlighet att ta emot flera simultana pekpunkter. Det är även så att pekaskärmar har egenskapen att de operationer en användare utför på ett objekt, med hjälp av pekaskärmen, är direkt manipulation av objektet. För en ovan användare blir det då lättare för dennes hjärna att koordinera händerna och synintrycket vid användande av pekaskärmar än vid användande av tangentbord och mus. [30] [31]

2.1.1 Simultan interaktion

Möjligheten att kunna använda två händer när ett problem utforskas är naturligt. Det går därför fortare att utföra en viss åtgärd med två händer än med en [25].

Två personer samtidigt vill manipulera objekt i ett system

Ej begränsad till att en person gör operationerna

2.1.2 Mjukvaruanpassning

Här måste mjukvaran utvecklas eftersom hårdvaran finns – OS-saker är fortfarande för små.... saffer skr

2.1.3 Tillämpningar

Flerpunktspeksskärmar kan ha många olika användningsområden.

De lämpar sig exempelvis bra som interaktiva whiteboards då de har möjlighet att ta emot inmatningar från flera samtidiga användare. Detta gör att flera personer samtidigt kan använda tavlan. Till skillnad från en vanlig whiteboard får man dessutom möjlighet att kunna lagra den information som matats in och på så sätt göra informationen på tavlan bestående.

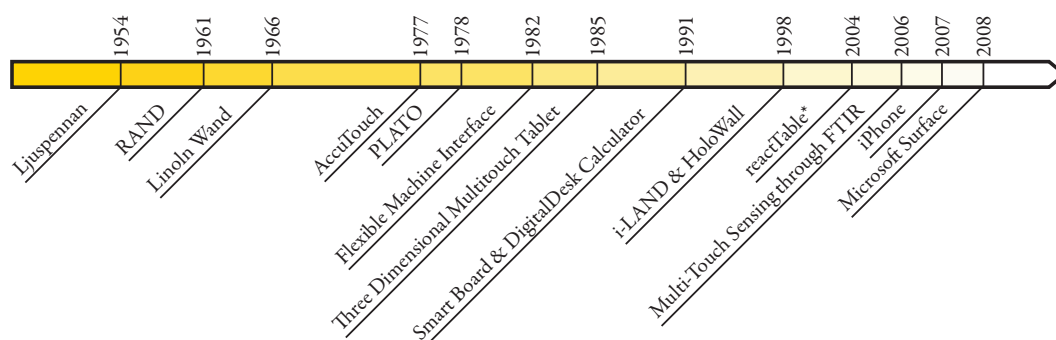
Från whiteboard är steget inte alls långt till anslagstavlor. Det skulle kunna gå att tillverka stora anslagstavlor där flera personer samtidigt kan läsa och sätta upp digitala anslag. Anslagstavlorna skulle kunna användas på företag, skolor eller i köpcentrum.

I presentationssammanhang finns flera andra tänkbara användningsområden. I nyhetssändningar analyseras olika scenarion i direktända program. Exempelvis skulle en väderkarta inte behöva vara animerad i förväg för att visa närmare vy-er av områden. Presentatören skulle i stället kunna navigera i kartan med bara händerna.

Tekniken lämpar sig också inom design. Det finns en fördel med att kunna arbeta med båda händerna när man gör kurvor. Det mer naturligt att arbeta med två händer och därmed går det också både enklare och fortare att utföra uppgifter [25]. Exempel på en teknik som passar in här är "TapeDrawing" och man slipper då problemet med att omsätta analoga ritningar digitalt senare.

2.2 Historia

Alternativa inmatningsmetoder till tangentbordet har funnits i över 50 år. I detta avsnitt presenteras historiska hållpunkter inom områden som kan relateras till pekskärmar och flerpunktspeksskärmar samt lite information kring dem. Figur 2.1 visar hållpunkter i historien.



Figur 2.1: Historisk tidslinje för inmatningssystem

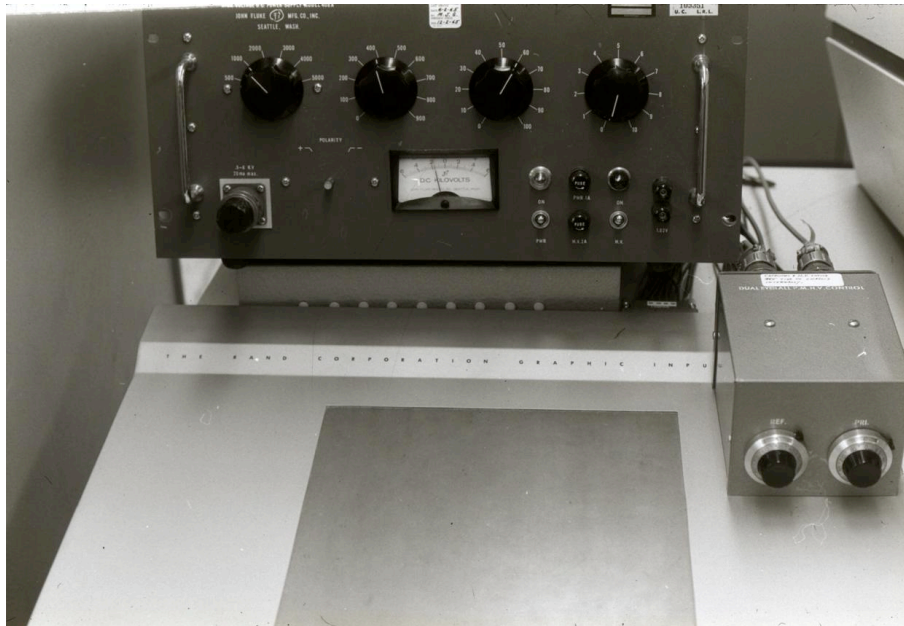
2.2.1 En-punktsinmatning

I detta avsnitt kommer inmatningsmetoder relaterade till pekskärmar att presenteras.

1954 introducerades ljuspennan som inmatningsmetod [23]. Med ljuspennan arbetade användaren direkt på skärmen med en för ändamålet utvecklad penna. Pennan hade en inbyggd ljussensor som kände av när CRT-skärmens katodstråle träffade densamma. Katodstrålens position är känd och utifrån denna information kan man härleda pennans position i två dimensioner.

THE RAND TABLET, se figur 2.2, lanserades 1961 [3] och hade en 10 gånger 10 tum stor yta vilken kunde läsa diskreta punkter i planet. Ytan bestod av en elektriskt laddad platta. På ytan användes en speciell penna (sond) för att göra inmatningar. Pennan hade en tryckkänslig spets för att simulera en verklig penna

och på så sätt mata in data i tre dimensioner. Då inmatning och utmatning var fysiskt separerade kan THE RAND TABLET främst liknas vid dagens ritbord¹[5].



Figur 2.2: The Rand Tablet

1966, fem år efter RAND, utvecklades en in-dataenhet kallad THE LINCOLN WAND [18], där användaren kunde peka direkt på skärmen. Fortfarande används en speciell penna för att göra kopplingen mellan pekpunkten på skärmen och datorn. Pennan var utrustad med en ultraljudsmottagare och dess position gick att utläsa i tre dimensioner, till skillnad från ljuspennans två dimensioner. Tekniken fungerar, i kort, så att skärmen har fyra stycken inbyggda ultraljudssändare som periodiskt sänder ut en ultraljudssignal. När användarens penna tar emot ljuden från sändarna kan tiden beräknas för varje sändare. På så sätt kunde positionen för pennan beräknas.

Elva år senare, 1977, tillverkades en transparent yta kallad AccuTouchTM[14]

¹Engelska: Tablet

som kunde avläsa en punkt baserat på fingrets placering på skärmen. Systemet består av två lager av elektriskt ledande material. När ett finger trycker på ytan sluts en elektrisk krets mellan de två lagren. Fingrets position beräknas genom att mäta potentialen, horisontellt och vertikalt, i punkten [7].

Året efter AccuTouch, 1978, användes en pekskärm i en utbildningsprodukt vid namn PLATO [8], se figur 2.3. PLATO hade en plasmaskärm om 512 gånger 512 bilpunkter som dessutom var utrustad med möjligheten att ta emot data genom att placera ett finger på skärmen. En användare kunde på så sätt göra val i olika applikationer utan att använda tangentbordet. Pekskrmen hade en upplösning på 16 gånger 16 punkter. Detta innebar att det var möjligt att känna av 256 olika områden på skärmen. Runt skärmen placerades infraröda ljuskällor och mottagare. Dessa skapade ett rutnät av ljusstrålar över skärmen. När ett finger sattes på skärmen bröts två ljusstrålar och detta gav fingrets position som vidarebefordrades till applikationen [1].

Från 1980-talet och framåt började pekskrmar leta sig in i allt fler kommersiella produkter [29]. Hewlett-Packard lanserade kommersiella arbetsstationer för personligt bruk med inbyggd pekskrm. Dessa byggde på samma teknik som den i PLATO.

Under nittioalet, närmare bestämt 1991, presenterades Smart BOARD™. Smart BOARD, är en interaktiv whiteborard som är menad att användas i utbildningssammanhang.

2.2.2 Fler-punktsinmatning

Det här avsnittet kommer presentera inmatningsmetoder relaterade till flerpunktspekskrmar.

Redan 1982 togs en prototyp fram för att kunna känna av flera simultana inmatningspunkter från en eller flera användare. Prototypen fick namnet The



Figur 2.3: PLATO 1978

Flexible Machine Interface [29] och fungerade genom bildinläsning. En yta läses av med en kamera och fingrarna uppträder som svarta punkter som kameran kan avläsa.

Tre år senare, 1985, presenterades en yta med möjlighet att ta emot flera samtidigt punkter och som dessutom talar om graden av kontakt [19]. Ytan, som var indelad i 64 gånger 32 punkter, tillät att användaren arbetade direkt med fingrarna på ytan. Avläsningen skedde genom att ytan periodiskt skannades av med hjälp av en algoritm där punkter med lägre elektrisk laddning upptäcks. Ytan rapporterade plankoordinater samt hur hårt användaren tryckte på punkten. Detta gav en tredimensionell inmatning. Ytan kunde dock inte visa någon bild och bildskärmen var därför tvungen att placeras separat vilket gör att denna yta inte riktigt

kan klassas som en äkta pekskärm.

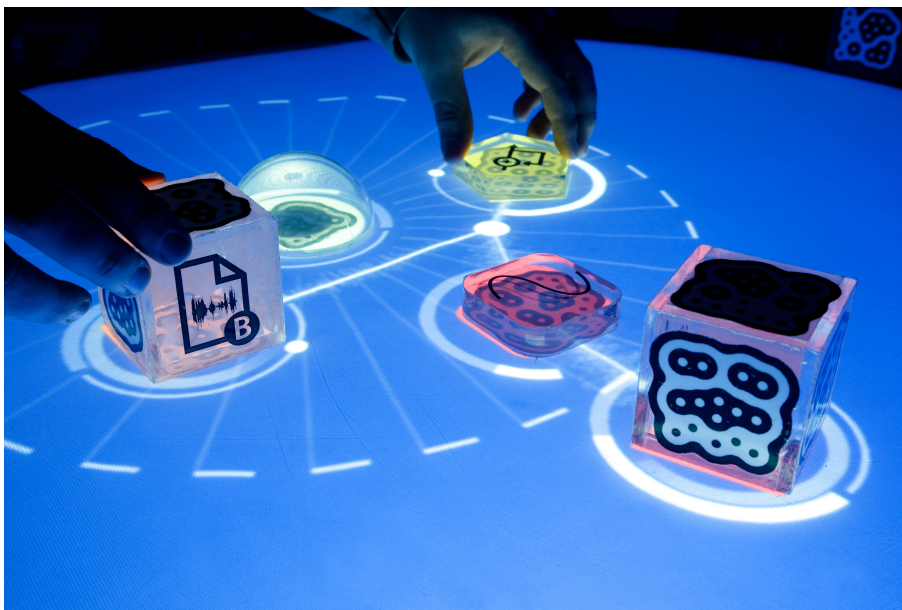
The DigitalDesk Calculator [34] presenterades 1991. Enheten projicerade ett gränssnitt på toppen av ett bord. Bredvid projektorn fanns även en kamera som registrerade användarens rörelser över bordet och dessutom objekt på bordet. Systemet kunde då ta emot flera samtidiga inmatningar och känna igen dokument placerade på bordet.

Sju år senare, 1998, presenterades projektet i-LAND [32]. i-LAND var ett projekt där två enheter togs fram. DynaWall och InteracTable. DynaWall var en stor interaktiv whiteboard som hade möjligheten att ta emot information från flera samtidiga användare och InteracTable var ett bord runt vilket upp till sex personer kunde stå. Bordet kunde ta emot in-data från flera samtidiga användare.

Samma år presenterades även HoloWall [21]. HoloWall var en vägg designad för att kunna ta emot fler samtidiga inmatningar men även känna igen föremål som hölls upp framför ytan. HoloWall fungerar genom att en yta projekteras bakifrån. Vid projektorn finns en kamera som filmar ytan och det infraröda ljus som belyser upp ytan. Kameran kan registrera det infraröda ljuset som reflekteras och på så sätt räkna ut pekpunkter.

Under tjugohundratalet, närmare bestämt 2004, presenterades en fungerande prototyp av reactTable*[17], figur 2.4. reactTable* är ett interaktivt musikinstrument som har stöd för att kunna känna igen flera fysiska objekt placerade på ytan samtidigt. Objekt identifieras optiskt med hjälp av en kamera och märkningar på varje objekt. Utifrån objekten, dess riktning och deras inbördes förhållande genereras musik [15].

De tre åren 2006, 2007 och 2008 innebar stora språng på marknaden för flerpunktspeksskärmar. 2006 presenterade Jefferson Y. Han en flerpekspunktsyta som kunde konstrueras både enkelt och icke kostnadskrävande [9]. Han presenterade sina idéer på konferensen TED (Technology, Entertainment, Design)[33] vilket



Figur 2.4: reactTable*

fick världen att öppna ögonen för vad som kan åstadkommas.

Apple iPhone™ [10], se figur 2.5, lanserades i början av 2007 och var en av de första telefonerna med en inbyggd flerpunktspeksskärn.

2008 levererades Microsoft® Surface™ [2], se figur 2.6, för första gången. Microsoft Surface består av att en bild projekteras bakifrån på en yta. Ytan belyses även med infrarött ljus från projektorns håll. Kameror på undersidan registrerar ytan och kan känna igen flera samtidigt användare och dessutom objekt.

2.3 Pekskämar och flerpunktspeksskärmar idag

Marknaden för flerpunktspeksskärmar börjar idag ta fart. Microsoft® Surface™ finns idag i ett antal AT&T-butiker [2] och Apple säljer iPhone framgångsrikt [11].

Utöver dessa produkter är interaktiva whiteboards populärt i utbildningssammanhang. Här är det dock främst traditionella peksskärmar som är populära och



Figur 2.5: iPhone 3G

inte flerpunktspeksskärmar.

När det gäller flerpunktspeksskärmar levererar ett företag vid namn N-trig ytor för detta. N-trig levererar skärmar till bland annat Dell och Intel [24] som har implementerat dem i sina datorer.

Jämförade av hårdvaran vi har med ex. MS Surface

2.4 Sammanfattning

Flerpunktspeksskärmar ger flera fördelar före peksskärmar. Bland annat låter de användaren använda flera händer och fingrar när denne utforskar ett problem och dessutom låter de flera personer samtidigt interagera med ett system. Flerpunktspeksskärmar passar inte i alla situationer. Dock kan de med fördel användas i flera specifika tillämpningar.



Figur 2.6: Microsoft Surface

Pekskärmar är ingen nyhet på marknaden och så inte heller flerpunktspek-skärmar. Det har funnits varianter av pekskärmar sedan 1954 och redan från början har de använts för att mata in data till olika applikationer. På senare år har pekskärmarna utvecklats så att de kan ta emot fler än en in-datapunkt.

Kapitel 3

Potentiella lösningar

I det här kapitlet kommer olika lösningsförslag ges till problemet med att skapa hård- och mjukvara för flerpunktspekskärmar. Först tas aspekter angående hur hårdvaran kan konstrueras upp, därefter tas metoder för både bildåtergivning och pekpunktsregistrering upp. Senare i kapitlet diskuteras olika lösningsförslag för mjukvaran och hur denna kan bli expanderbar.

3.1 Hårdvara

3.1.1 Bildåtergivning

För att flerpunktspekskärmen skall fungera måste ut-data, i form av grafik, presenteras på den yta som registrerar pekpunkter. Det finns flera sätt att lösa detta på och det går kombinera olika sorters bildåtergivningsmetoder med pekpunktsregistrering.

3.1.1.1 Digitalprojektor

Ett bildåtergivningssätt är att använda sig av en digitalprojektor¹. En digitalprojektor monteras så att den projicerar en bild på en duk eller ett plexiglas². Digitalprojektorn kan monteras i sitt ursprungliga skick men kan behöva kompletteras

¹Engelska: Video projector

²Polymetylmetakrylat

med speglar eftersom avståndet för projicering kan bli kort. [9] Vissa digitalprojektorer kan också behöva kompletteras med ett filter för infrarött ljus vilket tar bort oönskat infrarött ljus från projektorns lampa [21].

3.1.1.2 Plattskärm

Ett annat alternativ, som i mindre skala, är billigare än projektorn är att använda sig av en plattskärm³. Plattskärmen kan behöva förberedas för att kunna användas tillsammans med olika typer av pekpunktsregistrering.

Om optisk pekpunktsregistrering, se avsnitt 3.1.2.2, skall användas måste skärmen kunna släppa genom ljus. Plattskärmen måste monteras isär så att bakstycket kan tas bort och så att kretskorten som sitter monterade bakom skärmen kan flyttas, se avsnitt 3.1.2.2 [22].

3.1.2 Pekpunktsregistrering

För att flerpunktspeksskärmen skall kunna användas måste samtliga pekpunkter som befinner sig på skärmen registreras av systemet. Det finns ett flertal olika sätt att genomföra denna registrering på.

3.1.2.1 Kapacitiv avläsning

Genom historien har flera olika produkter använt sig av kapacitiv avläsning av pekpunkter [19] [14]. I detta fall måste den kapacitiva ytan vara transparent för att bildåtergivning skall vara möjlig. Inköp eller utvecklande av en sådan metod kan rimligtvis inte ligga innanför både budget och tidsrymd för detta projekt varpå detta alternativ inte kommer att behandlas vidare.

³LCD, med eller utan TFT

3.1.2.2 Optisk avläsning

Ett annat sätt att läsa av aktiva pekpunkter på skärmen är genom optisk avläsning.

En kamera monteras bakom ytan som skall registreras, eller på annat plats med hjälp av speglar, så att den kan registrera hela bildytans baksida. Avläsning av ytan sker bakifrån eftersom att användarens händer och armar inte är i vägen för avläsningen på detta sätt. Det enda som syns på kameran kommer därför att vara de punkter som är nära skärmens yta eller i kontakt med ytan.

För att kameran skall kunna se pekpunkterna måste dessa belysas. För att inte störa skärmens bild bör ett, för ögat, osynligt ljus användas för att skapa pekpunkter. Detta gör att kameran enbart skall se detta, för ögat, osynliga ljus. [21] [28] Exempel på osynligt ljus är infrarött ljus, vilket återfinns i exempelvis fjärrkontroller till hemelektronik.

Hur upplysningen av pekpunkterna sker kan varieras.

Utspridd upplysning

Ett sätt att lysa upp punkterna är med en teknik som kallas utspridd upplysning⁴. Ytan som skall registreras belyses underifrån med infrarött ljus [21]. När ett finger placeras på ytan kommer detta att reflektera ljuset nedåt så att detta kan registreras av kameran, se figur 3.1.

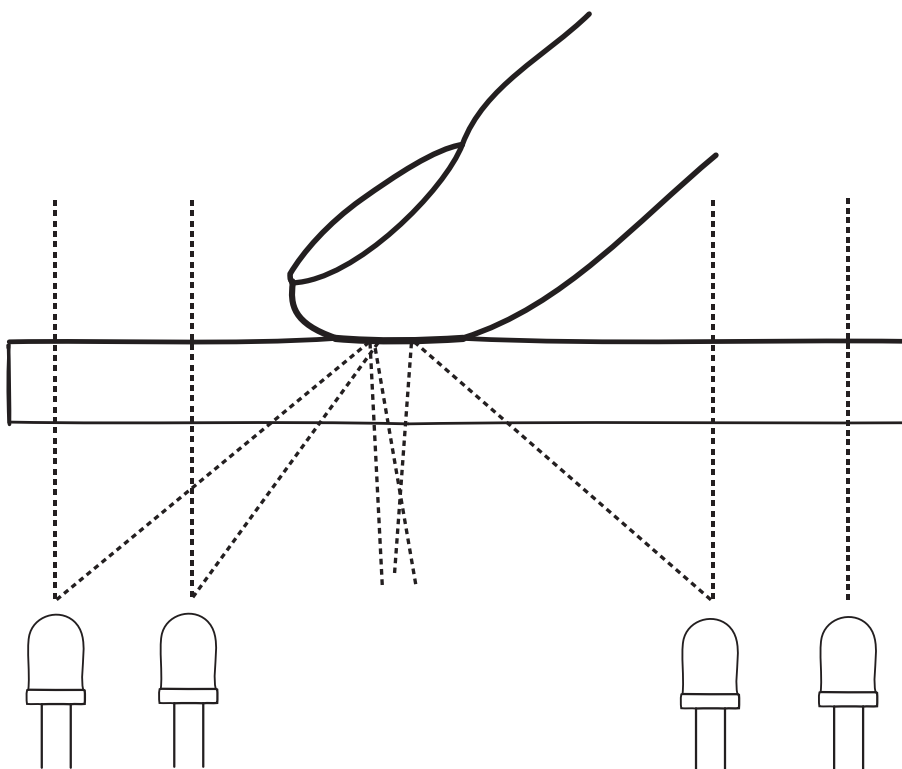
Frustrerad total intern reflektion

Ett annat sätt att belysa pekpunkter är via så kallas frustrerad total intern reflektion⁵ (FTIR) [9].

Tekniken går ut på att man sänder in infrarött ljus i en plexiglasskiva. Ljuset sänds in med en vinkel lika med eller större än den så kallade *kritiska vinkeln* för

⁴Engelska: Diffused Illumination

⁵Engelska: Frustrated Total Internal Reflection



Figur 3.1: Utspridd upplysning

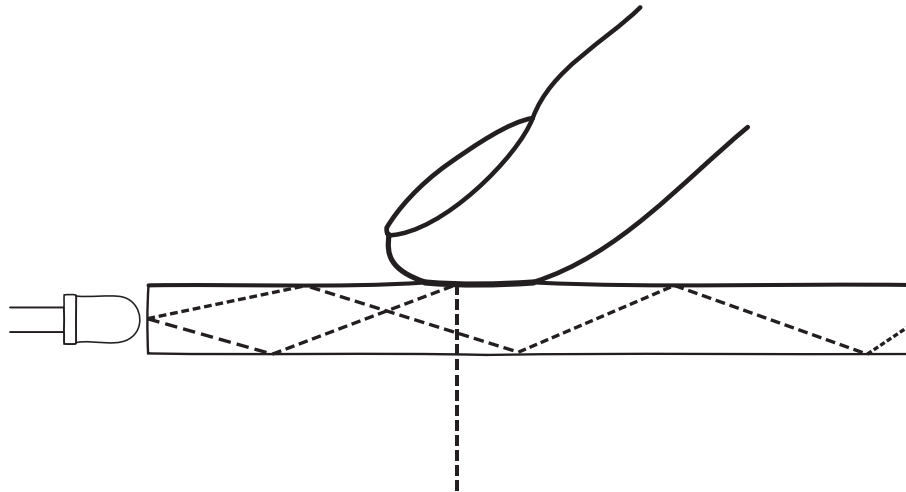
total reflektion. När ljuset försöker bryta sig ut från plexiglasskivan kommer det att träffa på luft som har ett lägre optiskt brytningsindex än plexiglas. Detta gör att allt ljus kommer att reflekteras och stanna kvar inuti plexiglasskivan.

Den *kritiska vinkeln*, θ_c , för när total reflektion uppstår kan härledas med hjälp av Snells lag [26], se ekvation 3.1. Total reflektion uppstår endast när refraktionsvinkeln är 90 grader. I ekvationen är n_1 brytningsindex för plexiglasskivan och n_2 brytningsindex för luft.

$$n_2 \sin \theta_c = n_1 \sin 90^\circ \Leftrightarrow \theta_c = \arcsin \left(\frac{n_1}{n_2} \right) \quad (3.1)$$

När ett finger sätts mot plexiglasskivans yta medför det en förändring av den kritiska vinkeln vid det aktuella området. Detta innebär att ljuset kommer att

lämna plexiglas et och reflekteras mot fingret, se figur 3.2



Figur 3.2: Frustrerad total intern reflektion

Laserplansupplysning

Laserplansupplysning⁶, se figur 3.3, fungerar genom att lyser med

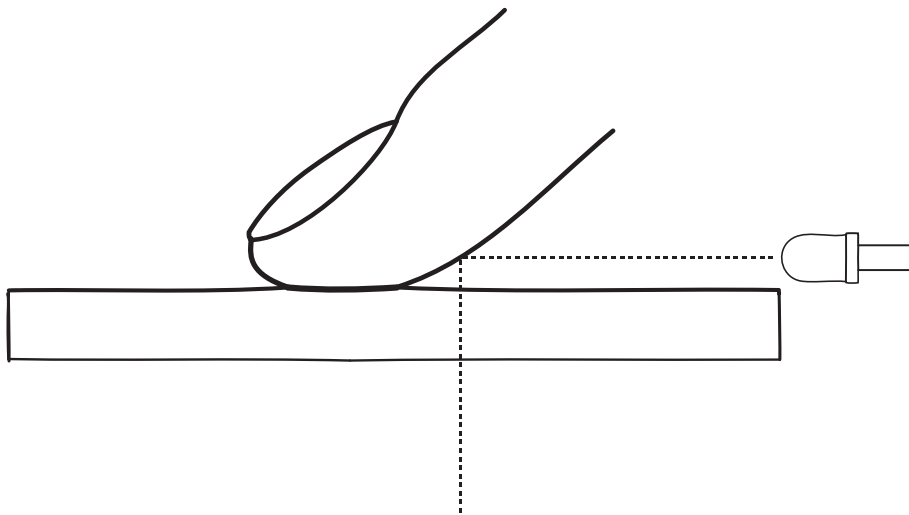
Källa saknas, ingen text ännu

Penna

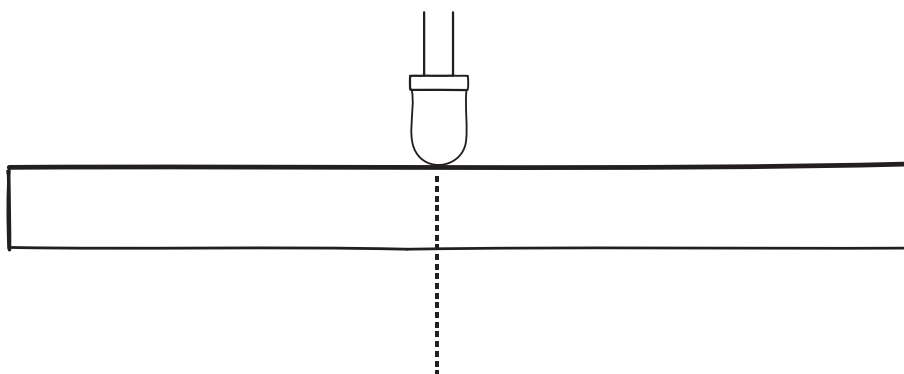
Eftersom pekpunkter registreras med hjälp av en kamera är det möjligt att belysa kameran direkt med en infraröd ljuspunkt, istället för indirekt via reflektion. Detta är en säker metod eftersom att ljuset kommer ha tillräcklig kraft för att träffa kameran. Metoden går också att kombinera med de övriga metoderna för optisk avläsning utan att göra modifikationer i hårdvaran.

Genom att bygga in en infraröd lysdiod i en penna kan pennan användas för att peka på skärmen. Ljuset från pennans infraröda lysdiod kommer att uppfattas som en pekpunkt av kameran.

⁶Engelska: Laser Light Plane Illumination



Figur 3.3: Laserplansupplysning



Figur 3.4: Lysdiodspenna

3.1.2.3 Sensorbaserad avläsning

Den sensorbaserade metoden går ut på att plattskärmen har inbyggda sensorer för att känna av ljus. När ett finger placeras på skärmen kommer registreras detta av TFT-fotoceller inbyggda skärmens pixlar [6].

Denna metod kräver antingen att en hårdvara med tekniken köps in eller kan konstrueras. Detta alternativ kommer, på grund av dess komplexitet, inte att vidare behandlas i denna uppsats.

3.1.3 Chassi

Design, ritningar, med mera.

3.2 Mjukvara

Mjukvaran är otroligt viktig när det gäller pekpunktsbaserad inmatning. Mjukvarudelen består av ett grafiskt användargränssnitt, spåraren⁷ samt logik som tolkar uppgifter från spåraren och exempelvis gör ändringar i användargränssnittet.

3.2.1 Pekpunktsinformation

Beroende på vilken typ av pekpunktsavläsning som används i hårdvaran erhålls olika soters information.

3.2.1.1 Optisk avsläsning

När man använder en optisk pekpunktsavläsning är den data man erhåller en sekvens av stillbilder. För tillämpningen är de erhållna bilderna oanvändbara i sitt ursprungliga format. Användbar information utläses ur bilden genom att behandla och tolka bilden, detta är spårarens uppgift. Spårarens ska åtminstone lämna i från sig information om koordinater för pekpunkter.

Olika spårare

Det finns flera spårare som kan användas. Spårarna implementerar olika sätt att lämna i från sig data, gemensamt för de som studerats är att de implementerar TUIO, se avsnitt 3.2.3.2. I tabell 3.1 finns exempel på ett antal spårare som implementerar TUIO.

⁷Engelska: Tracker

<i>Spårare</i>
BBTouch
Touché
TouchLib
Reactivision

Tabell 3.1: Olika spårare som implementerar TUIO

3.2.1.2 Övriga avläsningstyper

Det finns, som nämnt i avsnitt 3.1.2, fler sätt än optisk pekpuntsregistrering. Alla dessa sätt rapporterar data i olika format vilket gör att det inte finns något generellt angreppssätt. Gemensamt är dock att vi fortfarande enbart är intresserad av koordinater av pekpunkter samt eventuellt om det är nya eller gamla punkter.

3.2.2 Klient

Klientens uppgift är att ta emot pekpunktsinformation och tillämpa informationen. Klienten ritar upp gränssnittet på skärmen och innehåller logik för att reagera på pekpunktsinformationen.

3.2.2.1 Grafisk uppbyggnad

Klientapplikationen kan ha ett antal lager som ritas på skärmen. Varje lager innehåller information om vad som skall ritas, dess position samt metoder för att tala om för lagret om omgivande händelser.

På detta sätt kan ett lager skapas för varje grafiskt objekt som skall ritas, vilket är bra ur ett expansions- och separeringsperspektiv.

Bild på hur lager kan tänkas ligga på skärmen

3.2.2.2 Datahantering

Klientapplikationen tar emot pekpunkter från hårdvaran och översätter dessa till tre stycken olika meddelande-typer, se tabell 3.2. Meddelandetyperna används oavsett vilket protokoll eller API-som använts mot hårdvaran. Dessa tre meddelandetyper räcker fullgott för att utföra operationer på objekt i planet eftersom de innehåller information om var en punkt skapas, flyttas och tas bort.

<i>Typ</i>	<i>Argument</i>
Aktivera pekpunkt	Pekpunktens identitet, koordinat
Flytta pekpunkt	Pekpunktens identitet, koordinat
Avaktivera en pekpunkt	Pekpunktens identitet

Tabell 3.2: Meddelandetyper, internt i klient

Ett *Aktivera pekpunkt*-meddelande kommer att orsaka en kollisiondetektering mellan pekpunktens koordinat samt de olika lager som finns i systemet. Om en kollision mellan pekpunkten och lagret uppstår kan meddelandet skickas vidare till detta lager.

När det gäller meddelandena *Flytta pekpunkt* samt *Avaktivera pekpunkt* så vet systemet redan vilket lager som är associerat med den aktuella pekpunktsidentiteten. Meddelanden skickas då vidare direkt till detta lager.

Det finns en stor fördel med att låta varje lager ta hand om inkomna meddelanden. I stället för att en generell funktion skall ta emot meddelanden, detektera operationen och sedan tala om för lagret vad det skall utföra kan lagret detektera operationen själv. Detta medför att nya lager-typer, med nya operationer, kan introduceras utan att omgivande programkod måste skrivas om.

Exempel

Låt ett lager ta emot meddelandet *Rotera X grader*. För att detta skall fungera måste klienten först ta emot någon av de tre meddelandetyperna i tabell 3.2. Efter

detta skall en beräkning ta plats för att räkna ut hur många grader som lagret skall roteras och slutligen skall meddelandet *Rotera X grader* skickas till lagret.

Introducera ett nytt lager som inte skall kunna roteras men spela upp ett ljud. Ett sådan lager skulle kunna ta emot meddelandena *Spela* och *Stoppa*. Nu måste en del av klienten skrivas om för att kunna veta vilka lager som svarar på vilka meddelanden samt lägga till kunskap för att veta om ett *Spela*- eller *Stoppa*-meddelande skall skickas.

Om istället alla denna funktionalitet enbart finns i lagret behöver klienten enbart ta reda på vilket lager som skall ha ett visst meddelande och därefter vidarebefordra detsamma. Varje lager kan själv bestämma vad som skall hända när meddelandet tas emot.

3.2.3 Spårare och klientkommunikation

Klienten måste få information om pekpunkter från spåraren. Hur spåraren tillhandahåller klienten denna information kan variera beroende på implementation.

3.2.3.1 API

Ett sätt att låta klienten kommunicera med spåraren är via ett API. APIets utformning varierar mellan olika spårare och måste därför anpassas efter vilken spårare som används. Detta innebär att applikationen måste skrovas om för varje spårare som skall användas.

3.2.3.2 TUIO

Ett annat sätt är att använda sig av ett välspecifierat protokoll. TUIO [16] är ett protokoll framtaget specifikt i syfte att förmedla information om optiskt identifierade objekt. TUIO arbetar över valfritt transportprotokoll, men UDP används

med fördel i ett IP-nätverk. Eftersom TUIO arbetar över nätverket är det enkelt att låta klienten byta mellan olika spårare, som implementerar TUIO, då denna inte behöver programmeras om.

Exempel på spårare som implementerar TUIO finns i tabell 3.1.

3.3 Begränsningar och krav

3.3.1 Hårdvara

Ska kunna... Visa bild -> VGA/DVI/DISPLAYPORT eller liknande ska vara en källa Ta emot p

3.3.2 Mjukvara

Ska kunna... Ta emot data Hantera den Göra lämpliga åtgärder

Kapitel 4

Hårdvaruprototyp

En prototyp, se figur 4.1, har tagits fram för detta projekt enligt specifikationen i avsnitt 1.3. Hårdvaruprototypen kommer att användas för att testa mjukvaruprototypen. I det här kapitlet kommer utförliga beskrivningar att ges över hur hårdvaruprototypen konstruerad. Först kommer chassit att beskrivas, därefter kommer tillvägagångssättet för pekpunksregistrering att förklaras. Sist förklaras metoden för bildåtergivning.

4.1 Pekpunksregistrering

4.1.1 Val av lösning

Hårdvarulösningen har gjorts valts utifrån två aspekter. En stor aspekt är priset för utveckling och det andra är tiden för att utföra jobbet. Båda parametrarna är ytterst begränsade av detta projekt.

Den lösning som valts baseras på optisk pekpunksregistrering med hjälp av en kamera, se avsnitt 3.1.2.2.

4.1.2 Kamera

För att läsa in pekpunkter optiskt används en kamera. En webbkamera har inhandlats för ändamålet och specifikationerna kan ses i tabell 4.1. Man kan med



Figur 4.1: Hårdvaran

fördel använda en kamera av bättre kvalité för att få snabbare överföring av bilder samt bättre kvalité på bilderna. Detta kan ge bättre resultat eftersom ursprungsbilderna då är av en högre kvalité. Kameran, se figur 4.2, i detta projekt är konstruerad för 640×480 bildpunkters¹ upplösning i 30 bilder per sekund.

<i>Tillverkare</i>	<i>Modell</i>	<i>Inköpsställe</i>
Microsoft	Xbox LIVE Vision	GameStop Sweden AB

Tabell 4.1: Kameraspecifikationer

¹Engelska: Pixels



Figur 4.2: XBOX Live Vision

4.1.2.1 Modifiering för infrarött ljus

För pekning kommer infrarött ljus att användas, se avsnitt 4.2.1.1. Detta gör att kameran kan modifieras för att enbart registrera infrarött ljus. Fördelen med detta är att inget annat ljus kommer att störa kameran.

Filter

Ett filter monteras in i kameran för att enbart släppa genom infrarött ljus. Bästa resultat fås genom att använda ett filter som enbart släpper genom ljus av den våglängd som används i ljuskällan, se avsnitt 4.2.1.1. Ett sådant filter är dock relativt dyrt och kan kosta över 100 US-dollar [20].

I stället kan fullt exponerad och framkallat negativ användas [27]. Kameran monteras isär och filtret, i detta fall fullt exponerat och framkallat negativ, placeras framför kamerans bildsensor. Efter testning framkom att *tre (3) lager* av sådant

negativ behövdes för att blockera så mycket synligt ljus som möjligt. Fler lager innebar ingen märkbar skillnad för varken synligt eller infrarött ljus.

4.1.2.2 Fotografering

Bilder som nu tas med kameran kommer enbart att registrera infrarött-ljus eller ljus som ligger mycket nära det infraröda ljuspektrumet. Allt annat ljus bör släckas ut av filtret.

4.1.3 Beräkningar

Kameran måste monteras inuti chassit om den skall kunna registrera pekpunkter underifrån, enligt 3.1.2.2. För att kunna bestämma hur minsta möjliga avstånd mellan kameran och bildskärmens yta måste dess synfält beräknas. Detta kommer också att behövas för att beräkna chassits höjd.

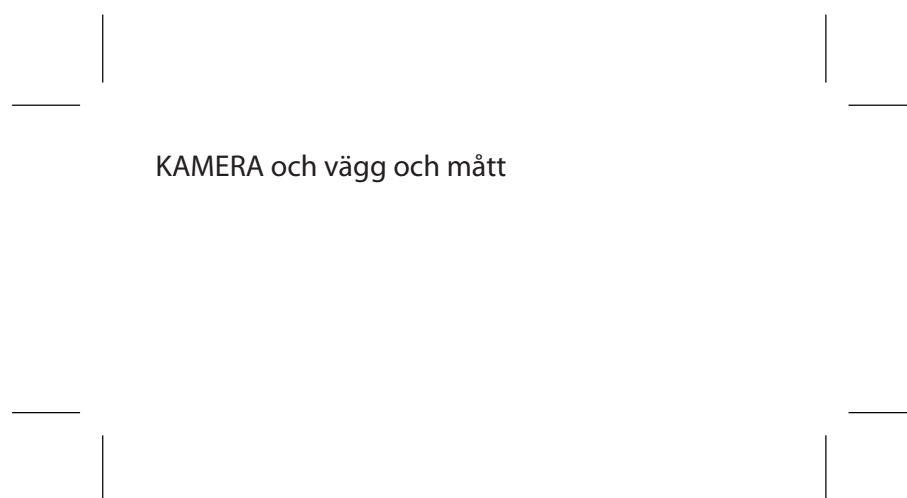
4.1.3.1 Kamerans synfält

Det *exakta* måttet för förhållandet mellan kamerans avstånd och ytan som registreras är inte viktigt, utan ett ungefärligt mått är tillräckligt. I prototypen kan det vara bra att ha lite utrymme för justeringar i efterhand samt möjligheten att korrigera för fel.

Låt A vara avståndet från kamerans lins till ytan som skall registrera. Låt vidare B vara bredden på det område som kan registreras av kameran vid avståndet A . För att kunna ge chassits höjd är det nu intressant att veta förhållandet mellan A och B , se figur 4.3.

Mätvärden

Tabell 4.2 visar olika mätpunkter samt förhållandet mellan A och B , se figur 4.3.



Figur 4.3: Kamerans synfält

A	B	A/B
100	75	$4/3$
90	65	$18/13$
80	60	$4/3$
70	53	$70/53$
60	45	$4/3$
50	40	$4/3$
40	30	$4/3$
30	22	$15/11$

Tabell 4.2: Mätvärden för kamerans synfält

Som synes i tabell 4.2 kretsar alla värden kring $\frac{4}{3}$. Formel 4.1 erhålls från dessa värden.

$$A = \frac{4}{3}B \quad (4.1)$$

4.1.3.2 Chassits höjd

Det är nu intressant att veta hur långt från skärmen som kameran måste sitta för att kunna registrera hela skärmens yta. Eftersom bildskärmen är i bredbildsformat² och kameran registrerar en nästan kvadratisk yta så är bredden på skärmen det intressanta.

Utifrån formel 4.1 kan vi nu härleda kamerans avstånd, A, eftersom B är känd från skärmens specifikationer.

$$\begin{cases} A = \frac{4}{3}B \\ B = 485mm \end{cases} \Rightarrow A = \frac{485 \cdot 4}{3} = \frac{1940}{3} \approx 647mm.$$

Detta ger att chassits höjd måste som allra minst vara 647mm högt. Utöver detta tillkommer kamerans egen höjd samt den höjd som en monteringsanordning upptar.

Minska chassits höjd

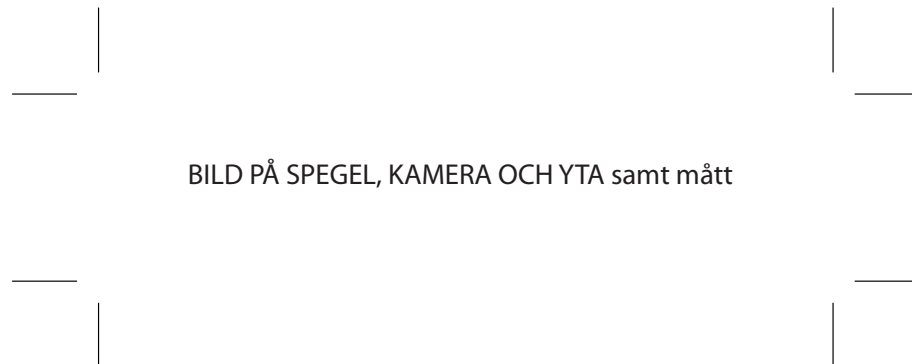
För att reducera chassits höjd kan en spegel användas. Spegeln reflekterar inkommande ljus från pekpunkter och förlänger på så sätt avståndet mellan ytan och kameran.

Teoretiskt kan höjden på chassit halveras med hjälp av en plan spegel. Dock är detta inte praktiskt möjligt eftersom

1. inget får skymma ytan.
2. kameran inte kan monteras exakt i toppen på chassit

Kameran kan dock monteras på sidan av chassit och vinklas för att titta neråt. Om spegeln också vinklas kan en vy över toppen på lådan åstadkommas enligt

²Engelska: Wide screen



Figur 4.4: Kamerans synfält utökas genom en spegel

figur 4.4. Det finns på grund av detta två alternativ, det ena alternativet är att kameran monteras på kortsidan och då var som helst i höjddled. Det andra alternativet är att montera kameran på långsidan, men då måste denna placeras ovan eller under plattskärmens kretskort.

Enligt ritningarna, se avsnitt 3.1.3, kommer en stor del av chassits fasta långsida att upptas av ett kretskort tillhörande plattskärmen. Att montera kameran på kortsidan kräver att en monteringsanordning byggs eftersom kameran måste monteras på högkant och kan då inte stå på foten, se figur över kamera 4.2.

Antag att chassits höjd halveras, detta ger att $\frac{647}{2} \approx 321mm$. För att inte göra konstruktionen mer komplex monteras kameran sittandes på sin fot. Detta innebär att kameran måste placeras på chassits långsida. En bra plats för detta är precis ovan kretskorten i chassit, eftersom kameran här kan få stöd av kretskortets hållare vilket underlättar montering.

Höjden på chassit beslutades till $400mm$ för att ge utrymme för justeringar och dessutom för att ge möjlighet att montera kameran vinklad.

4.1.4 Bildinhämtning

Kameran ansluts till en dator med hjälp av USB. En mjukvara kan nu, via en kompatibel drivrutin, använda sig av kameran för att inhämta bilder.

Spåraren kan här hämta in bilder och sedan berabeta dessa för att utläsa relevant information. Mer om denna procedur presenteras i avsnitt 5.1.

4.2 Pekning

4.2.1 Val av lösning

Då optisk pekpunktsregistrering används finns flera sätt att skapa detekterbara pekpunkter.

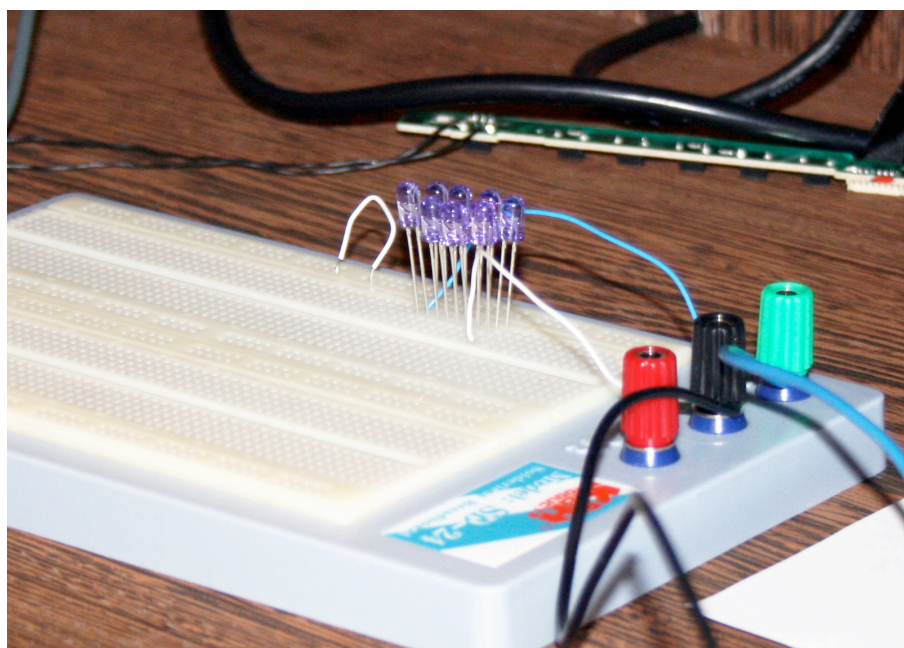
4.2.1.1 Ljuskälla

Enligt avsnitt 3.1.2 är infrarött ljus en lämplig ljuskälla att använda för pekpunktsregistrering. För denna prototyp har infraröda lysdioder använts med en våglängd av $880nm$ [27].

Utspridd upplysning

Bland beskrivna metoder i avsnitt 3.1.2.2 är tekniken med *utspridd upplysning* den teknik som är enklast att genomföra. Detta eftersom ett antal lysdioder enbart behöver monteras upp i ett jämt mönster.

Ett antal infraröda lysdioder monterades upp på en laborationsplatta som sedan placerades inuti chassit. Eftersom undersidan av bilåtergivningen är blank uppstår reflektioner från ljuskällan. En sådan reflektion är svår överröstad av reflektioner från pekpunkter som exempelvis reflektioner av fingrar. För att reducera reflektionen placeras ett tunt skisspapper under den blanka ytan. Reflektio-



Figur 4.5: Utspridd belysning uppmonterad

nerna reduceras därmed en aning.

De infraröda lysdioderna måste placeras ut med ett jämnt avstånd mellan varje lysdiod. Detta för att kunna ge en jämn belysning över hela ytan som skall registreras.

4.2.2 Problem

4.2.2.1 Utspridd upplysning

Ett problem med den utspridda upplysningen var att det inte var möjligt att med enbart fingrar överrösta de reflektioner som kvarstod efter att skisspappret hade monterats. Det gör det svårt att använda denna teknik i den aktuella konstruktionen. För att komma vidare måste en annan teknik att användas.

4.2.3 Infraröd ljuspenna

Som lösning på de problem som uppstått med pekpunktsinmatningen, se avsnitt 4.2.2, har två infraröda ljuspennor har tillverkats för att användas med flerpunktspeksskärmen. Pennorna har en inbyggd infraröd lysdiod, en strömknapp samt ett batteri med motstånd.

Pennorna emitterar själva ljus i stället för att reflektera ljus, detta gör ljuspunkterna mycket starkare från dessa. En fördel med att använda dessa pennor är att de fungerar med alla metoder som beskrivits i avsnitt 3.1.2.2.

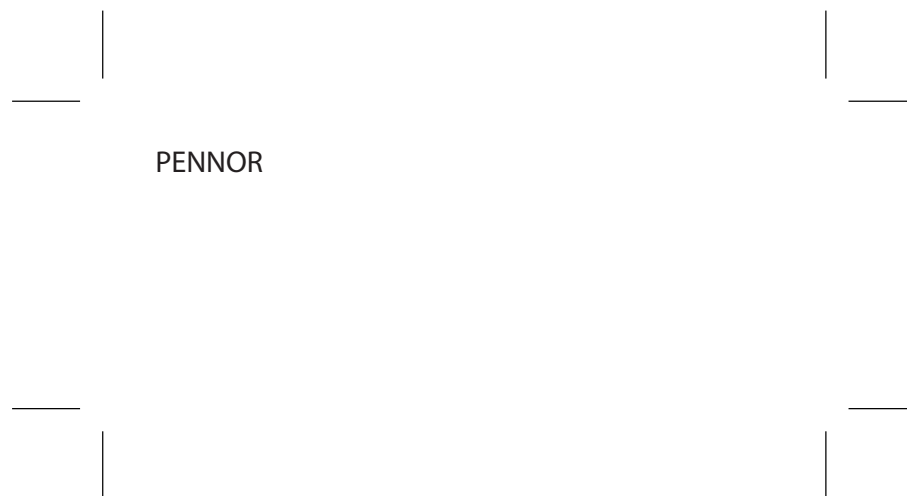
4.2.3.1 Uppbyggnad

Pennorna använder en whiteborard-penna som grundstomme. Inuti pennan finns ett AAA-batteri, en infraröd lysdiod, en brytare, ett motstånd, en plexiglasstav samt en metallbricka. Samtliga komponenter finns i tabell 4.3.

Lysdioden har placerats i pennans spets, se figur 4.6. När lysdioden trycks in sluts brytaren och dioden tänds. Detta medför att pennan kan användas för pekning genom att den enbart sätts ned mot ytan. När pennans spets inte är tryckt mot någon yta är lysdioden släckt vilket medför att pennan inte kan registreras som en pekpunkt.

<i>Komponent</i>	<i>Tillverkare och modell</i>	<i>Inköpsställe</i>
OH-penna	Edding 363	
Infraröd lysdiod	Osram SFH485-2	ELFA AB
Tangentbordsswitch	Okänd	Kjell & Company AB
Batteri 1,5 Volt AAA	Duracell	Clas Ohlson
Motstånd, 48 Ohm	Okänd	ELFA

Tabell 4.3: Komponenter för konstruktion av infraröda ljuspennor



Figur 4.6: Pekpennor

4.3 Bildåtergivning

4.3.1 Val av lösning

För bildåtergivning har en plattskärm valts eftersom det är det billigaste alternativet, i valet mellan projektor och plattskärm.

4.3.1.1 Plattskärm

Plattskärmen har ett antal icke önskvärda egenskaper, som dessutom kan variera från modell till modell. Eftersom skärmen måste kunna släppa genom infrarött ljus måste bakstycket avlägsnas och så även de kretskort som är monterade på samma ställe.

Plattskärmen är uppbyggd av en bildpanel ytterst, ett antal filtrerande lager samt en bakgrundsbelysning, se figur 4.7.

Genom att separera alla dessa delar kan icke önskade egenskaper tas bort. I

Bild på skär- mens delar

Figur 4.7: Plattskärmens lager

figur 4.8 syns bildpanelen på skärmen och dessutom ett av de två kretskorten. Lagret som sprider ut ljuset tas bort, detta lager kommer att förstöra alla möjliga pekpunkter som tar sig genom skärmen.



Figur 4.8: Plattskärmens bildpanel

Lagret för att ge vit bakgrundsfärg, se figur 4.9, lämnas kvar. Utan detta får bilden på skärmen avsevärt sänkt kvalitet. Detta lager släpper dessutom genom

infrarött ljus och utgör därför inget större problem.

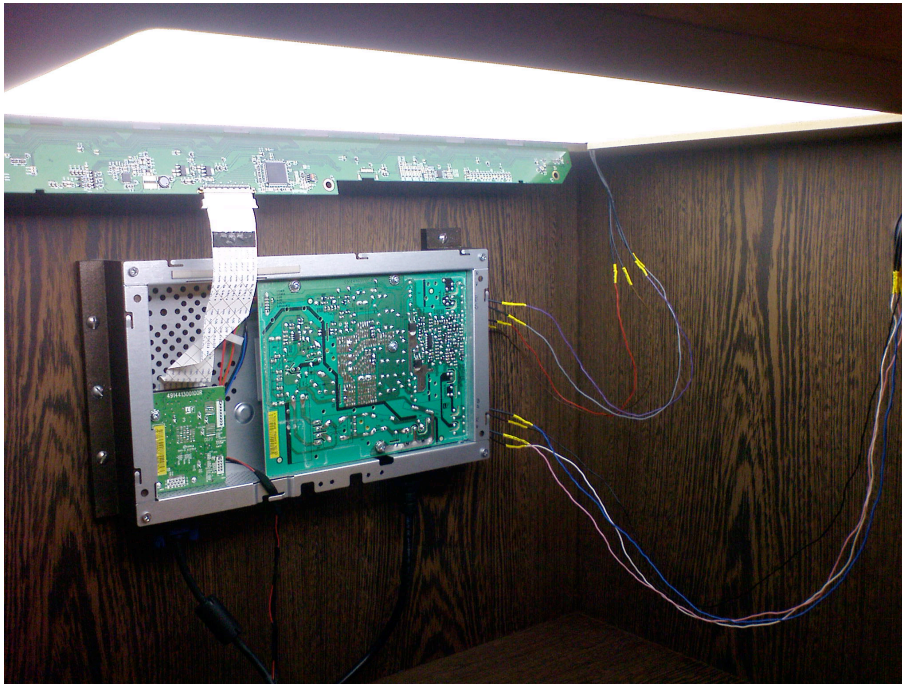


Figur 4.9: Lager för att bland annat ge vit färg

Plattskärmen monteras i chassit, se avsnitt 4.4, tillsammans med bakgrundsbelysningen, se figur 4.10. Kretskorten, som tidigare var monterade på baksidan av bakgrundsbelysningen, har nu monterats på chassits vägg. Detta eftersom de inte får vara i vägen för den optiska bildinläsningen, se avsnitt 4.1.2.

Märke, modell och storlek

För ändamålet valdes den billigaste plattskärmen i storlek 22 tum som kunde hittas. Specifikationerna över vilken skärm som används finns i tabell 4.4. Storleken 22 tum valdes utifrån att det är en vanligt förekommande skärmstorlek vilket gör det möjligt att få ett bra pris. Priset söktes med hjälp av webbsidan <http://www.prisjakt.se>.



Figur 4.10: Plattskärm monterad i chassi. På bilden syns undersidan av skärmen.

<i>Tillverkare</i>	<i>Modell</i>	<i>Inköpsställe</i>
LG	W2234S	Nya Internetworking Data i Gbg AB (Inet)

Tabell 4.4: Plattskärmsspecifikationer

4.4 Chassi

Ett chassi har konstruerats i trä. Chassit rymmer all hårdvara utom den dator som tar emot pekinformation och visar gränssnittet.

4.4.1 Mått

Chassits inre höjd är 400mm , denna höjd beslutades utifrån de beräkningar som gjordes i avsnitt 4.1.3.2. Bredden och djupet på chassit ges av plattskärmens storlek. Djupet på chassit måste dessutom utökas med bredden på kretskorten och kameran som skall monteras i bakkant. Chassits bredd är $??\text{mm}$ och djupet är

??mm.

4.4.2 Intern uppbyggnad

Inuti chassit sitter kontrollerkort för plattskärm, en kamera, en spegel samt kablar för anslutning. En översiktsbild av chassits interna uppbyggnad finns i figur 4.11



Figur 4.11: Hårdvarans uppbyggnad internt

4.4.2.1 Plattskärm

Som tidigare nämnts används en plattskärm, se avsnitt 4.3, för bildåtergivning. Plattskärmen är monterad i chassits topp med sin bildpanel uppåt. Plattskärmen har ett antal kretskort som inte får vara monterade i vägen för kamerans synfält. Dessa kretskort har därför monterats mot chassits bakre vägg, se figur 4.10.

4.4.2.2 Kamera

Bla kamera bla bla bla, monterad ovan plattskärmens kretskort. Se figur 4.12.



Figur 4.12: Kameran är monterad ovan plattskärmens kretskort

4.4.3 Anslutningar

Chassit har anslutningar för elektricitet, USB samt en Mini-D-Sub-15. Electricitetskontakten används för att ge skärmen ström och skärmen har ett intag monterat för detta, se figur 4.13. Chassit har även ett intag för USB där en standard USB-kabel (Typ A till Typ B) kan anslutas mellan datorn och chassit. Internt är kameran ansluten till denna kontakt via en USB Typ A, se figur 4.13.

För att minimera risken för störningar i signalen till plattskärmen finns inget intag för videosignal monterad på chassit. I stället har videosignalskabeln monterats genom chassit och ger möjligheten att anslutas direkt till den dator som skall visa användargränssnittet. Det finns då en möjlighet att ansluta plattskärmen utan att behöva göra en skarvning av videokabeln.



Figur 4.13: T.v intag till chassit och t.h. intern koppling

4.4.4 Start och avstängning

För att kunna starta plattskärmen måste det vara möjligt starta upp skärmen via den inbyggda start- och avstängningsknappen. För att göra det lättare att utföra denna åtgärd så har en knapp monterats vid anslutningarna på chassit, se röd knapp i figur 4.13. Denna knapp har internt kopplats över den befintliga start- och avstängningsknappen på plattskärmen.

4.5 Sammanfattning

En hårdvaruprototyp har konstruerats utifrån en plattskärm, en webbkamera samt infraröda lysdioder. Initialt var planen att använda utspridd upplysning men detta misslyckades. I stället skapades ett antal pennor med infraröda lysdioder som kan användas för pekning.

Kapitel 5

Mjukvaruprototyp

I detta kapitel kommer detaljerade specifikationer om mjukvaruprototypens konstruktion att förklaras. Utöver detta kommer bakomliggande tekniker för mjukvaran att förklaras. Mjukvaruprototypen kommer att använda en befintlig spårare.

5.1 Spårare

Hårdvaruprototypen som presenterats i denna uppsats använder optisk avläsning, se avsnitt 4.1. Den information som erhålls är en sekvens av stillbilder. Spåraren måste tolka bilderna och generera pekpunkter utifrån dessa.

5.1.1 Funktionalitet

Bilden innehåller en mängd information. Enbart en delmängd av denna information är intressant för tillämpningen, exempelvis är färg inte intressant. Genom att applicera lämpliga filter på bilden gallras irrelevant information bort. Den intressanta information har olika utseende beroende på vilka förhållanden som rådde när bilden togs. På grund av detta behövs olika filter vid olika förhållanden.

Efter filtrering skall spåraren tolka och analysera den kvarvarande information. Spåraren måste identifiera pekpunkter och följa dessa mellan olika bilder i sekvensen. Till sist lämnar spåraren ifrån information om de aktiva pekpunkterna

och deras position.

5.1.1.1 Filtrering

För att kunna finna de pekpunkter som är intressanta finns, förutom den aktuella bilden, en referensbild. Referensbilden är tagen av kameran när inga pekpunkter finns aktiva. Genom att subtrahera informationen i referensbilden från den aktuella bilden har all bakgrundsinformation tagits bort [28].

Nivåer, kontrast

5.1.1.2 Tolkning och analys

Spåraren har nu en bild innehållandes noll eller fler punkter. För varje ljus punkt måste nu spåraren analysera om punkten

5.1.1.3 Val av spårare

Till denna mjukvaruprototyp har spåraren *Touché* valts, men vilken spårare som helst, som implementerar TUIO, kan användas. *Touché* har valts på grund av att den fungerar på Apple® MacOS® X samt att den upplevs enkel att konfigurera.

5.2 Klient

Den klient som har implementerats för att demonstrera tekniken har kallats *Lime - A Slight Touch of Lime*. Källkoden för denna klient finns i Bilaga C.

5.2.1 Val av programspråk och plattform

Mjukvaruprototypen skrivs i Objective-C® för Apple MacOS X. Anledningen till detta val är ... ? Macägare

5.2.1.1 Bibliotek

Klienten använder sig av fyra huvudbibliotek. Dessa tillhandahåller funktionalitet för grafik, kommunikation och datahantering.

Cocoa

Cocoa [12] är en objektorienterad miljö som används för programutveckling i Apple MacOS X. Cocoa består främst av två ramverk; Foundation Kit och Application Kit. Dessa ramverk tillhandahåller bland annat strängar, värdemanipulering, containers samt det grafiska gränssnittet.

Quartz

Quartz [13] är ett två-dimensionellt API för att rita grafik i Apple MacOS X. Quartz 2D tillhandahåller Portable Document Format (PDF)s ritmodell. Quartz tillhandahåller också flera av de visuella effekter som MacOS X användargränssnitt använder, exempelvis transparens och skuggor.

En skiss över Quartz lagerstruktur finns i figur 5.1.

Core Animations

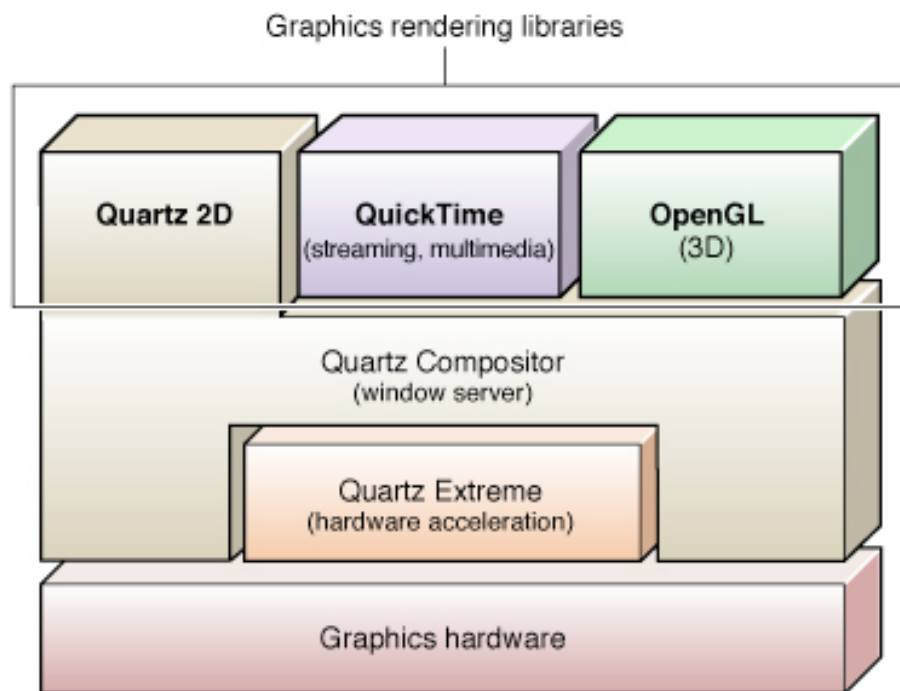
III

BBOSC

BBOSC används för att ta emot data från spåraren, ytterligare information finns i avsnitt 5.3.2.1.

5.2.2 Programdesign

test



Figur 5.1: Quartz uppbyggnad

ÖVERSIKTSBILD ÖVER SYSTEMET UPPBYGGNAD

5.2.2.1 Rotationslager - rotera kring ankarpunkt

Beskrivning

test

Ritningar

test

KOD

test

5.2.2.2 Flyttlager - flytta lager mellan två punkter

Beskrivning

test

KOD

test

5.2.2.3 Skalningslager - skala lagrets innehåll

Beskrivning

test

KOD

test

5.2.2.4 Egendefinierade lagertyper

Beskrivning

test

5.2.2.5 Lagerinnehåll

test

Bilder

test

Former

test

Film

test

Text

test

5.2.2.6 Lagerhanterare

Test

5.2.2.7 Problem

Ett antal problem uppstod under programmeringen logik för olika lager. Nedan beskrivs problematiken samt lösningen. Utförligare beskrivningar av hur varje lagertyp är implementerade finns längre ned i detta avsnitt.

Rotering av lager kring ankarpunkt

Problem med förskjutning av lager

Mappning av ankarpunkt mellan två koordinatsystem

Inbyggd metod gör inte som man vill?

5.2.3 Användargränssnitt

5.2.3.1 Navigering

ATT TÄNKA PÅ TIPS.... IDÉER

5.2.4 Gester

5.3 Spårare och klientkommunikation

Den här prototypen använder sig av protokollet TUIO för att hämta in information om pekpunkter. VARFÖR HAR VI VALT DET? = STÖD FÖR FLERA SPÅRARE ÄN ET

5.3.1 Om TUIO

TUIO[16] utvecklades som en del av reactTable* [17], se avsnitt 2.2.2, och bygger på OSC-protokollet. Detta gör det möjligt att använda TUIO på alla enheter som har stöd för OSC, exempelvis Flash [16].

Innan ytterligare information om TUIO presenteras kommer en kort presentation om OSC-protokollet och hur det är uppbyggt.

5.3.1.1 OSC

OSC [36], OpenSound Control, är ett protokoll för kommunikation mellan datorer och multimediaenheter som exempelvis synthar. OSC är optimerat för nätverkskommunikation och är konstruerat för klient/server-kommunikation. OSC är utöver detta oberoende av vilket transportprotokoll som används.

OSCs datablock kallas för paket och ett paket innehåller ett eller flera meddelanden. Ett meddelande består av en adress, en typ samt noll eller fler argument. De datatyper som stöds enligt OSC är ASCII-strängar, 32-bitars flyttal ,heltal (in-

teger) samt 64-bitars tidstämplar och binär data [36] [35].

Adressen i meddelandet ger destinationen [35], adresserna är inte förbestämda i protokollet vilket gör att man kan skapa nys sorters adresser. Adressen är uppbyggd av noder som är separerade med ett “/”, ex. “/sverige/karlstad/universitetsgatan”. Sista noden i adressen är mottagaren av meddelandet.

5.3.1.2 TUIO över OSC

TUIO definierar en uppsättning meddelanden i OSC, se figur 5.2, med uppgift att förmedla information om objekt. I huvudsak finns två meddelandetyper. En meddelandetyper, *alive*, talar om vilka objekt som befinner sig på ytan. En annan meddelandetyper, *set*, talar om information om de objekt som är placerade på ytan. Utöver dessa definierar TUIO ett meddelande som talar om sekvensnummer (fseq) samt ett meddelande med namnet på servern (source). [16]

```
/tuio/[profileName] source name  
/tuio/[profileName] set [sessionId parameterList]  
/tuio/[profileName] alive [list of active sessionIDs]  
/tuio/[profileName] fseq [int32]
```

Figur 5.2: TUIO, Meddelandetyper

TUIO-meddelanden är uppdelade i profiler, se figur 5.3. Profilnamnet finns lagrad i OSC-adressen och är där ändnoden. Profilen föregås av “/tuio/”. [16]

Varje meddelandetyper har ett antal argument, dessa framgår av figur 5.1. [16]

5.3.2 TUIO i mjukvaruprototyp

Mjukvaruprototypen använder sig enbart av en profil från TUIO. Profilen som används är “/tuio/2Dcur” eftersom unikt identifierbara objekt inte är intressant i denna implementation.

2D Interactive Surface

```
/tuio/2Dobj set s i x y a X Y A m r
/tuio/2Dcur set s x y X Y m
```

2.5D Interactive Surface

```
/tuio/25Dobj set s i x y z a X Y A m r
/tuio/25Dcur set s x y z X Y Z m
```

3D Interactive Surfaces

```
/tuio/3Dobj set s i x y z a X Y Z A B C m r
/tuio/3Dcur set s x y z X Y Z m
```

raw profile

```
/tuio/raw_[profileName]
/tuio/raw_dtouch set i x y a
```

custom profile

```
/tuio/_[formatString]
/tuio/_sixyP set s i x y 0.5
```

Figur 5.3: TUIO, Profiler

s	sessionID, temporary object ID, int32
i	classID (e.g. marker ID), int32
x, y, z	position, float32, range 0...1
a, b, c	angle, float32, range 0..2PI
X, Y, Z	movement vector (motion speed and direction), float32
A, B, C	rotation vector (rotation speed and direction), float32
m	motion acceleration, float32
r	rotation acceleration, float32
P	free parameter, type defined by OSC packet header

Tabell 5.1: TUIO, Meddelandeargument

5.3.2.1 BBOSC

För mjukvaruprototypen har biblioteket BBOSC används för att ta emot och behandla OSC-meddelanden över nätverk. Biblioteket tar emot alla slags OSC-meddelanden och en filtrering måste göras för att se till att endast läsa den profil som är intressant.

```
for(BBOSCMessage* theMessage in messages)
{
    if([[[theMessage address] address] isEqualToString:@"tuio/2Dcur"])
    {
        //Ta hand om meddelande
    }
}
```

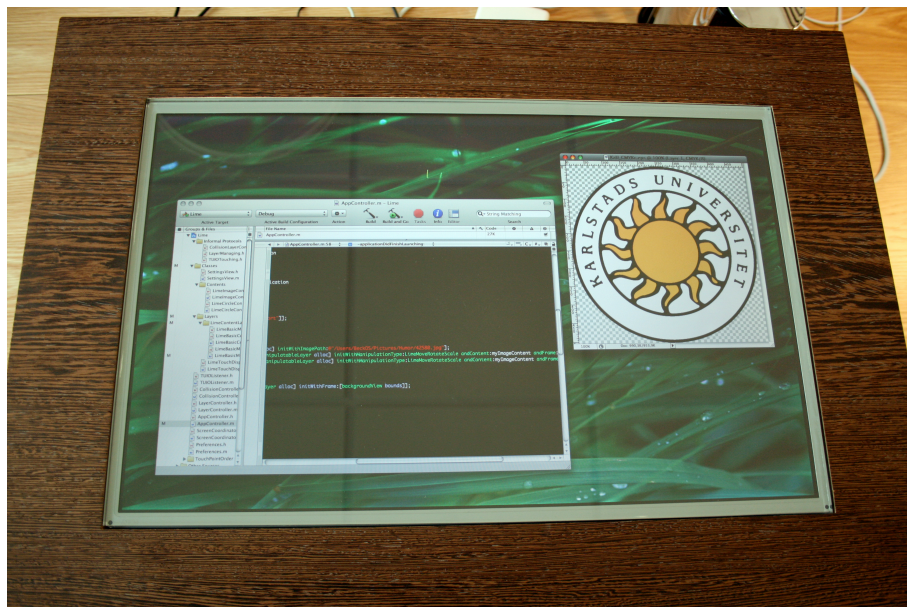
```
    }  
}  
  
for(int i = 0; i < 10; i++)  
{  
    printf(i);  
}
```

5.4 ÖVRIGT

Läckor?

Kapitel 6

Utvärdering av prototyp



Figur 6.1: Bildåtergivning ger godkänt resultat

Utvärdering 1 - Rektanglar och trianglar

Utvärdering 2 - Utökad

Sammanfattande utvärdering

Användandet av Stylus inte så bra som fingrar.

Kapitel 7

Slutsats

7.1 Översikt

7.2 Detta projekt

7.3 Framtida vidareutveckling

7.3.1 Hårdvara

Massor av förbättringar

7.3.2 Mjukvara

7.3.2.1 Lager

Lätt att göra fler lager p.g.a programstruktur

Litteraturförteckning

- [1] Bill Buxton. Internetsida, Sep 2008. Multi-Touch Systems that I Have Known and Loved.
- [2] Microsoft Corporation. Microsoft Surface, Sep 2008. <http://www.microsoft.com/surface>, läst Sept. 2008.
- [3] RAND CORPORATION. Rand 2007 annual report. Årsrapport, 2007.
- [4] Anders Lotsson DATATERMSGROUPEN KTH. E-post mellan anders lots-son och henrik bäck. Sep 2008.
- [5] M.R Davis and T.O Ellis. The rand tablet: A man-machine communication device. pages 1–21, May 1964.
- [6] W den Boer, A Abileah, P Green, and T Larsson. 56. 3: Active matrix lcd with integrated optical touch screen. *SID Symposium Digest of Technical Papers*, Jan 2003.
- [7] Morrison et al. Passive touch system and method of detecting user input. Patent 20 070 075 982, United States Patent and Trademark Office, Apr 2007.
- [8] M Grossman and D Walter. Teaching with interactive computer capabilities (plato: Computer-based education for animal breeding). *Dairy Sci*, (61):1308 – 1311, Sep 1978.
- [9] Jefferson Y Han. Low-cost multi-touch sensing through frustrated total internal reflection.
- [10] Apple Inc. Pressmedelände, SAN FRANCISCO, MACWORLD, Jan 2007. <http://www.apple.com/pr/library/2007/01/09iphone.html>, läst Sept. 2008.
- [11] Apple Inc. Pressmedelände, California, CUPERTINO, Jul 2008. <http://www.apple.com/pr/library/2008/07/14iphone.html>, läst Sept. 2008.

- [12] Apple Inc. Leopard Guides, Cocoa, Aug 2008. <http://developer.apple.com/documentation/Cocoa/index.html>, läst Nov. 2008.
- [13] Apple Inc. Leopard Reference: Graphics Imaging, Quartz, Aug 2008. <http://developer.apple.com/reference/GraphicsImaging/idxQuartz-date.html>, läst Nov. 2008.
- [14] Tyco Electronics Inc. Internetsida, Sep 2008. <http://www.elotouch.com/AboutElo/History>, läst Sept. 2008.
- [15] S Jordà. Sonigraphical instruments: from fmol to the reactable. *Proceedings of the 2003 conference on New interfaces for ...*, Jan 2003.
- [16] M Kaltenbrunner, T Bovermann, and R Bencina. Tuio: A protocol for tabletop tangible user interfaces. *Proc. of the The 6th Int'l Workshop on Gesture in Human- ...*, Jan 2005.
- [17] M Kaltenbrunner, G Geiger, and S Jordà. Dynamic patches for live musical performance. *Proceedings of the 2004 conference on New interfaces for ...*, Jan 2004.
- [18] G Roberts Lawrence. The lincoln wand. *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 223–227, Sep 1966.
- [19] S.K Lee, W Buxton, and K.C Smith. A multi-touch three dimensional touch-sensitive tablet. *CHI '85 PROCEEDINGS*, pages 21–25, 1985.
- [20] Asahi Spectra Co. Ltd. Produkt, California, Torrance, Jan 2007. http://www.asahi-spectra.com/opticalfilters/syousaik2_ver2.asp?key=YBPA880, läst Nov. 2008.
- [21] Nobuyuki Matsushita and Jun Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. *UIST 97*, pages 209–210.
- [22] Nima Motamedi. Hd touch: multi-touch and object sensing on a high definition lcd tv. *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, Apr 2008.
- [23] Brad A Myers. A brief history of human computer interaction technology. *ACM interactions*, 5(2):44–54, Mar 1998.
- [24] N-trig. Pressmedelände, N-trig's DuoSense™ Digitizer Embedded in Intel's New UrbanMax Mobile Computer, Aug 2008. <http://www.n-trig.com/Content.aspx?Page=PressReleases&PressReleaseId=315>, läst Sept. 2008.

- [25] Russell Owen, Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. When it gets more difficult, use both hands – exploring bimanual curve manipulation. page 8, Mar 2005.
- [26] Frank L. Pedrotti and Leno D. Pedrotti. *Introduction to Optics*. Pearson Education LTD., 1993.
- [27] Robert Lubkoll Björn Bollensdorff Prof. Marc Alexa, Karl Wessel. LEDs and IR Filters, Project Multi-touch Display, Jan 2008. <http://user.cs.tu-berlin.de/~karlw/LEDSpdf2.pdf>, läst Nov. 2008.
- [28] J Rekimoto and N Matsushita. Perceptual surfaces: Towards a human and object sensitive interactive display. *Workshop on Perceptual User Interfaces (PUP97)*, Jan 1997.
- [29] Dan Saffer. *Interactive Gestures: Designing Gestural Interfaces*. O'REILLY, 2008.
- [30] B Shneiderman. Touch screens now offer compelling uses. *IEEE Software*, 8(2):93–94, Mar 1991.
- [31] Ben Shneiderman and Cathrene Plaisant. *Designing the user interface*. Addison-Wesley, fourth edition.
- [32] Norbert A Streitz, Jörg Geißler, Torsten Holmer, Shinichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. i-land: an interactive landscape for creativity and innovation. *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 120–127, 1999.
- [33] Entertainment Design TED, Technology. Speakers Jeff Han: Human-computer interface designer, Sep 2008. http://www.ted.com/index.php/speakers/jeff_han.html, läst Sept. 2008.
- [34] P Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. *ACM UIST*, pages 27–33, Nov 1991.
- [35] Matthew Wright. Open Sound Control 1.0 Specification, Sep 2002. http://opensoundcontrol.org/spec-1_0, läst Okt. 2008.
- [36] Matthew Wright, Adrian Freed, and Ali Momeni. Opensoundcontrol: State of the art 2003. *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, pages 154–159, May 2003.

Bilaga A

Figurreferenser

Figurer nämnda i denna bilaga har ägs av nämnda nedan, övriga figurer är framtagna för denna uppsats och får användas fritt.

Karlstads universitets sol: Karlstads universitet Figur 2.2: RAND Corporation

Figur 2.3: Wikipedia, GNU Free Documentation License

Figur 2.4: Xavier Sivecas, ReactTable*

Figur 2.5: Apple Inc.

Figur 2.6: Microsoft Corporation

Figur 4.2: Microsoft Corporation

Bilaga B

Ritningar för chassi

Bilaga C

Programkod för klient