



Fakulteten för ekonomi, kommunikation och IT  
Avdelningen för Datavetenskap

Mathias Andersson och Henrik Bäck

# Flerpunktspekskärm 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. Utveckla?



# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
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
<b>2</b>	<b>Bakgrund</b>	<b>5</b>
2.1	Introduktion till flerpunktspekaskärmar . . . . .	5
2.1.1	Simultan interaktion . . . . .	5
2.1.2	Tillämpningar . . . . .	6
2.2	Historia . . . . .	6
2.2.1	En-punktsinmatning . . . . .	6
2.2.2	Fler-punktsinmatning . . . . .	9
2.3	Pekskärmar och flerpunktspekaskärmar idag . . . . .	13
2.4	Sammanfattning . . . . .	14
<b>3</b>	<b>Potentiella lösningar</b>	<b>15</b>

3.1	Hårdvara . . . . .	15
3.1.1	Bildåtergivning . . . . .	15
3.1.2	Pekpunktsregistrering . . . . .	16
3.1.3	Chassi . . . . .	19
3.2	Mjukvara . . . . .	20
3.2.1	Pekpunktsinformation . . . . .	21
3.2.2	Klient . . . . .	22
3.2.3	Spårare och klientkommunikation . . . . .	23
3.3	Begränsningar och krav . . . . .	24
3.3.1	Hårdvara . . . . .	24
3.3.2	Mjukvara . . . . .	24
<b>4</b>	<b>Hårdvaruprototyp</b>	<b>25</b>
4.1	Pekpunktsregistrering . . . . .	25
4.1.1	Val av lösning . . . . .	25
4.1.2	IR-kamera . . . . .	25
4.1.3	Beräkningar . . . . .	25
4.1.4	Bildinhämtning . . . . .	26
4.2	Pekning . . . . .	26
4.2.1	Val av lösning . . . . .	26
4.2.2	Problem . . . . .	26
4.2.3	IR-Penna . . . . .	27
4.3	Bilåtergivning . . . . .	27
4.3.1	Val av lösning . . . . .	27
4.4	Sammanfattning . . . . .	28
<b>5</b>	<b>Mjukvaruprototyp</b>	<b>31</b>
5.1	Spårare . . . . .	31

5.1.1	Funktionalitet . . . . .	31
5.2	Klient . . . . .	32
5.2.1	Val av programspråk och plattform . . . . .	32
5.2.2	Programdesign . . . . .	33
5.2.3	Användargränssnitt . . . . .	33
5.2.4	Gester . . . . .	33
5.3	Spårare och klientkommunikation . . . . .	33
5.3.1	Om TUIO . . . . .	33
5.3.2	TUIO i mjukvaruprototyp . . . . .	35
5.4	ÖVRIGT . . . . .	36
<b>6</b>	<b>Utvärdering av prototyp</b>	<b>37</b>
<b>7</b>	<b>Slutsats</b>	<b>39</b>
	<b>Litteraturförteckning</b>	<b>41</b>



# Figurer

2.1	Historisk tidslinje för inmatningssystem . . . . .	7
2.2	The Rand Tablet . . . . .	8
2.3	AccuTouch™ 1977 (License pending) . . . . .	9
2.4	PLATO 1978 . . . . .	10
2.5	reactTable* . . . . .	11
2.6	iPhone 3G . . . . .	12
2.7	Microsoft Surface . . . . .	13
3.1	Utspridd upplysning . . . . .	17
3.2	Frustrerad total intern reflektion . . . . .	19
3.3	Laserplansupplysning . . . . .	20
3.4	Lysdiodspenna . . . . .	20
4.1	Plattskärmens lager . . . . .	28
4.2	Plattskärmens bildpanel . . . . .	29
4.3	Lager för att bland annat ge vit färg . . . . .	29
4.4	Plattskärm monterad i chassi. På bilden syns undersidan av skärmen. . . . .	30
5.1	TUIO, Meddelandetyper . . . . .	34
5.2	TUIO, Profiler . . . . .	35





# Tabeller

3.1	Olika spårare som implementerar TUIO . . . . .	21
3.2	Meddelandetyper, internt i klient . . . . .	22
4.1	Komponenter för konstruktion av IR-pennor . . . . .	27
5.1	TUIO, Meddelandeargument . . . . .	35



# Kapitel 1

## Inledning

Den här uppsatsen kommer att visa hur man kan lösa problemet med att flera användare samtidigt kan interagera med ett system via en och samma in-dataenhet. Som in-dataenhet kommer en flerpunktspekskärm<sup>1</sup> att användas. Flerpunktspekskärmen kommer att konstrueras från standardkomponenter. En applikation kommer att utvecklas för att demonstrera delar av den funktionalitet som flerpunktspekskärmen tillhandahåller.

I det här kapitlet ...

### 1.1 Problem

Hur gör man ett system som ger illusionen att flera manipulationer sker samtidigt?

Interaktion med system, samma yta

En eller flera användare

Flera pekpunkter -> En sådan förändring skulle medföra nya möjligheter för hur inmatning kan ske och vilka operationer som kan utföras [22]

Direkt manipulation [26].

### 1.2 Motivering

Varför blir detta bra?

Bättre än ngn annan?

---

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

Likvärdigt?

Varför görs jobbet?

Dyrt med andra?

## 1.3 Mål

Målet med den här uppsatsen är lösa problemet med att flera användare simultant kan interagera med ett och sammas system via samma in-data enhet. Målet består av två delar, en hårdvaru- och en mjukvarudel. allmänt om vad hårdvaran och mjukvaran är

## 1.4 Genomförande

Låda-Skärm-Mjukvara

## 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.

Utvärdering kap6

Slutsats kap7

## **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 [27]. 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. [26] [27]

#### 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 [22]. Här måste mjukvaran utvecklas eftersom

Två personer samtidigt vill manipulera objekt i ett system

## 2.1.2 Tillämpningar

Flerpunktspekskärmar 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. Anslagstavlor 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 vyer 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 [22]. Exempel på en teknik som passar in här är "TapeDrawing" och man slipper då problemet med att omsätta analoga ritningar digitalt senare.

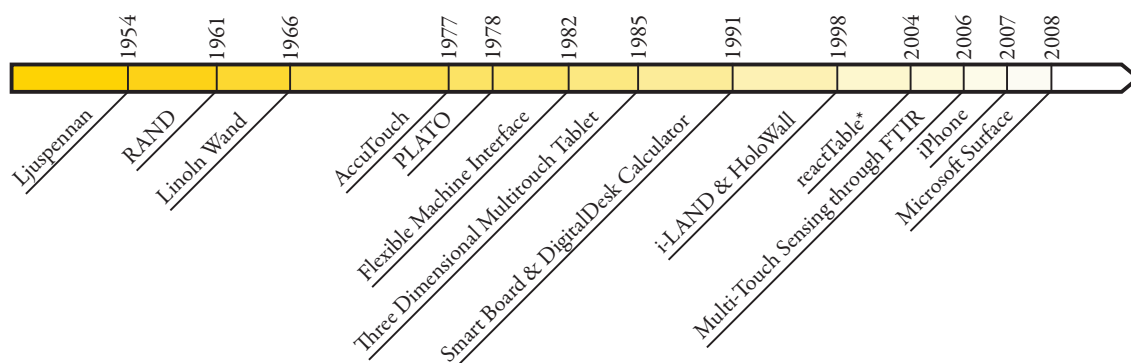
## 2.2 Historia

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

### 2.2.1 En-punktsinmatning

I detta avsnitt kommer inmatningsmetoder relaterade till pekskärmar att presenteras.



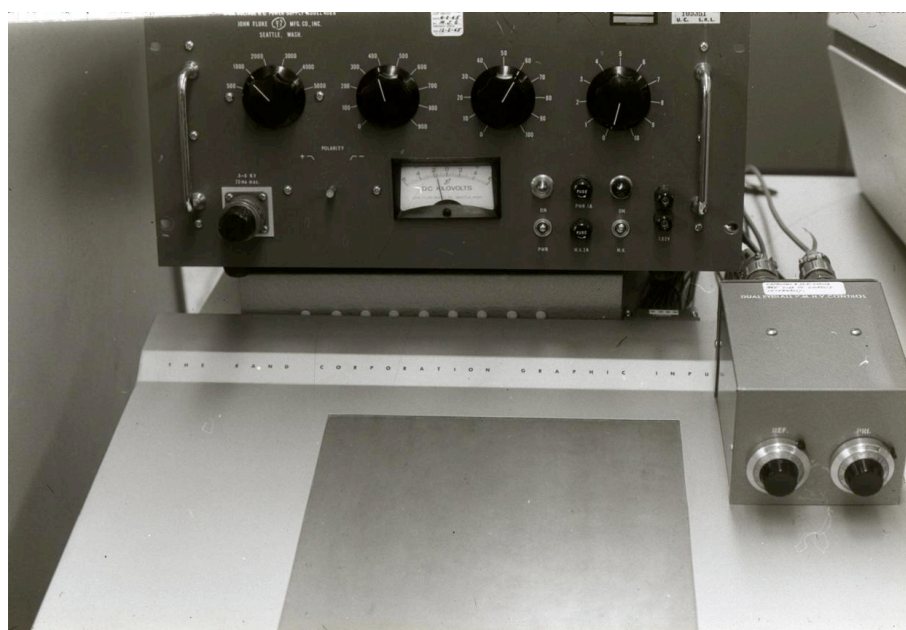


Figur 2.1: Historisk tidslinje för inmatningssystem

1954 introducerades ljuspennan som inmatningsmetod [20]. 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 ritbord (tablet)[5].

1966, fem år efter RAND, utvecklades en in-dataenhet kallad THE LINCOLN WAND [16], 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

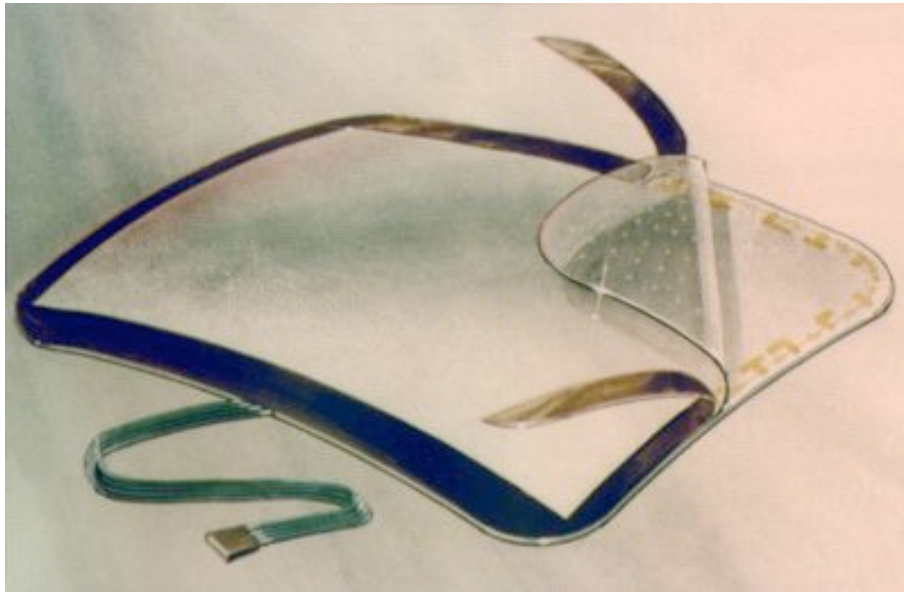


Figur 2.2: The Rand Tablet

för pennan beräknas.

Elva år senare, 1977, tillverkades en transparent yta kallad AccuTouch™[12], se figur 2.3, 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 2.4. 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 IR-Ljussändare 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 appli-



Figur 2.3: AccuTouch™ 1977 (License pending)

kationen [1].

Från 1980-talet och framåt började pekskärmar leta sig in i allt fler kommersiella produkter [25]. Hewlett-Packard lanserade kommersiella arbetsstationer för personligt bruk med inbyggd pekskärm. Dessa byggde på samma teknik som den i PLATO.

Under nittiotalet, 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 flerpunktspekskärmar.

Redan 1982 togs en prototyp fram för att kunna känna av flera simultana inmatningspunkter från en eller flera användare. Prototypen fick namnet The Flexible Machine Interface [25] 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 samtidiga punk-



Figur 2.4: PLATO 1978

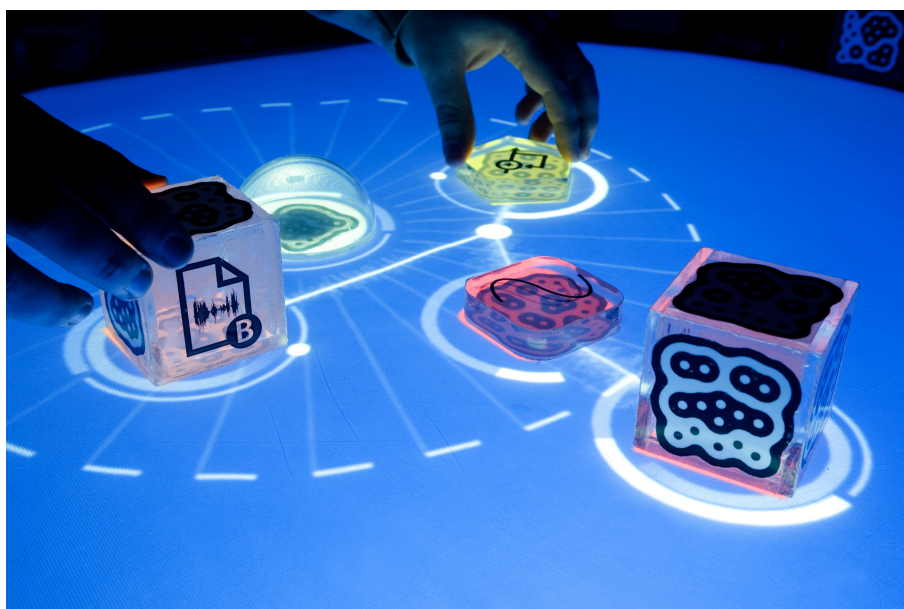
ter och som dessutom talar om graden av kontakt [17]. Ytan, som var indelad i 64 gånger 32 punkter, tillät att användaren arbetade direkt med fingrarna på ytan. Avläsningen skedde genom att ytan periodiskt skannades av med hjälp av en algoritm där punkter med lägre elektrisk laddning upptäcks. Ytan rapporterade 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 [30] 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 [28]. 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 samtidigt användare och InteracTable var ett bord runt vilket upp till sex personer kunde stå. Bordet kunde ta emot in-data från flera samtidigt användare.

Samma år presenterades även HoloWall [18]. HoloWall var en vägg designad för att kunna ta emot fler samtidigt 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 IR-ljus som belyser upp ytan. Kameran kan registrera det IR-ljus som reflekteras och på så sätt räkna ut pekpunkter.

Under tjugohundratalet, närmare bestämt 2004, presenterades en fungerande prototyp av reactTable\*[15], figur 2.5. 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 [13].



Figur 2.5: reactTable\*

De tre åren 2006, 2007 och 2008 innebar stora språng på marknaden för flerpunktspekskärmar. 2006 presenterade Jefferson Y. Han en flerpekpunktsyta som kunde konstrueras både enkelt och icke kostnadskrävande [9]. Han presenterade sina idéer på konferensen TED (Technology, Entertainment, Design)[29] vilket fick världen att öppna ögonen för vad som kan åstadkommas.

Apple iPhone™ [10], se figur 2.6, lanserades i början av 2007 och var en av de första telefonerna med en inbyggd flepunktsspekskärm.



Figur 2.6: iPhone 3G

2008 levererades Microsoft® Surface™ [2], se figur 2.7, 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 samtidiga användare och dessutom objekt.



Figur 2.7: Microsoft Surface

## 2.3 Pekskämar och flerpunktspekskärmar idag

### JÄMFÖR med våra idéer enl kap1

Marknaden för flerpunktspekskärmar 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 pekskärmar som är populära och inte flerpunktspekskärmar.

När det gäller flerpunktspekskärmar levererar ett företag vid namn N-trig ytor för detta. N-trig levererar skärmar till bland annat Dell och Intel [21] som har implementerat dem i sina datorer.

## 2.4 Sammanfattning

Flerpunktspekskärmar ger flera fördelar före pekskärmar. 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. Flerpunktspekskärmar passar inte i alla situationer. Dock kan de med fördel användas i flera specifika tillämpningar.

Pekskärmar är ingen nyhet på marknaden och så inte heller flerpunktspekskärmar. 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 expanderar.

### 3.1 Hårdvara

#### 3.1.1 Bildåtergivning

För att flerpunktspekskärmen skall fungera måste in- och utdata presenteras på samma yta. Det finns flera sätt att lösa detta på och det går kombinera olika sorters bildåtergivningsmetoder med pekpunktsavläsning.

#### Digitalprojektor

Ett bildåtergivningssätt är att använda sig av en digitalprojektor<sup>1</sup>. En digitalprojektor monteras så att den projicerar en bild på en duk eller ett plexiglas<sup>2</sup>. Digitalprojektorn kan monteras i sitt ursprungliga skick men kan behöva kompletteras med speglar eftersom avståndet för projicering kan bli kort. [9] Vissa digitalprojektorer kan också behöva kompletteras med ett IR-filter vilket tar bort oönskat IR-ljus från projektorns lampa [18].

---

<sup>1</sup>Engelska: Video projector

<sup>2</sup>Polymetylmetakrylat

## Plattskärm

Ett annat alternativ, som i mindre skala, är billigare än projektorn är att använda sig av en plattskärm<sup>3</sup>. Plattskärmen kan behöva förberedas för att kunna användas tillsammans med olika typer av pekpunktsregistrering.

Om optisk pekpunktsregistrering, se avsnitt 3.1.2, 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 [19].

### 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.

#### Kapacitiv avläsning

Genom historien har flera olika produkter använt sig av kapacitiv avläsning av pekpunkter [17] [12]. I detta fall måste den kapacitiva ytan vara transparent för att bilåtergivning 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 utredas.

#### 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

---

<sup>3</sup>LCD, med eller utan TFT

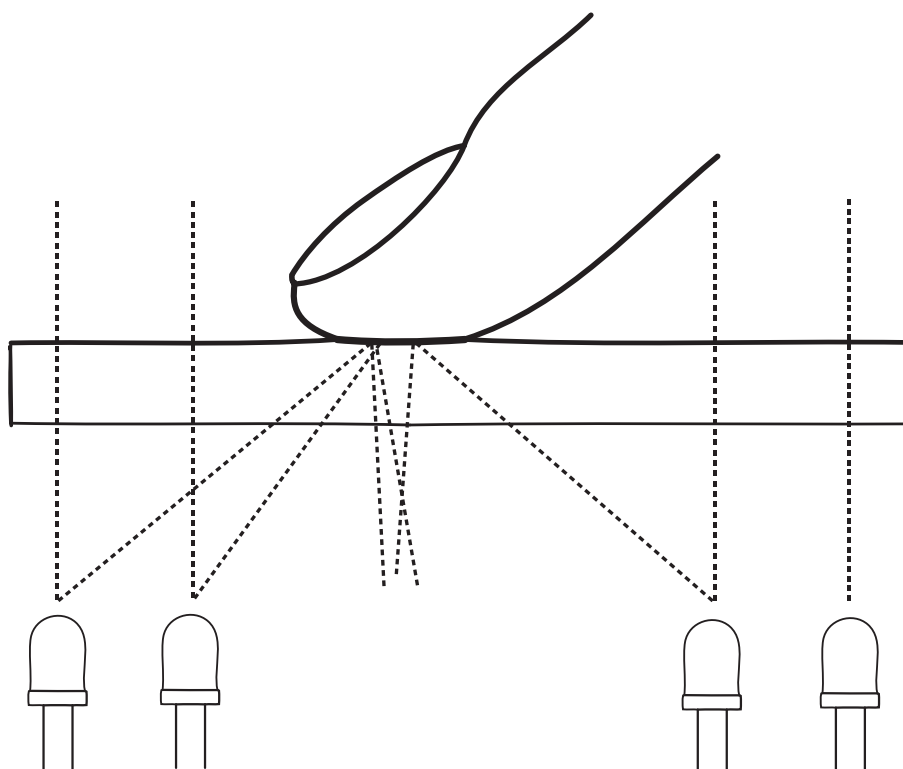
i kontakt med ytan.

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. [18][24]

Hur upplysningen av pekpunkterna sker kan varieras.

### Utspridd upplysning

Ett sätt att lysa upp punkterna är med en teknik som kallas utspridd upplysning<sup>4</sup>. Ytan som skall registreras belyses underifrån med infrarött ljus [18]. När ett finger placeras på ytan kommer detta att reflektera ljuset nedåt så att detta kan registreras av kameran, se figur 3.1.



Figur 3.1: Utspridd upplysning

---

<sup>4</sup>Engelska: Diffused Illumination

### Frustrerad total intern reflektion

Ett annat sätt att belysa pekpunkter är via så kallad frustrerad total intern reflektion<sup>5</sup> (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 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*,  $\theta_c$ , för när total reflektion uppstår kan härledas med hjälp av Snells lag [23], 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 plexiglasets och reflekteras mot fingret, se figur 3.2

### Laserplansupplysning

Laserplansupplysning<sup>6</sup>, se figur 3.3, fungerar genom att lyser med

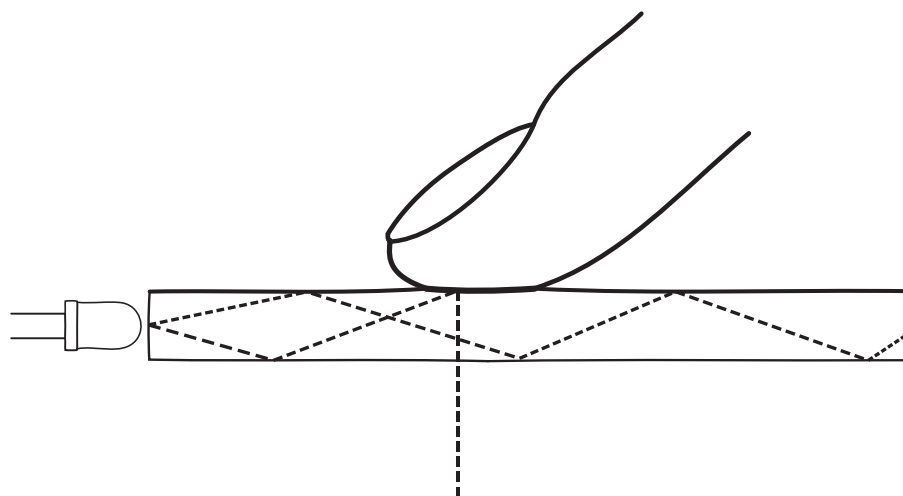
Källa saknas, ingen text ännu

### Penna

Eftersom pekpunkter registreras med hjälp av en kamera är det möjligt att belysa kameran direkt med en IR-ljuspunkt, istället för indirekt via reflektion. Detta är en säker metod eftersom

<sup>5</sup>Engelska: Frustrated Total Internal Reflection

<sup>6</sup>Engelska: Laser Light Plane Illumination



Figur 3.2: Frustrerad total intern reflektion

det är garanterat att ljus kommer 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 IR-lysdiod i en penna kan pennan användas för att peka på skärmen. Ljuset från pennans IR-lysdiod kommer att uppfattas som en pekpunkt av kameran.

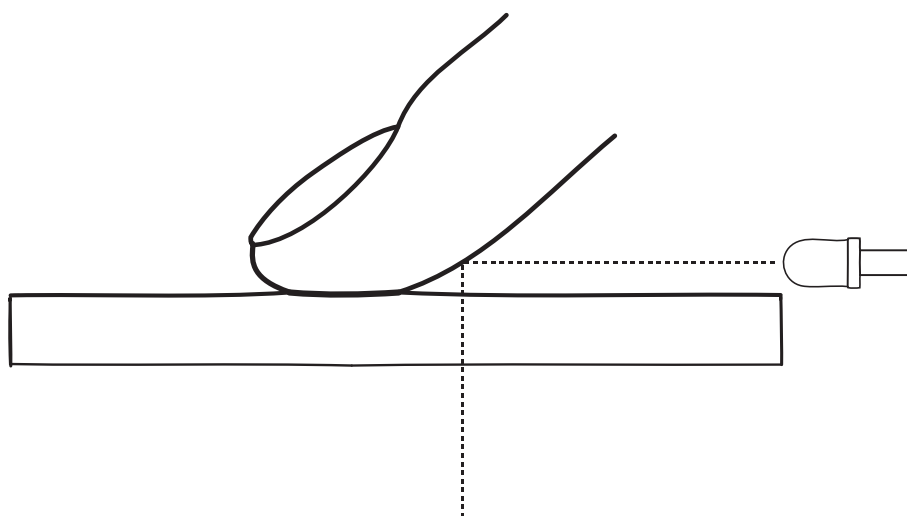
### Sensorbaserad avläsning

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].

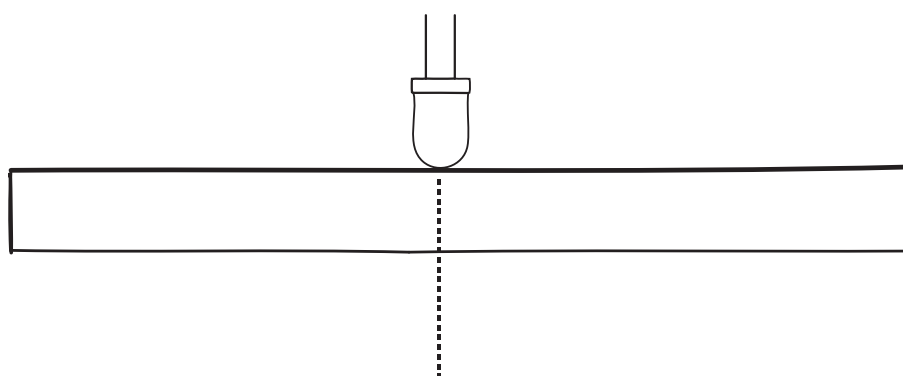
Denna metod kräver antingen att en hårdvara med tekniken köps in eller kan konstrueras. Detta alternativ kommer på grund av dess komplexitet inte att utredas i denna uppsats.

### 3.1.3 Chassi

Design, ritningar, med mera.



Figur 3.3: Laserplansupplysning



Figur 3.4: Lysdiodspenna

## 3.2 Mjukvara

Mjukvaran är otroligt viktig när det gäller pekpunktsbaserad inmatning, dock består mjukvrudelen av mer än just det grafiska gränssnittet. Här finner vi spåraren<sup>7</sup>, som har en speciell uppgift samt klienten som erhåller information från spåraren och tillämpar denna.

---

<sup>7</sup>Engelska: Tracker

### 3.2.1 Pekpunktsinformation

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

#### Optisk avsläsning

När man använder en optisk pekpunktsavläsning är den data man erhåller är 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.

#### 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. I tabell 3.1 finns exempel på ett antal spårare som implementerar TUIO.

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

Tabell 3.1: Olika spårare som implementerar TUIO

#### Övriga avläsningstyper

Det finns, som nämnt i avsnitt 3.1.2, fler sätt än att optiskt registrera pekpunkter på. Alla dessa sätt rapportera data i olika format vilket gör att det inte finns något generellt angreppssätt.

### 3.2.2 Klient

Klientens uppgift är att ta emot data från spåraren och tillämpa informationen. Klienten ritlar dessutom upp gränssnittet på skärmen. Oavsett hur data förmedlas till applikationen behöver denna en bra design för att kunna byggas ut efter behov.

#### 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.

#### 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 kollisionsdetektering 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 kan



på så sätt skickas vidare direkt till detta lager.

Det finns en stor fördel med att låta varje lager ta hand om meddelandena. I stället för att en generell funktion skall ta emot meddelanden, beräkna åtgärden och sedan tala om för lagret vad det skall göra kan lagret göra detta själv. Det gör att nya lager-typer kan introduceras utan att omgivningen i programmet 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 sådant lager skulle kunna ta emot meddelandena *Spela* och *Stoppa*. Nu måste en del av klienten skrivas om för att kunna veta vilka meddelanden 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 denna information till klienten kan variera beroende på implementation.

#### API

Ett sätt att låta klienten kommunicera med spåraren är via ett API. API varierar mellan olika spårare och måste därför anpassas efter vilken spårare som används. Detta innebär att applica-

tionen måste anpassas efter varje spårare som skall användas.

## **TUIO**

Ett annat sätt är att använda sig av ett väl-specifierat protokoll. TUIO [14] är ett protokoll framtaget enbart för syftet att förmedla information om optiskt identifierade objekt i de här sammanhangen. 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 Ta emot pekpunkter

### **3.3.2 Mjukvara**

Ska kunna...

Ta emot data Hantera den Göra lämpliga åtgärder

## Kapitel 4

# Hårdvaruprototyp

En prototyp 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 tillvägagångssättet för pekpunktsregistrering att förklaras. Efter detta förklaras metoden för bildåtergivning.

### 4.1 Pekpunktsregistrering

#### 4.1.1 Val av lösning

#### 4.1.2 IR-kamera

Bilder tas med kameran

Kameran har filter enligt modellen i kap 3

Bilderna skickas till datorn via USB

#### 4.1.3 Beräkningar

Kamera monteras ovan krestkorten, spegel

#### Kamerans synvinkel

Hur långt från ytan måste den sitta

## Chassits höjd

Föregående ger oss hur hög lådan måste vara

## Spegelns storlek

Minska lådans storlek med spegel

### 4.1.4 Bildinhämtning

Kamera

## 4.2 Pekning

### 4.2.1 Val av lösning

Då optisk pekpunktsregistrering används finns flera sätt att skapa pekpunkter.

Vi provar DI

### 4.2.2 Problem

DI

Gick inget bra, vi såg inte mycket

FTIR

Potentiellt fungerar det, inte utrett ännu Hoppas på att hinna prova det i början på Nov, annars tas detta s

### 4.2.3 IR-Penna

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

#### Uppbyggnad

IR-pennorna använder en whiteborard-penna som grund. Inuti pennan har...

Mathias skriver resten

Genomskärbild + fotografi

<i>Komponent</i>	<i>Tillverkare och modell</i>	<i>Inköpsställe</i>
OH-penna	Edding 363	
IR-lysdiod	Osram SFH485-2	ELFA
Tangentbordsswitch 6x6x4,3 mm	Okänd	Kjell & Company
Batteri 1,5 Volt AAA	Duracell	Clas Ohlson
Motstånd, 48 Ohm	Okänd	ELFA

Tabell 4.1: Komponenter för konstruktion av IR-pennor

## 4.3 Bilåtergivning

### 4.3.1 Val av lösning

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

#### Plattskärm

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

Plattskärmen är uppbyggd av en bildpanel ytterst, ett antal filtrerande lager samt en bakgrundsbelysning, se figur 4.1.

# Bild på skärmens delar

Figur 4.1: Plattskärmens lager

Genom att separera alla dessa delar kan icke önskade egenskaper tas bort. I figur 4.2 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.

Lagret för att ge vit bakgrundsfärg, se figur 4.3, lämnas kvar. Utan detta får bilden på skärmen avsevärt sänkt kvalité. Detta lager släpper dessutom genom IR-ljus och utgör därför inget större problem.

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

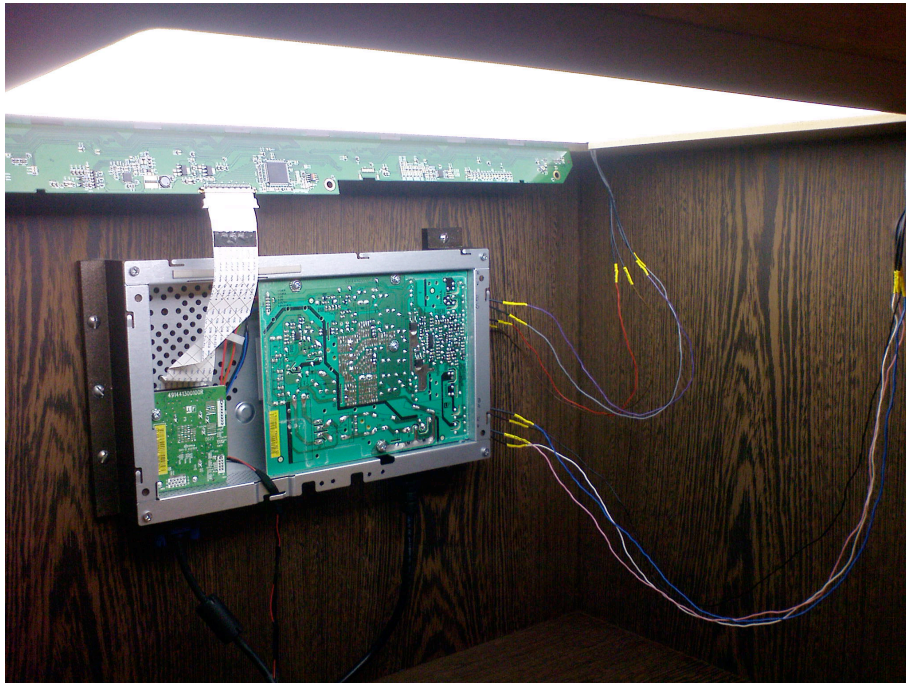
## 4.4 Sammanfattning



Figur 4.2: Plattskärmens bildpanel



Figur 4.3: Lager för att bland annat ge vit färg



Figur 4.4: Plattskärm monterad i chassi. På bilden syns undersidan av skärmen.



## 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.

## Filtrering

### Filtreringstekniker

Ta bort bakgrund

Nivåer, kontrast

## Tolkning och analys

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 [24].

För att spåraren skall kunna

Behandla med filter Tolkar resultatet Meddelar vilka blobar (kontrulös r

## Val av spårare

Till denna mjukvaruprototyp har spåraren **Touché** valts men vilken spårare som helst, som implementerar TUIO, kan användas.

## 5.2 Klient

Den klient som har implementerats för att demonstrera tekniken har kallats *Lime - A Slight Touch of Lime*.

### 5.2.1 Val av programspråk och plattform

Mjukvaruprototypen skrivs i Objective-C<sup>®</sup> för Apple<sup>®</sup> MacOS<sup>®</sup> X. Anledningen till detta val är

## Bibliotek

CoCoa

Quartz

Core Animations

### 5.2.2 Programdesign

Lager

Problem

Innehåller punkt

containsPoint fungerar inte som den skall, orask okänd. Egen lösning

### 5.2.3 Användargränssnitt

### 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 EN...GOTT SKÅL NO

### 5.3.1 Om TUIO

TUIO[14] utvecklades som en del av reactTable\* [15], 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 [14].

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

## OSC

OSC [32], 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 (integer) samt 64-bitars tidstämplar och binär data [32] [31].

Adressen i meddelandet ger destinationen [31], adresserna är inte förbestämda i protokollet vilket gör att man kan skapa nya sorters adresser. Adressen är uppbyggd av noder som är separerade med ett “/”, ex. “/sverige/karlstad/universitetsgatan”. Sista noden i adressen är mottagaren av meddelandet.

## TUIO över OSC

TUIO definierar en uppsättning meddelanden i OSC, se figur 5.1, 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). [14]

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

Figur 5.1: TUIO, Meddelandetyper

TUIO-meddelanden är uppdelade i profiler, se figur 5.2. Profilnamnet finns lagrad i OSC-adressen och är där ändnoden. Profilen föregås av “/tuio/”. [14]

```

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 5.2: TUIO, Profiler

Varje meddelandetyper har ett antal argument, dessa framgår av figur 5.1. [14]

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 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.

## 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
    }
}
```

## 5.4 ÖVRIGT

Instruments har används för att kontrollera läckor

## Kapitel 6

# Utvärdering av prototyp

Utvärdering 1 - Rektaglar och trianglar

Utvärdering 2 - Utökad

Sammanfattande utvärdering

Användandet av Stylus inte så bra som fingrar.





## Kapitel 7

# Slutsats



# 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 DATATERMSGRUPPEN KTH. E-post mellan anders lotsson 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] Tyco Electronics Inc. Internetsida, Sep 2008. <http://www.elotouch.com/AboutElo/History>, läst Sept. 2008.
- [13] S Jordà. Sonigraphical instruments: from fmol to the reactable. *Proceedings of the 2003 conference on New interfaces for ...*, Jan 2003.

- [14] M Kaltenbrunner, T Bovermann, and R Bencina. Tuio: A protocol for table-top tangible user interfaces. *Proc. of the The 6th Int'l Workshop on Gesture in Human-...*, Jan 2005.
- [15] M Kaltenbrunner, G Geiger, and S Jordà. Dynamic patches for live musical performance. *Proceedings of the 2004 conference on New interfaces for ...*, Jan 2004.
- [16] G Roberts Lawrence. The lincoln wand. *Proceedings of the AFIPS Fall Joint Computer Conference*, pages 223–227, Sep 1966.
- [17] S.K Lee, W Buxton, and K.C Smith. A multi-touch three dimensional touch-sensitive tablet. *CHI '85 PROCEEDINGS*, pages 21–25, 1985.
- [18] Nobuyuki Matsushita and Jun Rekimoto. Holowall: designing a finger, hand, body, and object sensitive wall. *UIST 97*, pages 209–210.
- [19] 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.
- [20] Brad A Myers. A brief history of human computer interaction technology. *ACM interactions*, 5(2):44–54, Mar 1998.
- [21] 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.
- [22] 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.
- [23] Frank L. Pedrotti and Leno D. Pedrotti. *Introduction to Optics*. Pearson Education LTD., 1993.
- [24] J Rekimoto and N Matsushita. Perceptual surfaces: Towards a human and object sensitive interactive display. *Workshop on Perceptual User Interfaces (PUI'97)*, Jan 1997.
- [25] Dan Saffer. *Interactive Gestures: Designing Gestural Interfaces*. O'REILLY, 2008.
- [26] B Shneiderman. Touch screens now offer compelling uses. *IEEE Software*, 8(2):93–94, Mar 1991.
- [27] Ben Shneiderman and Cathrene Plaisant. *Designing the user interface*. Addison-Wesley, fourth edition.

- 
- [28] 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.
- [29] Entertainment Design TED, Technology. Speakers Jeff Han: Human-computer interface designer, Sep 2008. [http://www.ted.com/index.php/speakers/jeff\\_han.html](http://www.ted.com/index.php/speakers/jeff_han.html), läst Sept. 2008.
- [30] P Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. *ACM UIST*, pages 27–33, Nov 1991.
- [31] Matthew Wright. Open Sound Control 1.0 Specification, Sep 2002. [http://opensoundcontrol.org/spec-1\\_0](http://opensoundcontrol.org/spec-1_0), läst Okt. 2008.
- [32] Matthew Wright, Adrian Freed, and Ali Momeni. Opensoundcontrol: Stateof theart 2003. *Proceedings of the 2003 Conference on New Interfaces for Musical Expression*, pages 154–159, May 2003.