

Documentatie multi-touchscreen TDI-case

Fuel 2.0

Leon de Jong
503733
V1TDI3

Inleiding

Er zijn 2 mogelijkheden: een informatieve applicatie of een spelconcept. Bij een informatief systeem zaten we met het probleem dat je mensen niet eenvoudig op een leuke manier van informatie kunt voorzien. Een spelconcept kan mensen wel entertainen (en dus aantrekken en 'vasthouden'), maar daarbij blijft de informatievoorziening toch een beetje achterwege. Na wat research kwam ik erachter bedrijven die een USP willen overbrengen door middel van een computergame daar vaak in falen. De game is of goed uitgebalanceerd en leuk om te spelen, maar maakt niet duidelijk welke USP de uitgevers van hun product willen overbrengen, of het hele concept is enkel gebaseerd op de informatievoorziening, wat als gevolg heeft dat de game saai, onspeelbaar of zelfs ronduit irritant is. Wij hopen deze 2 mogelijkheden dan ook op een leuke maar zinvolle manier te kunnen combineren. Niet door ze te fuseren in een geheel, maar door ze naast elkaar te laten bestaan in een enkele applicatie (interface) die als één geheel aanvoelt.

Algemeen

Auto

Als auto hebben we de McLaren F1 gekozen. Een van de meest pure en spartaanse productiewagens ooit, géén ABS, géén traction control, geen stuur- en rem bekrachtiging, maar wel zorgvuldig met de hand gebouwd uit duurzame materialen en hoogstaande technieken. De wagen is gebouwd met een enkel doel: presteren. Dit in tegenstelling tot bijvoorbeeld Italiaanse sportwagens als Ferrari en Lamborghini, overgewaardeerde super-sportwagens met een luxe interieur een gestroomlijnd uiterlijk en de wegligging van een stuk zeep. Sportwagens die als in 'overvloed' bestaan. Waar vind je een 2^e McLaren F1?

Concept

We willen een interface maken waarin diverse soorten media aanwezig zijn, zoals tekst, afbeeldingen, video, audio en eventueel games. Deze interface bestaat uit één grote pagina, meerdere malen zo groot als de eigenlijke schermruimte die we ter beschikking hebben. Dat betekent dat je maar een deel van de interface tegelijk ziet, en de focus dus op de inhoud van dat deel ligt. Je kunt door de interface navigeren door het te verslepen (drag en drop elementen). De interface bestaat zoals eerder gezegd maar uit één pagina, geen onduidelijke navigatie met links en terugwegen. Gewoon een pagina waar alles direct beschikbaar is. De interface wordt getoond op een touchscreen die we zelf gaan bouwen. Zo willen we de interface als het ware 'tastbaar' maken. Het zou helemaal mooi zijn als je het scherm niet eens aan zou hoeven raken, en je alles kunt sturen d.m.v. lichaamsbeweging, maar waarschijnlijk is dat iets teveel gevraagd.

Doelgroep

De doelgroepen die de AutoRAI komende beurzen meer wil aantrekken bestaan uit Jongeren onder de 18, allochtonen en 50-plussers. De auto die wij hebben gekozen is hier niet perse geschikt voor. Wij hebben dan ook geen specifieke doelgroep onder ogen, anders dan autoliefhebbers. (Wat heb je überhaupt op de AutoRAI te zoeken als je niet een beetje geïnteresseerd bent in auto's?).

Doelen

- De aandacht trekken met je installatie;
- Minstens één USP overbrengen van je auto merk / type;

Randvoorwaarden / mogelijkheden

De randvoorwaarden van dit project gesteld in de manual:

- Het eindproduct is óf een game, óf een interactieve (audio)visuele beleving.
- Bij de interactie wordt geen gebruik gemaakt van toetsenbord en muis. Het is wel mogelijk om een ander inputdevice via een toetsenbord of muis aan een computer te koppelen.
- De installatie bestaat uit een softwarematig en hardwarematig deel.
- De installatie gebruikt maximaal 5 bij 5 vierkante meter inclusief de ruimte die de bezoekers nodig hebben.
- De installatie mag gebruik maken van een projectiescherm en beamer.
- De installatie is geschikt voor grote bezoekersaantallen.

Soorten gebruikers

Er is slechts één soort gebruiker, en dat is degene die de interface bedient. Aangezien de applicatie maar een enkele auto tentoonstelt waar alle informatie inmiddels al over bekend is, hoeft de applicatie niet geüpdate te worden. Onderhoud is niet nodig. De applicatie draait hooguit enkele weken, een periode waarin slijtage niet zal voorkomen, mits er duurzame materialen gebruikt worden. Dit alles betekent dat een beheerder, in welke zin dan ook, niet nodig is.

Lijst functionele eisen

Wat natuurlijk sowieso moet werken is de (onder het volgende kopje beschreven) hardwarematige deel. Lijst van functionaliteiten ziet er als volgt uit:

Must haves

Info:

- Naam auto's
- Omschrijving auto
- Specificaties auto

Customize:

- Kleuren veranderen auto
- Specificaties veranderen auto

Media:

- Slideshow afbeeldingen auto

Should Haves

Info:

- Gewonnen awards autos

Media:

- Slideshow video auto

Could Haves

Customize:

- Uiterlijke kenmerken veranderen auto (anders dan kleur)

Media:

- Race- of andere game

Would Haves

Motion detection

Wireframe(s)

Het onderstaande wireframe toont de structurele opbouw van de applicatie. De bestaat uit informatie i.v.m. de auto. Deze informatie omvat minimaal de naam van de auto, de omschrijving en de specificaties.

Het 'customize' gedeelte bevat een of meerdere afbeeldingen van de auto. De rail onder dit gedeelte bestaat uit een kleurenwaaier, waarmee de kleur van de McLaren aangepast kan worden. Ook kun je kiezen uit de verschillende versies van de wagen en upgrades die hiervoor in de loop der tijd uitgebracht zijn. Aanpassingen aan deze gegevens hebben direct invloed op de weergave van de naam omschrijving en de specificaties die mee veranderen met je keuze.

Het 'pictures' gedeelte is simpelweg een slideshow van verschillende afbeeldingen de McLaren en diverse andere zaken die daarmee te maken hebben (bijvoorbeeld een focus op bepaalde bijzondere en/of belangrijke onderdelen). Vanuit de rail onder de weergave van de afbeeldingen kun je je keuze maken.

Het 'video' gedeelte is in wezen hetzelfde als het picture gedeelte met als verschil, inderdaad hoe kan het ook anders, video's i.v.m. met McLaren i.p.v. afbeeldingen.



De basiskleuren die we gaan gebruiken voor de applicatie zijn zwart/donkergrijs en oranje. Wanneer je kijkt naar de wireframe hierboven lijkt de afbeelding uit meerdere delen of 'secties' te bestaan. Dit is in wezen ook zo, maar het is de bedoeling dat alle secties naadloos in elkaar 'overvloeien' zonder dat de gebruiker door heeft dat en waar er een sectie begint of eindigt. Dit willen we verwezenlijken door middel van ons design. De secties, maar ook de inhoud hiervan zal ook niet rechthoekig zijn, maar kan elke vorm aannemen, simpelweg bepaald door het design. Voor meer informatie hierover verwijst ik naar Anwar.

Sitemap

Aangezien de verschillende secties/delen van de applicatie een naadloos geheel vormen en dus uit maar een pagina bestaan is een sitemap overbodig.

Flowchart

Hetzelfde geldt voor een flowchart. Mocht deze toch nodig zijn vind ik dit meer bij het interactie ontwerp horen (waar Ahsan mee bezig is).

Class Diagram

Hardware

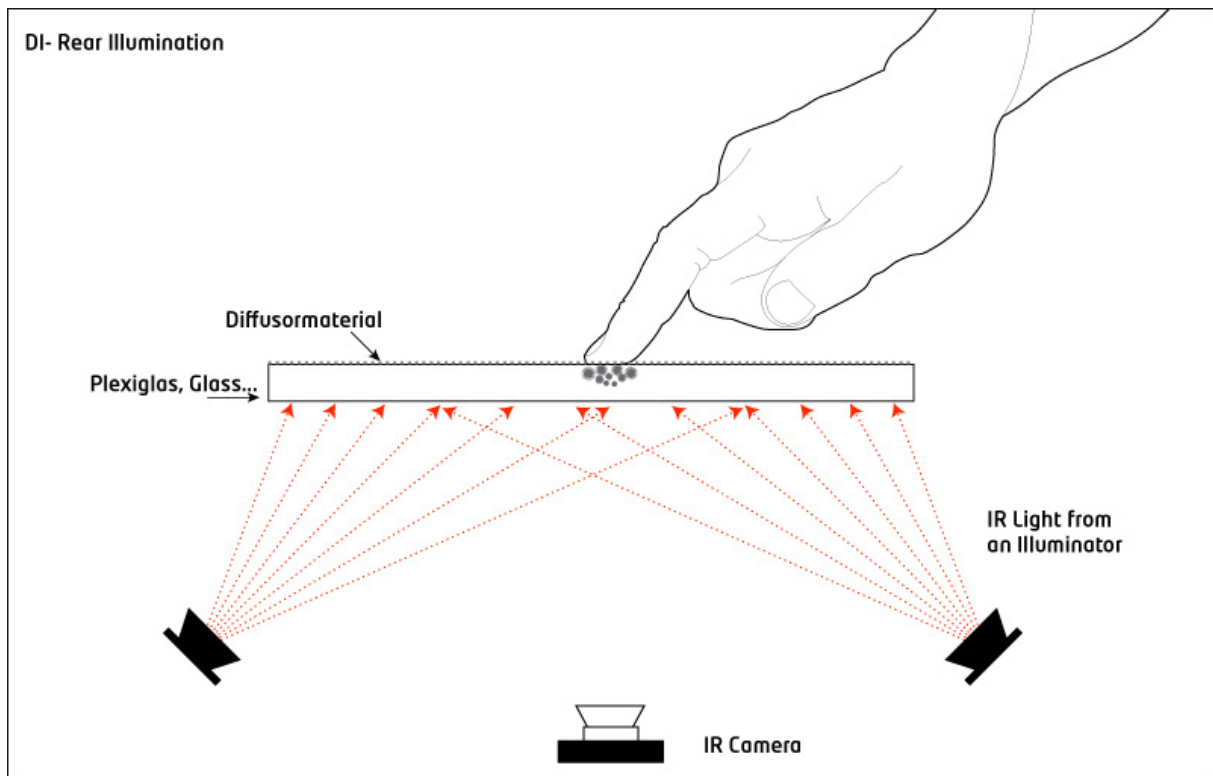
De hardware van het project bestaat uit een multi-touchscreen, welke we zelf gaan bouwen. Er zijn 2 soorten Multi-touchscreens welke met een relatief laag budget te bouwen zijn. Beiden zijn gebaseerd op infrarood licht en volgen min of meer hetzelfde principe. De benodigheden om één van beide schermen of zelfs een hybride versie te maken zijn dan ook vrijwel hetzelfde. Zie hieronder.

Benodigheden

- Transparante plexiglas plaat van minimaal 8 mm dik;
- Videoprojector;
- Webcam;
- Spiegel;
- IR emitters (LEDjes, Osram SFH485);
- Infrarood lens;
- Vloeibaar siliconen rubber / rubbermat o.i.d.;
- Weerstand;
- Materiaal voor frame (hout, metaal, kunststof, ...);
- Voeding;
- Computer;
- Projectiemateriaal (Rosco Grey, tracing paper, ...);
- Gereedschap (soldeerbout, boor, schroevendraaiers etc.).

Met deze benodigheden zijn dus 2 soorten schermen te maken: en DI (Diffused Illumination) en een FTIR (Frustrated Total Internal Reflection) scherm.

Diffused Illumination Multi-touchscreen

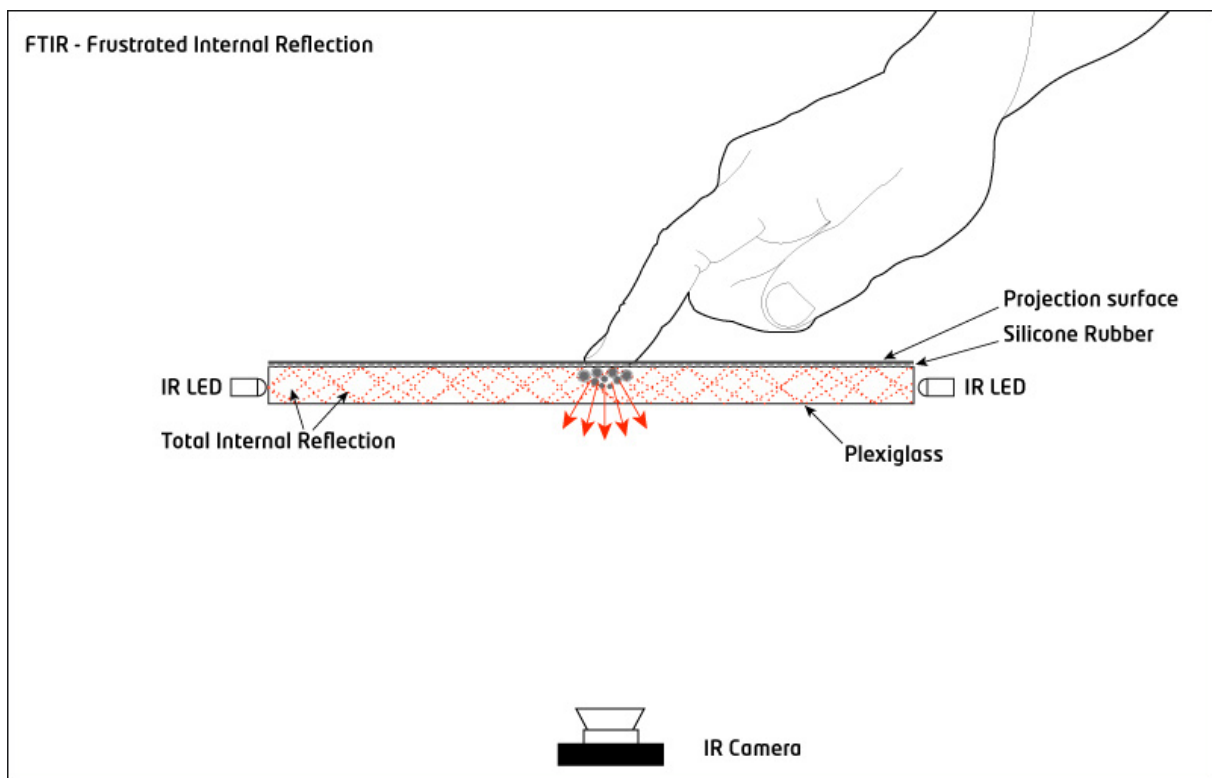


Een DI opstelling bestaat uit een plexiglasplaat waarover een projectiedoek of overtrekpapier gespannen is. Onder deze plexiglas plaat staan meerdere infraroodstralers opgesteld, dit kunnen bijvoorbeeld infrarood LEDjes zijn, maar ook andere infrarood emitters (bijv. een gloeilamp). Naast deze infraroodstralers is een webcam of een videocamera geplaatst, gericht op de plaat plexiglas. Het filter dat infrarood tegenhoudt in deze camera is vervangen door een filter dat infrarood licht juist doorlaat en zichtbaar licht tegenhoudt. Een videoprojector verzorgt ten slotte het beeld door deze, al dan niet via een spiegel, op het projectiedoek of doortrekpapier te projecteren.

Het infrarode licht, welke wordt afgegeven door de LEDjes of andere emitters, schijnt door het projectiemateriaal waardoor deze diffuus wordt. Wanneer een hand of voorwerp dus boven het oppervlak zweeft vangt deze relatief weinig IR-licht op. Dit verandert wanneer het oppervlak daadwerkelijk aangeraakt wordt. Voordat het IR-licht verspreid kan worden door het projectiemateriaal, kaatst het via je vinger terug naar de webcam, die dit vervolgens opvangt. Dit is te zien als (ronde) witte vlekken in een donkere omgeving, zogenaamde 'blobs'. Na diverse filters (bijvoorbeeld om de achtergrond zoveel mogelijk te verwijderen en het contrast van de afbeelding te verhogen) kunnen deze blobs verwerkt worden in de software die de coördinaten hiervan weer doorstuurt een applicatie. Deze applicatie zet deze data vervolgens om in beeld, welke via de videoprojector geprojecteerd wordt op het doek.

Door het gebruik van infraroodlicht IR-filters wordt de input (de blobs) van de output (het licht van de videoprojector) gescheiden.

Frustrated Total Internal Reflection Multi-touchscreen



Ook bij deze methode wordt de input en output gescheiden met behulp van infrarood licht en filters. Het verschil is dat de infrarood emitters (LED's) aan de zijkanten geplaatst zijn en het IR-licht zo dóór de plexiglasplaat zelf gaat. Omdat de dichtheid van de plexiglas plaat hoger is dan de lucht waar deze zich in bevindt, kaatst het licht van rand (overgang) naar rand heen en weer door de plaat. Zodra een vinger het oppervlak aanraakt wordt dit onderbroken, en kaatst het licht via de vinger in de webcam, om zo een 'blob' te produceren.

Zodra een projectiescherm of overtrekpapier geplaatst wordt op het plexiglas gaat dit effect verloren, omdat dit materiaal geen sluiting maakt wanneer je dit met je vinger op de plaat drukt. Omdat projectiemateriaal toch noodzakelijk is omdat je toch ergens op moet projecteren, wordt er een extra laag van ongeveer 1 mm dik tussen de plexiglas plaat en het projectiemateriaal geplaatst, bestaande uit siliconen rubber. (een z.g.n. 'compliant surface') Deze maakt wel luchtdichte 'sluiting' neemt de taak van je vinger over. Dit materiaal is helaas niet kant en klaar te krijgen en zul je dus zelf moeten produceren uit vloeibare componenten. Deze kosten aardig wat geld. Daarbij is het proces niet bepaald gemakkelijk en kunnen er makkelijk onvolkomenheden ontstaan in je materiaal (luchtbellen, het dof uitslaan, of oneffenheden).

FTIR versus DI

Voordelen FTIR

- Betere prestaties, blobs zijn beter te onderscheiden;
- Makkelijker werkend te krijgen dan DI als de setup eenmaal staat.

Nadelen FTIR

- Compliant surface vereist;
- Meer complexe constructie;
- Moeilijker te bouwen;
- Duurder.

Voordelen DI

- Meer eenvoudige constructie;
- Makkelijker te bouwen;
- Goedkoper;
- Geen compliant surface vereist.

Nadelen DI

- Minder goede prestaties, 'blobs' zijn minder goed zichtbaar;
- Moeilijker werkend te krijgen dan FTIR als de setup eenmaal staat.

Software

Er zijn 2 softwareprogramma's die met regelmaat worden gebruikt voor het interpreteren van de data ('blobs' of fiducials) vanuit de webcam.

- Touchlib (met name voor blobs);
- reacTIVision (met name voor fiducials, later meer hierover).

Deze worden vaak een of meerdere van deze programma's/talen gekoppeld voor de uiteindelijke applicatie/user interface:

- C, C++ of C#;
- OpenGL GLUT;
- Java;
- Flash (ActionScript);
- Processing;
- VVVV;
- OpenCV;
- Pure Data;
- Max/MSP;
- SuperCollider;
- Quartz Composer;
- Squeak (Smalltalk);
- Reaktor;
- Ableton Live;
- EyesWeb.

Van deze laatste programma's en talen weet ik niet van allemaal precies wat ze inhouden. De meeste heb ik wel in werking gezien in video's van andere mensen die bezig zijn met software voor een Multi-touchscreen. Wij gaan in ieder geval aan de slag met Touchlib en Flash. Touchlib omdat we met blobs gaan werken en waarschijnlijk niet met 'fiducials.' Flash, omdat dat vrijwel het enige programma is waar we enigszins thuis in zijn. Het nadeel is dat Flash een van de traagste oplossingen is.

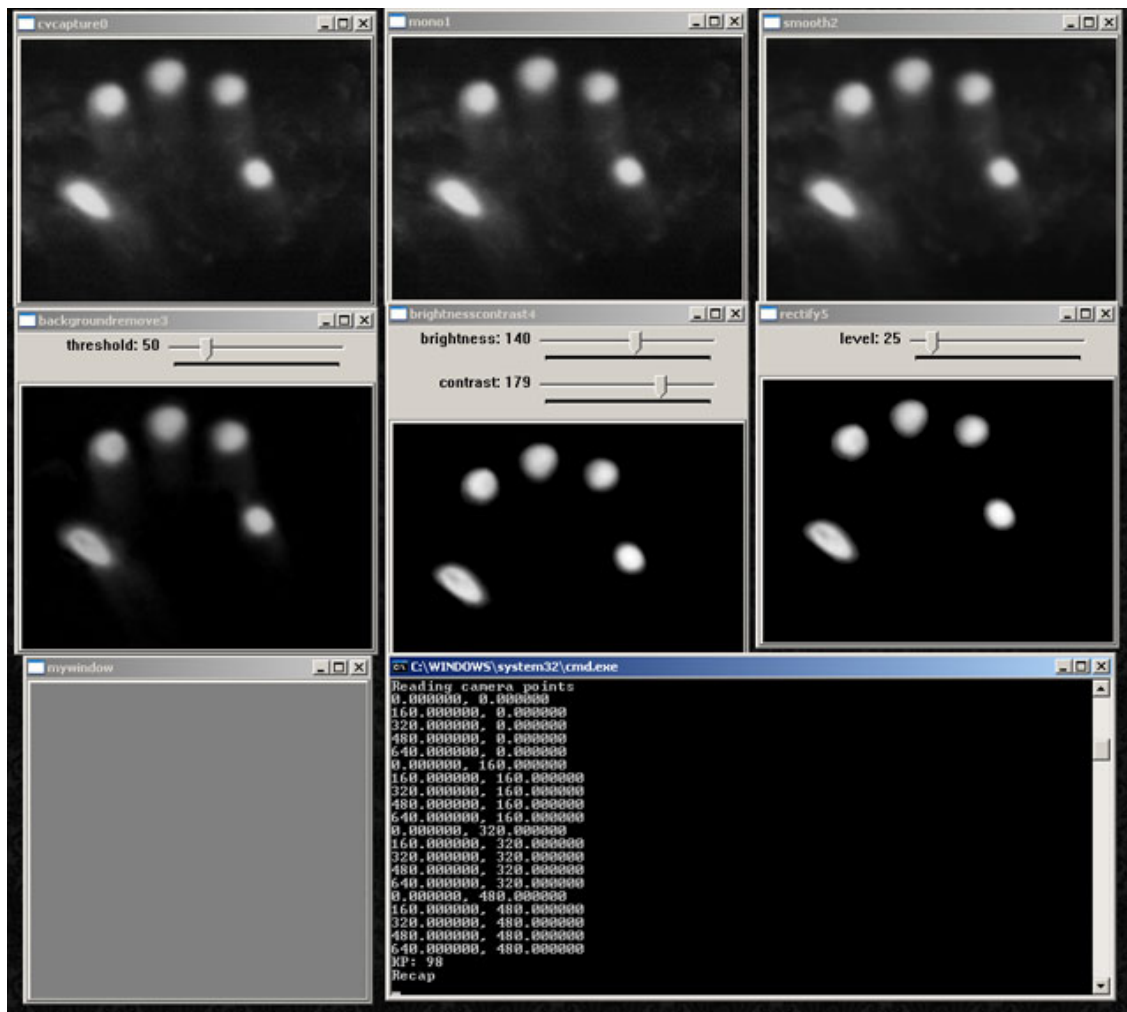
TouchLib

Dit programma verwerkt de 'blobs' die vanuit je webcam komen. De applicatie maakt gebruik van een aantal filters om de herkenning van 'blobs' te verbeteren. Deze filters zijn o.a:

- Backgroundremove (verwijdert de achtergrond, alleen nieuwe elementen in het beeld zijn nu nog zichtbaar, bijvoorbeeld de fluctuerende helderheid van de 'blobs');
- Brightnesscontrast (hiermee kun je de helderheid en het contrast van je plaatje aanpassen);
- Highpass (laat enkel signalen boven een bepaalde, vooraf ingestelde waarde door, wordt gebruikt om de vage delen en onscherpe randen in je plaatje te verwijderen);
- Invert (draait je plaatje om: zwart wordt wit en andersom);
- Rectify (verwijdert alle pixels onder een bepaalde vooraf ingestelde helderheidswaarde);
- Resize (verandert de resolutie van je plaatje);
- Smooth (gooit gaussian blur op je plaatje, waardoor de boel vervaagt);

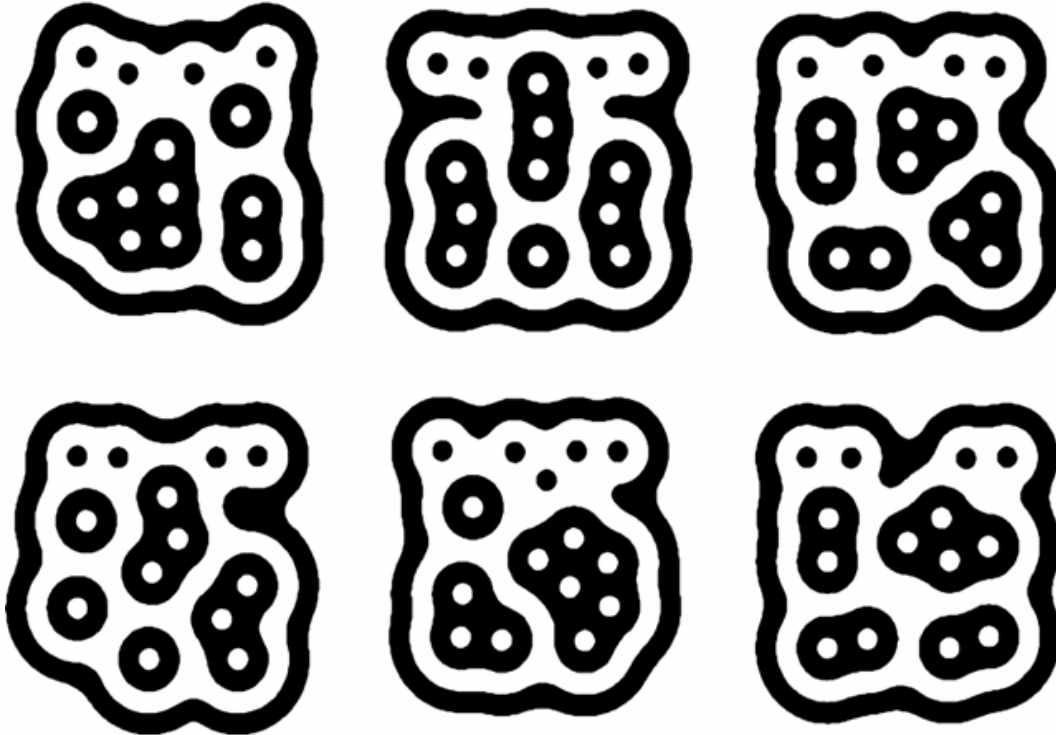
Na goed gebruik van een aantal van deze filters (al naar gelang je opstelling) blijft er als het goed is een zwart vlak over met witte rondjes op de plaats waar je het touchscreen aanraakt. De aanwezigheid, posities en bewegingen kunnen via OSC (Open Sound Control) met het TUIO (protocol voor tangible user interfaces) doorgestuurd worden naar andere applicaties om daar verder verwerkt te worden.

TouchLib ondersteunt ook patroonherkenning, alhoewel minder uitgebreid dan reactIVision.



reactIVision

Dit programma kan net als TouchLib gebruik maken van 'blob' detection, maar reactIVision is met name bedoeld voor patroonherkenning. De applicatie kan namelijk vast bepaalde patronen identificeren, zogenaamde fiducials. Zie de afbeelding hieronder.



reactIVision kan deze fiducials identificeren, de aanwezigheid ervan bijhouden, en de positie en rotatie ervan volgen. Al deze parameters worden standaard via OSC in het TUIO protocol gestuurd naar applicaties die dit protocol ook ondersteunen.

De connectie met Flash

Voor de connectie tussen TouchLib en Flash zijn in totaal 4 stukjes software nodig:

- TouchLib die blobs detecteert en verwerkt;
- OSC.exe die deze data van TouchLib omzet in het TUIO protocol en deze verstuurt via OSC op UDP poort 3333;
- FLOsc die het TUIO signaal op UDP poort 3333 omzet in een XML codering via een TCP poort (bijvoorbeeld poort 3000) zodat Flash de data kan benaderen;
- Flash Player 9 om de data op te vangen en om te zetten in een applicatie/GUI.

