

Beteckning: _____



Institutionen för matematik, natur- och datavetenskap

Visualisering av multidimensionella data med hjälp av parallella koordinater

Marcus Bergsten
juni 2009

Examensarbete, 15 högskolepoäng, C
Datavetenskap

Datavetenskapliga programmet, inriktning visiomatik
Examinator: Stefan Seipel
Medbedömare: Anders Hast

Visualisering av multidimensionella data med hjälp av parallella koordinater

av

Marcus Bergsten

Institutionen för matematik, natur- och datavetenskap
Högskolan i Gävle

S-801 76 Gävle, Sweden

Email:

Ndv06mbn@student.hig.se

Abstrakt

Problem består ofta av flera variabler som behöver sammanvägas för att vi ska kunna fatta ett beslut om vilket alternativ som är bäst. Detta gör vi var och varannan dag. Med detta arbete görs ett försök att skapa en integrerad webblösning som hämtar information och visualiserar den på ett sätt som förenklar ett beslutsfattande. Till hjälp för detta användes parallella koordinater.
Nyckelord: informationsvisualisering, parallella koordinater, webb intergrering

Innehåll

1. Inledning	1
1.1 Problem	1
1.2 Syfte	1
1.3 Frågeställning	1
1.4 Förväntningar	2
2. Teoretisk bakgrund	2
2.1 Tidigare forskning	2
2.2 Perspektiv och teoretisk ansats	4
2.3 Problemavgränsning	5
3. Metod och genomförande	6
3.1 Val av metod för visualiseringsdelen	6
3.2 Genomförande	6
4. Resultat	9
4.1 Scan.php	9
4.2 Show.html	9
5. Diskussion	12
5.1 Metod och genomförande	12
5.2 Resultat	12
5.3 Reflektioner	13
5.4 Slutsats	13
Bilagor	15
Bilaga A: scan.php.....	15
Bilaga B: Show.html.....	18
Bilaga C: Go.bat	19
Bilaga D: grantacc.txt	19

1. Inledning

I dagens samhälle är det svårt att undvika problem som kräver att man bedömer dem på flera parametrar. Det är sällan som vi har beslut som bara behöver jämföras på en parameter eller egenskap. När vi ska köpa en bil t.ex. är det sällan vi hittar två bilar som är exakt jämställda på alla punkter utom en. Till exempel två bilar som har gått lika många mil, är tillverkade samtidigt, har samma extrautrustning och grundalternativ, och endast skiljer markant på priset. I en sådan situation hade det inte varit svårt att göra en bedömning. Dessa typer av problem ställs vi inför nästan dagligen, dock inte alltid i svårighetsgraden eller samma konsekvenser som med att köpa en bil. Även mindre, vardagliga, problem kan vara svåra att överblicka och det är då vi behöver visuella hjälpmedel som underlättar beslutstagandet. Det är med denna grundtanke som målet är att undersöka om det går att skapa ett integrerat system som utnyttjar befintlig webbt teknik för att visualisera och stödja ett beslut adekvat.

1.1 Problem

Det finns flera problem att beakta innan arbetet sätts igång. Hur skall insamlandet av information göras? Vilken metod skall användas för att koppla denna information till visualiseringen? Och vilka tekniker finns som är avancerade nog för att kunna göra de saker som vi vill i form av visualisering?

Även rena visualiseringsproblem är tvunget att beaktas, finns det några andra tekniker än parallella koordinater som var värda att överväga som alternativ. Exakt hur skall visualiseringen se ut, fanns det några sidospår om modifikationer som kunde utforskas för att främja arbetet?

För att införskaffa informationen krävs en analys av hur ett par hemsidor på nätet (www.blocket.se, www.bytbil.se och www.mobile.se) i avseende på hur deras html sidor är utformade. Går det att utföra datamining eller informationsutvinning på dessa. Vilka parametrar kan vi utvinna från dem och hur är de i allmänhet uppbyggda tekniskt, går detta att utnyttja?

1.2 Syfte

Syftet med detta arbete är att undersöka huruvida ett integrerat system kan skapas som utnyttjar tillgänglig webbt teknik för att visualisera och stödja beslutstagande. För att möjliggöra detta måste även informationshämtningen undersökas och struktureras. En förutsättning för att kunna göra en utvärdering av de visuella teknikerna krävs att vi har bra data som skapar en stabil grund med bra representativa värden. Även en undersökning huruvida visualisering med hjälp av parallella koordinater kan vara ett alternativ till befintliga tjänster, som i dagsläget använder en mer traditionell teknik för vidarebefordran av information från säljare eller andra källor skall genomföras. Syftet är att demonstrera konceptet med parallella koordinater, genom att göra en fungerande demonstrator tillämpning.

1.3 Frågeställning

- Kan en lösning skapas som är en interaktiv visualisering av multidimensionella dataset och möjliggör selektering och utforskning av data för valfria attribut?
- Kan en sådan lösning integreras i ett webbgränssnitt?
- Hur kan information integreras från befintliga webbtjänster?
- Vilka verktyg eller mjukvarulösningar är lämpligast för visualisering?

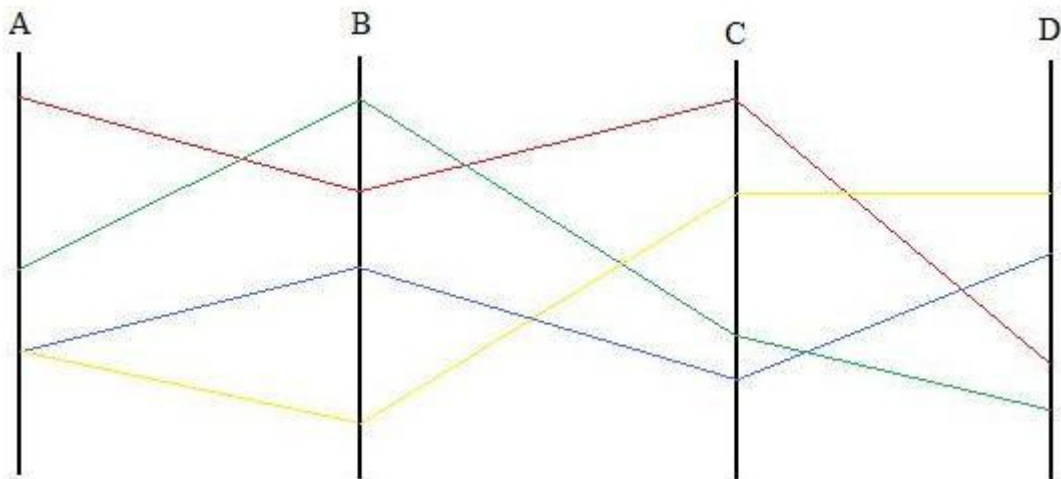
- På vilket sätt kan visualisering med hjälp av parallella koordinater underlätta förståelsen av komplexa data?

1.4 Förväntningar

Arbetet kommer att leda till en visualisering som kommer förenkla överblickandet av vald information. Denna lösning kommer att vara konstruerad så att den inte kräver ytterligare installation av speciell mjukvara på användarsidan, utöver de vanligaste mjukvaruprogrammen för webbläsare. Framst Java räknas och beaktas i denna bemärkelse som en nödvändig och normalt installerad mjukvara. Förväntningen är även att informationsutvinningen kommer vara ganska lätt att genomföra, även om sidornas kommer vara konstruerade för att försvåra åtkomsten till all deras information för att förhindra att externa sidor utnyttjar deras databaser.

2. Teoretisk bakgrund

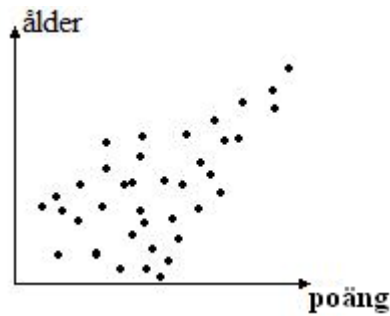
Parallella koordinater är ett välkänt och dokumenterat sätt att visualisera data av multidimensionell karaktär[1]. Den visar relationerna mellan två värden som representeras som punkter på axlarna A-D som sedan binds tillsammans med ett linjesegment, för var ny dimension som man vill ha med så lägger man till en axel i sin plot. Max och min för varje axel är normalt max och min för hela datasetet.



Figur 1. Ett enkelt exempel på parallella koordinater med 4 dimensioner.

2.1 Tidigare forskning

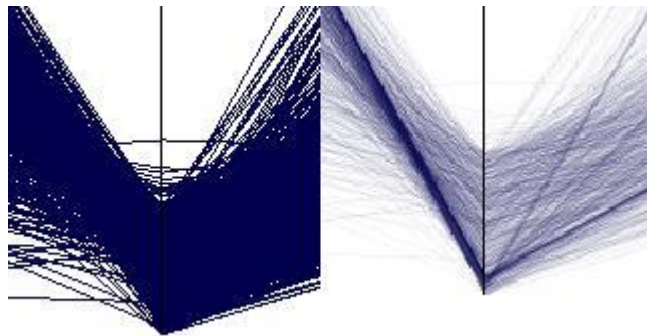
Artikeln av J Li, J-B Martens och J J van Wijk [2] gör en jämförelse mellan scatterplots, ett punktdiagram och parallella koordinater och kommer fram till några mycket intressanta slutsatser. De konstaterar att man kan urskilja fler nivåer av korrelation i en scatterplot än vad vi kan i en uppritning med parallella koordinater. De uppmärksammar även att trots att de test personer som de använde har fått bekanta sig med konceptet av parallella koordinater, och flera av användarna har tidigare erfarenhet av konceptet, så är det flera som tolkar de korrelationer som visas fel. En positiv korrelation ses som en avsaknad av samband, samt en avsaknad av korrelation kan blandas ihop med en negativ korrelation som har en viss spridning (figur4).



Figur 2. Ett exempel på en scatterplot som visar förhållandet mellan ålder och poäng i en tävling.

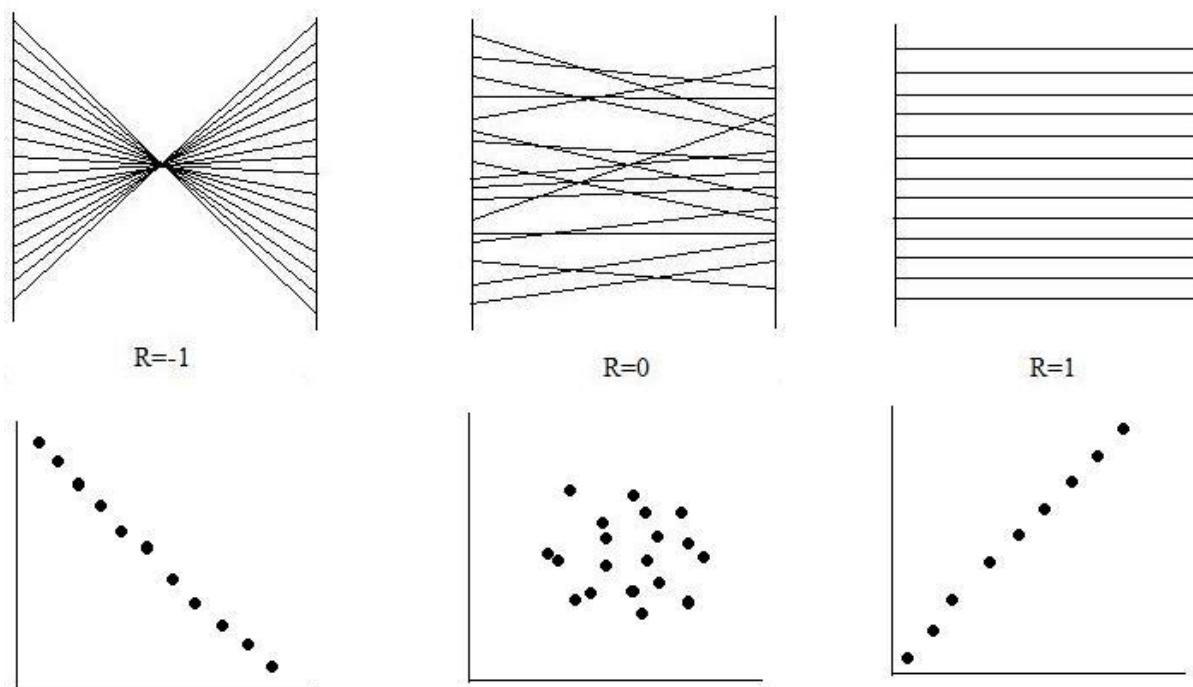
Ett annat problem är ”cluttering”, detta uppstår när vi har ett dataset som innehåller mycket data. När alla de objekt som ligger i datasetet ritas ut kommer man få linjer som ska visualiseras på en så liten yta att inget går att urskilja. En lösning som ofta används är blanding av de olika lagren så att man kan få en upplevd täthet i olika delar av ploten, detta gör att man ändå kan få en bra uppfattning om datastrukturen även vid stora datamängder.

Blanding är en teknik som används för att lägga ihop de olika komponenternas visuella bidrag till utritningen[4]. Det kan jämföras med att varje komponent ritas ut på ett OH-papper. Färgens genomskinlighet går då att reglera med att man väljer varje komponents alphavärde, vad som i exemplet skulle motsvaras av genomskinligheten på färgen på detta OH-papper. Den slutgiltiga färgen som målas ut på skärmen blir då den sammanlagda färgen av alla komponenter.



Figur 3. Bilden till vänster visar tung ”cluttering” och ingen blanding. Bilden till höger visar samma data, med mycket blanding.

Något som även läggs fram som mycket viktigt är möjligheten till interaktion. En av de naturliga nackdelar med parallella koordinater är att det är bara den korrelation som uppstår mellan två intilliggande värden som går att bedöma. Detta ger ju upphov till ett behov av att ha de värden man vill gör en bedömning på bredvid varandra. Om inte detta går, eller för att ge användaren möjligheten att själv bedöma vad som är som mest relevant, bör man kunna flytta de olika parametrarnas position för att kunna lägga dem i denna ordning.



Figur 4. Tre olika relationerna i parallella koordinater (ovan) och som scatterplots (nedan) från vänster till höger korrelationen: negativ, ingen och positiv.

Även en avgränsningsförmåga för att kunna begränsa ”träffarna” för att ge en mer detaljerad vy anses behöva upplevas som snabbt, det får inte hacka eller gå för segt. De förändringar som sker måste upplevas som momentana och kännas som det flyter.

Transferfunktioner istället för eller som komplement till vanliga blendingfunktioner är något som hjälper användaren att uppfatta de mönster som uppträder i datan. Dessa transferfunktioner skulle se till att blendingen inte har en rak funktion, utan en högre grad av genomskinlighet i de områden med mycket tung cluttering och mindre genomskinlighet i de områden där genomskinligheten bara skulle minska tydligheten. För stor genomskinlighet kan till och med resultera i att linjerna försvinner in i bakgrunden helt och hållet.

Det finns även andra tekniker som används, något som redan nämnts är scatterplots, som är en tvådimensionell uppritning av punkter. Man kan även projicera ner data till 2D eller 3D och då göra den möjlig att visualisera på mer traditionella vis. Glypher eller ikoner är även de ett sätt för att förmedla flera dimensioner av data, dessa kommer dock inte tas upp eller behandlas i denna uppsats.

2.2 Perspektiv och teoretisk ansats

Antalet bilar på en modern populär bilförsäljningssida kan i dagsläget komma upp i ett väldigt stort antal. Därför blir det ett måste att ha en metod som kan visualisera dessa bilar och därmed kunna ge en bra bild av vilket som är det bästa köpet när man selekterat sin grupp med bilar. När vi ska leta efter den bästa bilen så vill vi hitta bilar som går emot strömmen, om alla andra bilar har en positiv korrelation mellan hästkrafter och pris, vill vi hitta bilen som går mot strömmen och har en negativ korrelation. Detta borde vara tecken på ett bra köp. I andra fall där majoriteten av de övriga objekten har en negativ korrelation vill vi hitta positiva korrelationer. Målet blir att hitta en visualisering som stödjer detta och underlättar letandet.

2.3 Problemavgränsning

Den första delen av arbetet gick ut på att hitta en passande teknik och sedan inhämta informationen om hur det skulle appliceras i praktiken, processen krävde mycket tid och krävde att jag var tvungen att sätta mig in i först vad som var lämpligt att använda, för att sedan sätta mig in i exakt hur det skulle implementeras. Efterforskningen ledde till att det bestämdes att låsa det grafiska till att ha en fast uppställning av själva visualiseringen av parallella koordinater. Yttligare en avgränsning fick lov att göras, en egen bedömning av resultatet kunde bara göras då tidsbrist gjorde att någon kvantitativ undersökning inte gick att genomföra.

3. Metod och genomförande

Eftersom jag varit ensam i utvecklandet har det bestått av funderande och informationssökande på nätet, i böcker och artiklar. Den tekniska delen av arbetet utfördes med hjälp av en iterativ metod, där jag själv står för testning, utveckling, planering och utvärdering i varje steg. Återkommande avstämningar och diskussioner har fortgått genom arbetet.

3.1 Val av metod för visualiseringsdelen

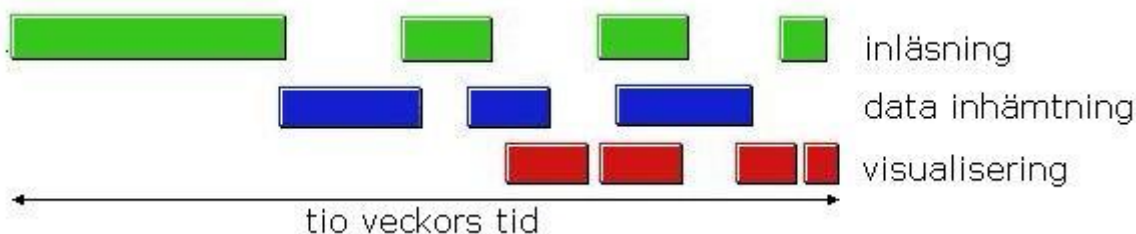
Valet att använda just parallella koordinater kändes naturligt, en av de stora fördelarna med en implementering av detta är att själva tekniken bakom det hela inte kräver mycket tekniskt eller programmeringsmässigt. För den enklaste formen av implementering behövs bara förmågan att måla ut linjer, medans flertalet alternativa tekniker kräver betydligt mera. Den är därför mycket lättimplementerad om man bara har vissa grundförutsättningar.

För att kunna se de tendenser och mönster som finns när vi tittar på den stora massan av data väljer vi att lägga en viss grad av opacitet på de data vi vill visa upp, det ger oss en bra bild av hur det ser ut och minskar den destruktiva effekten av den cluttering som uppstår. Genom att även visa vår selektion i en den vyn som normaliseras på selektionens max och min värden får vi en inzoomning på det vi gör, vilket gör det lättare att se detaljer på det axlar där mycket cluttering sker. Möjligheten att ta bort "förgrunden" ur ploten för att bara visa selektionsdata ger möjligheten att se den datan obehindrat.

På detaljnivå när man bara har 15 eller mindre objekt i sin selektion finns möjligheten att förtydliga varje objekt för sig. Genom att föra musen över de olika representationerna av objekten så förtydligas det objektet upp i ploten och genom detta ska man tydligare kunna se vilka som bryter trenderna hos datan.

3.2 Genomförande

Arbetet började med en planering samt informationsinsamlingsfas. Under denna tid undersöktes olika alternativa metoder för hur data skulle hämtas från respektive sidor. PHP och dess enkla sätt att koppla upp sig mot en annan sida visade sig mycket lämpligt. Dess syntax är även lätt att sätta sig in i och den har många bra funktioner för att modifiera och hantera strängar, vilket blir grunden till att utvinna informationen ur sidorna.

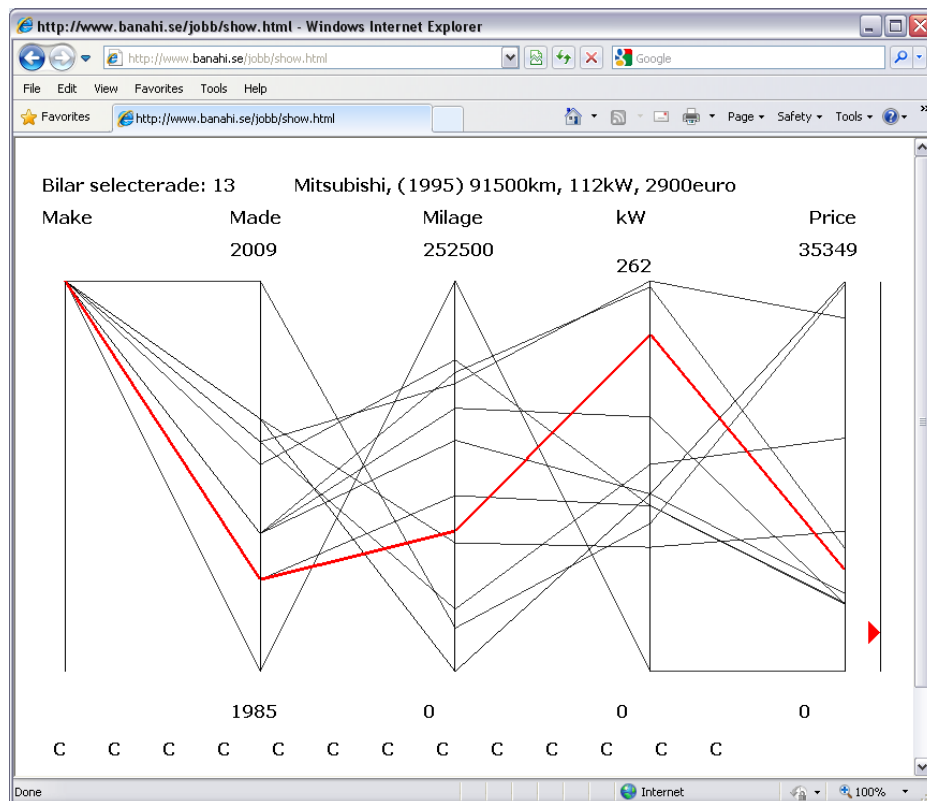


Figur 5. Tidsfördelningen under arbetet.

En första utgåva av parsingskoden konstruerades och testkördes mot sidan www.bytbil.com. Ganska snart skapades en funktion som hämtade in det data som gick att få tag på. En databas skapades i PHPmyadmin av ett mycket enkelt slag. Tabellen bestod av ett fält för varje parameter plus ett automatiskt ökande ID-fält, de första attributen som sparades var märke, tillverkningsår, rullade mil, färg och pris.

När denna del stod färdigt gjordes en enkel implementering av visualiseringen. Med hjälp av den informationssökning som gjorts togs beslutet att göra visualiseringen med hjälp av Javas openGL- implementering[9]. Till stor del för att openGL har goda möjligheter att utföra tekniska avancerade operationer, men även för att det är ett system som känns bekant och som jag har vana med. Det började med att bara måla ut de data som fanns, utan någon blanding eller några möjligheter till att selektera data. Arbetet i denna fas skede till största del med jGRASP[11], ett lättviktigt utvecklingsalternativ för java. Möjlighet till selektering arbetades efterföljande in i applikationen, när detta blev klart så lades även en möjlighet till blanding med kontrollslider till.

I nästa fas gjordes valet att omarbete inhämtningen så att den även sparade en länk till originalannonsern på nätet. Visualiseringen ändrades så att när man snävat åt selektionen så pass mycket att det bara var 15 bilar kvar, kom symboler för varje bil fram och när musen förs över dessa så förstärks den bil som symboliseras med dess motsvarande symbol, detta genom att de linjesegmenten gjordes tjockare och i en annan färg. En förklarande text dyker även upp för objektet.



Figur 6: en förtydligande vy av en markerad bil.

Vid denna tid innehöll databasen ett fält med attributet färg, detta är ett attribut som är svårt att visualisera med parallella koordinater, som bäst lämpar sig till användandet av värden på en ordinalskala, samt att färgen på en bil troligen inte är en viktig parameter i ett val inför ett köp. Färg var även en av de få värden som gick att få tag på från www.bytbil.se på grund av hur de valde att presentera data av bilar. Utläsning av andra parametrar än de nuvarande implementerade attributen skulle kompliceras mycket tack vare att det skulle kräva att man gick in på varje individuell annons istället för att läsa den från sidan där alla bilar listas. I detta skede togs beslutet att omarbete inläsningen av data och rikta den mot sidan www.mobile.de istället. Här fanns attribut som motoreffekt att tillgå och deras utbud av bilar var även mycket större. I samband med detta gjordes den synliga delen av scan.php om för att underlätta inhämtningen av data eftersom databasen behövdes göras om helt med de nya fälten. Efter att man skickat in en sida till parsningsfunktionen laddas den följande sidan med

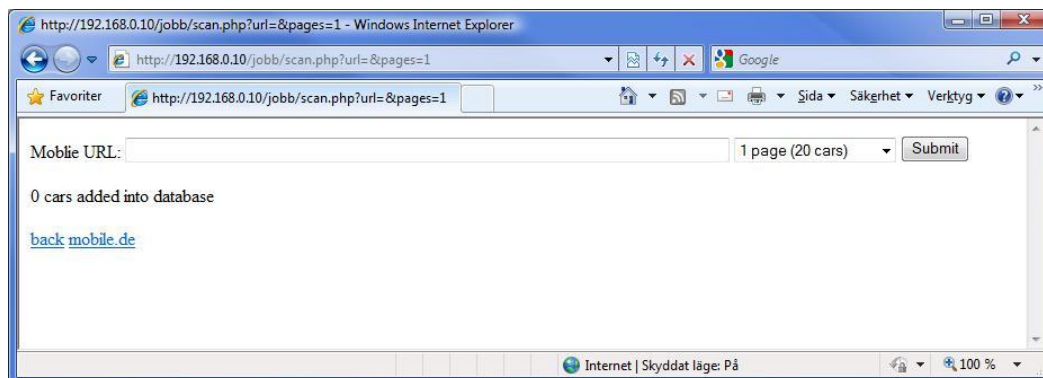
den nästkommande sidan i ordningen som standardvärde. Så att man direkt kan klicka på knappen submit och nästa sida läses av.

Vissa delar av visualiseringen arbetades här om och i ett försök att förbättra upplevelsen av datan och försöka förtydliga de områden där cluttering framkom genom att införa linked views. I bakgrunden visades då inte den nertronsade representationen av de bortvalda objekten. Istället visades en version av de selekterade objekt, som normaliserats mot de max och min värden som fanns i det nya datasetet som utgjordes av dessa bilar.

Detta är den version som i nuläget är den gällande.

4. Resultat

Sidan är uppbyggd med en minimal indexfil som länkar till en PHP sida[4] (scan.php) och en vanlig htmsida(show.html), Denna innehåller ett javascript som kör java webbstart[6] med hjälp av en launcher gjord för just detta syfte[7].



Figur 7. En skärmdump från scan.php

4.1 Scan.php

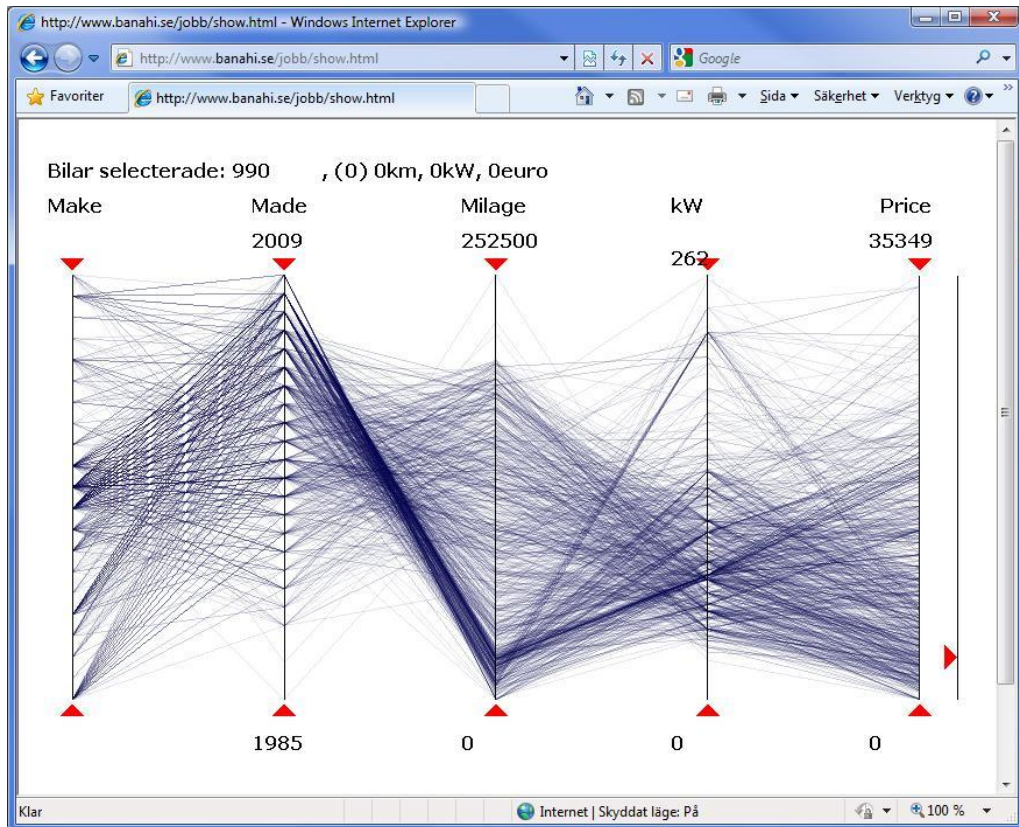
Scan.php (bilaga 1) har till syfte att ta emot en URL, från sidan mobile.de. Detta är en stor och populär sida för försäljning av bilar i Tyskland. PHPscriptet tar här en länk som är den adress som fås vid en sökning och hämtar hem det data som finns på denna sida. Efter detta parsas(syntastiksanalys, letandet efter en sekvens av tecken, i detta fall ord) denna info till ett märke, tillverkningsår, antal mil, motorstyrka i kW och pris. När dessa data har hittats skickas den upp till en MySQL[8]databas som i sin tur lägger till ett ID som blir dess primärnyckel i den relationsdatabas som skapats i projektets syfte. Man har här även ett val att parse igenom mer än en sida från mobile.de åt gången. Om man i switchboxen väljer ”2sidor(40) bilar” kommer scriptet, efter den sidan man lagt till i URL boxen parsats, att parse den följande sidan. Efter egen erfarenhet kan detta bli problem om detta görs på en dator med dålig uppkoppling eller om det av någon anledning är stor fördröjning i kommunikationen mellan servern som kör PHP-scriptet och mobile.de. Detta på grund av att PHP kastar ett exception om den exekveringstid är längre än 30 sek, vilket den lätt kan bli under de tidigare nämnda förutsättningarna och många valda sidor. När en sökning av en länk är slutförd korrekt visas de tillagda bilarna och en summeringsrad som talar om hur många bilar som lagts till. På sidan finns även en länk till mobile.de för smidighet. Länken som skickas in från mobile.de måste innehålla ”pageNumber=” följt av ett nummer, detta görs lättast genom att man klickar på sida två i den följd av sidor som blivit sökresultatet.

4.2 Show.html

Den andra stora delen av sidan är grundad i show.html (bilaga 2). Här initieras det applet som ska via JOGL[9] rita upp våra parallella koordinater. Här laddas diverse hjälpbibliotek för att kunna köra vår appletkod, bland annat stödfunktioner för att kunna skapa en koppling till vår MySQL server m.m. Javaprogrammet som sköter renderingen består av två klasser plus en internklass i Pcpplot.java (java koden finns att tillgås på www.banahi.se/jobb för en kortare framtid, därefter går den att få skickad till sig på begäran).

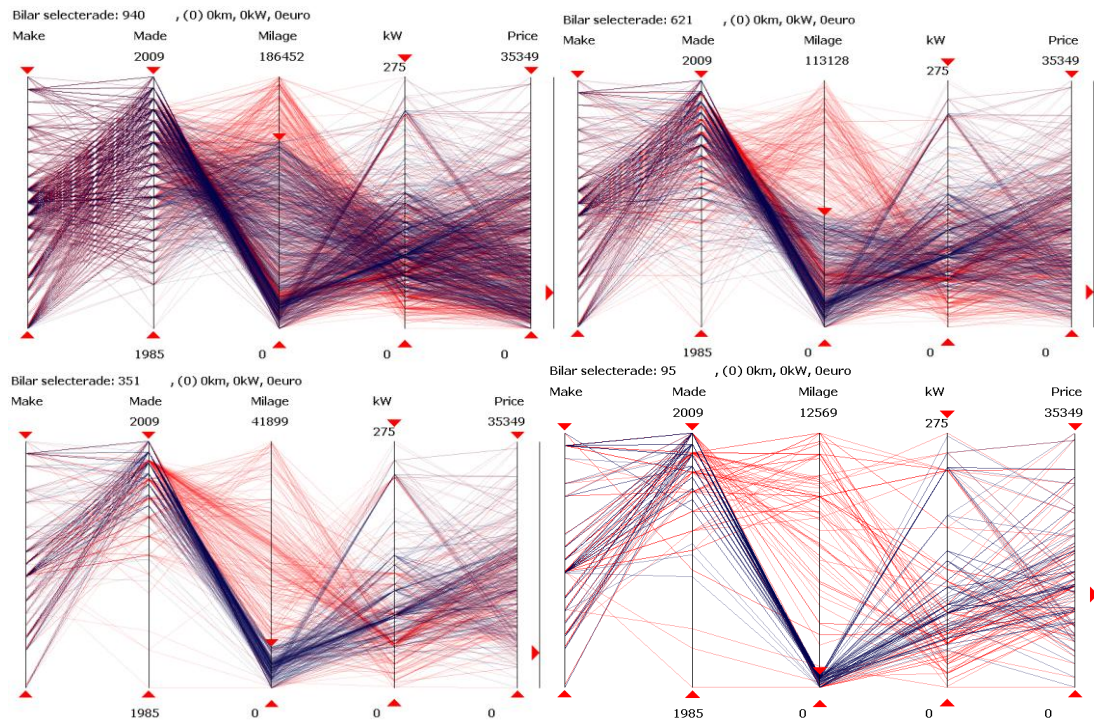
Klassen PcpApplet agerar applet och initierar det som ska ritas ut. Klassen skapar en instans av Pcpplot. Pcpplot i sin följd startar en tråd och använder sig av OpenGL för att rita ut våra parallella koordinater. Det första som sker är att en koppling görs till MySQL servern och all

data hämtas och läggs i en länkad lista. För att detta ska fungera när applikationen körs som applet måste rättigheter sättas för att lov ska ges till en socketkoppling. Tyvärr används en bat-fil och text fil (bilaga 3 och 4) för att ge appleten fulla rättigheter. Skulle man använda detta på en kommersiell sida skulle man ge bara de rättigheter som behövs för att minimera säkerhetsriskerna. Kopplingen till SQL databasen måste alltså ske över internet då javaapplikationen räknas som den körs lokalt, men de data som används ligger lokalt under körningen. När väl datan är inläst visualiseras den utefter de 5 parametrarna.



Figur 8. Show.html i sitt grundläge

Den finns markörer för max och min värde som skall visas på varje axel. Dessa kan flyttas för att selektera delar av data. När ett urval av data är selekterat data ritas detta ut i bakgrunden i röd färg fast normaliserad mot max och min på sin selektion istället för mot det max och min värde som den totala data volymen har (se figur 8). För att kunna förbättra resultatet så finns även en slider för att kunna reglera alphavärdet för varje linjesegment. I det översta vänstra hörnet finns en rad som talar om hur många bilar man har selekterat just nu. Klickar man på ordet "Bilar" här så tas förgrunden ut så att man lättare kan se enbart sin selektion. När man avsmalnat en selektion så pass mycket att man bara har 15 eller mindre bilar kvar dyker det upp en rad med "C" under ploten. Genom att hålla sin muspekare över ett av dessa C så får man dels upp dess info i raden överst, dessutom markeras den valda bilen för att ytterligare underlätta jämförelse med de andra bilarna. väljer man sedan att klicka på C tas man till den annons som representeras. Dock så kommer inte databasen vara dagsaktuell så länge inte visualiseringen är direkt kopplad till mobile.de's egen databas vilket gör att bilar som finns i visualiseringen inte nödvändigt finns kvar som annons.



Figur 9: en selektion av bilar med lågt milantal.

För att underlätta utvecklandet användes en vanlig text editor och verktygen wampserver[10] som är en samling av apache, MySQL och PHP.

Inga utav de sidor som skrivits är validerade eller följer rådande konventioner för html.

5. Diskussion

I min mening är den största generella bristen hos parallella koordinater är att man måste bekanta sig ganska mycket med det för att kunna få en intuitiv känsla av vilka objekt som är ”bra” och ”dåliga”. Efter att ha suttit många timmar med att titta på det i olika former och i olika sammanhang så känns det inte som man får en känsla för hur resultaten är genom en snabb överblick såsom man kanske skulle få med en vanlig graf. Kanske beror detta på att man är så van vid alla de klassiska formerna av presentation av data i 2 eller på sin höjd 3 dimensioner. Eller så är detta något man får stå ut med när man vill visualisera något med flera dimensioner. Men jag tror inte att en vanlig bilköpare skulle ta sig tid till att lära sig och kunna uppskatta de fördelar som parallella koordinater erbjuder. Det känns helt enkelt något som gemene man inte skulle ta till sig. Inte förrän det blivit vanligare i vår vardag eller på våra arbetsplatser.

Tekniken ger dock en möjlighet att titta på en större volym, se vilka trender som finns och se vart det finns mycket att välja mellan. Här kan man nog se bilar som man inte annars skulle se att de fanns. Man får en bil som ligger precis utanför ramarna för det man skulle vilja se hos en framtida bil, kanske skulle denna väljas bort om man tog ett fast värde i en drop-down list. Även när man har smalnats av sin sökning så ger parallella koordinater en bättre överblick. Det man skulle vilja visualisera på ett klarare sätt är ”trendbrott”, i jakten på det perfekta bilköpet är man inte ute efter att se vilken korrelation som finns mellan hästkrafter och pris. Eller år och mil. Man ville se bilar som sticker ut. Hitta den där bilen som är som alla andra i mil och pris och hästkrafter men som ligger de där tusenlapparna lägre än alla andra.

5.1 Metod och genomförande

Utvecklingsmetoden för datautvinningen känns som den enda vettiga för ett arbete av denna storlek där man jobbar själv. Valet att utnyttja JOGL känns som ett bra val, OpenGL blir ett kraftfullt hjälpmedel, och under utvecklingens gång har jag varit överraskad vid ett flertal tillfällen över hur snabbt och smidigt alla funktioner utförs. Detta beror mycket på att man kan utnyttja så pass mycket av datorns kraft när vi har möjlighet att sköta utritandet accelererat. En svaghet i visualiseringsmetoden är helt klart att den inte kan anpassas dynamiskt. Praktiskttaget alla funktioner i programmet är hårdkodade och om bara det minsta i programmet ändras så behöver flertalet omständiga omdefinitioner göras av parametrar i programmet. Den sista utgåvan av visualiseringen känns som ett steg bakåt. Den variant av programmet som fanns innan var tydligare, att två dataset ritas ut på samma yta ger för mycket cluttering trots att det ritas ut i olika färger. Något som hade varit önskvärt var att ha ett utritningsområde som var separerat från området där selektionen skedde. Tack vare att programmet var så pass statiskt kodat hade detta betytt att för mycket av koden hade behövts göras om i en för sen stund i utvecklingen. Av samma anledning skulle en omstrukturering av informationen på mobiles sida vara förödande för informationsutvinningen, denna bygger på att sidan ser ut just som den gör vid detta tillfälle.

5.2 Resultat

Målet att lyckas göra en webbaserad lösning på problemet är löst tillfredställande, att användaren måste ha java installerat känns rimligt. Det känns som den rent tekniska biten funkar bra. Java appleten fungerar oväntat bra prestandamässigt, selektionen sker fort och känns inte som man behöver vänta på resultatet när man flyttar min- och maxvärden. En automatisk förändring av alphavärdet skulle kanske varit en trevlig implementering. När antalet visade objekt sjunker försvinner nästan linjerna om man inte ändrar opaciteten.

Det känns även som de röda datasetet som visas i bakgrunden blir aningen svårtydd och otydliga och i många fall skymda av förgrunden, vilket leder till att man inte kan se det så väl som skulle önskas. Även om en simultan vy av de båda är bra så skulle det gynnas mer om man hade haft separata vyer. En vy med originaldata som även visar den bortvalda datan i en transparent bakgrund. Och en vy där man ser bara valda data som anpassas.

Något som skulle förstärkt visualiseringen skulle varit om man implementerat axlar som går att byta plats på. Den nuvarande fasta konfigurationen kanske inte är den bästa. Att låta användaren själv välja i vilken ordning de skall ligga skulle bidra till att man får en bättre förståelse vad data visar. Och även hjälpa till att tyda vissa data.

En av de problem som inte gått att lösa är när en klickning på ett enskilt objekt gjorts och man ska slussas till den enskilda bilens annons. Här skulle en bättre lösning vara att man kom till ett nytt fönster men detta verkar inte stödjas på något sätt som hittats. Detta gör att om man hittar en bil man vill se annonsen till och sedan ska gå tillbaka till sitt ursprungsdata har eventuella urval av data försvunnit.

5.3 Reflektioner

Att genomföra en användarstudie där insamling av kvantitativ och kvalitativ data insamlats hade varit mycket intressant. För att kunnat genomföra detta hade dock den praktiska biten behövt vara mycket längre gånget i utvecklingen tidigare i arbetets gång. För att kunna göra test om huruvida man kan hitta ”den perfekta bilen” så hade en genomgång och plantering av manipulerad data behövts göras, vilket heller inte tiden räckte till för.

5.4 Slutsats

Visualiseringen känns lyckad, dock skulle en linked view som med separata utritningsområden varit ett bättre alternativ. Det finns klara fördelar med visualisering med parallella koordinater mot ett traditionellt alternativ. Det krävs troligen dock att metoden presenteras och introduceras på ett pedagogiskt sätt för att allmänheten skulle acceptera den som ett alternativ. Att göra en integrerad webblösning finns det uppenbarligen inte några hinder för. Lite synd är det att det olika tekniker kunnat utforskats, då det i nuläget endast kan spekuleras i huruvida det valda sättet är det bästa.

Referenser

- [1] Alfred Inselberg. The plane with parallel coordinates. *Visual Computer* 1(4): sidorna 69-91. 1985.
- [2] Jing Li, Jean-Bernard Martens, Jarke J van Wijk. Judging correlation from scatterplots and parallell coordinate plots. *Information Visualization* 1-18. 2008
- [3] Jimmy Johansson. Efficient Information Visualization of multivariate and Time-varying data, LiU-Tryck, Linköping 2008.
- [4] Schroeder Will, Martin Ken, Lorensen Bill. *The Visualization Toolkit, fourth Editio*. Kitware Inc. Colombia 2006.
- [5] *PHP: Hypertext Preprocessor*, www.php.net (2009-05-24)
- [6] *Java Web Start Techonlogy*.
<http://java.sun.com/javase/technologies/desktop/javawebstart/index.jsp> (2009-05-24)
- [7] *JNLP-based applet launcher*. [https:// applet-launcher.dev.java.net](https://applet-launcher.dev.java.net) (2009-05-24)
- [8] *MySQL*. www.mysql.com (2008-05-24)
- [9] *Java OpenGL*. <https://jogl-demos.dev.java.net/> (2008-05-24)
- [10] *Wampserver*, <http://www.wampserver.com/> (2008-05-24)
- [11] jGRASP, <http://www.jgrasp.org/>, (2008-05-28)
- [12] Robert Spence, *Information Visualization*, Addison Wesley, London 2000

Bilagor

Bilaga A: scan.php

```
<head>
</head>
<body>
<form name="input" action="scan.php" method="get">
Moble URL:
<input type="text" size="85" name="url" value="<?php
    $temp = $_GET['url'];
    if(strlen($temp)>5){
        $nr = (int)substr($temp, 64, 2);
        $nr++;
        if($nr<11)
            $next = substr_replace($temp, $nr, 64, 1);
        else
            $next = substr_replace($temp, $nr, 64, 2);
        echo $next;
    }
    else
        echo "";
?>" />

<select name="pages">
<option value=1>1 page (20 cars)</option>
<option value=2>2 pages (40 cars)</option>
<option value=3>3 pages (60 cars)</option>
<option value=4>4 pages (80 cars)</option>
<option value=5>5 pages (100 cars)</option>
<option value=6>6 page (120 cars)</option>
<option value=7>7 pages (140 cars)</option>
<option value=8>8 pages (160 cars)</option>
<option value=9>9 pages (180 cars)</option>
<option value=10>10 pages (200 cars)</option>
</select>

<input type="Submit" value="Submit" />
</form>

<?php

$sqlserv = mysql_connect('localhost','root','fisk');
mysql_select_db('exjobb', $sqlserv);

$cars_added =0;

$url = $_GET['url'];
$pages = $_GET['pages'];
scanmore($pages, $url);

echo $cars_added . " cars added into database";

function scanmore($sidor, $adress){
    if(strlen($adress)>10){
        $page = (int)substr($adress, 64, 2);
```

```

        for($i = $page; $i < $sidor+$page ; $i++){
            if($i<11)
                $newadd = substr_replace($adress, $i, 64, 1);
            else
                $newadd = substr_replace($adress, $i, 64, 2);
            scanbytbil($newadd);
            //echo $newadd . "<br>";
        }
    }
}

function scanbytbil($adress){
    $url = $adress;

    global $cars_added;

    $page = @fopen($url, "rt");          //öppnar sidan

    $replacements = array(" ", ".", " ", "Â", ","); //Chars som jag vill plocka bort i strängen senare för
att underlätta parsning

    //märke
    $car_add_start = 'td class="description"';

    $car_url_str = 'href="';
    $car_url = "";
    $car_make = "";
    $car_kw = "";
    $car_made = "";
    $car_milage = "";
    $car_price = "";

    for ($i = 1; $i <= 1000; $i++) { //går ner 1000 radet i dokumentet
        $source_code = fgets($page);
        //hämtar nästa rad
    }

    while($source_code = fgets($page)){          //hämtar nästa rad
        if(strpos($source_code, $car_add_start)== true){
            $source_code = fgets($page);
            $source_code = fgets($page);
            $source_code = fgets($page);
            $source_code = fgets($page);
            $source_code = fgets($page);
            $source_code = fgets($page);
            $source_code = fgets($page);
            $car_url = substr($source_code, 31, strlen($source_code)-49);
            $source_code = fgets($page);
            $source_code = fgets($page);
            $tempstring = substr($source_code, 0, strlen($source_code)-1);
            $car_make = str_replace($replacements, "", $tempstring);
        }

        if(strpos($source_code, "kW")== true){
            $max_string_length = strlen($source_code);
            for($o = 0; $o <= $max_string_length; $o++){
                if(substr($source_code, $o, 1)==" "){
                    $car_kw = (int)substr($source_code, 0, $o);
                    break;
                }
            }
        }
    }
}

```

```

//      echo "kw:";
//      echo $car_kw;
}

if(strpos($source_code, "registration")== true){
    $car_made = substr($source_code, 38, 43);
    //      echo "made:";
    //      echo $car_made;
}

if(strpos($source_code, "mileage")== true){
    $tempstring = "";
    $max_string_length = strlen($source_code);
    for($o = 25; $o <= $max_string_length; $o++){
        if(substr($source_code, $o, 1)=="&"){
            $tempstring = substr($source_code, 25, $o);
        }
    }
    $car_milage = (int)str_replace($replacements, "", $tempstring);
    //      echo "milage";
    //      echo $car_milage;
}

if(strpos($source_code, "priceBlack")== true){
    $tempstring = "";
    $max_string_length = strlen($source_code);
    for($o = 25; $o <= $max_string_length; $o++){
        if(substr($source_code, $o, 1)==" "){
            $tempstring = substr($source_code, 25, $o);
        }
    }
    $car_price = (int)str_replace($replacements, "", $tempstring);
    echo $car_make . " " . $car_made . " " . $car_milage . " " . $car_kw . " " .
$car_price . "<br>";

//echo $car_price;
$sql = "INSERT INTO data SET
id = NULL,
märke = '$car_make',
år = '$car_made',
mil = '$car_milage',
kw = '$car_kw',
url = '$car_url',
pris = '$car_price'";
if(mysql_query($sql)) {
    $cars_added++;
} else {
    echo "error adding car : " . mysql_error();
    echo "<br>";
}
}

if(strpos($source_code, "</html>")== true)
break;
}
}
//skicka all data !!!
?>

```

<p>back mobile.de</p>

Bilaga B: Show.html

```
<applet code="org.jdesktop.applet.util.JNLPAppletLauncher"
  width=800
  height=600
  archive="http://download.java.net/media/applet-launcher/applet-launcher.jar,
    http://download.java.net/media/jogl/builds/archive/jsr-231-webstart-current/jogl.jar,
    http://download.java.net/media/gluegen/webstart/gluegen-rt.jar,
    http://banahi.se/jobb/mysql-connector-java-5.1.7-bin.jar,
    http://banahi.se/jobb/pcp.jar">
  <param name="codebase_lookup" value="false">
  <param name="subapplet.classname" value="PcpApplet">
  <param name="subapplet.displayname" value="Parallel Coordinate Plot Applet">
  <param name="noddraw.check" value="true">
  <param name="progressbar" value="true">
  <param name="jnlpNumExtensions" value="1">
  <param name="jnlpExtension1"
    value="http://download.java.net/media/jogl/builds/archive/jsr-231-webstart-current/jogl.jnlp">
</applet>

<p><a href="index.html">back</a></p>
```

Bilaga C: Go.bat

```
appletviewer -J-Djava.security.policy=grantacc.txt show.php
```

Bilaga D: grantacc.txt

```
grant {  
    permission java.security.AllPermission;  
};
```