ISSN 0280-5316
ISRN LUTFD2/TFRT--5752--SE

EasySee – Ett verktyg för visualisering av mätdata från klimatprovning

Matilda Gustafsson
Emma Jönsson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Juni 2005

Department of Automatic Control
Lund Institute of Technology
Box 118
SE-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
June 2005

Document Number
ISRNLUTFD2/TFRT--5752--SE

Author(s)
Matilda Gustafsson and Emma Jönsson

Supervisor
Mats Andersson at Volvo in Gothenburg
Rolf Johansson Automatic Control in Lund

Sponsoring organization

Title and subtitle
EasySee – Ett verktyg för visualisering av mätdata från klimatprovning (EasySee – A Tool for Visualization of Data Obtained During Climate Testing).

Abstract
During the ongoing development of new car models, testing and simulation are used for validation, which involves a large amount of measured data being sampled. To use the tests and simulations in the best way possible, the data collection needs to be effectively worked on. Currently, the evaluation of sampled data is done primarily through studies of numerous parameters in graphs. To avoid problems choosing the appropriate parameters to study, a graphic representation of the system's functions can be used. The purpose of the Master's project is to develop a graphical interface, which visualizes data sampled during the development of a climate system and thereby eases the evaluation. The visualization is enabled through a method that visualizes a selection of measured data over a surface in an easy-to-grasp way. A model is developed to calculate temperatures in all places in the compartment as well as in parts of the climate unit. These temperatures are represented with colors. The data is primarily sampled in the car model Volvo XC90.The visualization tool EasySee is implemented in MATLAB 7.0 release 14 and Java 1.4.2 on a Windows platform. The interpolation is done by MATLAB with Java responsible for the graphical drawing in the user interface and also the user interaction. This Master's Thesis covers the development process of the visualization tool EasySee a Master's project presented by Volvo Technology Corporation, department 6280 Control and Simulation, in Gothenburg. The Master's Thesis consists of the following main chapters: prerequisites, theory of analyzing sampled data and graphical user interfaces, method of development, results and guidelines for future development. This Master's project resulted in an application with excellent opportunities to ease the daily work analyzing sampled data.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
Swedish

Number of pages
120

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through:University Library, Box 3, SE-221 00 Lund, Sweden Fax +46 46 222 42 43

Sammanfattning

Vid den ständiga utvecklingen av nya bilmodeller använder man sig av provning och simulering för validering av system. Detta innebär att en stor mängd mätdata samlas in och för att utnyttja provningarna och simuleringarna maximalt krävs att denna datamängd bearbetas effektivt.

Utvärderingen av mätdata görs i dagsläget främst genom att granska ett hundratal parametrar i graf form. För att undvika problem med att välja rätt parametrar att studera så kan istället en grafisk representation av systemets funktion utnyttjas.

Syftet med examensarbetet är att ta fram ett grafiskt gränssnitt som visualiserar data från utveckling av klimatsystem och därmed underlätta utvärderingen av dessa data. Visualiseringen möjliggörs genom att en metod utvecklas för att visualisera punktvisa mätdata på ett överskådligt sätt över en yta. En modell tas fram för att interpolera temperaturer för samtliga punkter i kupén och delar av klimatanläggningen. I gränssnittet representeras dessa temperaturer av färger. Mätdata är i första hand upptagen i bilmodellen Volvo XC90.

Visualiseringsverktyget EasySee är implementerat i MATLAB 7.0 release 14 och Java 1.4.2 på en Windows plattform. Interpoleringen utförs i MATLAB medan Java ansvarar för uppritningen i användargränssnittet samt hanterar interaktionen med användaren.

Rapporten beskriver utvecklingen av visualiseringsverktyget EasySee på uppdrag av Volvo Technology Corporation, avdelning 6280 Control and Simulation, i Göteborg. Rapporten avverkar följande huvudavsnitt: förutsättningar för arbetet, teori för tolkning och behandling av mätdata samt användargränssnitt, tillvägagångssätt vid utveckling av verktyget, resultat samt riktlinjer för framtida vidareutveckling.

Arbetet resulterade i en applikation med stora möjligheter att underlätta det dagliga arbetet med analys av klimatmätdata.

Förord

Denna rapport är ett resultat av vårt examensarbete inom civilingenjörsprogrammet i datateknik vid Lunds Tekniska Högskola, LTH. Examensarbetet har utförts vid Institutionen för Reglerteknik i samarbete med Volvo Technology Corporation, VTEC, under tidsperioden januari till juni år 2005.

Vi vill rikta ett stort tack till samtliga anställda på avdelningarna Control and Simulation samt Vehicle Electronics och särskilt till våra handledare Björn Mårdberg, Mikael Askerdal, Peter Sundqvist och Vincent Andreasson. Ett stort tack också till Dr. Rolf Johansson, handledare vid Reglertekniska Institutionen, LTH.

Göteborg den 17 juni 2005

Matilda Gustafsson och Emma Jönsson

Innehållsförteckning

Figurförteckning

1 Inledning

1.1 Bakgrund

Kupéklimatet i dagens bilar styrs ofta med en klimatanläggning som utnyttjar automatisk reglering. För att erhålla ett behagligt förarklimat och imfria rutor styr klimatregulatorn temperaturen och luftdistributionen i kupén. Volvo Technology Corporation, VTEC, har under flera år gjort mjukvaran till Volvos klimatanläggningar. På VTEC, som är ett bolag inom Volvo Group, sker Forskning och Utveckling till nytta för produktionsbolagen samt Volvo Cars. Inom avdelningen Elektronik och Programvara fokuserar man på inbyggda styrsystem för bland annat klimatanläggningen. Där återfinns gruppen för klimatreglering som utvecklar modeller och algoritmer för att kunna reglera klimatet, och i huvudsak temperaturen i kupén. Algoritmerna implementeras sedan i klimatanläggningens styrenhet.

Den ständiga utvecklingen av nya bilmodeller hos Volvo medför kontinuerlig testning och simulering där en stor mängd klimatdata samlas in. Dessa data ligger sedan till grund för beslut gällande regleringen i klimatsystemet. I framtiden antas dessa simuleringar bli än mer betydelsefulla då projekt ska genomföras med hjälp av färre provvagnar på en kortare projekttid än i dagsläget. Kraven ökar alltså samtidigt som systemen i dagens bilar blir alltmer komplexa.

För närvarande analyseras mätdata från dessa simuleringar med hjälp av ett stort antal grafer, men då komplexiteten växer ökar svårigheterna med att tyda beroendena i systemet. Dessutom kräver graferna mycket utrymme och blir fort oöverskådliga. Som ett alternativ till att studera flera signaler som enskilt ger liten information om systemet som helhet, uppstod ett önskemål om att istället grafiskt visualisera systemet för att få en överblick över dess beteende. Fördelarna med ett visualiseringsverktyg är många, dels elimineras problemen med att välja vilken data som är relevant, dels kan visualiseringen, genom att fokusera på det övergripande beteendet, ge ytterligare förståelse som skulle vara svår att uppnå med hjälp av bara enkla grafer. Visualiseringen erbjuder också en möjlighet att lätt se effekterna av förändrade förutsättningar.

Vid utveckling av reglersystem för klimatanläggningar har avdelningen tagit fram flera olika modeller över kupéklimatet samt även samlat på sig en stor mängd mätdata. Denna information, tillsammans med avdelningens erfarenheter från tidigare hjälpmedel, kommer att vara utgångspunkten för vårt examensarbete.

1.2 Syfte och problemställning

Syftet med examensarbetet är att underlätta utvärderingen av klimatmätdata i fordon. För att få en bättre förståelse och helhetsbild av hur temperaturen i kupén och klimatanläggningen är fördelad utvecklas ett visualiseringsverktyg. Vi vill med detta verktyg besvara frågan om det är möjligt att med hjälp av visualisering underlätta analys av klimatmätdata.

Verktyget är avsett att utgöra en inledande del i arbetet med kupéklimatet och ska ge en översikt över mätdata samt en känsla för klimatet som helhet. Dess huvudsakliga uppgift är att visa *hur* regleringen fungerar, men för att besvara frågan *varför* systemet förhåller sig på detta sätt måste signalernas grafer fortsatt detaljstuderas. Verktyget har dock i detta läge minimerat den delmängd signaler som är intressanta.

Målet med verktyget är att ta fram ett grafiskt gränssnitt som visualiserar data i ett antal vyer på ett representativt och lättförståeligt sätt, samt tillåter en enkel interaktion med användaren. Frågan blir följaktligen hur ett grafiskt användargränssnitt för ett visualiseringsverktyg bör utformas.

1.3 Målgrupp

Denna rapport riktar sig till personer med intresse för visualisering och dess betydelse för analys av data samt utveckling av användargränssnitt. Läsaren förutsätts ha en teknisk bakgrund för att till fullo kunna tillgodogöra sig alla resonemang.

Eftersom syftet med verktyget är att det ska användas i arbetet vid VTEC, ställs krav på att rapporten även ska vara tillgänglig för intressenter vid avdelningen.

1.4 Avgränsningar

Det finns flera tänkbara användningsområden för verktyget, till exempel vid simulering med SimuLink, vid Hardware In the Loop-testning samt vid uppspelning av filer från klimatprovning, både offline och online. Vi har i vårt arbete begränsat oss till uppspelning av mätdatafiler offline, då detta är en lämplig utgångspunkt för att bedöma om verktyget uppfyller de behov som avdelningen har. Om en utvärdering av verktyget visar på önskad funktionalitet kan resurser läggas på ytterligare utveckling för övriga användningsområde.

Nedan listade avgränsningar gäller för systemet:

- För att applikationen ska vara körbar har vi antagit att mätdata produceras enligt dagens format, MAT-filer. Dessutom förutsätter vi att viss mätdata, till exempel temperaturer för kupéns termoelement, finns tillgänglig för att applikationen ska vara exekverbar.

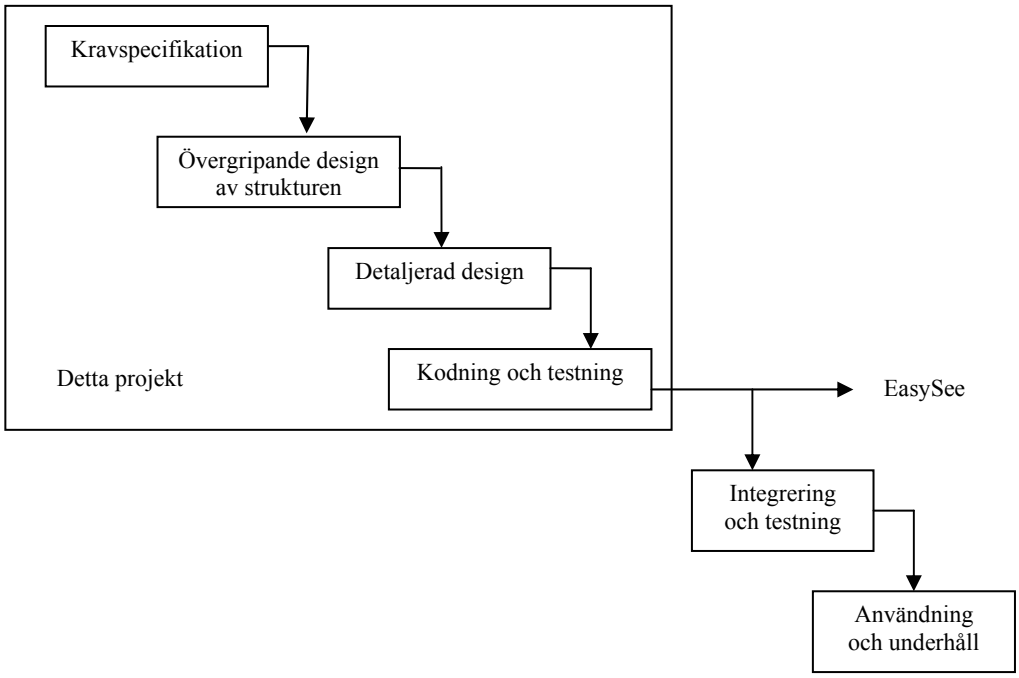
- Ingen hänsyn har tagits till att värme stiger vid framtagningen av modeller för att beskriva temperaturen i kupén och evaporatorn. Inte heller har hänsyn tagits till hur sätena i kupén påverkar flödena från ventilationerna.

- Inga fysikaliskt förankrade flödesberäkningar har gjorts för att bestämma ventilationernas påverkan på temperaturen i kupén.

- Visualiseringen sker endast i två dimensioner, där vi antagit att alla termoelement för respektive sida av kupén är placerade på samma djup i kupén. Vid bestämning av temperaturen för ena sidan tas ej hänsyn till den andra sidans termoelements eventuella påverkan.

- För att få en god upplösning samt utseendemässigt tilltalande visualiseringar med en rimlig tids- och arbetsinsats har flexibiliteten vad gäller byte av bakgrundsbilder i användargränssnittet begränsats något. Till exempel måste nya bakgrundsbilder vara i ungefär samma storlek samt innehålla lika många mätpunkter som ursprungsbilderna.

1.5 Metod



Figur 1. Projektets livscykel

Arbetet med EasySee inleddes med en verktygsutvärdering, vars syfte var att bestämma om ett verktyg som stödjer den funktionalitet vi trodde oss komma att behöva fanns tillgängligt samt hur hög dess inlärningströskel var. Efter jämförelse av alternativen beslutade vi att använda en kombination av Java och MATLAB. För mer detaljer se kapitel 3 Verktygsutvärdering.

Vidare identifierades flera områden där vi trodde oss kunna stöta på problem, dels på grund av bristande erfarenhet inom dessa områden, dels på grund av osäkerhet om vilka förutsättningar ett samarbete mellan Java och MATLAB egentligen innebar vad

gäller till exempel datadelning och prestanda. För alla dessa problemområden gjordes utförliga test för att bestämma våra möjligheter och vilka förutsättningar vi kunde utgå ifrån i det vidare arbetet med den övergripande designen av systemet.

Arbetet har fortskridit i en iterativ process där vi allteftersom behov uppstått gjort ändringar i systemets design, implementerat ny funktionalitet samt testat denna.

1.6 Notation

Definitioner

Ventilation	Ventilationsmunstycke, ur vilket luft tillförs kupén med hjälp av fläkten.
Termoelement	Mätpunkterna i kupén, klimatanläggningen respektive evaporatorn.

Förkortningar

Floor	Ventilation vid golvet.
Vent	Ventilation vid panel på instrumentbrädan.
Defroster	Defroster ventilation för framruta.

För alla hänvisningar till klassnamn och variabelnamn från koden används typsnittet `Courier New`, för att tydligt urskilja dessa i rapporten.

1.7 Disposition

Kapitel 2

Kapitlet innehåller grundläggande teori för de delar som behandlas i utformningen av verktyget EasySee. Kapitlet inleds med en genomgång av klimatsystemet i Volvo XC90 och fortsätter med ett avsnitt om tolkning och behandling av data. Detta avsnitt tar upp diverse visualiserings- och interpoleringstekniker. Avslutningsvis beskrivs designaspekter för ett grafiskt användargränssnitt

Kapitel 3

Verktygsutvärderingen inleds med en diskussion om visualiseringen av klimatmätdata bör utföras i två eller tre dimensioner. Därefter utvärderas aktuella verktyg utifrån

definierade huvuduppgifter för applikationen. Kapitlet avslutas med resultat och slutsats av verktygsutvärderingen.

Kapitel 4

Detta kapitel redogör för tillvägagångssättet vid utvecklingen av EasySee. De visualiserings- och interpoleringstekniker som valts för att åskådliggöra data i de olika delarna i applikationen beskrivs utförligt, tillsammans med motiveringar till varför just dessa tekniker använts. Kapitlet fortsätter med en redovisning av hur användargränssnittet utformats och avslutas med en översiktlig beskrivning av systemets design och struktur.

Kapitel 5

Kapitlet redogör för resultatet av arbetet med utvecklingen av EasySee samt besvarar de frågor som initialt togs upp i problemställningen.

Kapitel 6

Detta kapitel diskuterar vad i arbetet med EasySee som kunde ha gjorts annorlunda samt ger rekommendationer för en eventuell vidareutveckling och förbättring av systemet

2 Teori

2.1 Klimatsystemet i Volvo XC90

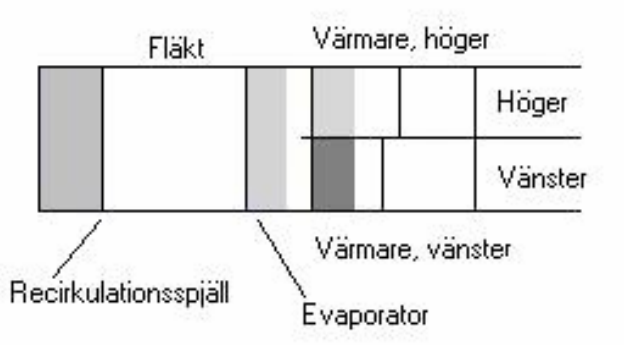
Volvos klimatanläggning ECC, Electronic Climate Control, styr recirkulationsgrad, fläkthastighet, AC, värme samt luftdistribution för att uppnå bra komfort för bilens passagerare. Klimatregulatorns huvudsakliga uppgift är att styra temperaturen i kupén, men även luftdistributionen är viktig för att få imfria rutor och ett behagligt klimat. Som insignaler till regulatorn kan bland annat nämnas utomhustemperatur, solintensitet, kylvattentemperatur, evaporatortemperatur, kupétemperatur, reglageinställningar samt bilhastighet [4].

Klimatanläggningen, som är lokaliserad under instrumentpanelen, kan delas upp i flera delar; en förenklad bild över dess struktur visas i figur 2. Det bör noteras att klimatanläggningen delas upp i en vänster- och högersida innan värmaren. Respektive sida, vars struktur återges i den nedre delen av figur 2, kan vidare delas upp i ytterligare delar efter funktion [4].

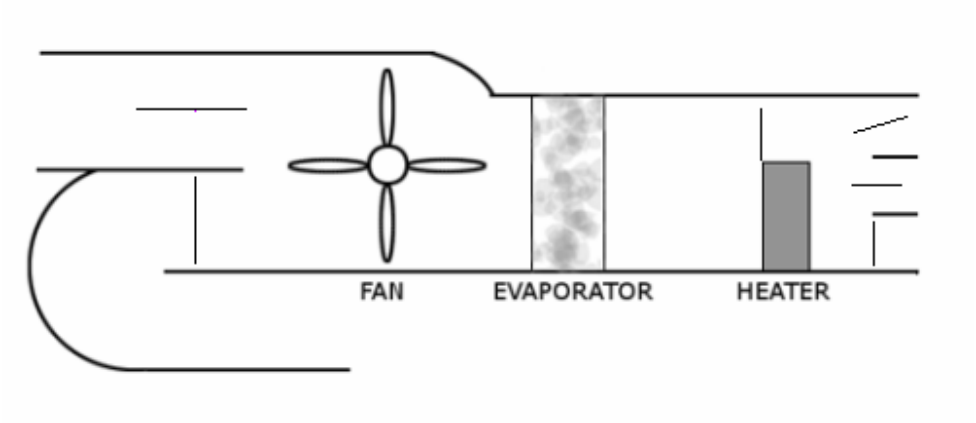
Nedan följer en kortfattad beskrivning av klimatanläggningens fysiska struktur samt dess funktion. Vi har dessutom valt att närmare beskriva evaporatorn.

2.1.1 ECC

Från ovan:

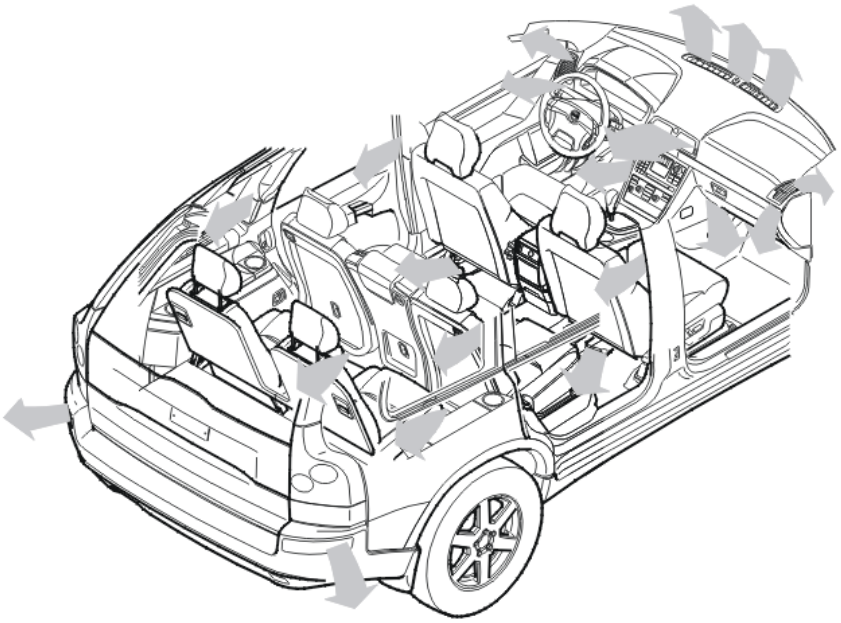


Från sidan:



Figur 2. Principiell uppbyggnad av klimatanläggningen

Det finns två luftintag till klimatanläggningen, utifrån eller som recirkulerad luft från kupén. Recirkulerad luft kan användas för att stänga ute dålig lukt, men den primära användningen är att återvinna kyla. I riktigt varmt klimat skulle det aldrig bli tillräckligt svalt i kupén om inte recirkulation användes. Utomhusluft och recirkulerad luft blandas och blåses genom systemet med hjälp av en fläkt. Första steget efter fläkten är evaporatorn där luften kyls ned, och samtidigt kondenseras luften och blir torrare. Klimatanläggningen delas därefter upp i en separat höger- och vänstersida. För respektive sida delar ett spjäll upp luften i två kanaler förbi värmaren. Luften i den nedre av kanalerna påverkas av värmaren innan kanalerna förenas igen, och genom att styra värmespjället påverkas alltså luftens temperatur in i kupén. Efter värmaren eftersträvas en luftblandning där temperaturen är fördelad så att luften vid golvventilationerna är varmare än luften vid defrosterventilationerna. Efter att luften blandats delas den upp i de tre kanalerna golv, panel och defroster. Ytterligare förgrening sker sedan till ventilationerna som är placerade inne i kupén enligt figur 3. Dessa ventilationer är även tabellerade i avsnittet 1.6 Notation [1][4][6].



Figur 3. De olika ventilationsmunstyckena i kupén för XC90

I figuren syns även de två munstyckena, placerade längst bak i kupén, som är förbundna med omgivningen för att släppa ut kupéluft. Vid de främre golvutrymmena finns munstycken för att recirkulera luft.

Då AUTO luftdistribution angetts av bilens passagerare på klimatenhetens panel kontrollerar regulatorn klimatanläggningens olika parametrar, till exempel spjällpositioner och fläktvarvtal. Det finns ett antal möjliga spjällkonfigurationer. Kortfattat innebär dessa att det finns tre huvudinställningar för luftdistributionen, FLOOR, VENT och DEFROSTER. Dessa luftdistributioner kan ses som reglertekniska arbetspunkter. Är regulatorn inställd på exempelvis luftdistributionen FLOOR-VENT blåser det ur både panelmunstyckena och golvmunstyckena. För att kupén ska upplevas som behaglig och för att avimningen ska fungera optimalt gäller följande regler. Vid kallt klimat eftersträvas att den varmaste luften kommer ut vid fötterna. Detta åstadkoms dels genom temperaturfördelningen som uppstår efter värmaren, dels genom lämplig reglering av de spjäll som sitter efter värmaren. Detta innebär att vid kallt klimat används troligtvis luftdistributionen FLOOR, dessutom är

DEFROSTER spjället något öppet för att minska imrisken. I varmt klimat används luftdistributionen VENT under nedkylningsfasen. Vid en stabiliserad, behaglig temperatur används i vissa körfall en luftdistributionen VENT-DEFROSTER. Denna kombination undviks dock när risken för kondens på utsidan bedöms vara för stor, exempelvis vid låg vagnhastighet i tropiska klimat [4][6].

2.1.2 Evaporatorn

En viktig del av klimatanläggningen är AC systemet, vars uppgift är att kyla luften som ska distribueras till kupén. Evaporatorn, som är ett annat namn för förångare, utgör en separat del i AC systemet [1].

Evaporatorn påverkas av utomhustemperatur, kupétemperatur, recirkulationsgrad, luftfuktighet och fläktspänning, där en given fläktspänning motsvarar ett visst luftflöde genom evaporatorn. Evaporatorn innehåller ett freonbesläktat ämne som fungerar som köldmedel då det omvandlas från vätska till gas. Köldmedlet håller konstant temperatur, till skillnad från värmaren där vätskan svalnar, men fasövergången mellan tillstånden, det vill säga från vätska till gas, kräver energi. Energin som förbrukas vid övergången tas från den omgivande luften som därmed kyls ner. Det sker alltså en värmeväxling mellan köldmedlet och luften, och dessutom kondenseras luften och blir torrare [6].

2.2 Tolkning och behandling av data

Ögonen utgör vårt viktigaste fönster mot verkligheten. För att förenkla utbytet av information mellan användare och dator kommer nästa generations system att ha ett stort visuellt innehåll där bilder och avbildningar kommer att förmedla information i allt större utsträckning. Redan i dagsläget utnyttjar man ofta så kallad vetenskaplig visualisering som är en metod med vars hjälp det icke-synliga kan göras synligt. Det kan till exempel röra sig om vindar, temperaturer eller molekyler som representeras med hjälp av geometriska figurer i olika färger istället för med symbolisk återgivning. Syftet med en visualisering är ofta att skapa en upplevelse hos användaren som motsvarar praktisk erfarenhet. [7]

Introduktionen av visuella informationssystem har möjliggjorts av flera faktorer, till exempel framsteg inom hårdvaruteknologin, förbättrade mjukvarumetodiker, nya metoder för att analysera visuell data samt framsteg inom digital kommunikation [3].

I avsnittet beskrivs vad begreppet visualisering innebär, dessutom presenteras några olika visualiseringstekniker.

2.2.1 Begreppet visualisering

Begreppet visualisering innebär omvandling av data eller information till bilder, med andra ord den process som transformerar data till en visuell form. Med hjälp av visualiseringar får användare möjlighet att observera information på ett enkelt och överskådligt sätt. Informationen i data får på så sätt större betydelse för betraktaren. Det handlar alltså om att visa komplex information i flera dimensioner på ett sådant sätt att förståelsen ökar [3].

Visualiseringar används för det första för att *förstå* information. En visualisering ska ge snabb insikt i information genom att använda människans goda perceptionsförmåga. Visualiseringar kan användas för att hitta samband och för att undersöka antaganden när man inte riktigt vet vad man söker. Med hjälp av visualiseringar kan man få en ökad insikt för att kunna fatta beslut [3].

För det andra används visualiseringar för att *förmedla* information till andra. Detta för att man lättare ska kunna illustrera något och försöka förvissa andra om vad man själv anser eller har upptäckt [3].

En bra visualisering ökar alltså förståelsen och förmedlingen av informationsmängden. Om visualiseringen dessutom är interaktiv där betraktaren själv kan manipulera visualiseringen får informationen ännu större betydelse. Genom animationer har man möjlighet att visa ett händelseförlopp och därmed ges ett helt nytt perspektiv på informationsflödet. Genom att förståelseprocessen blir enklare kan också förmedlingen av information till andra gå fortare. Det innebär att produktionstiden kortas drastiskt och därmed minskas kostnaderna. Detta medför billigare produkter som snabbare kommer ut på marknaden [3].

Det finns olika tekniker för att visualisera datamängder. Olika tekniker framhäver olika företeelser i ett dataset, och valet av visualiseringsteknik beror därför på vad för slags information man vill få fram och visualisera. Skaläralgoritmer är algoritmer för användning främst vid tvådimensionella visualiseringar, medan vektoralgoritmer fördelaktigen utnyttjas för visualisering i tre dimensioner [3].

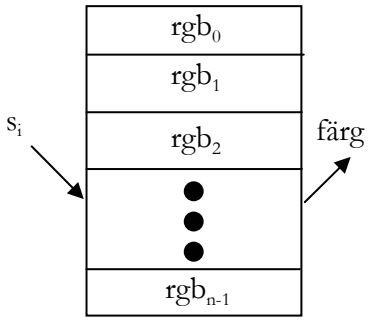
2.2.1.1 Skaläralgoritmer

En skalär är ett enskilt datavärde som är associerat med varje punkt eller cell i ett dataset. Skaläralgoritmer opererar på skalärdata. I följande avsnitt redogörs för de vanligaste skaläralgoritmerna.

2.2.1.1.1 Färgmappning

Färgmappning innebär att skalärvärden i punkten (x, y) eller (x, y, z) mappas till ett färgvärde. Mappningen sker genom en färguppslagstabell som innehåller färgvärden för all skalärdata. Varje skalärvärde, s_i, inom ett visst intervall motsvarar ett värde i uppslagstabellen.

Genom att använda sig av en överföringsfunktion istället för uppslagstabell får man en mer generell form av färgmappning. Det krävs mycket stor noggrannhet när man väljer uppslagstabell eller överföringsfunktion för att åstadkomma en effektiv och användbar färgmappning [3].



Figur 4. Färgmappning

2.2.1.1.2 Isoytor

Vid användning av isoytor skapas konturer i 2D som gränser mellan konstanta skalärvärden så att ett dataset delas in i regioner. Med hjälp av konturerna kan områden med samma värden som mappats till samma färg upptäckas, vilket gör det lättare att hitta samband i datasetet. I figur 5 nedan, har en tröskel med värdet fem valts för att skapa en kontur i 2D nätet. Eftersom det bara finns skalärvärden i ett begränsat antal punkter i datasetet krävs det att man använder sig av någon form av interpolering för att generera en kontur. Interpolering går ut på att man räknar fram uppskattade värden som saknas mellan mätpunkterna i datasetet.

Genom att välja olika tröskelvärden för konturer kan man visa olika nivåer av en bild. Tyvärr ges i allmänhet ingen direkt information om små detaljer och variationer [3].

3	6	6	3	1
7	6	7	3	2
7	8	6	4	3
7	6	8	3	2
2	3	4	3	3

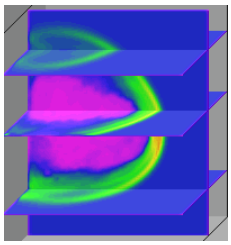
Figur 5. Isoyta med tröskelvärde 5

2.2.1.1.3 Slicing

Slicing tekniken innebär att ett eller flera tvådimensionella tvärsnitt görs genom en datamängd för att lättare få en inblick i denna. Den volym som avskiljs av snittytorna visas inte. Genom att interaktivt flytta snittplanen skapar man sig en uppfattning om

volymens totala uppbyggnad, vilket ger en tredimensionell känsla även om planen i sig är tvådimensionella [3].

I snittet kan man göra detaljstudier, varför metoden lämpar sig bäst för att studera en parameter i taget.

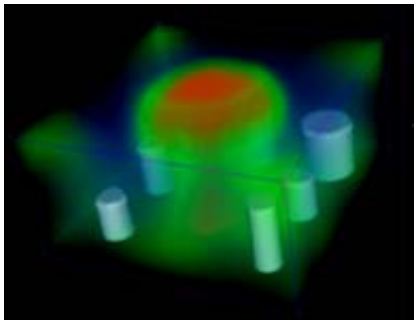


Figur 6. Slicing teknik

2.1.1.1.4 Volymrendering

Volymrendering innebär att hela datamängden visas med hjälp av att en ”stråle” skickas genom volymen. För varje bildpunkt, pixel, i bilden som ”strålen” går igenom integreras volymdata längs vylinjen. Det resulterande värdet visas med någon kodning, exempelvis färg eller intensitet. För att man ska uppfatta volymvisualiseringen på bästa sätt är volymens luminans och opacitet viktiga parametrar. Genom att styra genomskinligheten i volymen fås olika mycket inblick i volymens inre. En mängd varianter är baserade på denna teknik och motsvarar 2D fallets kontinuerliga bild där funktionsvärdet i varje punkt visas med någon form av kodning.

Volymvisualisering kan åstadkommas med hjälp av metoder som ger en 2D bild eller metoder som ger volymdata, volymrendering. Volymrendering ger ofta bilder som är svåra att tyda, men har som fördel att den använder originaldata och därför förbises inga väsentliga detaljer. Tekniken är ofta mycket beräkningskrävande och därmed långsam att använda [3].



Figur 7. Volymrendering

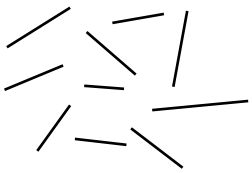
2.2.1.2 Vektoralgoritmer

Vektordata är en tredimensionell representation av riktning och storlek. Vektoralgoritmer opererar på vektorer. I följande avsnitt redogörs för de vanligaste vektoralgoritmerna.

2.2.1.2.1 Hedgehogs

En hedgehog är en riktad linje som visas för varje vektor, se figur x. Linjen börjar i den punkt som vektorn är associerad med och riktas i vektorkomponenternas (v_x, v_y, v_z) riktning. En hedgehoglinje är en approximation av en punkts rörelse över en tidsperiod. Hedgehoglinjen skalas proportionellt med vektorns storlek och ofta används färgkodning på linjerna för att representera skalärdata [3].

När alla hedgehoglinjer ritas ut kan det bli ett enda stort gytter av linjer och det kan då bli svårt att se i vilken riktning som de pekar och även att kunna urskilja den information som visualiseringen innehåller. För att undvika detta måste därför vissa linjer väljas ut ur datamängden, vilket kan vara svårt eftersom väsentlig information riskeras att skalas bort i visualiseringen [3].



Figur 8. Hedgehogs

2.2.1.2.2 Orienterade glyfer

En orienterad glyf är en riktad geometrisk form i två eller tre dimensioner, exempelvis en triangel eller en kon. Färg och storlek kan sättas för att representera vissa egenskaper hos skalärer i vektorn. Skalning av glyfer kan vara ett problem eftersom glyfen då inte får samma egenskaper som den representerar hos vektorn. På samma sätt som för hedgehogs, kan det då det gäller glyfer vara svårt att visa tillräckligt många utan att visualiseringen blir otydlig [3].

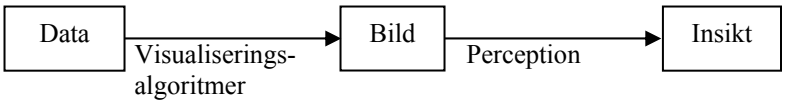
2.2.1.2.3 Strömlinjer

En annan metod att visualisera vektordata är att använda sig av kurvor i tre dimensioner, så kallade strömlinjer. Strömlinjer används för att följa spår eller banor i ett vektorfält. En lämplig tillämpning för strömlinjer kan vara att visualisera flöden, till exempel vindflöden [3].

2.2.1.3 Perceptuell visualisering

Att använda perceptuell visualisering, att ta hjälp av speciella egenskaper hos människans uppfattningsförmåga när man ska förmedla ett budskap, är mycket användbart vid skapandet av visualiseringar. Av denna anledning är det viktigt att känna till hur människan uppfattar olika företeelser eller signaler.

Den visuella perceptionen kan ses som en fortsättning på synsinnet där hjärnan gör en tolkning av intryck utifrån bland annat jämförelser och erfarenheter. Hjärnan ser färg, djup och rörelse separat och gör sig sedan en egen helhetsbild av dessa. Perceptionen hjälper till att skapa förståelse då man studerar bilder i visualiseringar. Hur data via bilder leder till insikt visas i figur 9 [3].



Figur 9. Perceptuell visualisering

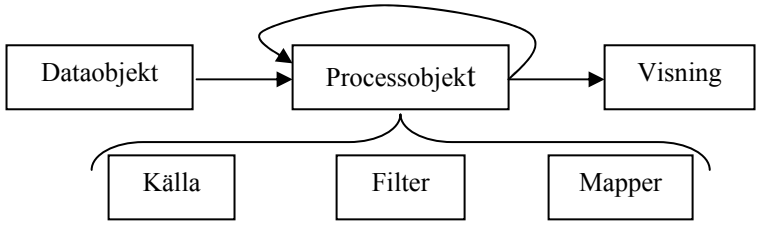
Det är väsentligt att veta hur människan tolkar exempelvis färgintryck när man tittar på en bild. En mättad, ljusstark färg påkallar mer uppmärksamhet än en mörk, omättad färg. Därför passar mörka färger bra som bakgrundsfärg då de upplevs som neutrala. Färger har också andra psykologiska effekter. Man skiljer på kalla färger såsom blått och violett och varma färger som rött och gult. Färgen rött uppfattar människan dessutom som en varningsfärg [3].

2.2.1.4 Visualiseringspipelinen

De flesta visualiseringsverktyg använder det som kallas visualiseringspipelinen. Denna beskriver förflyttningen av ursprungsdata till en visuell representation, och llustreras i figur 10 nedan. Pipelinen består av dataobjekt som representerar data, processobjekt (källobjekt, filterobjekt eller mapperobjekt) som opererar på data samt en angiven riktning av dataflödet (pilar som kopplar ihop olika objekt).

Källobjekten läser data från dataobjektet eller genererar data från lokala parametrar. En källa har ingen input från något annat processobjekt. Filtren i processobjekten utför transformationer på data och kan utvecklas för att manipulera, filtrera och

presentera data. Ett filter har både input från och output till andra processobjekt. Ett mapperobjekt avslutar dataflödet i visualiseringen och har därmed ingen output till andra processobjekt. Mapperobjekten konverterar data till grafiska primitiver så att det hela kan resultera i en visuell representation [3].



Figur 10. Visualiseringspipelinen

2.2.2 Begreppen interpolering och extrapolering

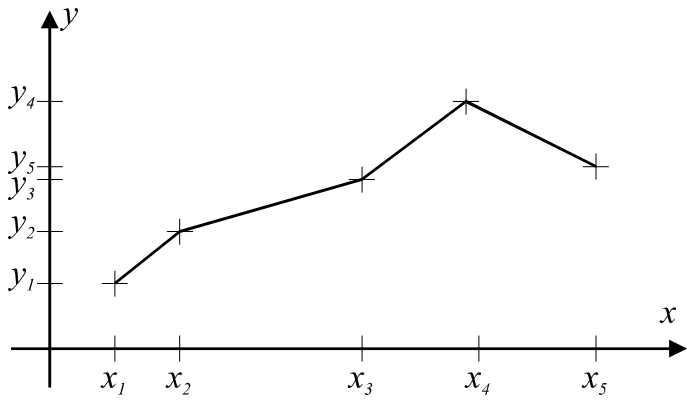
Den information som erhålls från ett system genom diverse provtagningar är mätdata upptagen i utvalda mätpunkter. Ofta finns dock ett intresse av att studera hur mätdata ser ut över en kontinuerlig yta. Interpolering är en metod för att uppskatta variationen över ytan genom användning av data från mätpunkterna. Ett flertal olika tekniker för att bestämma värden för mellanliggande punkter, det vill säga omvandla data från mätpunktsobservationer till kontinuerliga ytor, finns att tillgå.

Interpoleringen säger ingenting om hur en kurva ser ut utanför det intervall som spänns upp av mätpunkterna, utan beräkning av värden i sådana områden benämns extrapolering. Extrapolering innefattar olika tekniker för hur exempelvis en kurva eller yta förlängs utanför det område som innehåller mätpunkterna. Det är viktigt att ha i åtanke att vid alla former av interpolering och extrapolering ingår ett mått av godtycklighet.

I följande avsnitt kommer de vanligaste interpolerings- och extrapoleringsteknikerna beskrivas kortfattat.

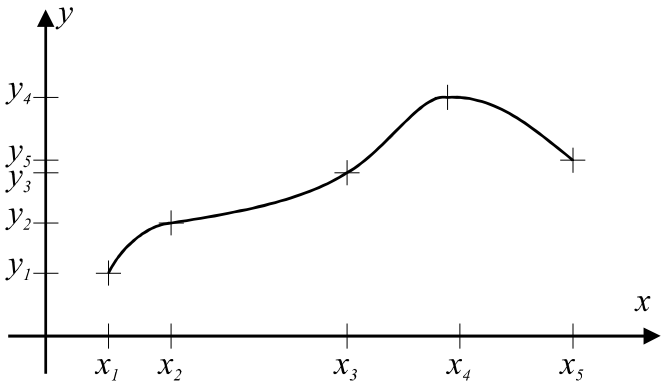
2.2.2.1 Interpolering i en dimension

Interpoleringsfunktionen bestämmer hur mätpunkterna sammanbinds för att bilda en kurva och därigenom ange värden för mellanliggande punkter. Interpoleringsfunktionen kan exempelvis vara ett polynom, ett styckvis polynom, ett trigonometriskt polynom eller en rationell funktion. Vid *styckvis linjär interpolering* bestäms värdet för en okänd mellanliggande punkt med hjälp av en linje genom de två närmsta mätpunkterna enligt figur 11 nedan [6].



Figur 11. Styckvis linjär interpolering

Det finns även teorier för hur man använder information från fler mätpunkter, exempelvis två mätpunkter på vardera sidan om interpoleringspunkten, för att bygga en bättre anpassad kurva. Bättre anpassad innebär att kurvan på ett mer riktigt sätt beskriver det verkliga beteendet, exempelvis genom en kontinuerlig derivata [6].



Figur 12. Interpolering med kontinuerlig derivata

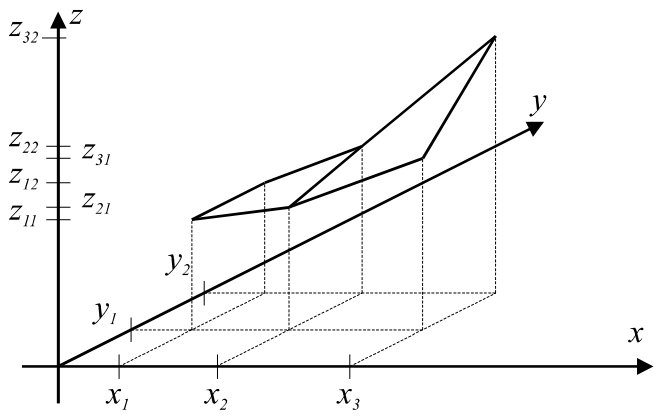
2.2.2.2 Interpolering i två dimensioner

När det gäller tvådimensionell interpolering bör man skilja på om mätpunkterna är rektangulärt eller godtyckligt utplacerade.

2.2.2.2.1 Rektangulärt placerade mätpunkter

Då mätpunkterna är utplacerade i ett rektangulärt rutnät, exempelvis sex punkter enligt figur 13, kan man precis som i fallet med en dimension välja olika interpoleringsstrategier. En enklare interpoleringsmetod som använder informationen från närliggande mätpunkter alternativt en mer avancerad metod som väger in fler mätpunkter. För mer avancerad interpolering hänvisas till splineteorin [6].

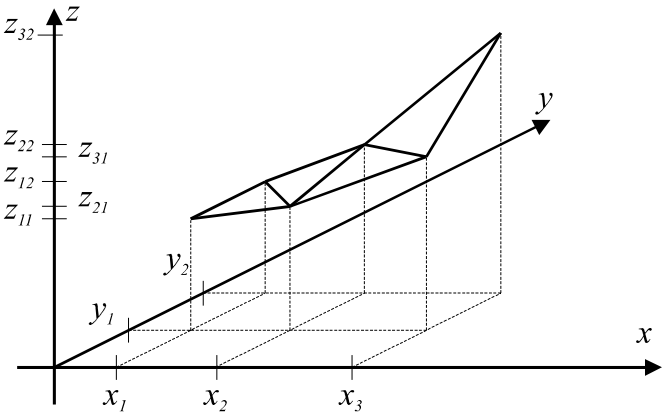
De interpoleringsmetoder som presenteras väger in mätpunkterna som utgör hörnen i den minsta rektangel som omsluter den punkt för vilken vi ska interpolera fram ett värde. För de linjer som utgör rektanglarnas kanter används exempelvis styckvis linjär interpolering. Då erhålls ett antal givna linjesegment enligt figur 13.



Figur 13. Rektangulärt placerade mätpunkter.

Inom varje rektangel kan punkternas värden bestämmas med hjälp av antingen *"styckvis linjära trianglar"* eller *viktning av de fyra hörnen* i rektanglarna.

Vid metoden "styckvis linjära trianglar" delas varje rektangel in i två trianglar. Eftersom en rät yta definieras av tre punkter, kan man därigenom bygga upp hela ytan av styckvis räta triangelytor [6].

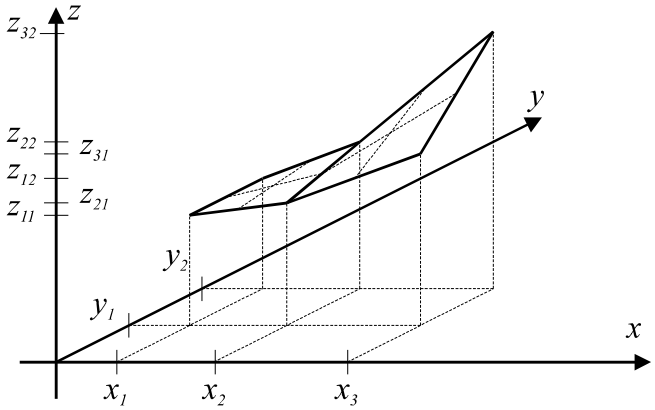


Figur 14. "Styckvis linjära trianglar".

Eftersom varje rektangel har två diagonaler, kan man välja mellan två trianguleringar i varje rektangel. I figuren valdes trianglarna så att de hörnpunkter som skiljde minst i z-led bands samman med en diagonal. De mellanliggande punkternas värden bestäms av det plan som spänns upp av de tre mätpunkter som utgör hörnen i respektive triangel.

Punkternas värden kan också bestämmas genom en viktning av de fyra hörnen i de ursprungliga rektanglarna, vilket är det vanligaste sättet att interpolera i två

dimensioner inom ett rektangulärt rutnät. För varje punkt beräknas då ett viktat medelvärde av z-värdena för de fyra punkterna i varje rektangel, där viktningen sker i proportion till avstånden i x- respektive y-led till mätpunkterna. Resultatet blir en yta som är styckvis linjär i både x-led och y-led. Dock är rektanglarna inte räta i alla riktningar. Denna typ av interpolering används ofta i tvådimensionella uppslagstabeller [6].



Figur 15. Viktning av hörnpunkter.

2.2.2.2.2 Godtyckligt placerade mätpunkter

Då mätpunkterna är godtyckligt utplacerade blir interpoleringen något svårare. Den ovan beskrivna metoden med styckvis linjära trianglar är dock att föredra framför viktning av rektangelns hörnpunkter, då trianglarna fortfarande får räta ytor och beräkningarna blir enklare [6].

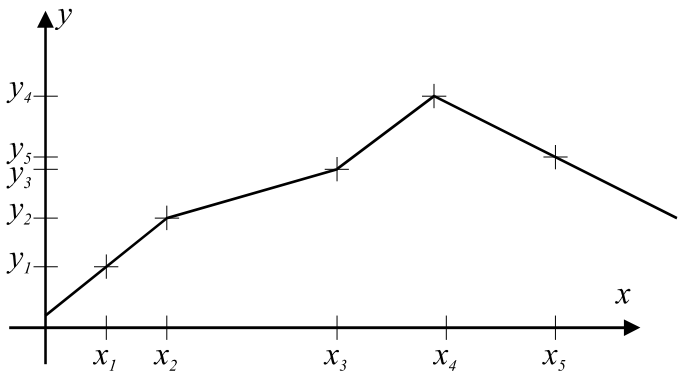
Vid godtyckligt placerade mätpunkter kan även andra metoder användas, se avsnitt *Närmsta granne interpolering* och *2.2.2.4 Avståndsviktad medelvärdesbildning*

Närmsta granne interpolering

Närmsta granne är en enkel interpoleringsmetod där varje okänd punkt ges samma värde som närmast belägna mätpunkt. Om mätpunkterna är oregelbundet spridda innebär detta att ytan kommer delas upp i homogena polygoner av olika storlek och form. Övergångarna blir skarpa, och det uppskattade värdet i en viss punkt avgörs endast av det värde som uppmätts i den mätpunkt som ligger närmast. Denna metod är enkel, men är i de flesta fall inte den lämpligaste för variabler som förändras gradvis över ytan [9].

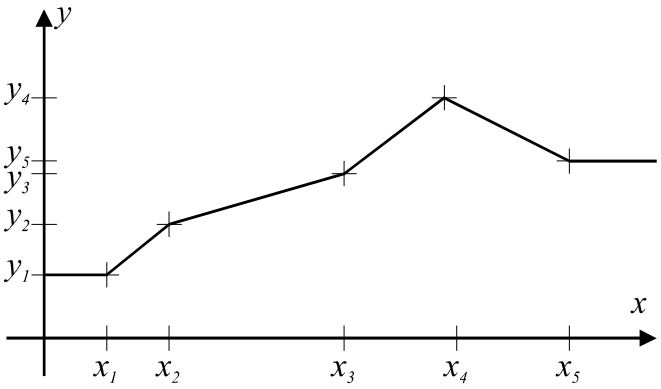
2.2.2.3 Extrapolering

Extrapolering innefattar olika tekniker för hur en kurva eller yta förlängs utanför det område som innehåller de givna mätpunkterna.



Figur 16. Linjär extrapolering.

Exempelvis kan linjär extrapolering från de två yttersta mätpunkterna, som i figuren ovan, användas. Denna sortens extrapolering kan verka mycket godtycklig i de fall där mätpunkterna ger en hackig kurva. Det anses ofta mindre spekulativt att bestämma värden i punkter utanför den kurva eller yta som mätpunkterna definierar genom att tilldela dessa punkter samma värden som de yttersta mätpunkterna, enligt nedanstående figur. Denna metod är ofta den som avses vid beräkningar utan extrapolering [6].



Figur 17. Extrapolering med konstant tilldelning av mätpunkternas värde.

2.2.2.4 Avståndsviktad medelvärdesbildning

Avståndsviktad medelvärdesbildning är kanske den mest använda interpoleringsmetoden och bygger, som namnet avslöjar, på viktade medelvärden. Metoden innebär i sitt grundutförande en kombination av interpolering och extrapolering.

Punkt för punkt i ett rutnät interpoleras genom att medelvärdet för ett antal närliggande mätpunkter beräknas. Dessa medelvärden viktas med avseende på avståndet till punkten. Mätpunkter som ligger nära punkten skall ha större inflytande än de som ligger längre bort varför det inverterade avståndet till mätpunkterna används som vikt [9]. Ovanstående kan sammanfattas med följande ekvation:

$$z(x_p) = \frac{\sum_{i=1}^{n} z(x_i)\frac{1}{d^k}}{\sum_{i=1}^{n}\frac{1}{d^k}}$$

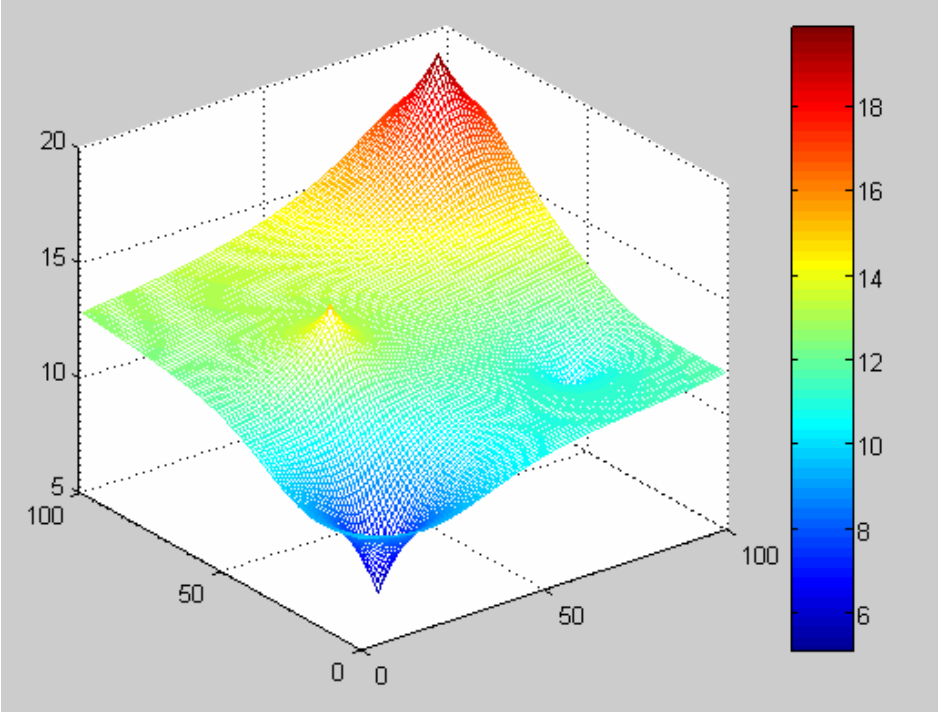
där

$z(x_p)$ = interpolerade värdet i punkt p
n = antalet mätpunkter
$z(x_i)$ = värdet för mätpunkt i
d = avståndet mellan mätpunkt i och punkt p
k = distansviktning

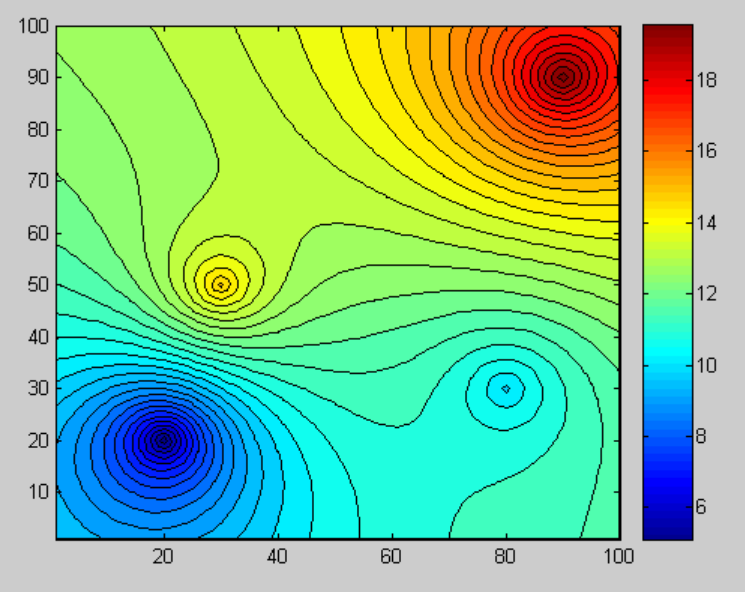
Ekvation 1. Avståndsviktad medelvärdesinterpolering.

Om man sätter k till ett blir varje punkt viktad i direkt proportion till sitt avstånd från aktuell mätpunkt. Om man ökar k till två blir mätpunkter som ligger nära punkten mer viktade än de som ligger längre bort, i proportion till kvadraten på det inverterade avståndet. På detta sätt kan man avgöra hur mycket man vill att närliggande mätpunkter skall påverka. Ett lågt värde ger dålig avspegling av lokala variationer medan ett för högt värde ger individuella mätpunkter överdrivet stort inflytande. Vanligtvis sätts värdet på k till mellan ett och tre. Oftast vill man inte inkludera alla mätpunkter när man beräknar ett värde i en punkt. Man anger därför en sökradie, där mätpunkter utanför denna radie exkluderas [9].

Figur 18 och 19 nedan visar ytor för den avståndsviktade medelvärdesbildningen applicerad på fyra godtyckligt utplacerade mätpunkter enligt tabell 1.



Figur 18. Avståndsviktad medelvärdesbildning



Figur 19. Nivåkurvor för avståndsviktad medelvärdesbildning.

Mätpunkt	x-koordinat	y-koordinat	Temperatur (°C)
1	20	20	5
2	80	30	10
3	30	50	15
4	90	90	20

Tabell 1.Data för utplacerade mätpunkter.

2.3 Design av grafiskt användargränssnitt

Ett gränssnitt definieras som en kontaktyta mellan två enheter. Grafiska användargränssnitt, GUI (Graphical User Interface), är kontaktytan mellan en användare och ett datorsystem och har som syfte att agera översättare, förmedlare samt guide för användaren såväl som datorn.

Gränssnittets design är avgörande för hur lättillgängliga systemets funktioner är för användaren. Det ska vara uppenbart för användaren vilka valmöjligheter som är tillgängliga, samt hur dessa ska användas. Endast ett användaranpassat gränssnitt kan användas med minimal ansträngning på ett enkelt och intuitivt sätt. Vad som är enkelt och intuitivt skiljer sig förstås från person till person, ju mer diversifierad den tänkta användargruppen är, desto svårare blir det att tillfredsställa alla [2][5].

Vid design av ett gränssnitt kan de principer som stödjer användarvänlighet delas in i tre kategorier: lärbarhet, flexibilitet samt robusthet. Dessutom är språk och layout väsentliga delar. Dessa **fem delar** beskrivs i korthet nedan [2][5].

2.3.1 Lärbarhet

Lärbarheten beskriver hur lätt det är för en ny användare att använda ett system och senare vad som krävs för att användaren ska få systemet att uppnå maximal prestanda. Lärbarhet uppnås genom att göra gränssnittet förutsägbart för användaren, genom att tydliggöra de funktioner som finns tillgängliga samt genom att hålla en följdriktighet för liknande funktioner genom hela layouten. Om något är konsekvent utformat så kan användaren uppfatta ett mönster och därmed lära sig strukturen och kännetecken. Det är konstaterat att vi lättare kan lära om vi kan känna igen mönster. Överensstämmande är därför en förutsättning för lärbarhet och gränssnittet bör designas att upprätthålla överensstämmelse vid alla tillfällen. Genom att använda accepterade standarder och konventioner underlättar man också för användaren då denna kan dra nytta av erfarenhet från sedan tidigare bekanta miljöer för att öka inlärningshastigheten. Dessutom bör varje del i gränssnittet där funktioner finns ha förklarande text och beteckningar för att guida användaren och minska dess minnesbelastning.

2.3.2 Flexibilitet

Ett flexibelt gränssnitt ska kunna anpassas och förändras, antingen av användaren eller av systemet självt, för att uppfylla särskilda behov. För användarens del innebär detta främst utseendemässiga förändringar medan systemet ska kunna ändra funktioner och innehåll i gränssnittet utifrån systemets aktuella tillstånd.

Flexibilitet är av stor betydelse för användarens känsla av kontroll över systemet. Därför är det en fördel om användaren tillåts att göra gränssnittet personligt med favoritfärger, mönster, fonter samt bakgrundsgrafik. Det kan också finnas fördelar med att låta användaren anpassa hur informationen presenteras, dess placering, storlek samt eventuellt organisering.

Vad man bör tänka på vid design av ett flexibelt gränssnitt är att flyttbara och föränderliga delar i gränssnittet i och för sig ger användaren flexibilitet, men proceduren kan bli omständlig och ta upp tid, eftersom användaren själv behöver gruppera skärmelementen, byta färger och så vidare. Dessutom kan användaren förstöra den design och de visuella koder som designern tänkt ut, varför det ofta är bättre att ge avkall på flexibiliteten och inte låta användaren kunna ändra skärmens layout i allt för hög grad.

2.3.3 Robusthet

För att ett gränssnitt ska anses vara robust gäller att det ska vara möjligt för användaren att utifrån gränssnittet avgöra i vilket tillstånd systemet befinner sig i, och så långt som möjligt, kunna återhämta sig från eventuella fel. Inga katastrofala fel får förekomma. Idealt ska det inte ens vara möjligt att begå några fel från gränssnittet. Feedback med bekräftelse på att en viss operation utförts är en viktig faktor för pålitligheten. Bekräftelsen får inte komma alltför lång tid efter den felaktiga inmatningen eftersom användaren då kan ha svårt att minnas vad som orsakade felet.

2.3.4 Språk

Ur programmerarens och företagets synvinkel är ett gränssnitt som endast stödjer ett språk, oftast engelska, att föredra då det kräver mindre arbete och därför är billigare att producera. För användaren är det dock en stor fördel att kunna välja att arbeta med det språk som man känner sig mest bekväm med.

2.3.5 Layout

En layout kan varieras i det oändliga men det finns vissa regler man bör följa för att uppnå ett bra resultat. Den skärmlayout man väljer ska utnyttjas genom hela applikationen då detta underlättar för användaren att förutspå placeringen av information och inställningar.

Det är viktigt att bestämma vilka skärmelement som användaren behöver ha tillgång till. Helst bör all relevant information för genomförandet av en uppgift kunna betraktas samtidigt. Huvuddelarna ska vara anpassade till skärmens storlek så att alla viktiga huvuddelar får plats samtidigt. Om det inte alltid är möjligt att visa all information samtidigt så är det viktigt att det på skärmen framgår var övrig information finns.

2.3.5.1 Interaktiva element

De interaktiva elementen ska lätt kunna identifieras, och helst arrangeras enligt vissa regler. Nedan följer information om de vanligaste interaktiva elementen och dess placering i gränssnittet [5]:

- Knappar bör ordnas i den sekvens de kommer att användas, samt placeras konsekvent. Knapparna måste vara tillräckligt lätta att träffa, de får inte vara för små eller ligga för tätt.

- Radioknappar är en grupp av två eller flera knappar för en viss inställning. Endast en knapp i gruppen kan vara aktiverad. Genom att aktivera en radioknapp i gruppen tänds denna och alla andra släcks. Radioknapparna skall placeras i rader eller kolumnvis så att de bildar en tydlig grupp.

- Kryssrutor används för att definiera om något är sant eller falskt, och används antingen fristående eller i grupp. Kryssrutor i grupp bör ha en grupprubrik som beskriver den aktuella gruppen. Även om kryssrutor bildar en grupp så är de fristående komponenter, där varje ruta kan vara vald eller icke-vald. När rutan är vald är den markerad med ett kryss eller en bock. Kryssrutorna bör placeras rad- eller kolumnvis så att de bildar en tydlig enhet.

- Kombinationsruta är en kombination av ett textfält och en listruta. Det finns olika varianter av kombinationsrutor, exempelvis kan själva fältet vara låst eller inmatningsbart. Är fältet inmatningsbart kan användaren mata in valet själv istället för att välja i listan, och i vissa fall behöver inte valet ens finnas i listan. Listrutan kan vara konstant nedfälld eller fällas ner genom att man klickar på knappen till höger om textfältet. Valen i listan skall vara sorterade i lämplig ordning.

- Symboler och ikoner kan utnyttjas för inlärning eftersom inlärningen påskyndas om informationen uttrycks i bilder jämfört med i text. Ikoner tar dessutom upp mindre utrymme än en språklig beskrivning av funktionen. Ikoner bör följa vissa formkrav för att fungera väl i användargränssnitt. Det är viktigt att det är hög kvalitet på bilderna så att de kan ses tydligt, varför det heller inte bör finnas för mycket detaljer i ikonen. Ikonen eller symbolen bör inte innehålla något utrymme för tolkningsproblem, utan det måste vara tydligt vad som avbildas och också vad avbildningen är tänkt att betyda. Språkoberoende ikoner är viktigt om systemet ska kunna användas över hela världen. Ikoner eller symboler kan till exempel användas i listrutor och på knappar.

3 Verktygsutvärdering

Idag finns ett flertal verktyg för visualisering tillgängliga på marknaden. Dessa kan vara både kommersiella och icke-kommersiella samt avse visualiseringar i två eller tre dimensioner. Vilket verktyg som ska användas beror bland annat på vad som ska visualiseras, vilket ändamål visualiseringen har samt vilka behov användaren har.

Som en första del i examensarbetet ”EasySee – Ett verktyg för visualisering av mätdata från klimatprovning” gjordes en verktygsutvärdering för att hitta det bästa angreppssättet på uppgiften. Innan sökningen efter och utvärderingen av potentiella verktyg kunde påbörjas togs beslut om visualiseringen skulle ske i två eller tre dimensioner. Resultatet av denna diskussion redovisas under nästföljande avsnitt. Därefter följer verktygsutvärderingen samt slutsatsen av denna.

3.1 Visualisering i två eller tre dimensioner?

Få undersökningar har gjorts angående fördelarna och nackdelarna med att använda 2D respektive 3D visualiseringar. Gällande det eventuella mervärdet med flerdimensionella visualiseringarna byggs antaganden mer på känslor än på empiriska undersökningar. Några undersökningar i frågan om valet av 2D eller 3D har dock genomförts och de flesta når samma slutsats, nämligen att det inte handlar om att strikt använda sig av antingen två eller tre dimensioner. Istället bör en kombination av dessa utnyttjas, där ändamålet och användningsområdet avgör vilken teknik som lämpar sig bäst. Det är viktigt att tänka på att 3D inte ska användas för sakens skull utan för att öka trovärdigheten, förståelsen och förmedlingen av informationen. Nedan följer en sammanställning av några av de utförda undersökningar, där de främsta för- respektive nackdelarna med två- och tredimensionella visualiseringar belyses [3].

Ny teknik kommer ofta från den akademiska världen och det har visat sig att det kan ta relativt lång tid innan den slår igenom kommersiellt. Enligt ett flertal studier verkar detta vara fallet med tredimensionella visualiseringar. I nuläget anses det ofta i

kommersiella sammanhang vara viktigare att kunna interagera i 2D än att kunna åskådliggöra data i tre dimensioner. Många tycker att 2D är lättöverskådligt och det är lätt att navigera i datamängden, och ställer sig frågande till om ytterligare en rumslig dimension verkligen ger så mycket extra information att det är värt att använda 3D. Samtidigt är andra övertygade om att den ytterligare dimensionen ger värdefull information som aldrig kan ersättas med kompletterande tekniker [3].

Tredimensionella visualiseringar visar mer information än tvådimensionella, och om den tredje rumsdimensionen reduceras bort kan viss information bli omöjlig att upptäcka. Den tvådimensionella bilden ger å andra sidan en tydlig helhetsbild, varför en kombination av de två bilderna tillsammans tillför mycket bra information om systemet. Eftersom 3D visualiseringar ger mer information är de väl lämpade för att presentera flerdimensionella och höginformativa datamängder. 2D visualiseringar visar endast utvalda snitt från det tredimensionella rummet och skalar i och med detta bort mycket data. Trots detta kan 2D ge tillräcklig information då den datareduktion som genomförs är erfarenhetsbaserad, och därmed väljs de snitt som tidigare har visat sig vara de mest intressanta. I undersökningarna anges möjligheten till att betrakta data ur fler perspektiv som en annan fördel med interaktiva 3D visualiseringar. Fördelen med interaktivitet är just att betraktaren själv får möjlighet att vrida, vända och navigera i bilden, något som inte är möjligt i en statisk tvådimensionell bild. Nackdelen är att fler frihetsgrader introduceras vilket kan försvåra navigeringen för användaren. Sammanfattningsvis ger 2D en god överblick men visar inte de detaljer som lätt kan upptäckas med 3D [3].

Ett problem med 3D visualiseringar som ofta tas upp i studierna är att de är dyra att använda. Den hårdvara och mjukvara som krävs för dessa visualiseringar är mycket kostsam jämfört med tvådimensionella visualiseringar. Prestandan på vanliga arbetsstationer är ofta tillräcklig för 2D medan dessa är alltför långsamma för att kunna presentera acceptabla 3D visualiseringar [3].

Efter att ha tagit del av ovanstående resonemang drog vi slutsatsen att utveckla en applikation med en tvådimensionell visualisering. Den huvudsakliga anledningen till detta val är att syftet med EasySee är att ge användaren en god överblick över mätdata samt en känsla för klimatet som helhet. Den ytterligare information som en tredje dimension skulle kunna tillföra är jämförbar med en 2D visualisering i kombination

med den redan idag tillgängliga grafrepresentationen, och vi valde därför att inrikta oss på att utveckla ett komplement till dessa. En annan bidragande orsak till valet är att EasySee ska distribueras till användare med tillgång endast till vanliga arbetsstationer, varför vi inte anser att de stora mängder data som behandlas skulle ge ett tillfredsställande resultat i tre dimensioner inom rimlig tid. Utöver det finns för få mätpunkter tillgängliga för att skapa en tillförlitlig 3D visualisering.

3.2 Verktyg

För att ett visualiseringsverktyg ska vara så effektivt som möjligt måste det stödja målgruppens behov. Det är därför mycket viktigt att man har identifierat behoven hos målgruppen för att kunna avgöra hur visualiseringsverktyget ska se ut. Designelement och gränssnitt komponeras sedan efter behoven.

Vi har identifierat följande huvuduppgifter i applikationen:

- Beräkningar och interpolering/extrapolering
- Presentation av data och layoutbyggnad
- Grafik och prestanda
- Filhantering och datadelning

Flera krav bör ställas på ett visualiseringsverktyg. För det första ska det vara enkelt och intuitivt att använda, det vill säga ingen dokumentation vara ska nödvändig och funktionaliteten ska vara självskriven. Vidare är det en fördel om systemet kan kompileras till en körbar fil för att minimera tidsåtgången innan applikationen kan köras. Om det visualiserade resultatet ska användas på en annan plattform än det ursprungligen designades för, bör det även vara möjligt att exportera källkoden och kompilera denna på den nya plattformen. Många av de verktyg som vi fann på marknaden kunde bortses från direkt då de krävde dyra licenser, var designade för specialområden som till exempel meteorologisk och geologisk data eller ej stödde den funktionalitet vi ansåg oss behöva för att hantera de ovan angivna huvuduppgifterna.

Nedan redovisas de verktyg som utvärderades noggrannare samt resultatet av denna utvärdering.

3.2.1 Visualization Toolkit

Visualization Toolkit, VTK, är ett komponentbibliotek för 3D grafik, bildbehandling och visualisering. Till skillnad från flera andra verktyg programmeras varje del av visualiseringen av utvecklaren själv. Paketet är open source och fritt tillgängligt. VTK är objektorienterat och baserat på ett C++ klassbibliotek som innehåller många klasser och avancerade algoritmer. Paketet har också kopplingar till Tcl, Java och Python. Egna användargränssnitt byggs med Tk (utvidgning av Tcl) eller Java GUI klassbibliotek [11].

Den underliggande visualiseringsmodellen baseras på visualiseringspipelinen. Genom att bygga filter och sedan koppla ihop dem med VTKs funktioner skapas applikationerna. VTK lämpar sig bäst för programmerare och visualiseringsutvecklare. En stor fördel med VTK är att det är testat på riktiga tillämpningar, det är alltså inget akademiskt språk som inte lämpar sig väl ute i industrin. En annan fördel är självklart att det är gratis och att det är relativt enkelt att skapa grafik och visualiseringsapplikationer [11].

Tyvärr är grafikmotorn i VTK långsam. Programpaketet är mycket stort och därför krävs det ett snabbt datorsystem med bra grafikkort för att visualiseringarna ska vara snabba att hantera.

3.2.2 MATLAB

MATLAB är ett interaktivt programpaket för tekniska beräkningar och visualiseringar av data, utvecklat av The MathWorks, Inc. MATLAB är kommandostyrt och programmerbart. Den öppna arkitekturen och det intuitiva språket gör programmet relativt enkelt att använda, och ett stort antal kommandon och funktioner finns tillgängliga, men kan även kompletteras med egna funktioner. MATLAB innehåller många verktygslådor för varierande användningsområden, till exempel finns det en verktygslåda för att visualisera 3D data [7].

En stor fördel med programmet är att det har en mängd olika tillämpningsområden och är mycket bra på matematiska beräkningar. Dess användningsområde är alltså inte enbart begränsat till visualiseringar. Matrisen är den grundläggande datatypen och utnyttjas med fördel i egna funktioner för optimerad prestanda.

En annan fördel är att MATLAB kan samverka med andra programspråk. Det går att anropa C och Fortran rutiner från MATLAB och vice versa. På så sätt kan snabba kompilerade program dra nytta av till exempel MATLABs avancerade matrisfunktioner. Omvänt kan flaskhalsar i ett MATLAB program undvikas om delar skrivs i C eller Fortran och därefter kompileras. Sedan MATLAB 6 kan även Java klasser användas. Det ger tillgång till en stor mängd färdigutvecklade verktyg, dels i Javas standardklasser, men även i alla andra klasser som skrivits. För att skapa nya egna Java klasser måste man använda verktyg utanför MATLAB. I MATLAB 6.5 kan man även anropa MATLAB från Java, men denna funktion är fortfarande i experimentstadiet och ingen dokumentation för detta finns i MATLAB [7].

Den största nackdelen med MATLAB är de dyra licenskostnaderna.

3.2.3 Java

Java är ett generellt programspråk avsett för lösning av naturvetenskapliga och tekniska problem. Java har fördelen att vara objektorienterat och säkert, samt även ett förhållandevis enkelt språk.

Java är ett säkert språk och den bytekod som kompilatorn genererar är inte bunden till något särskilt datorfabrikat. Java finns fritt tillgängligt, både utvecklingsmiljö, JDK, och kompilator, JRE, kan gratis laddas ner via Internet. Det finns även tillgång till ett mycket innehållsrikt standardbibliotek, med bland annat klasser för grafiska användargränssnitt, bildhantering och textbehandling.

3.2.4 Layoutverktyg

För konstruktion av användargränssnittet undersöktes möjligheten att underlätta arbetet genom att använda ett layoutverktyg. Vi begränsade utvärderingen till fritt tillgängliga verktyg som dessutom är kompatibla med de ovan nämnda aktuella visualiseringsverktygen.

3.2.4.1 GUIDE

I MATLAB finns utvecklingsmiljön GUIDE (Graphical User Interface Development Environment) för skapandet av grafiska användargränssnitt. GUIDE fungerar som ett

objektorienterat ritprogram, där fönster, knappar och andra komponenter finns att tillgå. Användargränssnittet skapas genom att komponenter valda i GUIDEs meny placeras ut med hjälp av ”drag-and-drop” enligt önskad layout. Komponenternas egenskaper anges sedan i en egenskapseditor. Interaktionen med användaren sker genom så kallade svarsfunktioner, som anropas när användaren aktiverar objektet [7].

Till GUIDEs nackdelar hör att komponentbiblioteket, och därmed layouten av gränssnittet, är begränsat.

3.2.4.2 AbaGuiBuilder

AbaGuiBuilder tillhandahålls av Abacus OpenSource Software Foundation och är ett visuellt layoutverktyg för att skapa användargränssnitt till Java applikationer. Målet med AbaGuiBuilder är att hjälpa utvecklaren att designa och skapa användargränssnitt i Java snabbt och enkelt. AbaGuiBuilder är ett WYSIWYG (what you see is what you get) verktyg som låter användaren placera ut Java Swing komponenter med hjälp av ”drag-and-drop”, och återger sedan detta som en fullt funktionell Java applikation med angivet utseende. Återgivningen sker med hjälp av en speciell tolkningsmotor som generar kod, vilken sedan importeras till systemet [12].

AbaGuiBuilder är gratis endast under en testperiod, vilket är en stor nackdel då arbeten med användargränssnitt ofta sker kontinuerligt under hela projektets gång.

3.2.4.3 LayoutBuilder

LayoutBuilder är ett layoutverktyg för att underlätta skapandet av ett grafiskt användargränssnitt med hjälp av Java komponenter och dess standard layouter. Komponenterna placeras ut med ”drag-and-drop” och samtidigt genereras Java kod för den aktuella designen [13].

LayoutBuildern innehåller endast ett begränsat antal av Javas grafiska komponenter, och är dessutom komplicerat och ej särskilt intuitivt att använda. På grund av de olika layoutmöjligheterna i Java är många layoutverktyg, och även LayoutBuilder, besvärliga och svåra att använda.

3.3 Slutsats

Det pågår mycket forskning för att ta fram hjälpmedel för visualisering som gör det möjligt att snabbt skapa kraftfulla visualiseringar utan att behöva programmera varje del av visualiseringen själv. Detta gör det självklart mycket enklare för användarna att utnyttja konceptet med visualiseringar. De alternativ vi begrundat innebär dock att all funktionalitet programmeras av utvecklaren själv, dels på grund av kostnadsfrågan och dels på grund av att ett beroende av en utomstående part så långt som möjligt vill undvikas.

VTK uteslöts eftersom god kännedom om och erfarenhet av MATLAB och Java fanns sedan tidigare projekt. Dessutom är VTK mer inriktat på tredimensionella visualiseringar samt dess programpaket för stort för att producera tillfredsställande visualiseringar med våra resurser.

Eftersom det finns en naturlig koppling mellan MATLAB och Java, och de båda har sina respektive fördelar för vår visualisering, valde vi att implementera systemet som en kombination av dessa. MATLABs många beräkningsmetoder samt avancerade matrishantering utnyttjas vid temperaturinterpoleringen. MATLAB är dessutom den dagliga arbetsmiljön för de i första hand tänkta användarna av vårt system och detta faktum vägde tungt i beslutet. Användaren får en lägre inlärningströskel samt kan enklare vidareutveckla systemet, dessutom finns det på avdelningen redan tillgång till MATLAB licenser. Till Javas starka sidor hör dess hantering av bilder och grafiska komponenter, vilka utnyttjas för presentation av resultatet. För samarbetet talar också den enkla datadelningen mellan verktygen. För mer detaljer och utförligare genomgång av för- och nackdelar hänvisas till Appendix B.

Vad gäller layoutverktyg föll valet på att egenhändigt implementera användargränssnittet med hjälp av Java Swing komponenter. Detta beslut är möjligen något mer tidskrävande än vid användandet av ett verktyg men fördelarna en egen implementation innebär överväger. Det grafiska användargränssnittet är komplext och innehåller många olika funktioner varför vi vill ha tillgång till alla grafiska komponenter samt kontroll över layouten.

4 Metod

4.1 Visualisering och interpolering

Det finns olika tekniker för att visualisera datamängder. Olika tekniker framhäver olika företeelser i ett dataset, och valet av visualiseringsteknik beror därför på vad för slags information man vill få fram och visualisera. Vid skapandet av visualiseringar finns det alltså en rad faktorer som bör beaktas:

- Vad vill man visa?
- Krävs det detaljerad eller översiktlig information?
- Vilket ändamål har visualiseringen?
- Vilken målgrupp riktar man sig till?

För att skapa den datamängd som ska visualiseras krävs ofta att olika interpolerings- och extrapoleringstekniker appliceras på mätpunkterna. Nästföljande avsnitt redogör för hur mätdata analys utförs på VTEC samt för de visualiserings-, interpolerings- och extrapoleringstekniker som används i EasySee.

4.1.1 Mätdata analys på VTEC

Under utveckling av nya bilmodeller använder man sig i dagsläget av provning och simulering. I och med detta samlar man på sig en stor mängd data. För att provningar och simuleringar ska kunna utnyttjas maximalt krävs att datamängden kan hanteras på ett bra sätt.

I dagsläget arbetar VTEC med att studera mätdata i ett flertal grafer. Behovet av ett visualiseringsverktyg uppkom eftersom man ville få en överblick av klimatet och komma runt problemet med att välja rätt parametrar att titta på och dessutom få en förståelse för hur regleringen fungerade. Visualiseringar kan underlätta i detta avseende för att hitta samband och för att undersöka antaganden kring regleringen.

Visualiseringar kan också användas för att förmedla information till andra. För VTEC innebär det att de får en möjlighet att visa resultat för kunder i produktionsbolagen som inte är insatta i hur klimatregleringen fungerar i detalj.

Genom att förståelseprocessen blir enklare kan också förmedlingen av information till andra gå fortare. Det innebär att produktionstiden kortas drastiskt och därmed minskas kostnaderna. Kortare projekttider möjliggörs genom att simuleringar kan användas mer effektivt då resultaten från dem visualiseras.

4.1.2 Visualiseringsteknik i EasySee

Då förutsättningarna för arbetet analyserats så valdes slicing tekniken bort eftersom metoden lämpar sig bäst för att studera en parameter i taget och VTEC redan har tillgång till bra studier av enskilda parametrar i form av grafer. Också volymrendering valdes bort då denna är beräkningskrävande och därmed inte lämpar sig för onlinevisualiseringar, ett framtida önskemål för VTEC.

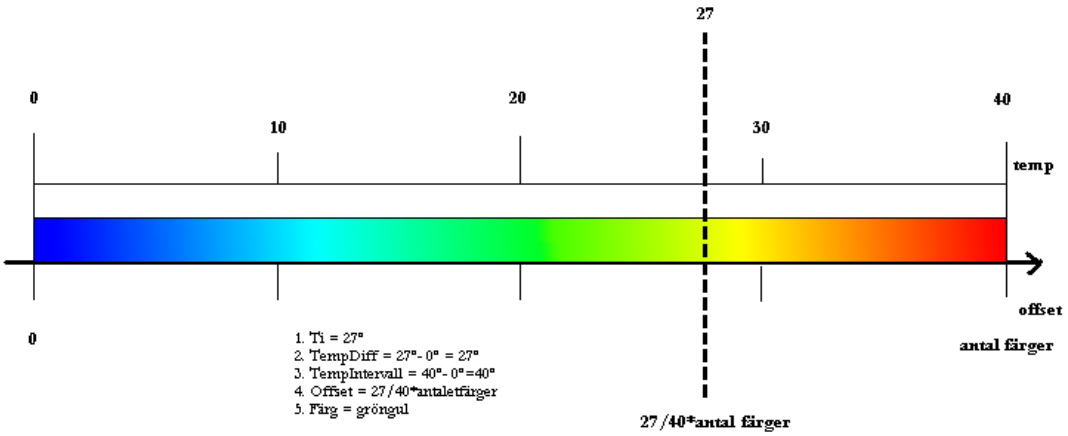
Det bästa alternativet bestämdes därmed till färgmappning med en linjär överföringsfunktion där varje skalärvärde, temperaturen t_i, motsvarar ett färgvärde som ges av överföringsfunktionen. Alla färger i RGB-färgskalan utnyttjas inte, utan de färger som är lämpliga har valts ut. De utvalda färgerna sträcker sig från blått till rött, via grönt, gult och orange. Dock har några av de gröna färgerna tagits bort då dessa annars tenderade till att bli överrepresenterade. Användaren väljer via gränssnittet vilken temperatur som ska motsvaras av blå färg och vilken temperatur som ska motsvaras av röd färg. Detta innebär att alla temperaturer som är lägre än den temperatur som ska motsvaras av blå färg blir blå och alla temperaturer som är större än den temperatur som är röd blir röda. De temperaturer som ligger i intervallet mellan temperaturerna som ställts in får linjärt tilldelade färger i färgintervallet, blått till rött.

Överföringsfunktionen från temperatur till färg fungerar enligt följande algoritm, se även figur 20:

1. Den aktuella temperaturen, T_i, läses in.
2. Temperaturen jämförs med den temperatur som ska motsvaras av blå färg och differensen mellan dessa beräknas.
3. Temperaturintervallet mellan blå och röd färg läses av från gränssnittet.
4. En offset, från färgen som motsvarar lägsta temperaturen, beräknas som (differensen/intervallet)*antalet färger som finns tillgängliga mellan ändfärgerna blått och rött. Denna offset är 0 om temperaturen är densamma som den temperatur som

enligt användaren ska motsvaras av blå färg. Om temperaturen är densamma som den temperatur som ska motsvaras av röd färg så är offseten samma som antalet tillgängliga färger.

5. Utifrån denna offset bestäms sedan vilken färg den aktuella temperaturen ska motsvaras av. Om offseten är mindre än eller lika med noll så är färgen blå, om offseten är större eller lika med antalet färger så är färgen röd. För offseter däremellan så blir färgen linjärt tilldelad utifrån offseten.



Figur 20. Överföringsfunktion för färgmappning

4.1.3 Interpolering och extrapolering

När mätdata för temperaturer i bilen registreras så görs detta endast i de fåtal punkter där termoelement finns placerade. Detta innebär att modeller för bestämning av temperaturen i övriga punkter i kupén respektive evaporatortvärsnittet har satts upp. Då förutsättningarna för kupén och evaporatorn skiljer sig åt har olika strategier för temperaturberäkningarna valts. Dessa strategier redovisas utförligt under följande avsnitt.

4.1.3.1 Kupén

I kupén finns för varje sida i bilen, förar- respektive passagerarsidan, sex stycken icke-symmetriskt utplacerade termoelement vid golv och tak samt ett flertal termoelement vid ventilationerna. Deras exakta placering åskådliggörs i Appendix G. Den avståndsviktade medelvärdesbildning valdes framför exempelvis de ”styckvis linjära trianglarna” för att undvika att behöva dela upp kupén i områden samt för att undgå en separat extrapoleringsstrategi.

För att bestämma temperaturen i en enskild punkt räknas dess avstånd till samtliga termoelement ut. Dessa avstånd avgör hur starkt motsvarande termoelement ska viktas vid temperaturtilldelningen i punkten. Termoelement som ligger nära punkten ska ha större inflytande än de som ligger längre bort, varför det inverterade avståndet till termoelementet används som vikt. Mätpunkterna i den avståndsviktade medelvärdesbildningen utgörs i vår applikation av termoelementen och deras temperaturvärden, där punkterna för vilka temperaturer bestäms är de pixlar i bakgrundsbilden som representerar kupén. Ekvation 2 beskriver hur temperaturer i kupén bestäms.

$$T(x_p) = \frac{\sum_{i=1}^{n} T(x_i) \cdot \frac{1}{d}}{\sum_{i=1}^{n} \frac{1}{d}}$$

där: $T(x_p)$ = interpolerade temperaturen i punkt p

n = antalet termoelement

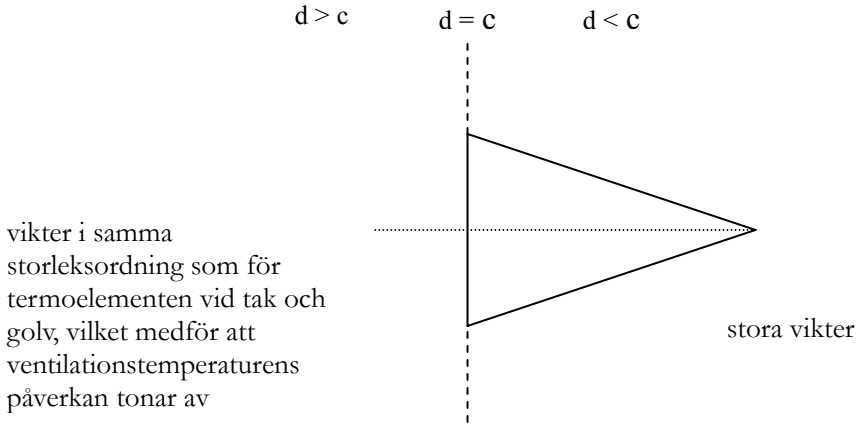
$T(x_i)$ = temperaturen i termoelement i

d = avståndet mellan termoelement i och punkt p

Ekvation 2. Temperaturtilldelning i kupén.

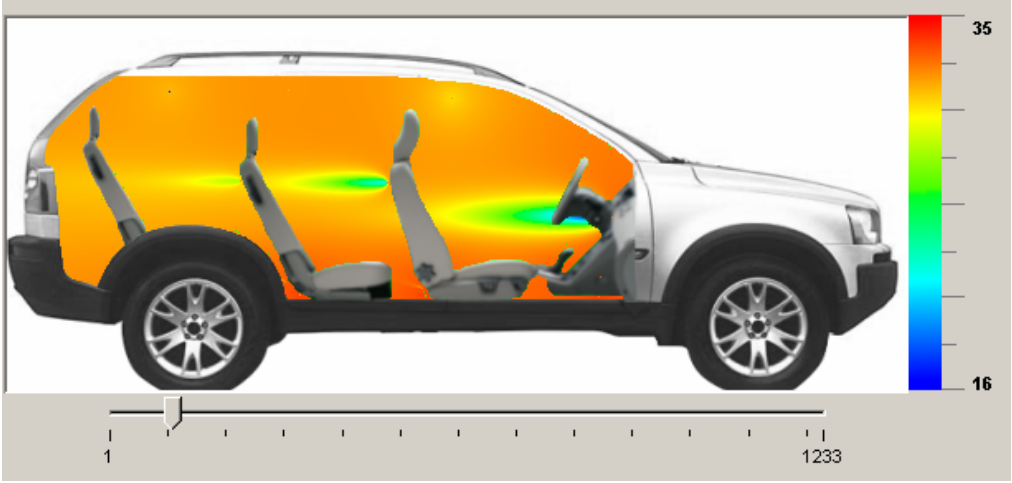
För termoelementen som är placerade vid tak och golv används sambandet som det visas ovan. För de termoelement som sitter vid ventilationerna kan viktningen ej bestämmas på samma sätt då dessa värden ej rimligen ska påverka punkter i motsatt riktning från ventilationsmynningen. För att likväl kunna använda ekvation 2 för alla typer av termoelement, används samma inverterade viktning även för termoelementen vid ventilationerna. Beräkningen av deras avstånd till punkten har dock modifierats något för att ge en korrekt representation av de rådande förhållandena. I punkten som motsvarar termoelementets exakta position sätts temperaturen till termoelementets temperatur, utan att påverkan från andra punkter viktas in.

För att beskriva ventilationerna samt avstånden till dessa utnyttjas den trigonometriska funktionen *cosh*, en funktion som har de egenskaper vi, enligt ovan, eftersöker. Till skillnad från termoelementen vid tak och golv, där avstånden beräknas enbart med Pythagoras sats, multipliceras det egentliga avståndet med cosh-uttrycket för den aktuella vinkeln. Den aktuella vinkeln utgörs av ventilationens position jämfört med ursprungsläget. Faktorerna i funktionen kan trimmas för att ge termoelementen vid ventilationerna en rimlig viktning jämfört med de termoelement som sitter vid tak och golv. Med hjälp av faktorerna kan också ventilationernas utsträckning bestämmas, för ytterligare detaljer angående cosh-funktionen hänvisas till Appendix E. Figur 21 visar en förenklad version av hur ett termoelement vid ventilation viktas jämfört med termoelement vid tak och golv.

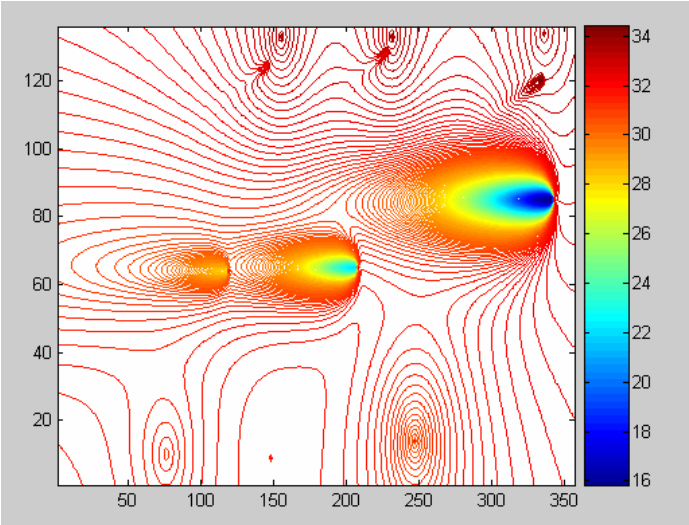


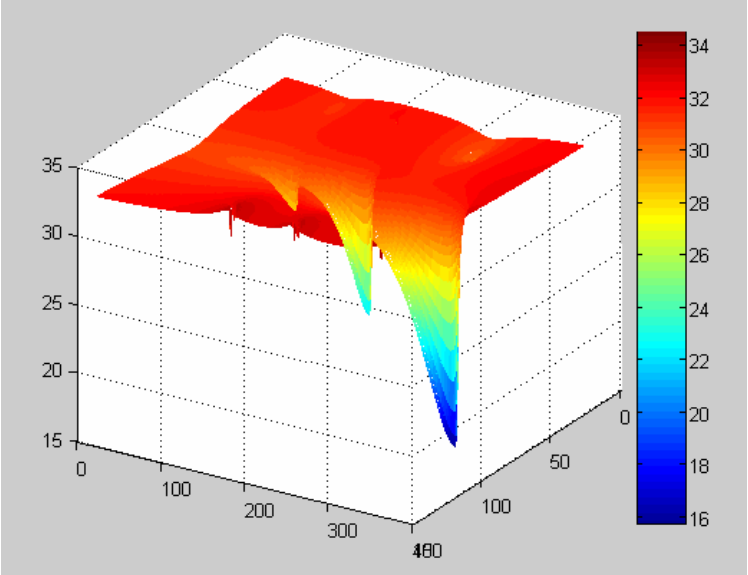
Figur 21. Principiell skiss av hur ventilationernas temperaturer viktas mot temperaturerna vid termoelementen vid tak och golv, där c är ventilationens kastlängd.

I nedanstående figur visas hur de interpolerade temperaturerna i kupén åskådliggörs i användargränssnittet vid det aktuella samplet. De ytor som återfinns i figur 23 visar på temperaturernas utseende med hjälp av plottar gjorda i MATLAB för att verifiera att inga felaktiga eller oväntade temperaturer bestämts av den avståndsviktade medelvärdesbildningen. Observera att den översta ytan i figur 23 bör roteras för att ej ge ett missvisande resultat. Det koordinatsystem vi använt vid visualiseringen utgår från bildens översta vänstra hörn, med ökade värden på x-axeln åt höger och ökande värden på y-axeln nedåt. I ytan i figur 23 motsvarar alltså temperaturer för låga y-värden temperaturer vid taket i bilen, och temperaturer i ytans ovankant motsvarar golvtemperaturer.



Figur 22. Kupén vid aktuellt sampel

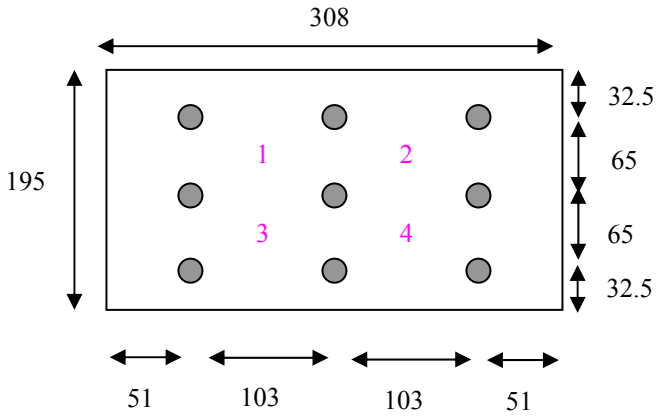




Figur 23. Ytor över interpolerade temperaturerna vid aktuellt sampel

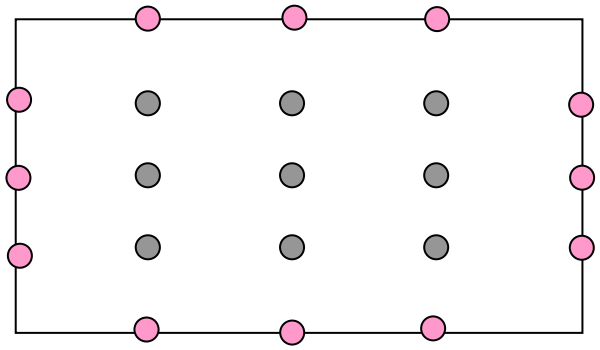
4.1.3.2 Evaporatorn

I evaporatorn finns nio stycken termoelement, rektangulärt utplacerad enligt figur 24 nedan.



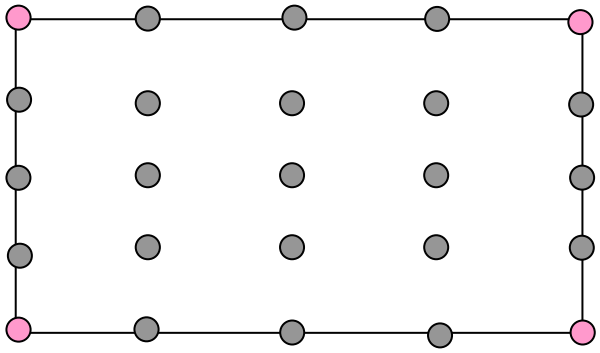
Figur 24. Termoelementens placering i evaporatorn. Avstånden i mm.

Alla övriga punkters temperatur bestäms genom att plana ytor bildas mellan fyra punkter. För de punkter som ligger mellan den nio termoelementen, i områdena 1- 4, definieras planen av de fyra närmast liggande termoelementen. För övriga punkter har hjälpelement adderats, enligt figur 25, för att på samma sätt kunna definiera plan även för dessa punkter.



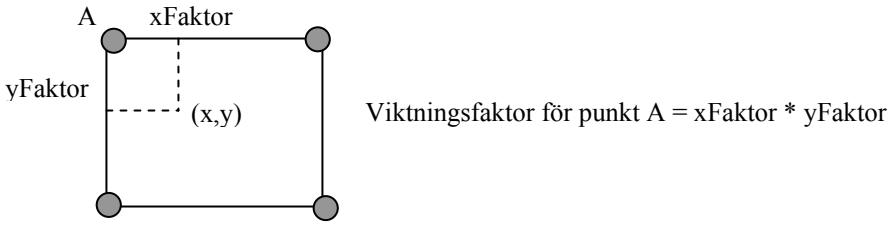
Figur 25. Evaporatorn med hjälpelement utplacerade.

Notera att hjälptermoelementen har lagts till utanför evaporatorns egentliga gränser, för att bilda ett symmetriskt rutnät där alla avstånd i x-led är lika och alla avstånd i y-led är lika. Slutligen adderas hjälptermoelement till hörnorna, som figur 26 visar, varefter ett komplett rutnät med termoelement finns tillgängligt.



Figur 26. Evaporatorn med hjälpelement utplacerade i hörnorna.

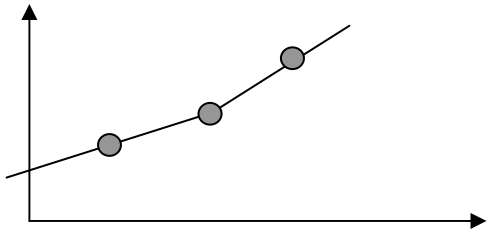
En enskild punkt tillhör det plan som spänns upp av de fyra mest närliggande termoelementen. Om en punkt skulle ha flera termoelement på samma avstånd väljs det element som ligger närmast i positivt x-led och/eller i positivt y-led. Positivt x-led definieras från vänster till höger, och positivt y-led definieras uppifrån och ner. För varje punkt sparas dess fyra tillhörande termoelement undan, tillsammans med viktningsfaktorer för respektive element. Dessa faktorer anger hur mycket varje termoelements temperatur ska viktas för att bestämma temperaturen i den aktuella punkten. Följande exempel beskriver hur en viktningsfaktor beräknas, faktorerna för övriga termoelement bestäms på samma sätt:



Figur 27. Bestämning av viktningsfaktor.

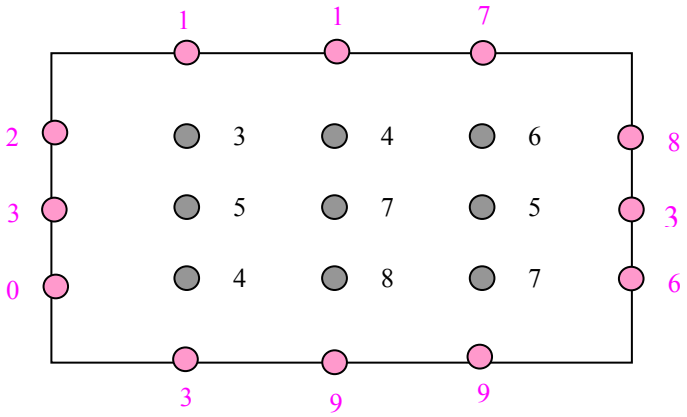
De ovan beskrivna förberedelserna sker i applikationens initieringsfas och utnyttjas sedan vid varje uppdateringstillfälle. Vid varje uppdatering läses temperaturerna för de nio ursprungliga termoelementen in från mätdata, och utifrån dessa bestäms sedan temperaturerna i hjälpelementen. Hjälpelementens temperaturer beräknas antingen med extrapolering eller utan.

Då extrapolering ska utnyttjas beräknas temperaturen i hjälptermoelementen med hjälp av linjär extrapolering, det vill säga temperaturen sätts utifrån de, i samma led, två närmaste termoelementens skillnad. Denna metod kan illustreras enligt figur 28.



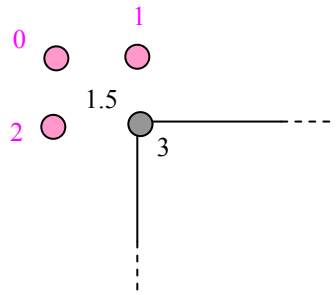
Figur 28. Illustration av temperaturtilldelning med extrapolering.

För ytterligare förtydligande, se exempel nedan.



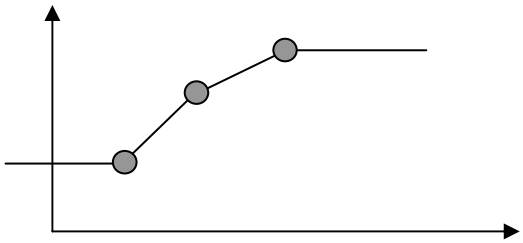
Figur 29. Exemplifiering av temperaturtilldelning med extrapolering Temperaturer i °C..

För temperaturtilldelning i hörnpunkterna utnyttjas det plan som definieras av de tre mest närliggande termoelementen, och temperaturen bestäms följaktligen så som nedanstående exempel anger.



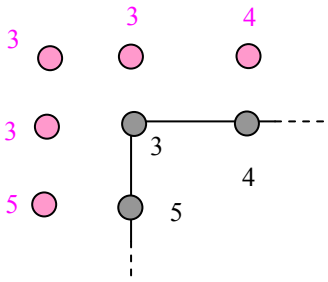
Figur 30. Exemplifiering av hörnpunkternas temperaturtilldelning vid extrapolering. Temperaturer i °C..

Vid de tillfällen då extrapolering ej används tilldelas hjälptermoelementen, även hörnorna, samma temperatur som närmsta ursprungliga termoelement. Eftersom varje enskild punkts temperatur fortfarande bestäms med hjälp av viktningsfaktorer undviks ett randigt utseende, och endast områden vid hörnpunkterna kommer att få en konstant temperatur. Nedanstående graf kan jämföras med illustrationen i figur 31 för att åskådliggöra skillnaden i temperaturtilldelning jämfört med när extrapolering används.



Figur 31. Illustration av temperaturtilldelning utan extrapolering.

För att ytterligare förtydliga hur termoelementens temperaturer bestäms följer ett enkelt exempel.



Figur 32. Exemplifiering av hjälpelementens temperaturtilldelning utan extrapolering. Temperaturer i °C..

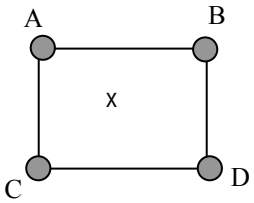
Det finns flera anledningar till att båda alternativen, med eller utan extrapolering, har implementerats, men det primära skälet är att då extrapoleringen alltjämt är godtycklig vill man kunna studera resultatet ur flera utgångspunkter för att kunna göra en korrekt analys av dess betydelse.

Den ovan nämnda strategin, att använda ett expanderat rutnät där alla termoelement i samma led är placerade på lika avstånd, medför att vi kan använda samma algoritm för alla punkter. Det vill säga att de punkter som egentligen finns på halva avståndet behöver inte specialbehandlas utan detta sköts automatiskt av algoritmen. Punkter utanför evaporatorns ursprungliga storlek visas ej upp i användargränssnittet.

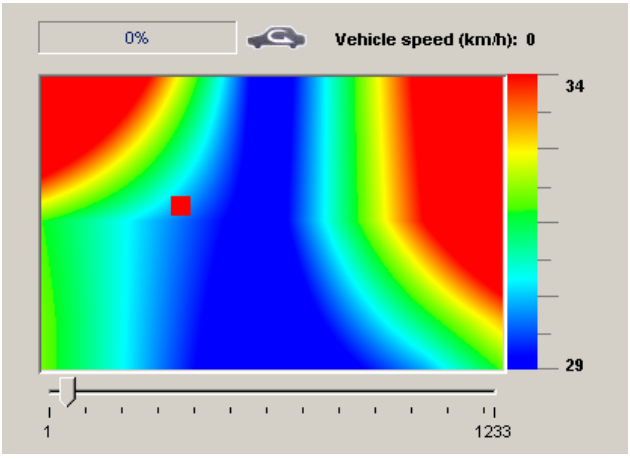
Temperaturen för varje punkt bestäms slutligen som en summering av respektive mätpunkts inverkan enligt ekvation 3.

$$temperatur_x = \sum_{i=0}^{4} factor_i \cdot temperatur_i$$

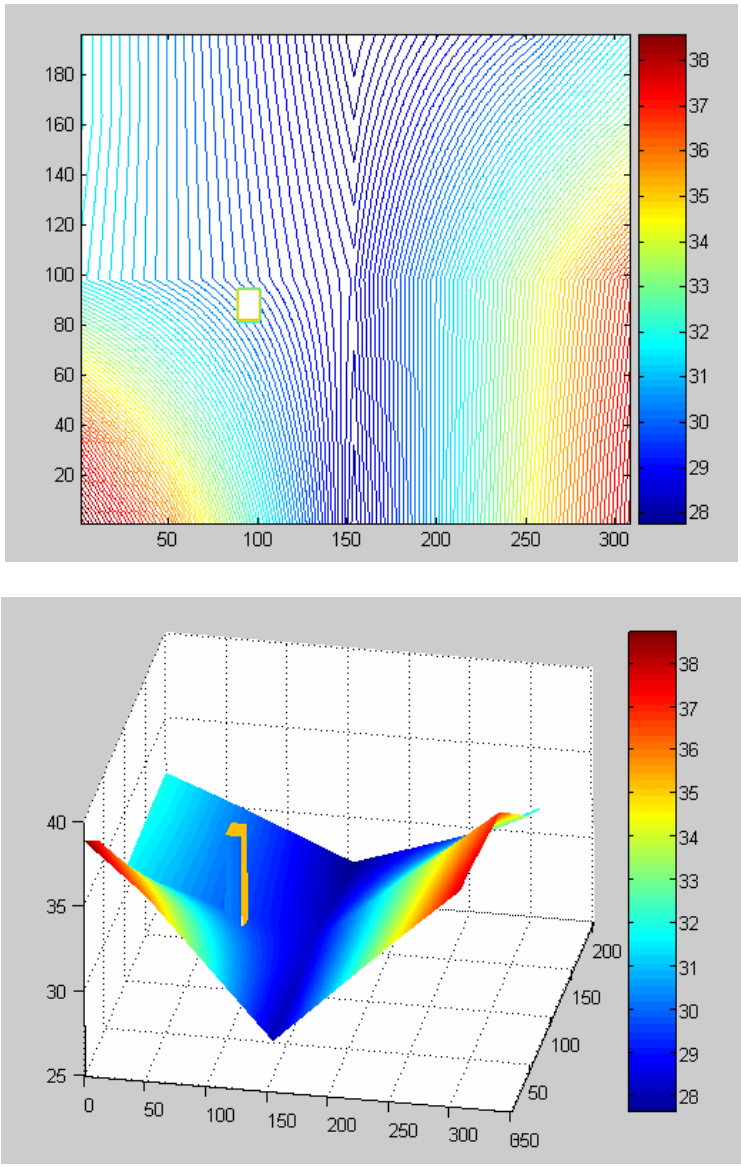
Ekvation 3. Temperaturbestämning



I nedanstående figur visas hur de interpolerade temperaturerna i evaporatortvärsnittet åskådliggörs i användargränssnittet vid det aktuella samplet. De ytor som återfinns i figur 34 visar på temperaturernas utseende med hjälp av plottar gjorda i MATLAB för att verifiera att inga felaktiga eller oväntade temperaturer bestämts med interpolerings- och extrapoleringsmetoderna. På samma sätt som för kupén bör den översta ytan i figur 34 roteras för korrekt återgivelse.



Figur 33. Evaporatorn vid aktuellt sampel.



Figur 34. Ytor över interpolerade temperaturer vid aktuellt sampel.

4.2 Gränssnittet till EasySee

Gränssnittet till applikationen utvecklades i en iterativ process där varje version tillkommit i samråd med de tänkta användarna. Faktorer som styrt utvecklingen har varit användarnas önskemål på funktioner och utseende, men också de vanligaste kognitiva aspekterna.

Från början presenterades flera olika idéer för hur gränssnittet skulle kunna se ut och därefter diskuterades en bra första layout fram. Denna har sedan successivt förändrats då ny funktionalitet har adderats till systemet, men huvudkomponenterna är fortfarande desamma.

En iterativ designprocess anses vara nödvändig för att åstadkomma ett bra interaktivt gränssnitt, men det finns ett stort problem: tid. Den tid som läggs på en iterativ design ska kompenseras med det faktum att den slutliga versionen är mindre sannolik att misslyckas då den har testats och förbättrats innan release.

Det är dock aldrig möjligt att fullt ut förutse hur ett gränssnitt kommer att användas eller vilka funktioner och valmöjligheter användaren förväntar sig. Flera visningar av de olika versionerna har lett fram till nya önskemål om funktionalitet som har gjort att layouten kontinuerligt förändrats. Varje ny version har därmed inneburit att gränssnittet förbättrats och gjorts mer komplext.

4.2.1 Designval

I ett försök att uppnå ett lärbart gränssnitt har de funktioner som finns tillgängliga tydliggjorts, dels genom att rama in funktionen med en beskrivande text, dels genom att samma text använts för motsvarande funktion på alla blad. Detta för att användaren snabbt ska uppfatta ett mönster och därmed lära sig strukturen på gränssnittet. Välkända standarder har utnyttjats då ikoner utformats för att användaren ska kunna öka inlärningshastigheten.

Flexibiliteten i gränssnittet har hållits begränsad i den mån att användaren ej kan ändra storlek på eller minimera delar i fönster. Inte heller kan bakgrundsfärg eller font på rubriker ändras. Detta beror på att projektet genomfördes på en begränsad tid, men

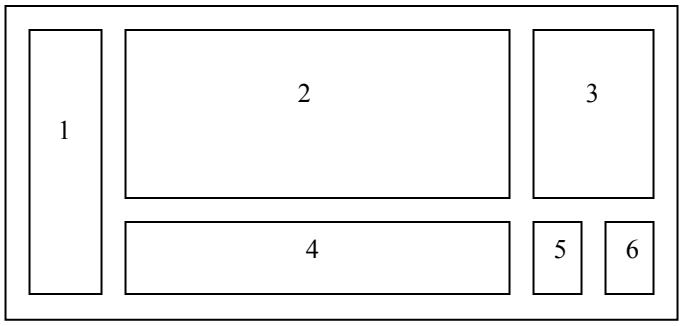
också på att användaren ej ska behöva lägga tid på att göra sådana inställningar. Istället har försök gjorts för att komma fram till den mest fördelaktiga layouten.

Interaktionen med användaren vad gäller bekräftelse av operationer visas kontinuerligt i gränssnittet där användaren alltid ser vilka inställningar som är aktuella. Istället för dialoger som kräver bekräftelse på valda inställningar så är det lätt för användaren att ändra tillbaka till tidigare inställningar. Inga katastrofala inställningar kan göras eftersom alla inmatningar i gränssnittet har definierade inställningsalternativ.

Gränssnittet använder engelska som arbetsspråk, samt där det är möjligt utnyttjas begrepp som de framtida användarna använder i sitt dagliga arbete.

4.2.2 Layout

Den layout som valts på användargränssnittet är främst motiverad av det faktum att programmet ska vara användarvänligt. Gränssnittet kräver dessutom mycket skärmyta då det är mycket information som ska visas. Därför består det av 3 blad, `ECC`, `Climate Unit` samt `Evaporator`, mellan vilka man bläddrar genom att välja respektive blads flik. Det första bladet med `ECC` visar slutresultatet från klimatanläggningen, den temperatur passagerarna i bilen omges av, samt vilka klimatinställningar passagerarna gjort på panelen. De övriga två bladen visar delsteg i klimatanläggningen, en översikt av AC:n samt en detaljstudie av tvärsnittet efter evaporatorn. Alla blad har samma uppdelning av fönstret för att användaren enkelt ska hitta de olika funktionerna på respektive blad.



Figur 35. Schematisk bild över gränssnittets layout

Eftersom mycket information ska finnas tillgänglig i gränssnittet krävdes bra val av komponenter.

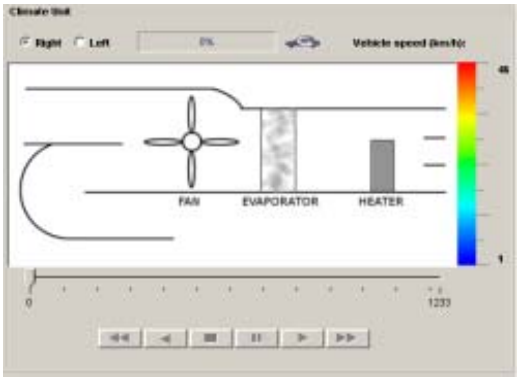
4.2.2.1 Del 1

För varje blad visas all mätdata som finns tillgänglig i mätfilen i en tabell i vänsterkanten (1 i figur 35). På vänstersidan visas signalnamnet och till höger visas det aktuella värdet på signalen.

4.2.2.2 Del 2

Den del som visar temperaturer (2 i figur 35) är utformad som en mediaspelare, se figur 36 och 37.

Figur 36. Mediaspelaren på ECC bladet



Figur 37. Mediaspelaren på Climate Unit bladet

Utformningen motiveras av de funktioner som önskats, såsom möjligheten att stoppa, pausa samt snabbköra uppspelningen av mätdata. Mediaspelaren har knappar för

rewind, reverse, stop, pause, play samt forward. Dessa knappar har de sedvanliga ikonerna för respektive funktion för att utnyttja igenkänning för inlärningen, se figur 38.

Figur 38. Knappsatsen med dess ikoner

Utöver knapparna har användaren också möjlighet att styra uppspelningen genom en list som förutom att ge användaren möjlighet att förflytta sig i mätfilen också visar hur mycket av mätfilen som avverkats och hur mycket som är kvar.

I denna del av användargränssnittet visas också recirkulationsgraden och bilens hastighet enligt figur 39 nedan. Recirkulationsgraden som varierar mellan 0 och 100 procent illustreras med en `ProgressBar` som är en lämplig komponent för procentillustrationer. Anledningen till att recirkulationsgraden visas är att den har stor påverkan på bland annat temperaturen och luftfuktighet i kupén, och därför kan ge stora skillnader i temperaturbeteendet. När man studerar mätdata är det nödvändigt att veta om beteendet beror på recirkulationsgraden eller någon annan faktor. Bilens hastighet är viktig då den avgör hur mycket värme som krävs för att erhålla en viss temperatur. Ju fortare man kör desto mer omges bilen endast av omgivningstemperaturen, medan om man kör långsamt så påverkar bilen omgivningstemperaturen. Vid kallt väder behövs det mer värme för att hålla önskvärd temperatur när man kör snabbt, medan vid varmt väder så värmer den varma fartvinden bilen och det behövs följaktligen mer kyla för att hålla önskvärd temperatur.

Figur 39. Recirkulationsgrad och vagnhastighet

I denna del av gränssnittet kan man även via radioknappar välja om man vill se höger eller vänster sidas temperaturer. Radioknapparna är uteslutande eftersom man endast kan se en av sidorna åt gången.

4.2.2.3 Del 3

På varje blad finns en del för användarinställningar (3 i figur 35) där användaren ställer in de förutsättningar som ska gälla för uppspelningen av data. Inställningarna underlättas av ”drop-down” menyer som sparar plats och förser användaren med de alternativ som är valbara.

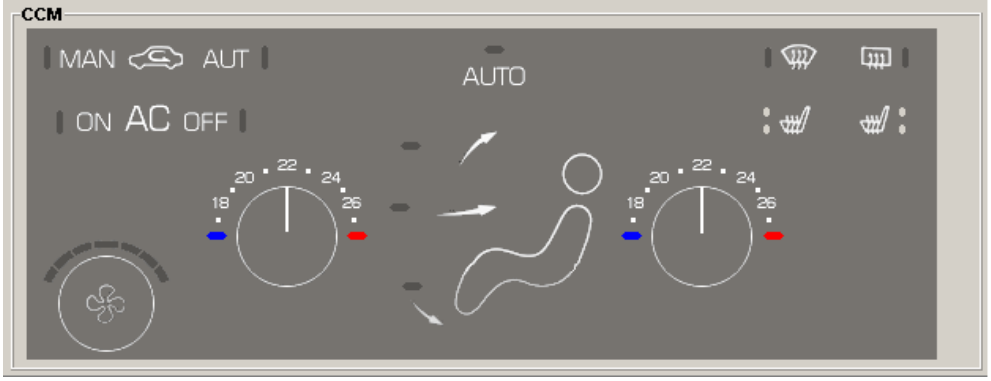
För `ECC` bladet så finns inställningsmöjligheter för färgtilldelning, ventilationsmunstyckenas utblåsvinkel samt visningstillstånd (displaymode), se figur 40. För `Climate Unit` och `Evaporator` görs inställningar istället för färgtilldelning samt grafutseende. För `Evaporator` finns möjlighet att också ställa in displaymode.

Figur 40. Inställnings del för ECC bladet, Climate Unit bladet samt Evaporator bladet

4.2.2.4 Del 4

I den nedre delen av gränssnittet (4 i figur 35) visas för ECC bladet en bild över reglageinställningarna på CCM panelen i bilen, se figur 41. Inställningarna som gjorts på panelen är en beskrivning av det kupéklimat passageraren önskar och det mål man vill uppnå med klimatregleringen. För att kunna analysera resultatet av regleringen måste passagerarens önskemål vara kända. Vid plötsliga temperaturförändringar i

kupén vill användaren snabbt kunna kontrollera om förändringarna beror på en, av användaren, ändrad inställning på panelen eller på en orsak som bör åtgärdas.

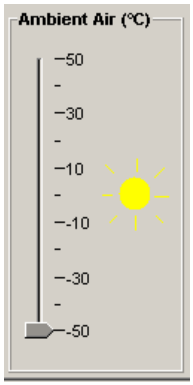


Figur 41. CCM panelen på ECC bladet

För de andra bladen visas istället en graf där utvalda signaler plottas. För `Climate Unit` bladet plottas värdet hos signalen för sensorn som beskriver evaporatortemperaturen tillsammans med signaler för medelvärdet av effekten hos defroster-, panel- respektive golvventilationerna. De tre sista är resultatet av arbetet i denna del av klimatanläggningen. På `Evaporator` bladet plottas värdet hos signalen för sensorn som beskriver evaporatortemperaturen tillsammans med min-, max- och medeltemperaturen av evaporatorns termoelement.

4.2.2.5 Del 5

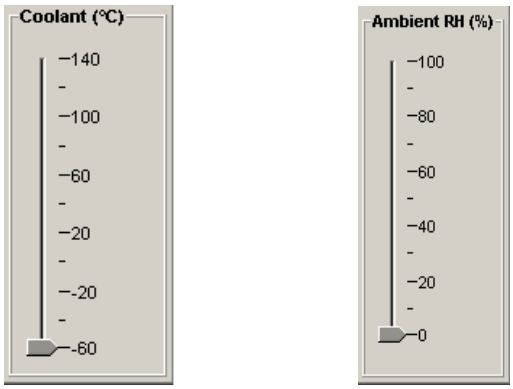
I nedre högra hörnet av användargränssnittet (5 i figur 35) visas utomhustemperaturen tillsammans med det aktuella vädret, se figur 42. Temperaturen representeras med lodräta `Sliders` för att ge intrycket av en termometer. Utomhustemperaturen är viktig av den enkla anledningen att omgivningen påverkar temperaturen i kupén, ett varmt utomhusklimat ger en ökning av kupétemperaturen och ett kallt utomhusklimat medför en temperatursänkning. Vädret, som visas med intuitiva ikoner, har betydelse för kupéklimatet eftersom sol värmer kupén oavsett utomhustemperatur medan regn visar på ökad luftfuktighet.



Figur 42. Representation av omgivningsväder

4.2.2.6 Del 6

För ECC bladet finns en termometer för kylarvattnet medan luftfuktigheten redovisas på de båda andra bladen i användargränssnittet (6 i figur 35), se figur 43. Kylarvattnets temperatur är intressant vid analys av klimatregleringen eftersom en kall motor aldrig kan avge värme, och det därför inte är meningsfullt att försöka öka temperaturen med hjälp av varm luft förrän motorn är varm.



Figur 43. Kylarvattnets temperatur respektive luftfuktigheten

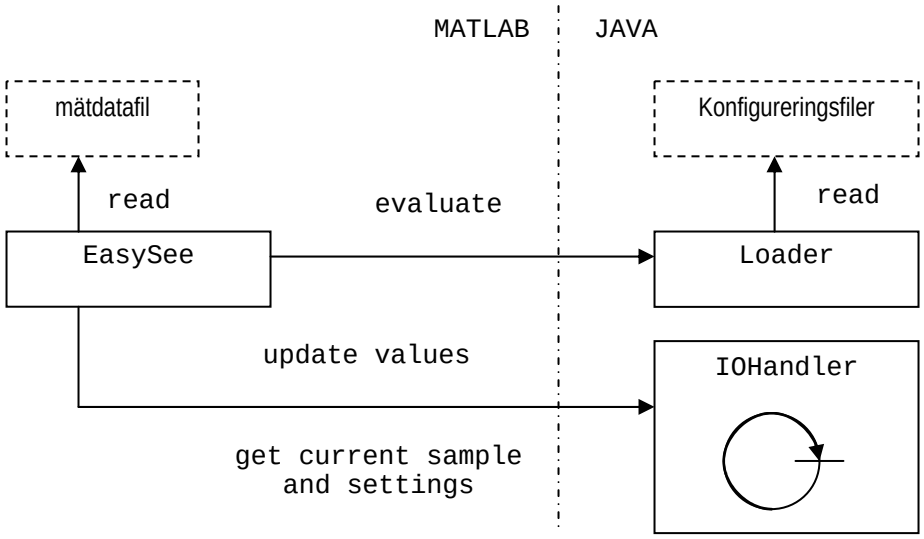
Luftfuktigheten visas eftersom varm, fuktig luft belastar AC systemet mer än varm, torr luft. Därför kräver AC systemet vid hög luftfuktighet ett högre varvtal och drar därmed mer energi för att nå en viss temperatur. Då AC systemets varvtal är begränsat medför detta att det inte alltid är möjligt att uppnå den önskvärda temperaturen.

4.4 Design och systemstruktur

Vad gäller design och systemstruktur utnyttjas följande strategi i verktyget EasySee: programmet startas med MATLAB scriptet `EasySee` och för varje sampel utförs här samtliga beräkningar genom anrop av diverse funktioner. De beräknade temperaturerna samt övriga värden skickas till Java applikationen, vilken sedan hanterar presentationen av dessa i användargränssnittet.

Nedan följer en principiell skiss över systemets uppbyggnad samt en översiktlig beskrivning av hur kommunikationen mellan de olika delarna sker. För fullständigt klassdiagram se Appendix D.

4.4.1 Kommunikation mellan MATLAB scriptet och Java applikationen



Figur 44. Principiell skiss över kommunikation mellan MATLAB scriptet och Java applikationen.

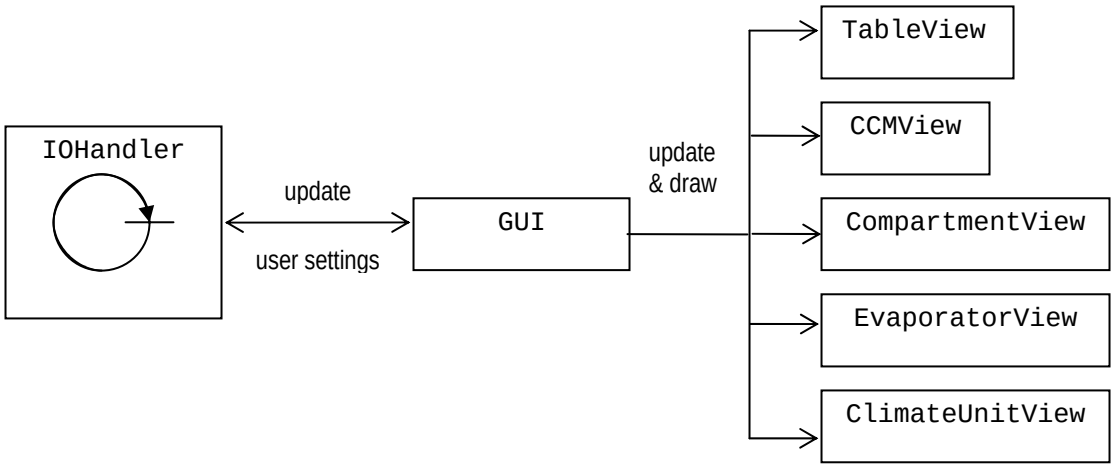
Systemets initieringsfas inleds med att önskad mätdatafil väljes. Signalerna i denna fil jämförs sedan med de signaler som `Loader` klassen läst från systemkonfigureringsfilen. Vi har valt att hantera konfigureringen med hjälp av Java

applikationen då MATLAB ej erbjuder någon funktionalitet för ett enklare tillvägagångssätt. I systemkonfigureringsfilen anges de signaler som är nödvändiga för att systemet ska kunna utföra fullständiga beräkningar, och om alla dessa finns tillgängliga fortsätter systemet att registrera de från användaren förinställda parametrarna som ska gälla vid exekveringen. Bland dessa parametrar i användarkonfigureringsfilen finns inställningar för ventilationernas utseende. Gränssnittet mellan MATLAB och Java, det vill säga `IOHandler`, skapas därefter. All kommunikation har samlats i `IOHandler` för att undvika cirkulära beroenden. Initieringsfasen avslutas med att uppgifter om termoelementens placering hämtas från `IOHandler` och sparas i MATLABs workspace.

För varje nytt sampel hämtas applikationens aktuella visningstillstånd, det vill säga vilka förutsättningar som gäller vid de beräkningar som ska utföras. Exempelvis ventilationernas utblåsvinklar samt om höger eller vänster sida av bilen ska åskådliggöras. För `ECC` beräknas och skickas temperaturer för kupén till `IOHandler`, dessutom skickas CCM inställningarna för aktuellt sampel. För `Climate Unit` skickas termoelementens temperaturer, värden för spjällens position samt data för grafrepresentation. För `Evaporator` skickas temperaturer för evaporatortvärsnittet samt data för grafrepresentation. Gemensamt för de tre huvuddelarna av användargränssnittet är att de förutsättningar som gäller för omgivningen samt tabellen uppdateras med värden för det aktuella samplet och skickas till `IOHandler`. Systemet kan även befinna sig i ett tillstånd där inga beräkningar behöver göras, exempelvis om användaren har stoppat eller avslutat applikationen.

Kommunikationen mellan `EasySee` och Java applikationen är enkelriktad. `EasySee` utnyttjar polling för att ta del av data från `Loader` och `IOHandler`, då anrop från Java klasser till MATLAB funktioner ej har något dokumenterat stöd. Eftersom verktyget ej hanterar några tidskritiska data är polling ett fullgott alternativ.

4.4.2 Presentation av data



Figur 45. Principiell skiss över dataflödet vid den grafiska presentationen.

`IOHandler` är en tråd och anger vilket sampel som är det aktuella. För varje sampel uppdateras `GUI` med de från `EasySee` erhållna värdena, det vill säga temperaturer, omgivningsdata, tabellvärden samt övriga inställningar. `GUI` uppdaterar i sin tur respektive vy med motsvarande värden och ritar därefter upp dessa komponenter i användargränssnittet. Innan uppritningen sker har vyerna behandlat mottagna värden enligt de regler som gäller för representation och färgkodning. Anledningen till att dessa vyer har brutits ut ur `GUI` är att de innehåller mer komplex logik och ett mer avancerat uppritningsförfarande än övriga grafiska komponenter.

För att säkerställa att `GUI` ej läser data från `IOHandler` samtidigt som `EasySee` skriver nya värden till samma variabel, skyddas de gemensamma datastrukturerna med `synchronized` monitorer.

När användaren gör en förändring av `GUI`s tillstånd skickas de nya förutsättningarna till `IOHandler` och hämtas vid nästa sampel av `EasySee` innan nya beräkningar utförs.

5 Resultat och slutsatser

Examensarbetet resulterade i visualiseringsverktyget EasySee med stora möjligheter att underlätta arbetet med analys av mätdata. Verktyget kommer antagligen att fungera som ett komplement till dagens hjälpmedel och bidra till en snabbare överblick av klimatet i fordon.

Att svara på problemställningen ”Är det möjligt att med hjälp av visualisering underlätta analys av klimatmätdata?” är svårt eftersom verktyget ej tagits i bruk ännu. Dock är uppfattningen från avdelningens sida att de kommer att kunna använda vår version av EasySee, men vidareutvecklar samtidigt gärna systemet till att fungera online och tillsammans med verktyg som används för simulering idag. Detta kommer eventuellt att innebära att de väljer att låta nya examensarbeten bygga på EasySee.

Vad gäller utformningen av användargränssnittet så har vi valt layout och funktioner enligt framtida användares önskemål samt efter vissa kognitiva förhållningsregler. Det har förhoppningsvis bidragit till att verktyget är användbart och därmed kommer att utnyttjas.

Verktyget har fungerat väl vid testning och användning, dock är mätdata i många fall ofullständig och saknar de nödvändiga signaler som programmet kräver. Därför har det varit svårt att hitta mätdata att testköra programmet med. Tidskraven på verktyget är relativt låga och fokus har legat på att presentationen ska se bra ut, inte att exakt alla sampel presenteras.

Generellt så kräver programmet mycket CPU men inga märkbara problem med andra program har märkts under exekveringen.

Tyvärr så har det ej funnits möjlighet att kontrollera om de temperaturer i kupé och evaporator som interpolerats fram stämmer överens med de verkliga temperaturerna. En sådan studie skulle innebära att ytterligare termoelement skulle ha behövts installeras i provvagnarna vid upptagning av mätdata. Då den elektriska utrustning som varje termoelement innehåller är kostsam fanns inga sådana möjligheter. Våra resultat har därför validerats utifrån de kunskaper och erfarenheter som våra

handledare skaffat sig under många års iakttagning av temperaturfördelningen i fordon.

6 Diskussion och rekommendationer

6.1 Alternativa tillvägagångssätt

Det som grundlade vårt angreppssätt på uppgiften var beslutet angående vilket verktyg vi skulle använda. Beslutet att använda en kombination av MATLAB och Java, med funktionsuppdelningen att MATLAB applikationen sköter interpoleringen medan Java applikationen hanterar användargränssnittet med uppritning och användaraktiviteter, begränsade våra möjligheter och flexibilitet. En mer noggrann verktygsutvärdering kunde ha utförts där dels fler verktyg utvärderats, dels möjligheten att implementera programmet helt i MATLAB testats ytterligare. Dock hade antagligen en fullständigt MATLAB baserad applikation tvingat oss att använda verktygslådan Image Processing Toolbox, vilken kräver dyra licenser. Alternativet att använda färdiga layoutverktyg hade inneburit mindre möjligheter att skapa ett skräddarsytt användargränssnitt.

Antalet alternativa lösningar till interpoleringen är oändligt många. Den avgörande anledningen till vårt val av interpoleringstekniker var att de ger ett tilltalande resultat vid uppritningen. Interpoleringsmetoderna för kupén och evaporatorn skiljer sig väsentligt åt i idé och utförande. Till kupén valdes en metod som baseras på avståndsviktning där ventilationernas påverkan innebär att avståndet manipuleras. Denna metod passar väl eftersom kupén har godtyckligt utplacerade termoelement men också områden som har en relativt okänd temperatur då inga termoelement är placerade där, till exempel längst bak i bilen. För evaporatorn så är termoelementen rektangulärt utplacerade och en interpoleringsmetod som delar in tvärsnittet i tydliga områden passade bättre. Inom varje område är samtliga punkter definierade av det plan som spänns upp av fyra termoelement i hörnen på området. Man hade kunnat tänka sig att göra en liknande interpolering i kupén, där termoelementen får spänna upp tydligt definierade plan och där man för punkter utanför dessa plan bestämmer temperaturen enligt någon extrapoleringsmetod. Problemet med att definiera områden är att det blir motsvarande mönster i färgläggningen med tydlig gränsmarkering mellan områdena. För evaporatorn hade man kunnat tänka sig att man använt MATLABs inbyggda interpoleringmetoder, såsom `linear`, `nearest` och `cubic spline`, för

området innanför de yttre termoelementen och enbart implementerat extrapoleringsmetoder för ramen kring detta område.

Användargränssnittets layout hade kunnat utformas annorlunda men utifrån de funktioner som önskades har vi tagit fram den layout som vi anser fungerar bäst. Den bakomliggande strukturen bygger på flikade sidor, `TabbedPanes`, men kunde ha valts till exempelvis interna fönster, `InternalFrames`, där respektive delfönster hade kunnat minimeras och flyttas runt individuellt. Risken är dock stor att en sådan struktur blivit något rörig och mindre lättanvänd.

Vad gäller systemstruktur, design och implementation har vi kontinuerligt tvingats att göra mindre ändringar allteftersom ny funktionalitet tillkommit. Det viktigaste beslutet som togs gällde var gränsdragningen för samarbetet mellan MATLAB och Java skulle placeras. Flera alternativ diskuterades, där utgångspunkten var att MATLAB applikationen skulle ansvara för interpoleringen för att utnyttja dess beräkningskraft och för att avdelningen lätt skulle kunna förbättra interpoleringsmetoderna. Istället för att, som i dagsläget, låta Java applikationen skapa visualieringsbilderna utifrån mätdata, kunde vi ha valt att låta MATLAB applikationen producera färdiga bilder som sedan skickas till Java applikationen för uppvisning. Alternativt kunde vi ha låtit MATLAB applikationen skapa matriser med färgkoder istället för temperaturmatriser.

För att undvika att onödig, i betydelsen samma, information skickas flera gånger kunde kontroller mot föregående temperaturmatris ha gjorts. Vi ansåg det dock mer tidskrävande att söka igenom och jämföra matriser än att skicka informationen igen, dessutom vill vi undvika att behöva spara undan onödigt stora datastrukturer för jämförelsen.

Implementationsmässigt kunde vi ha valt att bryta ut fler klasser ur GUI för att minimera storleken på denna klass och göra koden mer överskådlig. Vi har brutit ut och skapat egna klasser för de delar av bladen i användargränssnittet som kräver mycket logik, men i efterhand tycker vi att ett bättre alternativ hade varit att bryta ut varje blad i gränssnittet till separata klasser.

För att förenkla, och prestandamässigt förbättra, implementationen av interpoleringsmetoderna borde koordinaterna för termoelementen ha sorterats efter

position. Om hänsyn tagits till termoelement även i djupled vid beräkningen av temperaturer i kupén kunde valfritt snitt genom bilen ha gjorts tillgängligt för presentation i användargränssnittet.

6.2 Vidareutveckling och framtid

Inför framtiden har vi förslag på flera möjliga vidareutvecklingar och förbättringar. Nedan presenteras dessa uppdelade i områdena Visualisering, interpolering och extrapolering, Användargränssnitt, Funktionalitet, Systemstruktur samt Övrigt.

Visualisering, interpolering och extrapolering

- Interpolera och extrapolera även för den schematiska klimatanläggningen, `Climate Unit`.
- Byta interpoleringsmetod till någon man anser ger ett bättre resultat.
- Prestandamässigt optimera de interpoleringsmetoder vi utvecklat.
- Ändra överföringsfunktion för färgtilldelning samt addera möjligheten att förändra färgtilldelningen från linjär till s-form under exekvering.

Användargränssnitt

- Utöka användargränssnittet med fler blad och ytterligare funktioner.
- Implementera en bättre och mer tillförlitlig grafkomponent, i vilken även fler signalalternativ kan plottas.
- Göra gränssnittet mer flexibelt.
- Förenkla användargränssnittet ytterligare med exempelvis tooltips och kortkommandon.

Funktionalitet

- Bättre hantering vid inläsning av mätdata samt komplettering med data för de signaler som ej finns tillgängliga.
- Utveckla ett separat bildhanteringsprogram för utmarkering av termoelementens placering.

- Filtrera insignalerna till plottarna med till exempel ett första ordningens linjärt filter med valbar tidskonstant. Filterringen kan genomföras under exekvering eller som en del av initieringen innan uppspelning.
- Lägga till en funktion för att kunna markera ut termoelementen och dess namn om så önskas.
- Utveckla ett regressionsfilter för signalerna som plottas för att förbättra AUTO funktionen vid snabba värdeförändringar.

Systemstruktur

- Införa dubbelriktad kommunikation mellan MATLAB och Java applikationerna, och därmed möjliggöra anrop från Java till MATLAB. Implementera dessa delar i C alternativt med hjälp av JNI, Java Native Interface. Kontrollen av kommunikationen förbättras och exakta beräkningstider för interpoleringen etcetera kan erhållas.
- Tillåta fler parameterinställningar för användaren i konfigureringsfilen.

Övrigt

- Utöka systemet för onlinesimuleringar samt bygga nya interface mot andra applikationer såsom HIL och DTECS (Development Tool Embedded Control Systems).
- Läsa mätdata från andra format, exempelvis Excel filer.

7 Referenser

[1] Andersson, Jonas; Modellering av A/C-system i personbil med kompressor av typen externstyrt variabelt deplacement; Control and Automation Laboratory; Department of Signals and Systems Chalmers University of Technology; Göteborg; 2004; nr.: EX013/2004

[2] Fernström, Jesper; Commercial Graphical User Interface for Digital TV; Master's Thesis in Human Computer Interaction, KTH; 2004; http://cid.nada.kth.se/pdf/252.pdf

[3] Gelin, Lisa; Värdering av flerdimensionell visualisering i SMHIs verksamhet; Examensarbete vid SMHI och LiTH; Norrköping; 2002
http://hem.bredband.net/lisgel/exjobb/LisaGelinExjobbFinal.pdf

[4] Hammarlund, Henrik; Nordmark, Olof; Modellering av bilklimat i Volvo S80; Reg.nr: LiTH-ISY-EX-3149; 2001

[5] Hellström, Jessica; Nilsson, Maria D; *Riktlinjer vid design av användargränssnitt samt Internet Exploerer vs Netscape Navigator – en studie i design och användarvänligthet*; Institutionen för Informatik; Handelshögskolan vid Göteborgs Universitet; 1999;
www.handels.gu.se/epc/archive/00002215/01/hellstrom.nilsson.ia5840.pdf

[6] Mårdberg, Björn; Volvo Technology Corporation; Göteborg

[7] Pärt-Enander, Eva; Sjöberg, Anders; Användarhandledning för MATLAB 6.5; Elanders Gotab, Stockholm; 2003; ISBN 91-506-1690-0

[8] Schroeder, Will; Martin, Ken; Lorensen, Bill; The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics; 3rd edition; 2002; Kitware, Inc.; ISBN 1-930934-07-6

[9] Sjödin, Johnny; Riskbedömning av förorenad mark – en studie av statistiska hjälpmedel i analysfasen;
http://epubl.ltu.se/1402-1617/2005/118/LTU-EX-05118-SE.pdf

[10] Strobl, Andreas; Advanced Visual Information Systems with Content Based Image Querying: A User Interface and System Model; Master's Thesis at Victoria Institute of Technology; Australia; 2004;
www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2004/rapporter04/strobl_andreas_04132.pdf

[11] VTK User's Guide, Install, Use and Extend The Visualization Toolkit; authored and published by Kitware, Inc.; version 4.2; 2003; ISBN 1-930934-08-4

[12] http://www.openabacus.org/products.abaguibuilder.html

[13] http://www.jdc-software.com/docs/lbusage.php

8 Källförteckning

[1 Füher, Claus; Schroll, Achim; *Numerical Analysis - An Introduction*; Numerical Analysis, Center for Mathematical Sciences, Lund University; 4th edition; 2002

[2] Magnusson, Andreas; MIEX – *A Tool for Presentation of Radar Data*; Master's Thesis in Computer Science performed at Ericsson Microwave Systems AB; nr.: 9/0363-FCP104716; 2000

[3] Pärt-Enander, Eva; Sjöberg, Anders; *Användarhandledning för MATLAB 6.5*; Elanders Gotab, Stockholm; 2003; ISBN 91-506-1690-0

[4] Årzén, Karl-Erik; *Real-Time Control Systems*; Department of Automatic Control; Lund Institute of Technology; Lund; 2003

[5] http://w3.msi.vxu.se/multimedia/km/bild/farglara/farg1.htm

[6] http://home.student.uu.se/tova5083/Webb_F6.htm

[7] http://www.bigscreen.se/aob-mbf.htm

[8] http://java.sun.com/j2se/1.4.2/docs/api/index.html

[9] http://java.sun.com/docs/books/tutorial/uiswing/components/components.html

[10] www.vtk.org

Appendix A Användarmanual

Innehållsförteckning

1 Tekniska förutsättningar

För att kunna utnyttja programmet krävs att följande förutsättningar uppfylls:

- MATLAB version 7.0 R14
- EasySee.jar
- regler.jar
- visualisering.m, ECCInterpolation.m, EvaporatorInterpolation.m, initEvapInterpolation.m, initEvapCoord.m, completeTemp.m och completeTempNoExtra.m
- Skärmens upplösning bör vara minst 1024 x 768 pixlar

För att kunna komplettera eller förändra programmet krävs även:

- Java version 1.4.2_06

2 Typografi

De funktioner som finns tillgängliga i systemet EasySee beskrivs i användarhandledningen på nedanstående sätt:
Funktionen: Att …

- Beskrivning/kommandot

För alla hänvisningar till klassnamn och variabelnamn från koden används typsnittet `Courier New`, för att tydligt urskilja dessa i rapporten. För hänvisningar till funktioner i användargränssnittet används typsnittet **Microsoft Sans Serif**.

3 Installation

Att installera programmet EasySee

- spara EasySee.jar , regler.jar, samtliga m-filer, katalogen Images, userconfig.txt samt systemconfig.txt i arbetskatalogen
- ange att MATLAB ska startas i arbetskatalogen (under Properties fliken Shortcut anges sökvägen till arbetskatalogen på raden ”Start in”)

- lägg till sökvägen till EasySee.jar och regler.jar i MATLABs classpath.txt (`edit classpath.txt`)

Att starta verktyget EasySee

- Ange kommandot `run EasySee` i MATLABs prompt

Att avsluta verktyget EasySee

- Tryck på krysset i övre högra hörnet av användargränssnittet

Att packa upp programmet EasySee för att kunna göra ändringar i koden

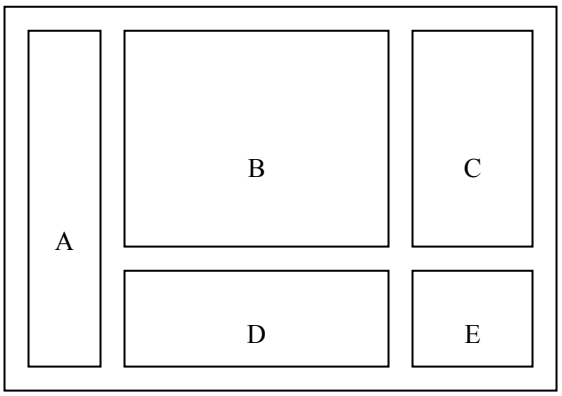
- Skriv kommandot `jar xfv EasySee.jar` i arbetskatalogen
- spara regler.jar i java_root\lib\ och i java_root\jre\lib\ext för att kunna kompilera ändringarna

Att packa programmet EasySee till en exekverbar .jar-fil efter ändringar

- Skriv kommandot `jar cfv EasySee.jar *.class *.png` i katalogen.
- Installera den nya .jar filen enligt ”Att installera programmet EasySee”

4 EasySee

EasySee består av tre blad, `ECC`, `Climate Unit` och `Evaporator`, vilka man växlar mellan genom att välja respektive blads flik i användargränssnittets ovankant. Bladen samverkar så att den uppdateringshastighet som valts på ett blad, gäller även på de båda andra. Uppdateringshastigheten kan ökas och minskas. Följande hastigheter finns tillgängliga för uppspelning av sampel framåt i tiden: x2, x5, x10, x20, x100. Om uppspelning önskas av sampel som ligger bakåt i tiden sett från det aktuella samplet så finns följande hastigheter tillgängliga: x2, x5, x10. Samtliga valbara hastigheter är multiplar av samplingshastigheten Temperaturintervall och övriga inställningar är individuella för respektive blad. Om ett värde utanför en komponents intervall anges kommer det senaste giltiga värdet på komponenten att användas istället. Varje blad är indelad i fem huvudområden enligt följande struktur:



A. Tabell för signalvärden

B. Spelare

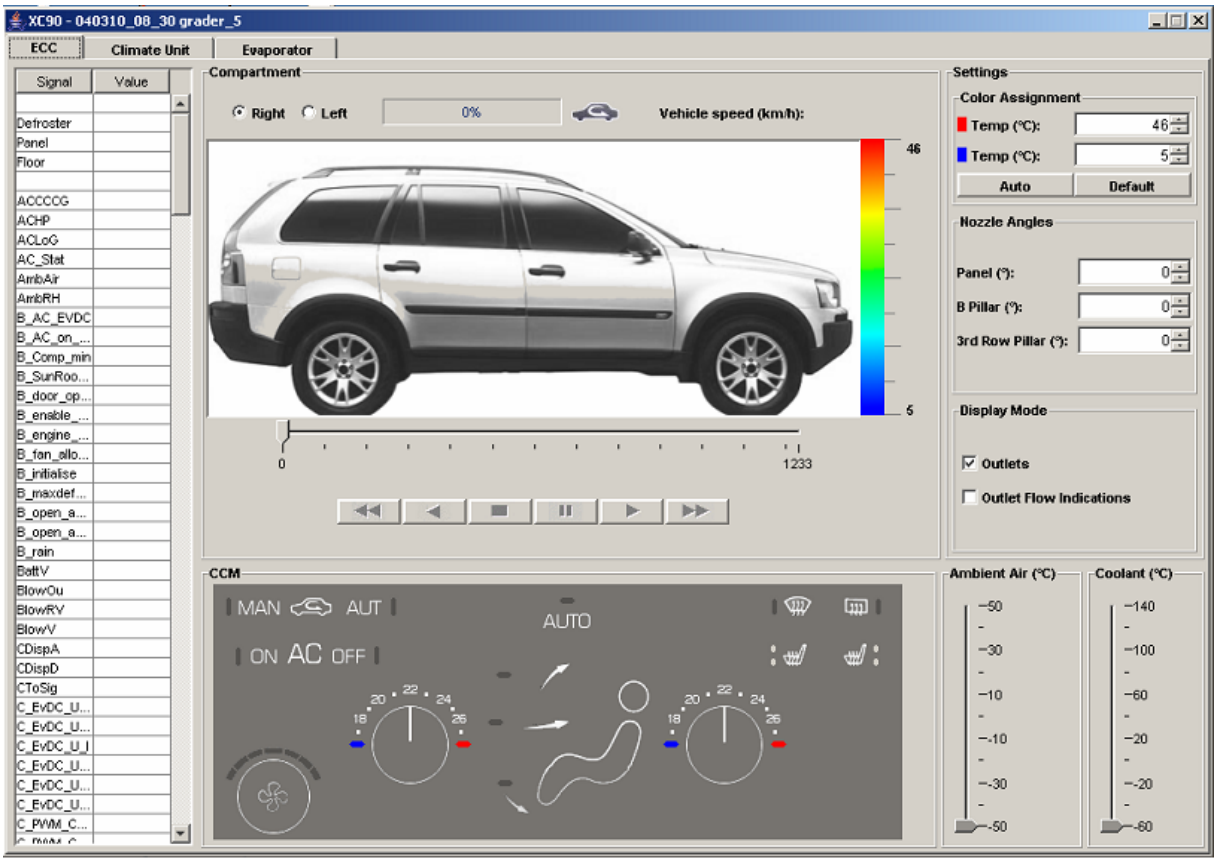
C. Användarinställningar

D. Panelinställningar för klimatenheten (gäller för `ECC` bladet) och graf för utvalda signaler (gäller för `Climate Unit` och `Evaporator` bladen)

E. Omgivningsdata

I detta kapitel återfinns de funktioner som finns på respektive blad, sorterat efter placering i gränssnittet**.**

4.1 ECC



4.1.1 Tabell

Att studera all data från mätfilen samt beräknade värden för andelen luft genom Defroster, Panel och Floor.

- Läs av värden i tabellen under **Value**

4.1.2 Compartment

Att välja uppritning av vänster (förar) sida

- Markera **Left**

Att välja uppritning av höger (passagerar) sida

- Markera **Right**

Att uppdatera temperaturerna med samplingshastigheten

- Tryck på play-knappen

Att öka uppdateringshastigheten ett steg

- Tryck på fastforward-knappen en gång

Att minska uppdateringshastigheten ett steg

- Tryck på rewind-knappen en gång

Att uppdatera temperaturerna med samplingshastigheten *men* för tidigare mätdata

- Tryck på reverse-knappen

Att stoppa uppritningen

- Tryck på stopp-knappen

Att pausa uppritningen

- Tryck på pause-knappen

Att ändra aktuellt uppritningssampel

- Dra listknoppen ovanför knappsatsen till önskat sampel

Att kontrollera aktuell recirkulationsgrad (%)

- Läs av värdet på komponenten framför symbolen för recirkulation

Att kontrollera fordonets aktuella hastighet (km/h)

- Läs av värdet vid **Vehicle Speed:**

4.1.3 Settings

4.1.3.1 Color Assignment

Att ställa in vilken temperatur som ska representeras med färgen röd

- Tryck på pilen uppåt respektive pilen nedåt vid den temperaturinställning **Temp (°C)** som föregås av en röd ikon för att öka respektive minska värdet. Alternativt skriv in det önskade värdet manuellt och tryck Enter.

Att ställa in vilken temperatur som ska representeras med färgen blå

- Tryck på pilen uppåt respektive pilen nedåt vid den temperaturinställning **Temp (°C)** som föregås av en blå ikon för att öka respektive minska värdet. Alternativt skriv in det önskade värdet manuellt och tryck Enter.

Att ställa in temperaturintervall för optimerat färgutnyttjande, utgående från termoelementens momentana max och min temperatur.

- Tryck på **Auto** för momentant optimerat temperaturintervall

Att ställa in temperaturintervall för optimerat färgutnyttjande, utgående från termoelementens max och min temperatur för alla sampel.

- Tryck på **Default** för optimerat temperaturintervall

4.1.3.2 Nozzle Angles

Att ställa in ventilationsmunstyckena på panelen (0° = rakt bakåt, 90° = mot taket, -90° = mot golvet)

- Tryck på pilen uppåt respektive pilen nedåt vid **Panel (°)** för att öka respektive minska vinkeln. Alternativt skriv in det önskade värdet manuellt och tryck Enter.

Att ställa in ventilationsmunstyckena på B-stolpen (0° = rakt bakåt, 90° = mot taket, -90° = mot golvet)

- Tryck på pilen uppåt respektive pilen nedåt vid **B Pillar (°)** för att öka respektive minska vinkeln. Alternativt skriv in det önskade värdet manuellt och tryck Enter.

Att ställa in ventilationsmunstyckena på tredje raden (0° = rakt bakåt, 90° = mot taket, -90° = mot golvet)

- Tryck på pilen uppåt respektive pilen nedåt vid **3rd Row Pillar (°)** för att öka respektive minska vinkeln. Alternativt skriv in det önskade värdet manuellt och tryck Enter.

4.1.3.3 Display Mode

Att visa temperaturen i kupén med hänsyn tagen till ventilationer

- Markera **Outlets**

Att visa temperaturen i kupén utan hänsyn tagen till ventilationer

- Avmarkera **Outlets**

Att visa markering för ventilationernas effekt

- Markera **Outlet Flow Indications**

4.1.4 CCM

Att kontrollera önskad temperatur på högersidan

- Läs av värdet på högra temperaturvredet

Att kontrollera önskad temperatur på vänstersidan

- Läs av värdet på vänstra temperaturvredet

Att kontrollera önskad fläktstyrka

- Läs av antalet tända dioder vid fläkt-knappen

Att kontrollera önskat tillstånd för AC

- Läs av tänd diod vid AC-knappen

Att kontrollera önskat tillstånd för recirkulation

- Läs av tänd diod vid recirkulations-knappen

Att kontrollera önskad luftdistribution

- Läs av tänd diod vid luftdistributionsknapparna

Att kontrollera önskat tillstånd för defroster

- Kontrollera om tänd diod vid defroster-knappen

Att kontrollera önskat tillstånd för bakre defroster

- Kontrollera om tänd diod vid bakre defroster-knappen

4.1.5 Omgivningsdata

Att kontrollera aktuell utomhustemperatur (°C)

- Läs av värdet på termometern för **Ambient Air**

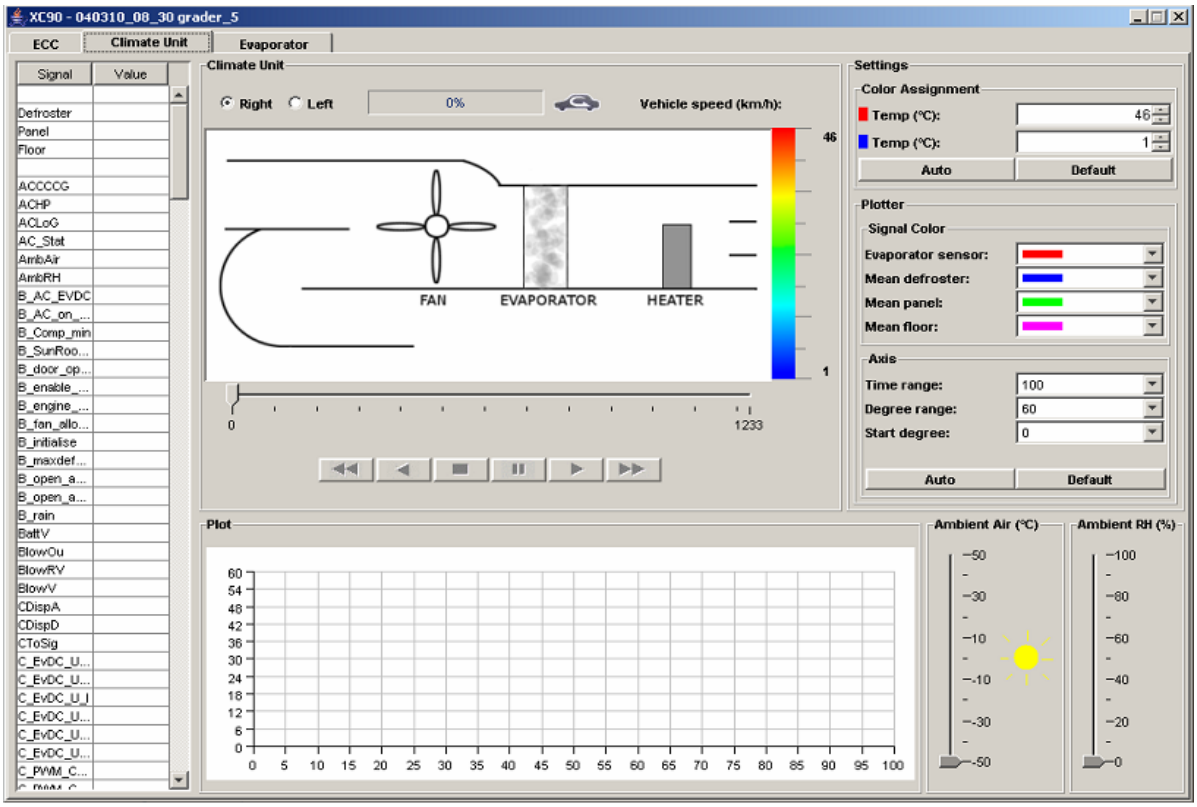
Att kontrollera aktuell temperatur för kylarvattnet (°C)

- Läs av värdet på termometern för **Coolant**

Att kontrollera aktuellt väder

- Kontrollera ikonen bredvid termometern för utomhustemperaturen

4.2 Climate Unit



4.2.1 Tabell

För beskrivning av tillgängliga kommandon se avsnitt *4.1.1 Tabell*

4.2.2 Climate Unit

Att kontrollera spjällens position

- Kontrollera positionen för önskat spjäll (lodrät position = luftvägen är helt stängd, vågrät position = luftvägen är helt öppen).

För beskrivning av övriga tillgängliga kommandon se avsnitt *4.1.2 Compartment*

4.2.3 Settings

4.2.3.1 Color Assignment

För beskrivning av tillgängliga kommandon se avsnitt *4.1.3.1 Color Assignment*

4.2.3.2 Plotter

4.2.3.2.1 Signal Color

Att välja vilken färg signalen för evaporator sensor temperaturen, `TC_evap`, ska representeras med i grafen

- Välj färg i listan för **Evaporator Sensor**

Att välja vilken färg signalen för medelvärdet av defrosterutblåsen, `mean(DefrX)`, ska representeras med i grafen

- Välj färg i listan för **Mean Defroster**

Att välja vilken färg signalen för medelvärdet av panelutblåsen, `mean(VenFX)`, ska representeras med i grafen

- Välj färg i listan för **Mean Panel**

Att välja vilken färg signalen för medelvärdet av golvutblåsen, `mean(FoutFX)`, ska representeras med i grafen

- Välj färg i listan för **Mean Floor**

4.2.3.2.2 Axis

Att välja hur många sampel som ska visas på tidsaxeln

- Välj intervall i listan för **Time Range**

Att välja starttemperatur på temperaturaxeln

- Välj temperatur i listan för **Start Degree**

Att välja hur stort temperaturintervall grafen ska visa med startpunkt i **Start Degree**

- Välj intervall i listan för **Degree Range**

Att ställa in temperaturintervall för optimerat utnyttjande av temperaturaxeln, utgående från termoelementens momentana max och min temperatur. Tidsaxeln visar 100 sampel.

- Tryck på **Auto** för momentant optimerad skalning av temperaturaxeln

Att ställa in temperaturintervall för optimerat utnyttjande av temperaturaxeln, utgående från termoelementens max och min temperatur för alla sampel. Tidsaxeln visar 100 sampel.

- Tryck på **Default** för optimerad skalning av temperaturaxeln

4.2.4 Plot

Att detaljstudera temperaturerna för evaporator sensorn, medelvärdet av defrosterutblåsen, medelvärdet av panelutblåsen samt medelvärdet av golvutblåsen

- Läs av signalernas värde i grafen. Representationsfärgen är angiven i **Signal Color**

4.2.5 Omgivningsdata

Att kontrollera aktuell utomhustemperatur (°C)

- Läs av värdet på termometern för **Ambient Air**

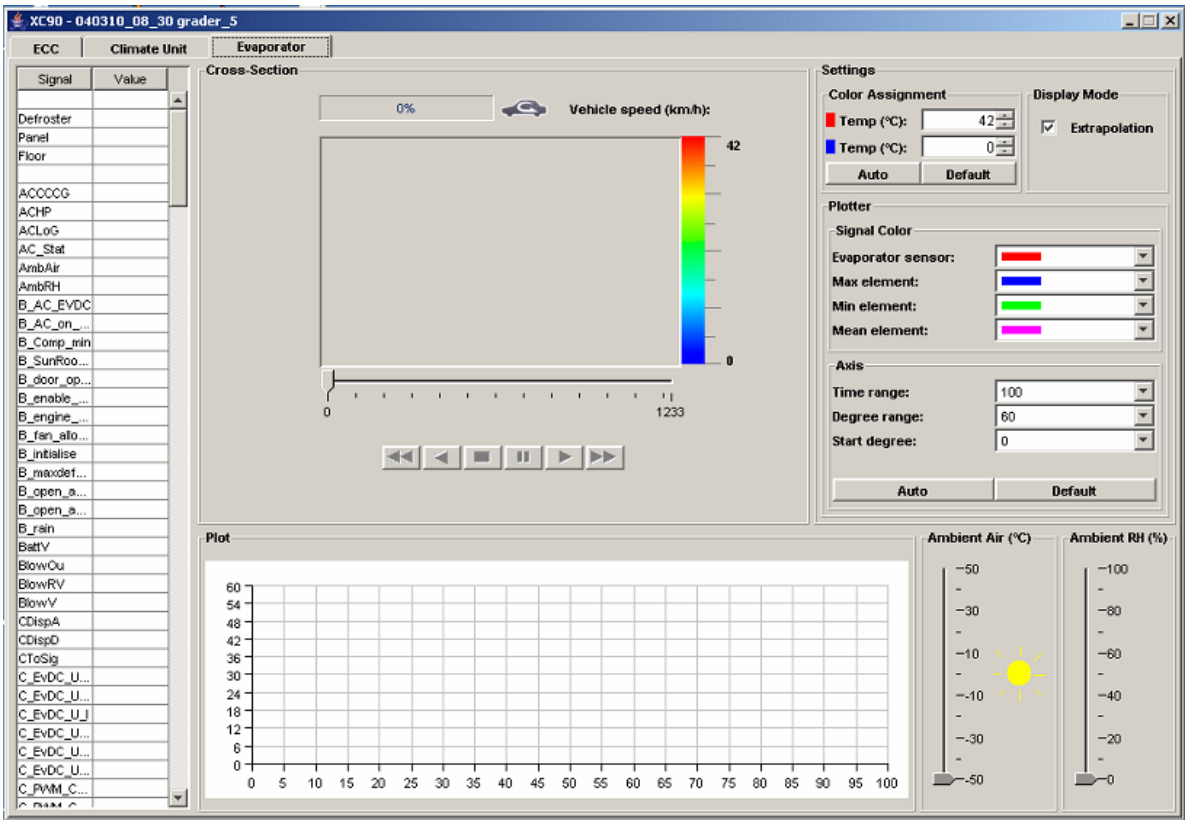
Att kontrollera aktuell relativ luftfuktighet (°C)

- Läs av värdet på hygrometern för **Ambient RH**

Att kontrollera aktuellt väder

- Kontrollera ikonen bredvid termometern för utomhustemperaturen

4.3 Evaporator



4.3.1 Tabell

För beskrivning av tillgängliga kommandon se avsnitt *4.1.2 Tabell*

4.3.2 Evaporator Crossection

För beskrivning av tillgängliga kommandon se under avsnitt *4.1.2 Compartment*. Dock finns ej i denna del kommandona **Left** och **Right** tillgängliga då luften delas först efter evaporatorn.

4.3.3 Settings

4.3.3.1 Color Assignment

För beskrivning av tillgängliga kommandon se avsnitt *4.1.3.1 Color Assignment*

4.3.3.2 Display Mode

Att välja att visa temperaturer beräknade med extrapolering

- Markera **Extrapolation**

Att välja att visa temperaturer beräknade utan extrapolering

- Avmarkera **Extrapolation**

4.3.3.3 Plotter

4.2.3.3.1 Signal Color

Att välja vilken färg signalen för evaporator sensor temperaturen, `TC_evap`, ska representeras med i grafen

- Välj färg i listan för **Evaporator Sensor**

Att välja vilken färg signalen för den högsta temperaturen av termoelementen i evaporatorn, `max(EvapXX)`, ska representeras med i grafen

- Välj färg i listan för **Max element**

Att välja vilken färg signalen för den lägsta temperaturen av termoelementen i evaporatorn, `min(EvapXX)`, ska representeras med i grafen

- Välj färg i listan för **Min element**

Att välja vilken färg signalen för medeltemperaturen av termoelementen i evaporatorn, `mean(EvapXX)`, ska representeras med i grafen

- Välj färg i listan för **Mean element**

4.2.3.3.2 Axis

För beskrivning av tillgängliga kommandon se avsnitt *4.2.3.2.2 Axis*

4.3.4 Plot

Att detaljstudera temperaturerna för evaporator sensorn, högsta temperaturen av termoelementen i evaporatorn, lägsta temperaturen av termoelementen i evaporatorn samt medeltemperaturen av termoelementen i evaporatorn

- Läs av signalernas värde i grafen. Representationsfärgen är angiven i **Signal Color**

4.3.5 Omgivningsdata

För beskrivning av tillgängliga kommandon se under avsnitt *4.2.5 Omgivningsdata*

5 Bakgrundsbilder

Uppritningen av temperaturfälten sker i bakgrundsbilder som läses in av programmet och sedan behandlas för att kunna visa upp de beräknade temperaturerna. Användargränssnittet är dynamiskt i avseendet att bakgrundsbilderna är utbytbara, de nya bilderna måste dock uppfylla vissa krav vilka specificeras nedan för respektive vy. Alla färger nedan anges med sin RGB-kod; alphakanalen sätts alltid till 0xff (opaque)

5.1 ECC

I `ECC` vyn finns två bilder som kan bytas ut; kupén som visas i `Compartment` samt panelen som visas i `CCM`. Bilden på kupén bör vara i storleksordningen 548 x 229 pixlar medan bilden på panelen bör vara i storleksordningen 577 x 210 pixlar. Programmet förutsätter att bilden inte innehåller några pixlar med den RGB-kod som används för förbehandlingen enligt nedan.

5.1.1 Compartment

Att markera ut termoelementens position i kupén

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan markeras med färg
- Markera ut kupén (det område för vilket temperaturer ska beräknas) i bilden genom att fylla denna med färgen 0xff00ff00
- Markera den pixel som motsvarar sensorns position i kupén med nedan angivna färg:

  HEAD3X: 0xffbb00ff
  HEADRX: 0xffcc00ff
  HEADFX: 0xffee00ff
  FLO3X: 0xffff00ff
  FLORX: 0xffff00ee
  FLOFX: 0xffff00dd
  VEN3X: 0xffff7700
  VENRX: 0xffff8800
  VENFX: 0xffff9900

FOUT3X: 0xffffbb00

FOUTRX: 0xffffcc00

FOUTFX: 0xffffdd00

DEFRCX: 0xffffff00

X står för höger respektive vänster sidas termoelement men då vi antagit att dessa sensorer har samma position i x,y-planet behöver endast positionen för en av dessa sensorer anges.

Att ersätta bakgrundsbilden på kupén

- Spara bilden med de nya ändringarna som ecc.png i arbetskatalogen
- Kompilera om med kommandot javac *.java (kräver att man har installerat jdk)
- Packa programmet enligt ”Att packa programmet EasySee till exekverbar .jar-fil efter ändringar”, se avsnitt *3 Installation*

5.1.2 CCM

Att markera ut diodernas position på panelen

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan färgläggas
- Markera de pixlar som motsvarar diodernas position i bilden med nedan angivna färg:

  Manuell inställning av recirkulationen: 0xff5500ff

  Automatisk inställning av recirkulationen : 0xff00ff00

  AC i ON-läge: 0xff00ffff

  AC i OFF-läge: 0xffff0055

  Automatisk inställning av ventilationerna: 0xffff00ff

  Defrosterventilationerna: 0xffffff00

  Panelventilationerna: 0xffff00aa

  Golvventilationerna: 0xffff00ee

  Defroster, fram: 0xffff00dd

  Defroster, bak: 0xffff00bb

  Första fläktdioden: 0xffff6600

  Andra fläktdioden: 0xffff7700

  Tredje fläktdioden: 0xffff8800

  Fjärde fläktdioden: 0xffff9900

  Femte fläktdioden: 0xffffbb00

Sjätte fläktdioden: 0xffffcc00

Sjunde fläktdioden: 0xffffdd00

Att ersätta bakgrundsbilden på panelen

- Spara bilden med de nya ändringarna som ecc.png i arbetskatalogen
- Kompilera om med kommandot javac *.java (kräver att man har installerat jdk)
- Packa programmet enligt ”Att packa programmet EasySee till exekverbar .jar-fil efter ändringar”, se avsnitt *3 Installation*

5.2 Climate Unit

I `Climate Unit` vyn kan bilden över den schematiska klimatanläggningen ersättas. Bilden på klimatanläggningen bör vara i storleksordningen 500 x 220. Programmet förutsätter att bilden inte innehåller några pixlar med den RGB-kod som används för förbehandlingen enligt nedan.

Att markera ut termoelementens position i klimatanläggningen

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan färgläggas
- Markera den pixel som motsvarar termoelementets position i bilden med nedan angivna färg:

  AMBAIR: 0xffff0055

  RECAIR: 0xffff0066

  EVAPSE: 0xffff0077

  DEF: 0xffff0088

  VENT: 0xffff0099

  FLOOR: 0xffff1100
- Spara bilden

Att markera ut de pixlar för vilka termoelementens temperatur ska visas

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan färgläggas
- Markera de pixel som motsvarar termoelementets område i bilden med nedan angivna färg:

  AMBAIR_AREA: 0xffff0000

RECAIR_AREA: 0xffffaa00

EVAPSE_AREA: 0xffffff00

DEF_AREA: 0xff00ff00

VENT_AREA: 0xff00ffff

FLOOR_AREA: 0xff0000ff

- Spara bilden

Att markera ut spjällens rotationsaxel i klimatanläggningen

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan markeras/förändras
- Markera den pixel som motsvarar rotationsaxelns position i bilden med färger enligt nedan:

  OSA: 0xffcc00ff

  REC: 0xff9900ff

  HEATER: 0xffee00ff

  DEF: 0xffff00ff

  VENT: 0xffff00dd

  FLOOR: 0xffff00bb

- Spara bilden

Att ersätta bakgrundsbilden i klimatanläggningen

- Spara bilden med de nya ändringarna som ecc.png i arbetskatalogen
- Kompilera om med kommandot javac *.java (kräver att man har installerat jdk)
- Packa programmet enligt ”Att packa programmet EasySee till exekverbar .jar-fil efter ändringar”, se avsnitt *3 Installation*

5.3 Evaporator

I `Evaporator` vyn kan bilden över evaporatortvärsnittet i `Evaporator Crossection` ersättas. Detta förutsätter dock att termoelementen är nio till antalet samt att de är symmetriskt utplacerade. Bilden över evaporatortvärsnittet bör vara i storleksordningen 308*196 pixlar. Programmet förutsätter att bilden inte innehåller några pixlar med den RGB-kod som används för förbehandlingen enligt nedan.

Att markera ut termoelementens position i evaporatorn

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan markeras/förändras
- Markera ut kupén (det område för vilket temperaturer ska beräknas) i bilden genom att fylla denna med färgen 0xff00ff00
- Markera den pixel som motsvarar termoelementets position i bilden med nedan angivna färg:

  EVAP11: 0xffff0000

  EVAP12: 0xffff7700

  EVAP13: 0xffffbb00

  EVAP21: 0xffffff00

  EVAP22: 0xff00ffee

  EVAP23: 0xff00ccff

  EVAP31: 0xff0000ff

  EVAP32: 0xffaa00ff

  EVAP33: 0xffee00ff
- Spara bilden

Att markera ut evaporator sensorns position

- Öppna bilden i ett bildhanteringsprogram där enskilda pixlar kan markeras/förändras
- Markera den pixel som motsvarar sensorns position i bilden med nedan angivna färg:

  SENSOR: 0xffff00aa
- Spara bilden

Att ersätta bakgrundsbilden på evaporatorn

- Spara bilden med de nya ändringarna som ecc.png i arbetskatalogen
- Kompilera om med kommandot javac *.java (kräver att man har installerat jdk)
- Packa programmet enligt ”Att packa programmet EasySee till exekverbar .jar-fil efter ändringar”, se avsnitt *3 Installation*

6 Krav på mätfiler

De mätfiler programmet ska hantera måste vara i formatet .MAT . Filerna ska innehålla signaler enligt *6.1 System konfiguration*.

6.1 Systemkonfiguration

För att programmet ska vara exekverbart så ställs krav på vilka signaler som måste finnas i mätfilen. Dessa redovisas i systemconfig.txt och är följande (måste vara namngivna i mätfilen enligt nedanstående):

- Head3R, Head3L, HeadRR, HeadRL, HeadFR, HeadFL
- Flo3R, Flo3L, FloRR, FloRL, FloFR, FloFL
- Ven3R, Ven3L, VenRR, VenRL, VenFR, VenFL
- Fout3R, Fout3L, FoutRR, FoutRL, FoutFR, FoutFL
- DefrCR, DefrCL
- EvaR11, EvaR12, EvaR13, EvaR21, EvaR22, EvaR23, EvaR31, EvaR32, EvaR33
- TC_evap, EvapSe
- AmbAir
- RecAir
- UN_osa_rec, UN_heat_right
- M_fv_def, M_floor_vent, VV_fan0
- N_rec_set
- B_enable_AC
- N_mode_set
- B_maxdef_set
- UB_rear_def
- N_fan_set
- TC_set_left, TC_set_right

Utöver dessa signaler så krävs ytterligare ett antal signaler för att alla funktioner i verktyget ska vara tillgängliga:

- Iw_Sun_filt

- B_rain
- AmbRH
- TC_coolant
- M_rec
- vk_vehicle

De signaler som finns i systemconfig.txt ska ej ändras av användaren utan endast vid eventuell vidareutveckling av programmet.

6.2 Användarkonfiguration

I filen userconfig.txt finns inställningar som kan ändras av användaren efter behov. Inställningarna som finns tillgängliga är följande:

- maximal påverkan av utblåsen
- fördelning mellan VEN utblåsen
- fördelning mellan FOUT utblåsen
- trimfaktor a, förklaring typ cosh a , för utblåsen
- trimfaktor theta, viktningsfaktor av utblås jämfört med HEAD- och FLO-termoelement
- vinklar för FOUT-utblåsens munstycken

7 Funktionsindex

Lista över alla tillgängliga kommandon utifrån sökord i alfabetisk ordning, de inledande siffrorna anger under vilket avsnitt funktionen återfinns.

AC:

- **4. 1.4** Att kontrollera önskat tillstånd för AC

Avsluta:

- **3** Att avsluta verktyget EasySee

Bakgrundsbilder:

- **5.1.1** Att markera ut termoelementens position i kupén
- **5.1.1** Att ersätta bakgrundsbilden på kupén
- **5.1.2** Att markera ut diodernas position på panelen
- **5.1.2** Att ersätta bakgrundsbilden på panelen
- **5.2** Att ersätta bakgrundsbilden i klimatanläggningen
- **5.2** Att markera ut termoelementens position i klimatanläggningen
- **5.2** Att markera ut spjällens rotationsaxel i klimatanläggningen
- **5.3** Att ersätta bakgrundsbilden på evaporatorn
- **5.3** Att markera ut termoelementens position i evaporatorn
- **5.3** Att markera ut evaporator sensorns position

Detaljstudie av data:

- **4.2.4** Att detaljstudera temperaturerna för evaporator sensorn, medelvärdet av defrosterutblåsen, medelvärdet av panelutblåsen samt medelvärdet av golvutblåsen
- **4.3.4** Att detaljstudera temperaturerna för evaporator sensorn, högsta temperaturen av termoelementen i evaporatorn, lägsta temperaturen av termoelementen i evaporatorn samt medeltemperaturen av termoelementen i evaporatorn

Färgmappning:

- **4. 1.3.1** Att ställa in vilken temperatur som ska representeras med färgen röd
- **4. 1.3.1** Att ställa in vilken temperatur som ska representeras med färgen blå

- **4. 1.3.1** Att ställa in temperaturintervall för optimerat färgutnyttjande, utgående från termoelementens momentana max och min temperatur
- **4. 1.3.1** Att ställa in temperaturintervall för optimerat färgutnyttjande, utgående från termoelementens max och min temperatur för alla sampel

Grafalternativ:

- **4.2.3.2.1** Att välja vilken färg signalen för evaporator sensor temperaturen, `TC_evap`, ska representeras med i grafen
- **4.2.3.2.1** Att välja vilken färg signalen för medelvärdet av defrosterutblåsen, `mean(DefrX`), ska representeras med i grafen
- **4.2.3.2.1** Att välja vilken färg signalen för medelvärdet av panelutblåsen, `mean(VenFX)`, ska representeras med i grafen
- **4.2.3.2.1** Att välja vilken färg signalen för medelvärdet av golvutblåsen, `mean(FoutFX)`, ska representeras med i grafen
- **4.2.3.2.2** Att välja hur många sampel som ska visas på tidsaxeln
- **4.2.3.2.2** Att välja starttemperatur på temperaturaxeln
- **4.2.3.2.2** Att välja hur stort temperaturintervall grafen ska visa med startpunkt i **Start Degree**
- **4.2.3.2.2** Att ställa in temperaturintervall för optimerat utnyttjande av temperaturaxeln, utgående från termoelementens momentana max och min temperatur
- **4.2.3.2.2** Att ställa in temperaturintervall för optimerat utnyttjande av temperaturaxeln, utgående från termoelementens max och min temperatur för alla sampel
- **4.2.3.3.1** Att välja vilken färg signalen för evaporator sensor temperaturen, T`C_evap`, ska representeras med i grafen
- **4.2.3.3.1** Att välja vilken färg signalen för den högsta temperaturen av termoelementen i evaporatorn, `max(EvapXX`), ska representeras med i grafen
- **4.2.3.3.1** Att välja vilken färg signalen för den lägsta temperaturen av termoelementen i evaporatorn, `min(EvapXX`), ska representeras med i grafen
- **4.2.3.3.1** Att välja vilken färg signalen för medeltemperaturen av termoelementen i evaporatorn, `mean(EvapXX)`, ska representeras med i grafen

Installation:

- **3** Att installera programmet EasySee

Kylarvatten:

- **4. 1.5** Att kontrollera aktuell temperatur för kylarvattnet (°C)

Mätfilsöversikt:

- **4.1.1** Att studera all data från mätfilen samt beräknade värden för andelen luft genom Defroster, Panel och Golv

Omgivningsklimat:

- **4. 1.5** Att kontrollera aktuell utomhustemperatur (°C)
- **4.2.5** Att kontrollera aktuell relativ luftfuktighet (°C)
- **4. 1.5** Att kontrollera aktuellt väder

Recirkulation:

- **4. 1.2** Att kontrollera aktuell recirkulationsgrad (%)
- **4. 1.4** Att kontrollera önskat tillstånd för recirkulation

Temperaturvred:

- **4. 1.4** Att kontrollera önskad temperatur på högersidan
- **4. 1.4** Att kontrollera önskad temperatur på vänstersidan

Uppdateringsalternativ:

- **4. 1.2** Att uppdatera temperaturerna med samplingshastigheten
- **4. 1.2** Att öka uppdateringshastigheten ett steg
- **4. 1.2** Att minska uppdateringshastigheten ett steg
- **4. 1.2** Att uppdatera temperaturerna med samplingshastigheten men för tidigare mätdata
- **4. 1.2** Att stoppa uppritningen
- **4. 1.2** Att pausa uppritningen
- **4. 1.2** Att ändra aktuellt uppritningssampel

Uppritningsalternativ:

- **4. 1.2** Att välja uppritning av vänster (förar) sida

- **4. 1.2** Att välja uppritning av höger (passagerar) sida
- **4. 1.3.3** Att visa temperaturen i kupén med hänsyn tagen till ventilationer
- **4. 1.3.3** Att visa temperaturen i kupén utan hänsyn tagen till ventilationer
- **4. 1.3.3** Att visa markering för ventilationernas effekt
- **4. 3.3.2** Att välja att visa temperaturer beräknade med extrapolering
- **4. 3.3.2** Att välja att visa temperaturer beräknade utan extrapolering

Uppstart:

- **3** Att starta verktyget EasySee

Vagnhastighet:

- **4. 1.2** Att kontrollera fordonets aktuella hastighet (km/h)

Ventilationer:

- **4. 1.3.2** Att ställa in ventilationsmunstyckena på panelen
- **4. 1.3.2** Att ställa in ventilationsmunstyckena på B-stolpen
- **4. 1.3.2** Att ställa in ventilationsmunstyckena på tredje raden
- **4. 1.4** Att kontrollera önskad fläktstyrka
- **4. 1.4** Att kontrollera önskad luftdistribution
- **4. 1.4** Att kontrollera önskat tillstånd för defroster
- **4. 1.4** Att kontrollera önskat tillstånd för bakre defroster
- **4. 2.2** Att kontrollera spjällens position

Vidareutveckling:

- **3** Att packa programmet EasySee till en exekverbar .jar-fil efter ändringar
- **3** Att packa upp programmet EasySee för att kunna göra ändringar i koden

Appendix B Verktygsutvärdering

Enligt kapitel 3 Verktygsutvärdering har vi identifierat följande huvuduppgifter i applikationen:

- beräkningar och interpolering/extrapolering
- presentation/layoutbyggnad
- grafik/prestanda
- filhantering/datadelning

För dessa punkter har för- och nackdelar, utifrån de förutsättningar som beskrivs i första avsnittet, för Java respektive MATLAB specificerats och presenteras i efterföljande avsnitt.

3.1 Förutsättningar

De förutsättningar som gäller för applikationen är:

- Vi har ej funnit några ”färdiga” verktyg som passade uppgiftens karaktär, utan dessa var alltför specialiserade och med alltför avancerade grafikmöjligheter för vår tillämpning. Dessutom vill vi ej vara beroende av någon utomstående programvara då detta kan innebära problem vad gäller möjligheter för till exempel underhåll och vidareutveckling.

- All mätdata finns i dagsläget i MAT-filer och avdelningen ser helst att denna hantering kan fortlöpa.

- Avdelningen jobbar dagligen med MATLAB/SimuLink. De ser gärna att de i framtiden kan förbättra interpoleringsmetoder samt underhålla programmet med en rimlig arbetsinsats.

- Den erfarenhet vi har av programmering av system och användargränssnitt är i Java.

- Vi är bekanta med MATLAB, vi har dock aldrig byggt några användargränssnitt med MATLAB och är ej heller vana vid bildhantering i denna miljö.

- Vi har inga förkunskaper i datorgrafik och bildhantering.

- Från och med version 6 av MATLAB kan Java-klasser användas i MATLAB. Det ger tillgång till en stor mängd färdigutvecklade verktyg, dels i Javas standardklasser, men även i alla andra klasser som skrivits. Java är integrerat i MATLAB och i varje installation finns en Java Virtual Machine (JVM) som kan användas direkt från kommandoprompten. Detta medför att objekt kan skapas och användas interaktivt i MATLAB, tillsammans med vanliga MATLAB variabler och funktioner. Till exempel kan man skapa en instans av en klass och skicka en referens till objektet till en MATLAB funktion som anropar objektets metoder

3.2 Beräkningar och Interpolering/Extrapolering

3.2.1 MATLAB

+ MATLAB är en kraftfull beräkningsmotor och har flera inbyggda bibliotek, vilka möjliggör komplicerade beräkningar genom enkla kommandon. Avancerade plottar och möjlighet att jämföra dessa finns tillgängligt, vilket skulle vara användbart vid jämförelse av flöden och temperaturer.

3.2.2 JAVA

+ Finns en del beräkningshjälpmedel att tillgå i JAI, Java Advanced Imaging, samt i standardbiblioteken

- Implementeringen kräver antagligen mer tid och arbete än i MATLAB.

3.3 Presentation/Layoutbyggnad

3.3.1 MATLAB

+ Finns färdigt layoutverktyg, GUIDE, som innebär snabb uppbyggnad av användargränssnitt

- Endast enkla grafiska komponenter finns tillgängliga, begränsade möjligheter att göra ett snyggt och användarvänligt användargränssnitt

3.3.2 JAVA

+ I princip obegränsade layoutmöjligheter.

- Byggandet av användargränssnittet i Java kräver en hel del programmering. För att reducera implementationsarbetet kan diverse byggverktyg för layouten användas. De vi hittat som är open-source innehåller tyvärr begränsat med grafiska komponenter, vilket vid utnyttjande skulle innebära att fördelen med Java elimineras.

3.4 Grafik/Prestanda

3.4.1 MATLAB

+ Speciella toolboxar, till exempel Image Processing Toolbox, finns att tillgå.

- I MATLAB finns prestandabegränsningar vid uppritning av samtliga pixlar i en bild, vilket kommer att innebära att vi måste flytta runt tidigare uppritade pixlar istället för att rita om dessa vid varje uppdatering (XOR-hantering). De toolboxar som är relevanta för vår uppgift har alltför dyra licenser.

3.4.2 JAVA

+ Java är ett bra presentationsverktyg och exempelvis har alla Swing komponenter automatisk dubbelbuffring. Standardklassen `BufferedImage` som används vid bildhantering utnyttjar också dubbelbuffring.

+ Java är plattformsoberoende.

- Finns risk för svårkontrollerade uppdateringsloopar av de grafiska komponenterna. Man måste vara väl medveten om vilka metoder i standardklasserna som är lämpliga att ersätta då man vill infoga egen funktionalitet vid uppritning.

3.5 Filhantering/Datadelning

Samarbetet mellan Java och MATLAB underlättas av att MATLAB har en inbyggd JVM, Java Virtual Machine, vilket innebär att MATLAB kan hantera .class filer. För att få en senare version av den inbyggda JVM:en (version 1.4.2) har vi installerat MATLAB 7.0 R14. Utöver detta är datadelning förberett så att man enkelt kan skicka variabler som finns i MATLABs workspace till Java metoder.

3.6 Slutsats

Var gränsen för samarbetet mellan MATLAB och Java dras är väsentligt för utvärderingen.
Vi anser att de olika utvecklingsmiljöerna utnyttjas maximalt då, i stora drag, MATLAB utför beräkningen av temperaturerna och sedan skickar resultatet till Java applikationen för presentation och användarinteraktion.

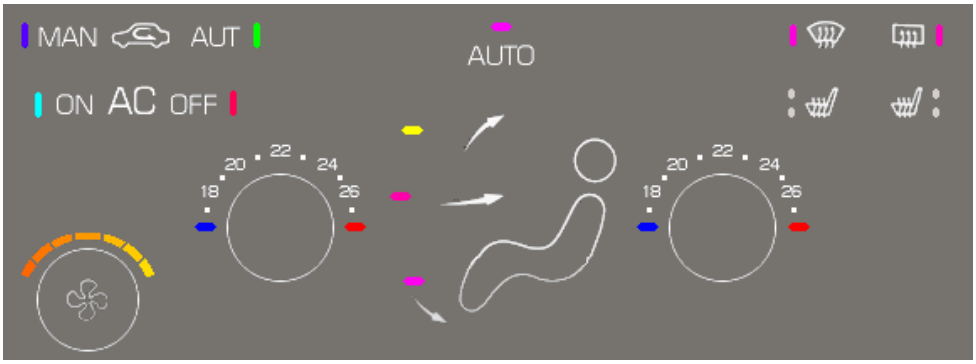
Ovanstående har framkommit genom testning samt genom deltagande av andra utvecklares erfarenheter.

Appendix C Bildbehandling i EasySee

Bilder i GIF och JPEG format är bitmapbilder. Den stora nackdelen med bitmap bilder är att vid bättre kvalitet krävs ett större antal pixlar vilket kräver större minneskapacitet. Alternativet till bitmapbilder är vektorbilder vilka inte begränsas av upplösningen. Vector bilder, eller med annat namn object graphics, är svårare att hantera och används mer sällan. Bilder bryts ner till matematiska objekt, punkter, linjer och kurvor som representerar informationen. Bilder i PNG (Portable Network Graphics) format är vektorbilder. Med PNG ges fördelen att det inte blir någon kvalitetsförsämring vid komprimering. Det är även möjligt att ange genomskinlighet utan att antalet färger behöver begränsas. För att beskriva genomskinligheten utnyttjas en parameter som kallas alfa. En pixels färg beskrivs då med färgformen ARGB (Alpha Red Green Blue) där genomskinligheten kan anta värden från 0 till 255 [10].

I applikationen används ett flertal bakgrundsbilder för att illustrera olika delar i bilen. Formatet på bilderna är .png för att erhålla en bra kvalitet utan bildförsämring vid komprimering. I bakgrundsbilderna finns ett eller flera områden som programmet färglägger under exekvering.

På `ECC` bladet är det dels området i kupén som färgläggs med färger som motsvarar temperaturer och dels dioderna på CCM-panelen vars färg ska ange om dioden lyser eller ej. För att bakgrundsbilderna ska kunna förändras på specifika positioner i bilden, där kupén är respektive där dioderna sitter, så krävs det att programmet kan särskilja de pixlar i bakgrundsbilden som är av intresse. Detta görs genom att bilderna är förbehandlade med unika specificerade färger för de olika intressanta områdena, se figur 1.

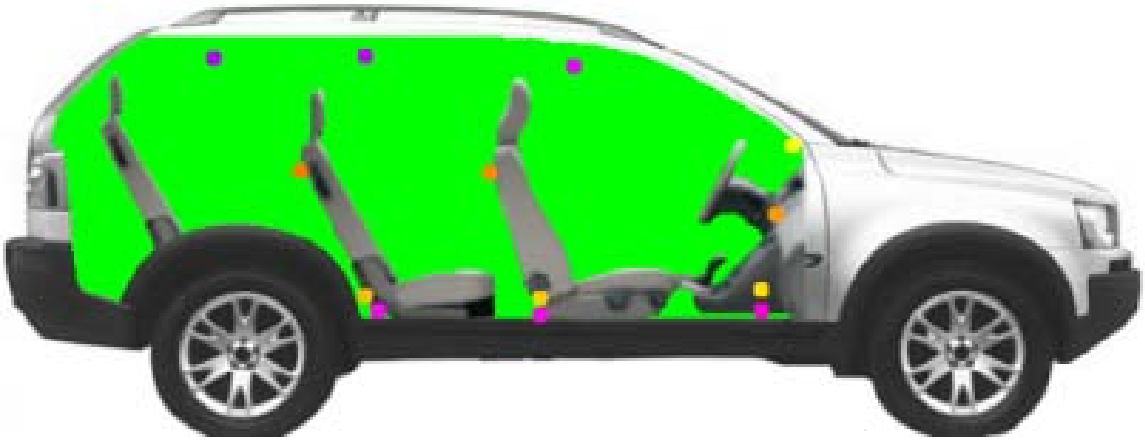


Figur 1. Förbehandling av CCM vy

Programmet läser vid start in bakgrundsbilderna och letar upp de förbehandlade färgerna. Eftersom det är just färgen som ska förändras i områdena så måste ett annat kännetecken kunna kopplas till respektive område, ett attribut som måste vara något annat än en färg eftersom denna är föränderlig under programmets körtid. Programmet utnyttjar därför den valbara opaciteten, ogenomskinligheten, hos de intressanta områdena. Vid initiering får pixlarna inom varje intressant område utifrån dess förbehandlade färg tilldelat ett värde i dess alfa-kanal som blir det kännetecken med vilket programmet kan känna igen pixlarna. Genom att välja värden på alfa-kanalen som ligger väldigt nära full ogenomskinlighet syns inte den lilla grad av genomskinlighet i pixlarnas färg som faktiskt existerar.

När programmet sedan läser i mätdatan att en viss diod ska tändas, letar den reda på diodens pixlar genom att söka efter tillhörande alfa- kanal på pixlarna i diodens område och därefter färglägga detta område så att dioden ser ut att lysa.

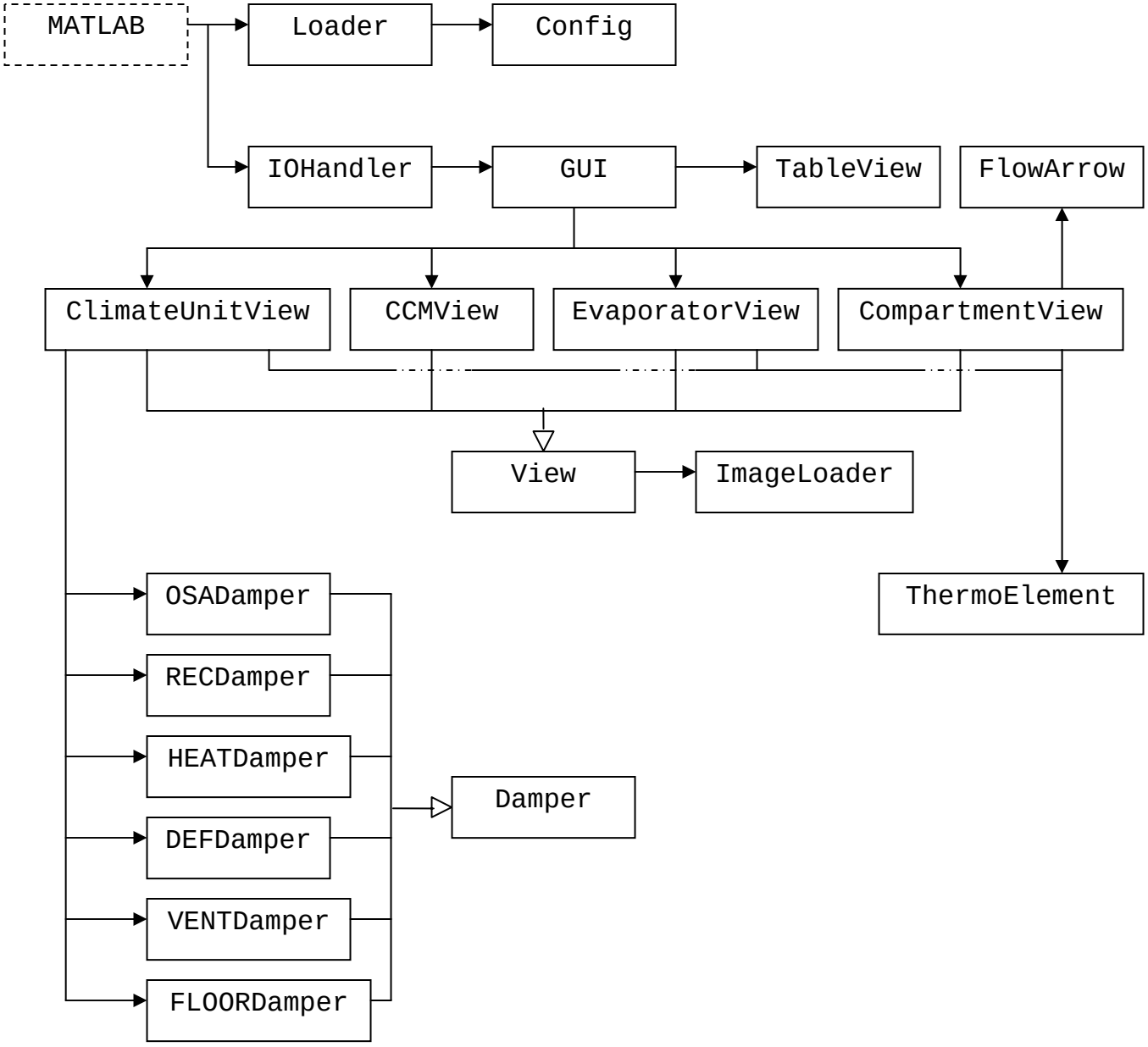
Samma strategi används för att veta var termoelementen är placerade i kupén, klimatenheten samt evaporatorn. Termoelementen markeras med unika färger, se figur 2, och när bilden läses in lagras koordinaterna för respektive termoelement. Informationen ligger sedan till grund för interpoleringen.



Figur 2. Förbehandling av termoelementens placering i kupén

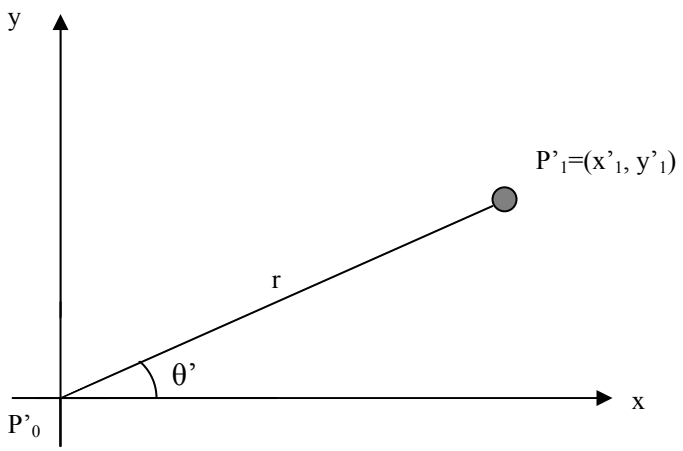
Genom att en klass, `BufferedImage`, ur Javas standardbibliotek används så kan en unik färg sättas för varje pixel i bakgrundsbilderna. Detta innebär också att en bra upplösning erhålls.

Appendix D Klassdiagram



Appendix E Viktning av ventilationer

Antag ventilationsmunstycke i punkten $P_0 = (x_0, y_0)$.



Ventilationstemperaturens påverkan i godtycklig punkt $P_1 = (x_1, y_1)$ beror av följande:

- det fysiska avståndet r till ventilationen
- fläkthastighet
- mode
- vinkel från munstyckets kastriktning i θ led.

P'_0 sätts till origo vilket ger $P'_1 = (x'_1, y'_1)$ där $x'_1 = x_1 - x_0$ och $y'_1 = y_1 - y_0$. För att räkna fram ett avstånd som ska vara underlag vid viktning av ventilationstemperaturen så används formeln:

$$s = r^2*(1+w*(\cosh(a*\theta')-1)) / (b*U_{fan}*mode)$$

där

$r = \sqrt{(x'^2_1 + y'^2_1)}$

$\theta = \text{atan}(y'_1 / x'_1)$;

$\theta' = \theta - \Delta\theta$ (koordinatsystemet roterat så att x-axeln ligger i ventilationsmunstyckets riktning, munstycket riktas med vinkeln $\Delta\theta$)

a = kalibrering av strålfokusering i höjdled

b = kalibrering av kastlängden

w = kalibrering av hur mycket ventilationstemperaturen ska viktas jämfört med temperaturer

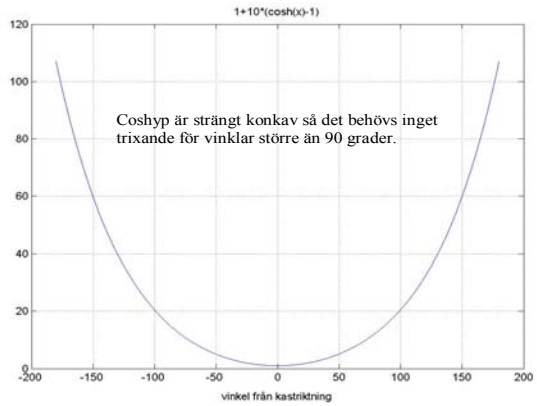
vid termoelementen vid golv och tak.

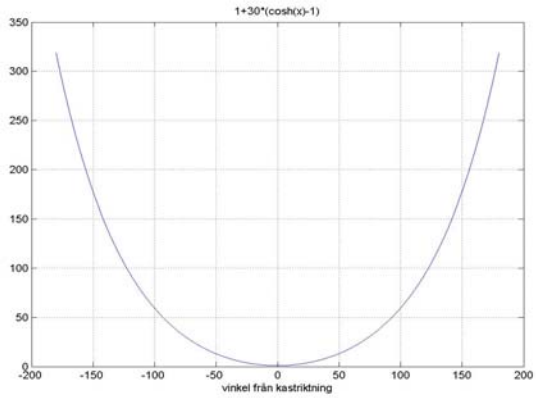
U_{fan} = fläkteffekt

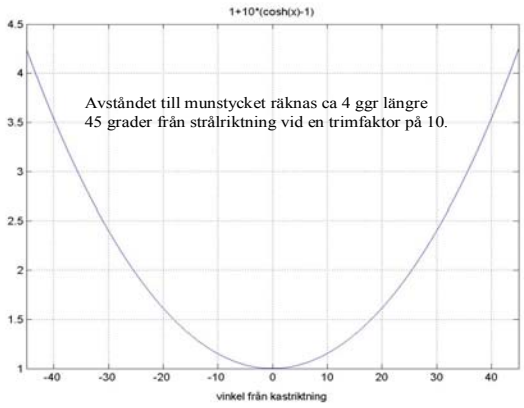
mode = andel av fläkteffekten

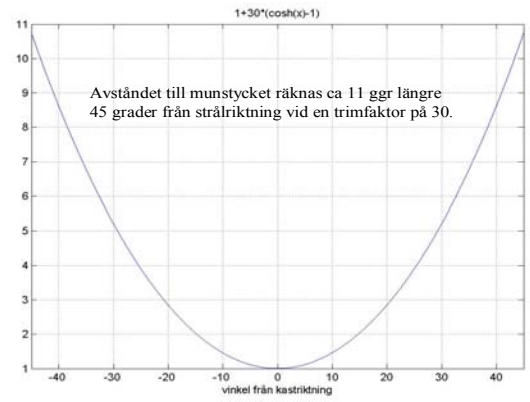
För termoelementen vid golv och tak bestäms avståndet för viktning till r, enligt Pythagoras sats.

Nedanstående kurvor visar på de av cosh-funktionens egenskaper som utnyttjas vid ventilationsberäkningarna.



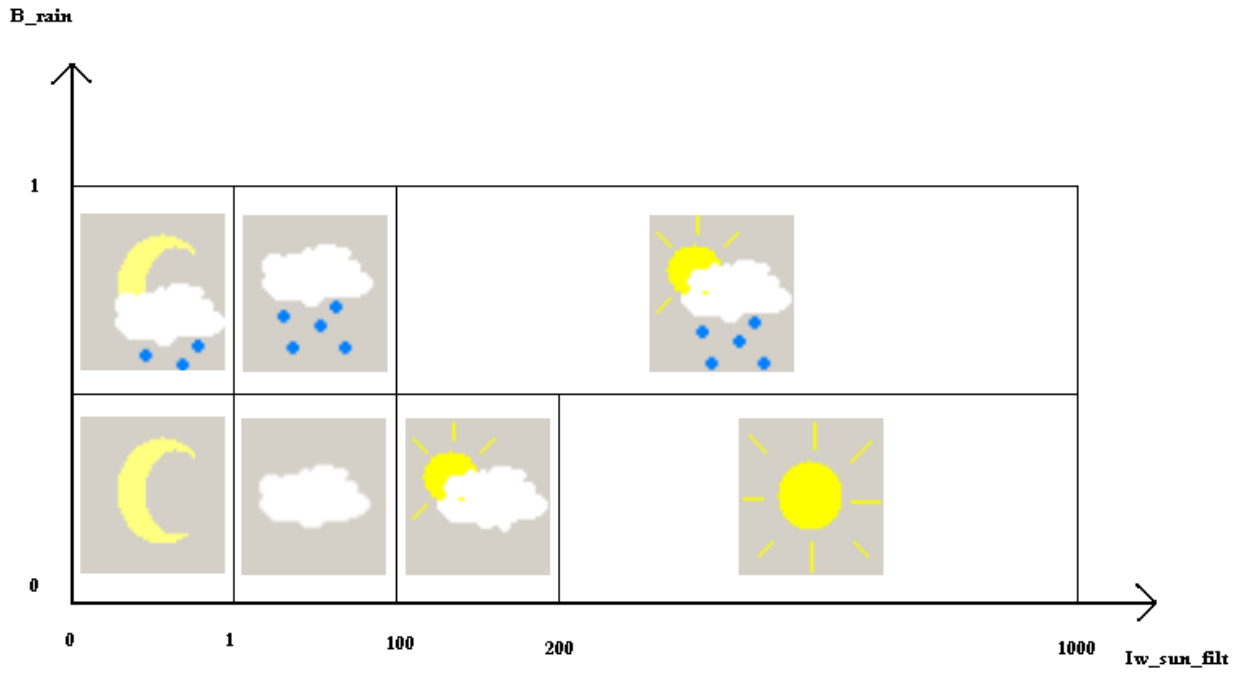






Appendix F Väderalgoritmen

För att uppskatta vilket väder som varit vid upptagningen av mätdata så studeras två signaler, `B_rain` och `Iw_sun_filt`. Med utgång från dessa signaler så har regler för vädret ställt upp enligt figur 1. Dessa regler är godtyckliga och har inte tagit alla aspekter i beaktning. Om man till exempel tillfälligt färdas i en tunnel så kommer solintensiteten, signalen `Iw_sun_filt`, att minska i värde och det kommer att illustreras med att vädret är molnigt, trots att vädret utanför tunneln mycket väl kan vara soligt.



Figur 1. Regler för väder

Appendix G Termoelementens placering i kupén

I figuren visas termoelementens placering för högersidan i kupén. Termoelementen på vänster sida har motsvarande position.