

TNM089: Bilder med High Dynamic Range

Cindy Truong (cintr523), Filip Kayar(filka428), Adam Nilsson(adani731)

24 oktober 2024



1 Introduktion

Dagens teknologi har utvecklats markant och blivit bättre men även mer avancerad. Speciellt inom foton och bildteknik har utvecklingen stigit och fört med sig nya och förbättrade metoder för att framställa bilder. En metod som är intressant och ska utforskas i denna rapport är framställningen av ”High dynamic range”-bilder, så kallade HDR-bilder.

En HDR-bild är en bild som ska kunna visa ett större dymaniskt omfång i en scen. En verlig scen kan innehålla delar som befinner sig i skugga med låg luminans medan andra delar kan befina sig i direkt solljus. En kamera kan inte ta en bild som både visar de mörka och ljusa delarna av en scen, vilket våra ögon gör bättre. Det ögonen gör är att anpassa ljusinsläppet för att se de mörka och ljusa delarna. Principen är densamma för HDR-bilder, att ta flera exponeringar som fångar de olika dynamikerna i en bild och sammanfogar dessa till en bild.

Det finns olika metoder för att skapa en HDR-bild, men följande rapport kommer främst utgå från metoden som beskrivs av Debevec och Malik [2]. Debevec och Malik skapar en HDR-bild från en serie av bilder på samma statiska scen, men med olika exponeringar för varje bild. Projektet kommer att implementera varje steg i HDR-kedjan och utforska olika metoder för att implementera de, samt utvärdera resultaten.

1.1 Syfte och frågeställning

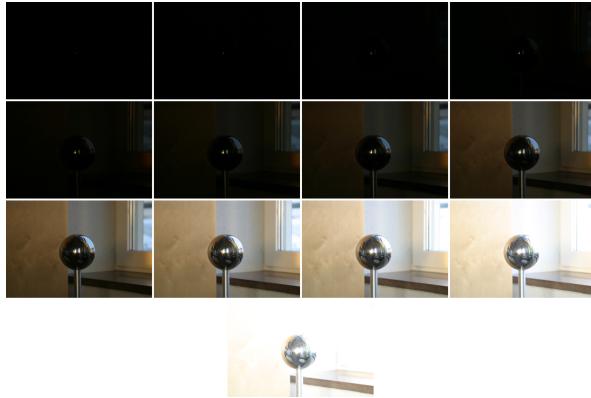
Denna rapport kommer att undersöka möjligheterna för att skapa en HDR-bild med olika metoder och tillvägagångssätt. Genom att använda olika metoder för att skapa en HDR-bild ska dessa jämföras för att framställa en så tillfredsställande bild som ”ser bra ut” för oss mänskor. Då det är beroende från mänsk till mänsk vad som ”ser bra ut” så kommer detta att diskuteras i diskussionen senare i rapporten. Rapporten kommer ta upp olika metoder att framställa olika känslighetsfunktioner (radiometric response functions) och utifrån de, bestämma hur en bra känslighetsfunktion ska se ut. Därefter ska en radiance map, d.v.s. en HDR-bild, tas fram, för att sedan konverteras för att kunna visas som en visbar bild på en vanlig skärm. Detta görs genom att använda tonmappning. Då det finns olika variationer av tonmappning med varierande resultat kommer dessa att implementeras och utvärderas för att bestämma den bästa.

Frågeställningarna som ska besvaras i denna rapport är:

- Hur väljs pixelvärdet på bästa sätt för att skapa en känslighetsfunktion? Hur många bilder i bildserien behövs i sådana fall, och hur ska de vara exponerade? Hur många pixelvärdet behövs?
- Vilka metoder finns det och vilken är bäst för att skapa en bra tonmappad bild?

1.2 Bildserie

För detta projekt valdes två bildserier med olika egenskaper för att kunna jämföra skillnaderna i den resulterande HDR-bilden. Den första bildserien, kallad *Stålkulans*, består av en reflekterande metallkula fotograferad mot en vägg i en kontorsmiljö. Denna serie har ett relativt litet omfång av både ljus och färger, med en övervägande grå färgton och små områden med hög ljusintensitet. I figur 1 visas bildserien som innehåller 13 så kallade exponeringsbilder (8-bit) med exponeringstider som sträcker sig från 1/4000 sekunder för de mörkaste bilden till 1.3 sekunder för den ljusaste. För denna bildserie finns färdiga känslighetsfunktioner, kalla dessa känslighetsfunktioner för referensfunktioner.



Figur 1: Bildserien *Stålkulan*.

Den andra bildserien, *Strömmen*, har ett betydligt större ljusomfång och högre färgmättnad. Bilderna är tagna utomhus längs Strömmen i Norrköping, med gula byggnader i bakgrunden, höstlös på gräset och ett träd med gröna löv i förgrunden. Trädet är fotograferat i motljus med solen som skiner genom löven och grenarna. Denna serie innehåller 9 bilder (8-bit) och har exponeringstider mellan 1/8000 sekunder och 1/50 sekunder. Bilderna illustreras i figur 2.



Figur 2: Bildserien *Strömmen*.

2 Teori och metod

Med bildserierna och dess exponeringstider i beredskap kan en HDR-bild skapas. I detta kapitel kommer teorin och implementationen redogöras för följande delmoment: att uppskatta en känslighetsfunktion, att skapa en radiance map (d.v.s. HDR-bilden) utifrån en känslighetsfunktion, och att utföra både global och lokal tonmappning på den resulterande radiance mappen. All implementation har utförts i MATLAB.

2.1 Känslighetsfunktion

Varje kamerasystem har ett unik sätt att mappa exponeringen från riktig infallande ljus till digitala värden. Denna systemspecifika mappning kallas för kamerans känslighetsfunktion.Debevec och Malik har definierat en metod att få fram denna känslighetsfunktion med hjälp av en bildserie av olika exponeringar, som exempelvis bildserierna i delkapitel 1.2. Mappningen från det att ett infallande ljus når kamerasensorn, digitaliseras, och ommappas till ett digitalt värde är sällan linjärt hela vägen - därför bör inte heller känslighetsfunktionen beskriva en linjär mappning. Känslighetsfunktionen kan snarare ses som en sammanfattnings av flera olinjära mappningar som sker när ljus omvandlas till pixelvärdet. När det gäller färgfotografering krävs det en känslighetsfunktion per R-, G-, och B-kanal.

Ett pixelvärde Z_{ij} kan beskrivas som en olinjär funktion av exponeringen $E_i \Delta t_j$, där i är pixelns position i en bild, och j är exponeringsindexet över alla exponeringstider Δt_j :

$$Z_{ij} = f(E_i \Delta t_j) \quad (1)$$

E_i avser relativa radians. Med omskrivning, där $g = \ln f^{-1}$, fås

$$g(Z_{ij}) = \ln E_i + \ln \Delta t_j \quad (2)$$

Funktionen g är alltså den sökta känslighetsfunktionen, som mappar logaritmen av exponeringen mot ett pixelvärde Z_{ij} . Både g och $\ln E_i$ är okända i ekvation (2), och kan estimeras med minsta kvadratiska fel genom att lösa ekvation

$$\mathcal{O} = \sum_{i=1}^N \sum_{j=1}^P \{w(Z_{ij})[g(Z_{ij}) - \ln E_i - \ln \Delta t_j]\}^2 + \lambda \sum_{z=Z_{\min}+1}^{Z_{\max}-1} [w(z)g''(z)]^2 \quad (3)$$

där N avser antalet relevanta pixelpositioner, och P är antalet exponeringar i bildserien. Den andra termen i ekvation (3) ser till att funktionen g blir mjuk med hjälp av dess andraderivata. Termen försöker "tvinga" de estimerade värdena att följa en mjukare kurva. λ är ett godtyckligt värde som styr effekten av andraderivatan, och därmed graden av mjukhet - ju högre λ -värde, desto mjukare blir funktionen g . I denna rapport betyder en funktions mjukhet både en mindre hackig, tandad kurva, och en överlag mer följsam kurva, d.v.s. extrema λ -värden gör alla funktioner g till linjära. För 8-bits kvantiserade bilder är $Z_{\min} = 0$ och $Z_{\max} = 255$, och viktfunktionen

$$w(z) = \begin{cases} z - Z_{\min} & \text{för } z \leq \frac{1}{2}(Z_{\min} + Z_{\max}) \\ Z_{\max} - z & \text{för } z > \frac{1}{2}(Z_{\min} + Z_{\max}) \end{cases} \quad (4)$$

är en hattfunktion som viktar mellanvärdena av pixelintervallet $[Z_{\min}, Z_{\max}]$ tyngre än ändvärdena Z_{\min} och Z_{\max} . Värden nära Z_{\min} och Z_{\max} i funktionen g är mindre tillförlitliga, eftersom många låga respektive höga exponeringsvärden förväntas mappas till ett få antal låga respektive höga pixelvärden. Funktionen g kommer därmed passa datan sämre nära ändvärdena, och därför bör dessa ändvärden ha mindre påverkan i utformningen av funktionen g [2].

Debevec och Malik har i sin artikel inkluderat en färdig MATLAB-funktion för ekvation (3) för att estima g och $\ln E_i$. MATLAB-funktionen kallas för *gsolve* och användes i sin helhet vid implementationen. *gsolve* kräver fyra inputs: en vektor med pixelvärden, en vektor med de logaritmatiserade exponeringstiderna för bildserien, ett λ -värde, och viktfunktionen från ekvation (4) i vektorform. De tre sistnämnda indata är triviala att implementera, men den förstnämnda är inte lika självtalar. Debevec och Malik noterar att inte alla pixelvärden i bildserien behövs, med tanke på hur komplex och beräkningstung estimeringen skulle bli. Därför bör endast några pixelvärden väljas på ett lämpligt sätt.

Två olika metoder implementerades för att välja pixelpositioner att sampla värden ifrån: genom att slumpa pixelpositioner, eller genom att "handplocka" pixelpositioner. För båda metoderna sampelas alla värden vid en viss pixelposition för alla exponeringsbilder; d.v.s. för pixelpositionen (1,1) får totalt 21 stycken pixelvärden om bildserien innehåller 21 stycken bilder, till exempel. För båda metoderna kan ett önskat antal pixelpositioner angas.

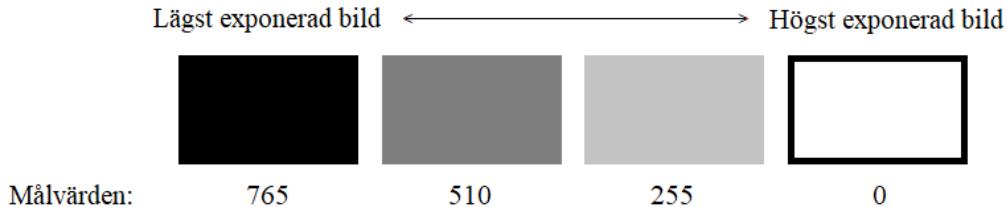
2.1.1 Slumpade pixelpositioner

Pixelpositionernas rad- och kolumnindex slumpsas enhetligt med hjälp av MATLAB-funktionen *rand*. Rad- och kolumnindexen sparas i varsin vektor, och vektorerna används sedan för att utvinna pixelvärdena från alla exponeringsbilder. Metoden är befogad, dåDebevec och Malik menar att pixelvärdena bör sampelas i ett spatialt jämnt avstånd mellan varandra i bildserien.

2.1.2 Handplockade pixelpositioner

Pixelpositionerna ”handplockas” på följande sätt: för den mörkaste exponeringsbilden hittas positionerna för de ljusaste pixlarna, för den näst mörkaste exponeringsbilden hittas positionerna för de näst mörkaste pixlarna, etc. Varje exponeringsbild kommer alltså ha ett unikt pixelvärde som mål, ett så kallat målvärde. Till exempel är målvärdet 0, svart, för den mörkaste exponeringsbilden - alltså väljs positionerna för pixlar med värden närmst 0 i denna bilden. Denna metoden implementeras med antagandet att pixelvärdet med ett större dynamiskt omfång kommer att väljas - för att därmed resultera i bättre estimerade känslighetsfunktioner - än metoden som slumper pixelpositioner från delkapitel 2.1.1. DeBevec och Malik noterar nämligen att pixelvärdena bör vara hyfsat jämnt fördelade från Z_{min} till Z_{max} , vilket inte kan säkerställas med den slumpade metoden. Genom att välja de ljusaste pixelvärdet i den mörkaste bilden, etc., fås ljusvariationerna från de mörkaste till de ljusaste punkterna av en scen - och därmed bör den ”handplockade” metoden resultera i ett stort dynamiskt omfång.

Ljusstyrkan för en exponeringsbild fås genom att summera bildens färgkanaler, kalla den resulterade bilden för summeringsbilden. Det betyder alltså att det längsta målvärdet blir 0, den mörkaste ljusstyrkan, och det högsta målvärdet blir $3 \cdot 255 = 765$, den ljusaste ljusstyrkan. Målvärdet fås med MATLAB-funktionen *linspace*, som delar upp målvärdet jämnt beroende på antalet bilder i bildserien, se figur 3. Sedan hittas alla unika ljusstyrkor i summeringsbilden, som vidare sorteras i stigande ordning utifrån hur nära varje unik ljusstyrka är målvärdet. Vidare beräknas förekomsten av varje unik ljusstyrka i summeringsbilden. Detta görs för att kunna säkerställa att exakt det önskade antalet pixelpositioner uppnås - varken fler eller färre.



Figur 3: Exempel på målvärden för en bildserie på 4 bilder.

Proceduren beskriven ovan görs alltså för varje bild i bildserien. Det innebär att varje bild bidrar med lika många pixelpositioner. Alla ”handplockade” pixelpositioner från alla bilder samlas i två vektorer, för positionernas rad- respektive kolumnindex, och därför kan pixelvärdet utvinnas på samma sätt som beskrivet i delkapitel 2.1.1. För att hindra att samma pixelpositioner väljs av nästkommande bild sätts ljusstyrkorna vid redan valda pixelpositioner till ett godtyckligt högt värde, så att dessa värden aldrig är tillräckligt nära målvärdet för att kunna väljas.

2.1.3 Utvärdering av linjäritet

Som tidigare nämnt bör den estimerade känslighetsfunktionen vara olinjär. För att bedöma linjäriteten krävs ett mått från linjär regressionsanalys: determinationskoefficienten R^2 . Med MATLAB-funktionen *fitlm* hittas den linje som bäst beskriver datan från känslighetsfunktionen. *Rsquared.Ordinary*-värdet från funktionen är R^2 -värdet som eftersöks [1]. R^2 beskriver hur väl *fitlm*-linjen beskriver känslighetsfunktionen: $R^2 = 1$ betyder det att *fitlm*-linjen beskriver känslighetsfunktionen perfekt, d.v.s. känslighetsfunktionen

är helt linjär. Är R^2 närmre 0 innebär det att känslighetsfunktionen snarare är olinjär. Då en bildserie resulterar i en känslighetsfunktion per R-, G-, och B-kanal fås tre stycken R^2 -värden. Bildseriens R^2 -värde beräknas som medelvärdet av dessa tre R^2 -värden.

2.2 Radiance map

Efter att känslighetsfunktionen g har tagits fram kan den användas för att omvandla pixelvärdet till relativ radians genom att använda en serie bilder med olika exponeringstider Δt_j . Exempelvis med stålkulorna från avsnitt 1.2. Denna omvandling gäller inte bara för de bilder som användes för att beräkna känslighetsfunktionen, utan kan även tillämpas på andra bildserier, förutsatt att samma kamera och inställningar används.

För att beräkna den relativa radiansen E_i vid en specifik pixel kombineras information från samtliga bilder tagna vid olika exponeringstider för samma pixelposition. Till skillnad från hur g beräknades, där endast ett fåtal pixelvärdet användes, används här alla tillgängliga pixlar för varje exponeringsbild. Radiansvärdet beräknas som en viktad summa, där exponeringstiden Δt_j och samma viktfunktion w , som användes vid framtagningen av g , tillämpas. Ekvationen (5) visar denna beräkning.

$$\ln E_i = \frac{\sum_{j=1}^P w(Z_{ij})(g(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^P w(Z_{ij})} \quad (5)$$

Precis som vid beräkningen av känslighetsfunktionen används viktfunktionen w här för att ge högre vikt åt exponeringar där pixelvärdet ligger närmare mitten av känslighetsfunktionen. Detta minimerar påverkan från över- eller underexponerade pixelvärdet, eftersom dessa inte bär någon användbar information. Med den här metoden kan brus och andra artefakter som exempelvis "blooming" reduceras, vilket resulterar i en mer representativ bild av scenens dynamiska ljusintensitet [2].

2.3 Tonmappning

När den färdiga radiance mappen (HDR-bilden) är skapad så återstår tonmappningen, vilket ska göra det möjligt att visa bilden på en skärm som har begränsad intervall av ljusstyrkenivåer. Då en radiance map kan anta värden från 0 och över 255 så måste den komprimeras och justeras så att innehållet kan visas med tillfredsställande resultat för en människa. Detta görs på olika sätt men i grunden så skalar tonmappningen om värdena för radiance mappen för att anta värden mellan 0 till 255 [4].

De två mest använda metoderna för detta är global tonmappning och lokal tonmappning. Den globala metoden applicerar samma metod på alla pixlar och tar ej hänsyn till hur pixlar relaterar till varandra. Lokal tonmappning är lite mer komplicerad och använder sig av filter för att ta hänsyn till pixlarnas position och relation. Dessa två metoder är olika men ska i grunden genomföra en mappning av radiance mappen för att ge ett "bra" resultat som användaren kan titta på.

2.3.1 Global tonmappning

I det här projektet användes fyra olika metoder för att tonmappa HDR-bilden globalt: linjär skalning i RGB, logaritmisk skalning i RGB, logaritmisk skalning i CIELAB och Reinhardmetoden.

Linjär tonmappning i RGB-färgrymden utfördes separat för varje färgkanal, som sedan sammanfogades efter att skalningen är genomförd. Denna metod normaliseras värdena till ett önskat dynamiskt omfång genom att först beräkna det maximala och minimala värdet i HDR-bilden. Värdena skalas sedan proportionellt ner till ett mindre acceptabelt omfång, vilket bevarar relationen mellan pixelvärdenes

ljusstyrka. Den linjär tonmappning resulterar ofta i bristande kvalité, med oftast områden som antingen är helt mörka eller helt ljusa.

Den andra metoden innebär logaritmisk skalning av HDR-bilden för att komprimera pixelvärdena till ett mindre omfång. Precis som i det linjära fallet utförs detta separat för de tre färgkanalerna (R-, G- och B-kanalerna). Skillnaden här är att alla pixelvärdet först multipliceras med den naturliga logaritmen (*log* i MATLAB) innan samma linjära operation från föregående stycke tillämpas. Den logaritmiska skalningen trycker ner omfanget på ett sätt som minskar skillnaderna mellan närliggande pixelvärdet i de ljusaste områdena, vilket innebär att dessa skillnader inte påverkar det nya omfanget lika mycket som skillnaderna i de mörkare partierna.

Den tredje metoden liknar den andra, men börjar med att konvertera HDR-bilden till CIELAB-färgrymden, som bättre motsvarar hur människor uppfattar färg och fångar perceptuella skillnader bättre. Genom att separera luminanskanalen (L) från de två färgkanalerna, a och b , kan ljusintensiteten justeras logaritmiskt utan att påverka färginformationen. Färgkanalerna behandlas separat och kan multipliceras med en faktor för att anpassa mättnaden, vilket gör det möjligt att både öka och minska färgmättnaden i bilden. Metoden kan ge godtagbara resultat, men för vissa bilder med lägre kontraster kan den ibland leda till sämre tonmappningar.

Den sista metoden för globala tonmappningar är en något mer avancerad teknik som kallas Reinhardmetoden. Informationen som används för detta baseras på artikeln [3]. I denna metod integreras både RGB-kanalerna och luminanskanalen, där processen inleds med att justera ljusintensiteten i HDR-bilden utifrån den genomsnittliga luminansen. Detta görs genom att beräkna den logaritmiska medelluminansen \bar{L}_w med hjälp av ekvation (6),

$$\bar{L}_w = \exp\left(\frac{\sum_{x,y} \log(\delta + L_w(x,y))}{N}\right) \quad (6)$$

där $L_w(x,y)$ är den HDR-bildens luminans för varje pixel (x,y) , δ är ett litet värde som förhindrar logaritmiska beräkningar med värden i intervallet $] -\infty, 0]$ och N är antalet pixlar i luminanskanalen. Genom att sedan använda \bar{L}_w , kan luminansen L_w skalias ner för att uppnå en mer anpassad genomsnittlig ljusstyrka, oberoende av olika bilders innehåll. Detta görs med ekvationen (7).

$$L(x,y) = \frac{a}{\bar{L}_w} L_w(x,y) \quad (7)$$

Variabeln a används här för att justera luminansen, så att över- och underexponering undviks. Problemet med denna beräknade luminans $L(x,y)$ är att vissa bilder kan ha områden med extremt hög luminans, vilket kan leda till oönskade starka ljuspunkter. För att hantera detta skalas luminansen ner ytterligare enligt ekvationen (8)

$$L_d(x,y) = \frac{L(x,y) \left(1 + \frac{L(x,y)}{L_{white}^2}\right)}{1 + L(x,y)} \quad (8)$$

I denna ekvation representerar L_{white} den högsta luminansen som ska mappas till ett rent vitt värde. Genom att justera detta värde kan man kontrollera hur mycket av de ljusaste delarna som tonas ner, vilket möjliggör en bättre hantering av ljuskontraster i bilden. När detta är gjort, kan det sista steget börja, där den justerade luminansen L_d används för att vikta den ursprungliga HDR-bilden. Denna process genomförs separat på varje RGB-kanal, och de resulterande kanalerna kombineras sedan. För att exempelvis beräkna den röda kanalen R kan ekvation (9) användas, där en variabel s kan väljas för att antingen öka eller sänka bildens färgmättnad.

$$R_{new} = \left(\frac{R_{org}}{L_w}\right)^s * L_d \quad (9)$$

Samma ekvation används för att beräkna de gröna och blå kanalerna, där R byts ut mot respektive kanal.

2.3.2 Lokal tonmappning

Lokal tonmappning använder sig av filter för att ta hänsyn till pixlars relation, vilket global tonmappning ej gör. Genom att använda dessa filter kan lokal tonmappning behålla detaljer i de flesta områdena som kan ge bättre bilder med detaljerade mörka och ljusa områden. De två olika metoderna som går igenom i denna rapport är metoder som använder två olika filterkärnor för att tonmappa bilderna.

Det generella fallet för hur lokal tonmappning fungerar är genom att först dela färgkanalerna i luminanskanal och kromakanaler. Själva kromakanalerna modifieras ej under tonmappningen utan behålls som de är tills de sätts ihop igen med luminanskanalen. För luminanskanalen så appliceras olika filter och modifieringar. Först lågpassfiltreras log luminanskanalen ($H_L(x, y)$) med filtret $B(x, y)$, som skapar ett grundläger som innehåller de låga frekvenserna, som kan ses i ekvation (10). Detta grundläger är en bild av luminanskanalens låga frekvenser, som sedan används för att skapa en högfrekvent bild genom att subtrahera det från log luminanskanalen, enligt ekvation (11). När detta är gjort finns det en bild av log luminanskanalens låga frekvenser men även en för de höga frekvenserna, H_L respektive H_H . Grundlagret, som innehåller de låga frekvenserna, är sedan skalad i logaritmisk skala med hjälp av filter för att få ett önskat resultat för ljusstyrkan, enligt ekvation (12). Det är när grundlagret skalias om som två olika metoder används, vilket är genom att filtrera med Gauss-filterkärna eller bilateral-filterkärna. Bildens höga frekvenser och den filtrerade bilden för de låga frekvenserna sätts samman för att skapa en ny log-luminans bild, ekvation (13) [4].

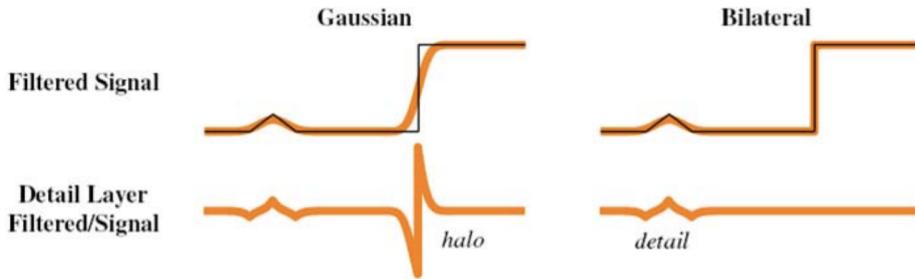
$$H_L(x, y) = B(x, y) * H(x, y) \quad (10)$$

$$H_H(x, y) = H(x, y) - H_L(x, y) \quad (11)$$

$$H'_L(x, y) = s \cdot H_L(x, y) \quad (12)$$

$$I(x, y) = H_H(x, y) + H'_L(x, y) \quad (13)$$

Som nämnt tidigare användes två olika filterkärnorna, Gauss och bilateral. I figur 4 ses det hur en signal blir filtrerad med de olika filterkärnorna. Här ses det att Gauss skapar en gloria kring skarpa kanter, medan bilateralfilter undviker det då de behåller kanterna vid filtreringen för grundlagret $H_L(x, y)$. Detta kommer att vara tydligt senare i resultatet. Detta beror på att bilateralfilterkärnor tar hänsyn till de spatiala komponenterna men även intensiteten, d.v.s. skillnaden på pixelvärdena.



Figur 4: Gauss- och bilateral filterkärna applicerade på en signal (Bild tagen från [4]).

3 Resultat

Detta kapitel kommer visa resultatet från de olika metoder som tagits upp i rapporten och används under projektets gång. De olika varianterna av känslighetsfunktioner, hur en radiance map ser ut och

resultatet av de olika tonmappningarna.

3.0.1 Känslighetsfunktion

Eftersom det fanns referensfunktioner för bildserien *Stålkulen* användes denna bildserie för att utvärdera resultatet av de estimerade känslighetsfunktionerna från båda metoderna som beskrivs i delkapitel 2.1.1 och 2.1.2. Alla 13 bilder i bildserien användes, samt 3 bilder i två olika scenarion: när de 3 bilderna är nära exponerade, d.v.s. med liknande exponeringstider nära mitten av bildserien, och när de är utspritt exponerade.

För att utvärdera totala antalet pixelvärden som behövs användes relativt få samt många värden: runt 14 stycken pixelvärden, samt runt 1400 stycken. För scenariot med 13 bilder krävdes alltså 1 respektive 100 pixelvärden per bild i bildserien, vilket resulterade i exakt 13 respektive 1300 stycken pixelvärden totalt. För båda scenarion gällande 3 bilder krävdes 5 respektive 500 pixelvärden per bild, som resulterade i 15 respektive 1500 stycken pixelvärden totalt.

Med variationen på metod, bildserie, och antal pixelvärden finns det alltså 12 olika fall av estimerade känslighetsfunktioner att undersöka. R-, G-, och B-kanalernas känslighetsfunktioner för respektive fall återfinns i kapitel 6, Bilagor, där x-axlarna motsvarar $\log E_i$, och y-axlarna motsvarar pixelvärden mellan 0 och 255. De svarta, tunnare kurvorna bakom de färgade kurvorna tillhör referensfunktionerna, för att underlätta jämförelsen av de estimerade känslighetsfunktionerna. Beteckningarna för varje fall redovisas i tabell tabell 1.

Av erfarenhet valdes $\lambda = 100$ för alla fallen, då valet gjorde vissa känslighetsfunktioner mjukare än andra, så att skillnaderna på de olika fallen kunde bedömas.

Tabell 1: Beteckningar för samtliga fall

	Antal pixelvärden	13 bilder	3 bilder, nära	3 bilder, utspritt
Slumpad	~14 st	A1	B1	C1
	~1400 st	A2	B2	C2
"Handplockad"	~14 st	A3	B3	C3
	~1400 st	A4	B4	C4

I tabell 2 sammanfattas varje falls R^2 -värde, som beskrivet i delkapitel 2.1.3, samt anmärkningar för enstaka fall. Fyra av fallen genererade en så kallad "Rank deficient"-varning i MATLAB, markerad som "R.D." i tabellen. Varningen uppkommer från *gsolve*-funktionen och påverkar validiteten av resultatet. Fallet A3 är markerad som "snirklig" i tabellen eftersom känslighetsfunktionerna för fallet är visuellt snirkliga, se Bilaga A3.

Tabell 2: R^2 -värdet för samtliga fall, samt anmärkningar

	Antal pixelvärden	13 bilder	3 bilder, nära	3 bilder, utspritt
Slumpad	~14 st	0.9570	0.9796	0.9767
	~1400 st	0.9277	0.9532	0.9229
"Handplockad"	~14 st	0.8885, snirklig	0.9745, R.D.	0.9987, R.D.
	~1400 st	0.9411	0.7682, R.D.	0.8959, R.D.

I tabell 3 har de fall utan anmärkningar i tabell 2 sorterats i stigande ordning med avseende på deras R^2 -värde.

Figur 5 illustrerar hur den slumpade metoden samt den "handplockade" metoden väljer 1300 stycken pixelpositioner för 13 bilder. De valde pixelpositionerna är färgade i rött.

Tabell 3: R^2 -värdet för fall utan anmärkningar, sorterat i stigande ordning

Fall	R^2
C2	0.9229
A2	0.9277
A4	0.9411
B2	0.9532
A1	0.9570
C1	0.9767
B1	0.9796



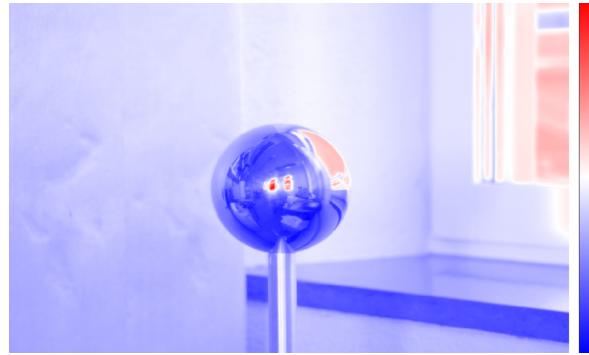
(a) Slumpade pixelpositioner.

(b) ”Handplockade” pixelpositioner.

Figur 5: De 1300 valda pixelpositionerna av de två olika metoderna, för 13 bilder.

3.0.2 Radiance map

Figur 6 visar resultatet av den radiance map som skapats från känslighetsfunktionen g på bildserien *Stålkulan*. De blå områdena representerar värden med låg radians, medan de röda områdena representerar hög radians.

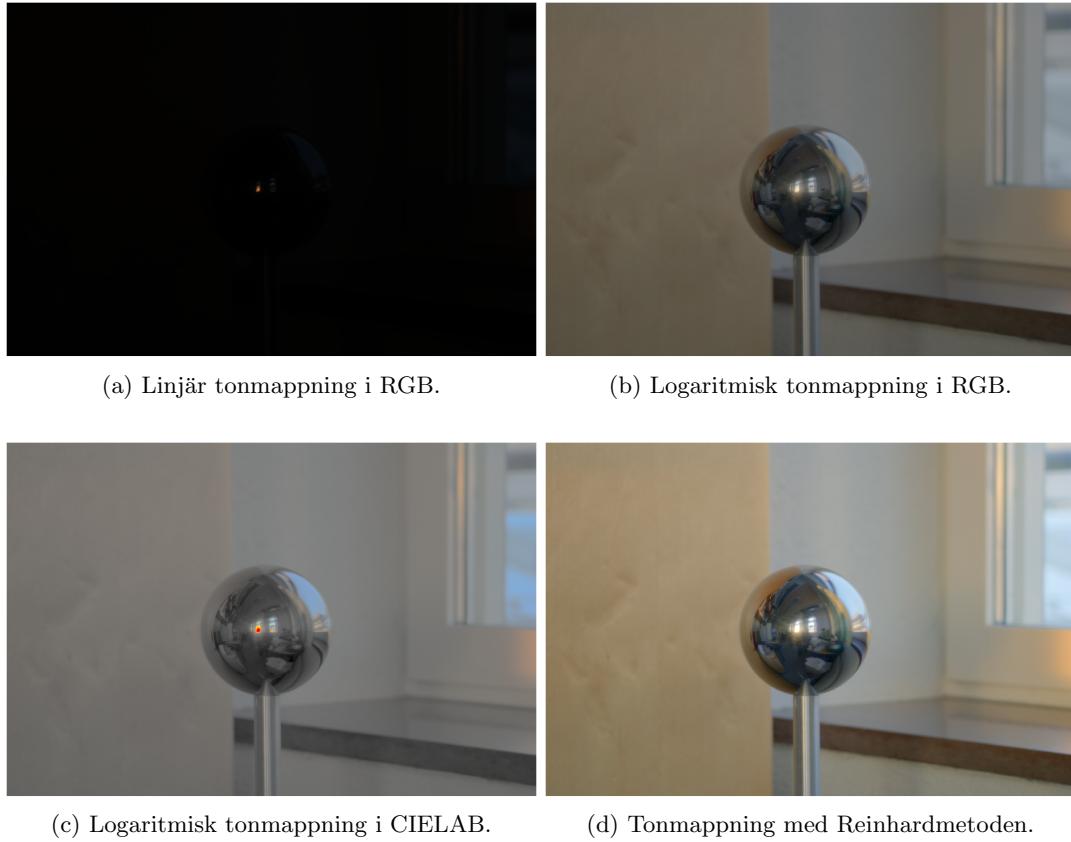


Figur 6: Radiance map som är färgmappad med färgen blå till röd.

3.0.3 Tonmappning

De metoder som presenterats i metoddelen 2.3 har använts för att skapa en HDR-bild med tonmappning. Detta resulterade i totalt åtta bilder från de två bildserierna för global tonmappning, samt fyra bilder för lokal tonmappning. Figur 7 visar resultatet av den globala tonmappningen för *Stålkulan*,

medan figur 8 illustrerar resultatet för *Strömmen*. Resultaten från den lokala tonmappningen visas i figur 9 för *Stålkulan* och i figur 10 för *Strömmen*.



Figur 7: Resultaten från de fyra olika metoderna för global tonmappning av *Stålkulan*.



(a) Linjär tonmappning i RGB.

(b) Logaritmisk tonmappning i RGB.



(c) Logaritmisk tonmappning i CIELAB.

(d) Tonmappning med Reinhardmetoden.

Figur 8: Resultaten från de fyra olika metoderna för global tonmappning av *Strömmen*.



(a) Lokal tonmappning med bilateral filtrering.

(b) Lokal tonmappning med Gaussian filter.

Figur 9: Resultaten från de två olika metoderna för lokal tonmappning av *Stålkulan*.



(a) Lokal tonmappning med bilateral filtrering.



(b) Lokal tonmappning med Gaussian filter.

Figur 10: Resultaten från de två olika metoderna för lokal tonmappning av *Strömmen*.

4 Diskussion

4.1 Känslighetsfunktion

Tabell 2 visar att fallen B3, B4, C3, och C4 genererade ”Rank deficient”-varningar, och därfor anses inte känslighetsfunktionerna för dessa fallen tillräckligt tillförlitliga att vidare överväga. Vidare kan även fall A3 bortses, eftersom den resulterade i känslighetsfunktioner som kunde mappa flera pixelvärden till en log exponering, se Bilaga A3. En unik exponering bör mappas till ett unikt pixelvärde för att anses vara en bra estimering.

Tabell 3 visar de 7 återstående, relevanta fallen. Enligt tabellen har fall C2 längsta R^2 -värde, och därfor är den också minst linjär. Som tidigare nämnt bör känslighetsfunktioner inte vara linjära. Innebär det då att det fallet med lägst R^2 -värde också korresponderar till de ”bästa” känslighetsfunktionerna? Inte nödvändigtvis! Se Bilaga C2: känslighetsfunktionerna är sannerligen olinjära, men de är något snirkliga. De är förvisso inte så pass smirklig att ett log exponering kan mappas till flera pixelvärden, men rimligtvis bör de estimerade känslighetsfunktionerna följa formen av referensfunktionerna. De plötsliga brantare backarna mitt i känslighetsfunktionerna för C2 tyder ju på att färre log exponeringar mappas till ett större spann pixelvärden, och därmed kommer närliggande exponeringar mappas till avsevärt olika pixelvärden i den digitala bilden. De mer horisontella delarna av kurvorna tyder på att flera log exponeringar mappas till ett mindre spann pixelvärden, vilket gör att dessa ljusstyrkor inte kommer vara väl representerade i den digitala bilden. Sammanfattningsvis kommer inte denna sortens mappning representera skillnaderna på ljusstyrkan i en scen väl. Tittar vi på bilagan för fallet med näst längsta R^2 -värdet i tabellen, fall A2, ser vi att känslighetsfunktionerna är desto mjukare och följer referensfunktionerna bättre. Det är därfor viktigt att inte endast förlita sig på R^2 -värdet, utan att även ta hjälp av sin visuella perception.

Dessutom, minns att λ styr graden av mjukhet för känslighetsfunktionerna, och att λ -värdet valdes till 100 för alla fallen. I stort sett alla känslighetsfunktioner kan korrigeras till mjukare funktioner om λ -värdet ökas, d.v.s. att ”tvinga” de estimerade värdena att följa en mjukare kurva. Vad innebär det för fall C2 och A2? Jo, de estimerade värdena för fall A2 behöver inte ”tvingas lika hårt” till att följa mjukare kurvor, d.v.s. pixelvärdena som valdes var tillräckligt bra för att producera stabila estimeringar, till skillnad från C2. Utifrån det generella utseendet av känslighetsfunktioner och R^2 -värdet bedöms fallet A2 ge mest tillförlitliga, mjuka, stabila, och därmed bäst estimerade känslighetsfunktioner.

Vi noterar att 6 av de återstående 7 fallen i tabell 3 avser den slumpade metoden att välja positioner att sampla pixelvärdena ifrån, som beskrivet i delkapitel 2.1.1. Endast ett av fallen avser den ”handplockade” metoden, från delkapitel 2.1.2. Vi kan därmed konstatera att metoden för att slumpa pixelpositioner är den bästa metoden, oavsett utformningen av bildserierna, och framförallt blir den bättre ju fler pixelvärden som skickas in i funktionen *gsolve*. Färre pixelvärden resulterar nämligen

i mer linjära känslighetsfunktioner, speciellt när bildserien innehåller färre bilder. Dessutom är den slumpade metoden enklare att implementera, och mindre beräkningstung - bättre i alla avseenden helt enkelt.

Vid fallen för bildserier med endast 3 bilder gynnas resultaten av fler pixelvärden, med avseende på R^2 -värdena i tabell 2. Detta gäller alltså fall B2 och C2. C2 må ha ett lägre R^2 -värde, men som tidigare diskuterat resulterar fallet i mindre tillförlitliga känslighetsfunktioner. Därför kan vi konstatera att mjukare känslighetsfunktioner fås vid få exponeringsbilder om bilderna är nära exponerade. Fallet med fler exponeringsbilder med färre pixelvärden, fall A1, presterar ungefär likvärdigt B2 och C2. Därför behövs färre pixelvärden om man har fler exponeringsbilder till sitt förfogande.

Som nämnt i delkapitel 2.1.2 implementerades den "handplockade" metoden med antagandet att den kommer prestera bättre än den slumpade metoden. Varför gjorde den inte det? En förklaring kan ges i figur 5. Den slumpade metoden väljer positioner jämnt spatialt utspritt från varandra över hela bilden, medan den "handplockade" metoden väljer positioner i koncentrerade områden. Troligtvis är det så att mer varierande ljusförändringar fångas in från pixlarna i den slumpade metoden, d.v.s. ett högre dynamisk omfang och mer information skickas in i funktionen *gsolve*. Eftersom positionerna i den "handplockade" metoden är så nära varandra kan inte de valda pixlarna beskriva ljusvariationerna lika väl, d.v.s. metoden skickar in för mycket liknande data till *gsolve*. Med andra ord skickas inte tillräckligt varierande data in till *gsolve*. Detta kan också förklara "Rank deficient"-varningarna som endast gällde fall med den "handplockade" metoden. Det enda "handplockade" fallet som presterade någorlunda bra var A4, med många bilder och pixelvärden. Antagligen presterade den bra just för att den fick så mycket information från alla 13 bilder och så många pixelvärden. Kanske skulle en kombination av de båda metoderna generera bästa resultat: den "handplockade" koden skulle behöva korrigeras för att välja samma pixelvärden men i mer utspridda områden av bilden.

4.2 Radiance map

Den resulterande radiance-mappen håller god kvalité och kunde därför användas för att testa olika tonmappningsmetoder, vilket resulterade i HDR-bilder med ett godtagbart dynamiskt omfang. Genom att analysera bilden, ser man att den verkar korrekt då den indikerar att de ljusaste partierna i bilden är fönstret och reflektionen av det i stålkulan.

4.3 Global och lokal tonmappning

De globala metoderna för tonmappningen gav generellt goda resultat för båda bildserierna, med undantag för den linjära mappningen. Detta för att den presterade dåligt eftersom pixlarna är mestadels svarta med ett lite ljust område. Den logaritmiska skalningen i RGB fungerade relativt bra för både *Stålkulan* och *Strömmen*. Nackdelen med denna metod är dock att färgerna tenderar att bli "utvattnade", vilket är särskilt tydligt i *Strömmen*, som har ett större omfang av färger. I *Stålkulan* är effekten mindre märkbar eftersom färgomfanget redan är begränsat. Detta visar att metoden kan fungera, men passar något bättre för bilder som inte behöver innehålla starka kontraster av färger.

Den logaritmiska skalningen i CIELAB-färgrymden gav liknande resultat som skalningen i RGB för *Stålkulan*. Dock uppstår en tydlig artefakt i det ljusaste området av kulan, vilket påverkar bildens kvalité negativt. För *Strömmen* resulterade denna metod i låg färgmättnad i flera delar av bilden medan vissa områden, som himlen, gav god färgmättnad. En fördel med att mappa i CIELAB är att den möjliggör justeringar i färginnehållet utan att påverka ljusintensiteten. Med det sättet som vi har implementerat metoden, visar det dock att det är svårt att hitta en lämplig skalfaktor för färgkanalerna.

Reinhardmetoden presterade bäst för våra resultat i båda bildserierna. Denna metod tillät mest justering genom de tre parametrarna a , L_{white} och s , vilket gjorde det enkelt att kontrollera både ljusintensiteten och färgmättnaden av HDR-bilden. På grund av detta anser vi att Reinhard ger de

bästa resultaten för båda bildserierna. Detta är dock en subjektiv bedömning, och andra personer kan föredra olika metoder beroende på sina personliga preferenser.

För lokal tonmappning fås resultaten enligt figur 9 och figur 10 som visar två olika scener med de två olika metoderna som använder sig av olika filterkärnor. Något som kan noteras direkt är att de är relativt lika jämfört med de fyra metoderna som användes i global tonmappning vilket gav större skillnader. För figur 9 ses det att i själva *Stålkulan* är detaljerna bättre för a) jämfört med b), detaljerna i b) är suddigare och oskarpa. För b) kan även en gloria ses kring *Stålkulan* vilket inte syns för bilaterala filter. Det bilaterala filtret bevarar detaljerna bättre än Gauss och ger skarpere och tydligare detaljer. Detta kan även ses lite på väggen, där a) har en gnutta bättre detalj i trädväggen jämfört med b). I figur 10 är de två olika filtren applicerade på *Strömmen* och här är skillnaden svårare att se. I a) kan vi se att det bilaterala filtret har lyckats bibehålla detaljerna i barken lite bättre än b) gjort. Det är även bättre skarpa på bilden generellt för det bilaterala filtret i a), texten på skylten är lite tydligare och bladen är skarpere. Detta kan däremot vara lite svårare att se om man inte zoomar in och kollar ordentligt. För Gaussfiltret är färgerna lite fylligare om man kollar på trädet och dess blad men även på husen i bakgrunden. Färgerna i b) ser djupare ut medan a) ser blekare ut, om man jämför de sida vid sida. Om man ska dra en slutsats från vilken metod som är bäst av bilaterala och Gauss så är det upp till varje person vad de tycker men enligt egna bedömningar väljs lokal tonmappning med bilaterala filter som den bättre.

Vid en jämförelse mellan global och lokal tonmappning framstår vissa skillnader. Båda metoderna ger bra resultat, men de har också sina svagheter som bör tas i beaktning innan man väljer tillvägagångssätt för att skapa HDR-bilder. För global tonmappning anser vi att Reinhard presterade bäst, medan det bilaterala filtret gav de bästa resultatet för lokal tonmappning. Frågan blir då vilken av dessa som överlag är bäst att använda. Återigen är detta subjektivt, där Reinhard kan vara att föredra för de personer som vill ha mer kontroll över färg och ljusintensitet. Denna metod är även mindre beräkningskrävande, vilket kan vara en anledning till varför den väljs. Men om prestandan inte är något problem och man uppskattar den visuella stilens som den lokala tonmappningen ger, kan den metoden vara bättre. Därför bedömer vi att båda metoderna är lika användbara då det är beroende på situation och personliga preferenser.

5 Slutssats

Sammanfattningsvis har rapportens syfte uppnåtts genom att ha tagit fram två olika metoder för att välja pixelvärdet för att skapa känslighetsfunktioner, och även testat olika metoder för tonmappning. Resultatet visar att metoden för att slumpa pixelpositioner ger de bästa känslighetsfunktionerna överlag. Vidare kan vi konstatera att ju fler exponeringsbilder och ju fler pixelvärdet, desto bättre resultat. Rapporten tar upp flera metoder för tonmappning med varierat resultat, men landar slutligen i att Reinhard är en bra metod för global tonmappning och bilaterala filter för lokal tonmappning. Rapporten framställer tillfredsställande resultat med god kvalité och varierande visuell stil, men då de två utvalda metoderna båda ger bra resultat är det svårt att avgöra den bästa. Det är då upp till observatörens preferenser och vad personen tycker ser bra ut.

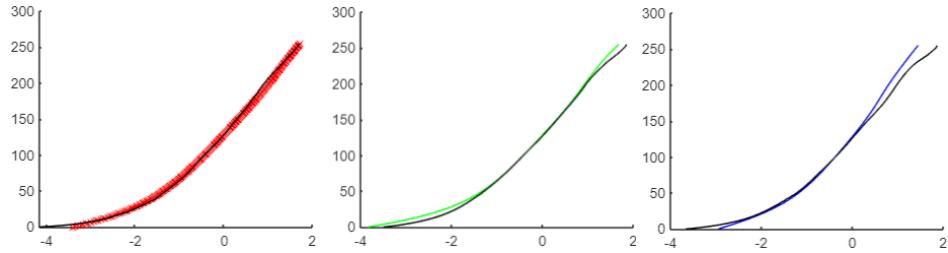
Referenser

- [1] MathWorks. Coefficient of determination (r-squared). <https://se.mathworks.com/help/stats/coefficient-of-determination-r-squared.html#btwmhik-5>, 2024.
- [2] Jitendra Malik Paul E. Debevec. *Recovering High Dynamic Range Radiance Maps from Photographs*. University of California at Berkeley, 1997.
- [3] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. 2002. URL: https://www.researchgate.net/publication/2908938_Photographic_Tone_Reproduction_For_Digital_Images.

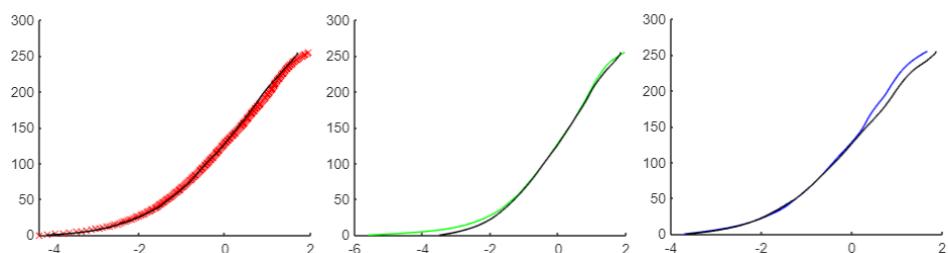
- [4] Richard Szeliski. Computer vision: Algorithms and applications. 2010.

6 Bilagor

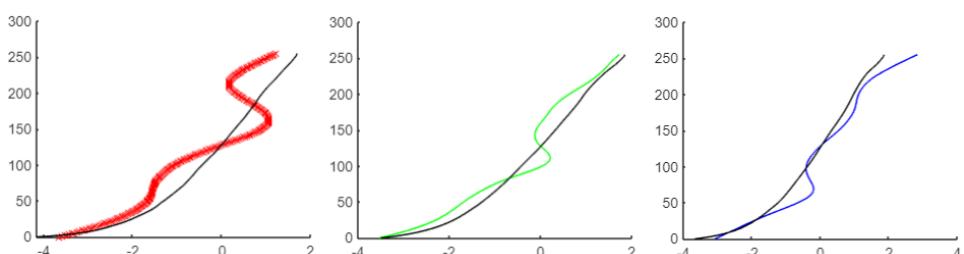
Bilaga A1



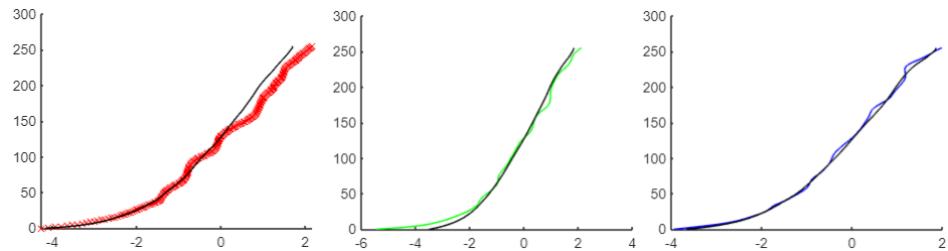
Bilaga A2



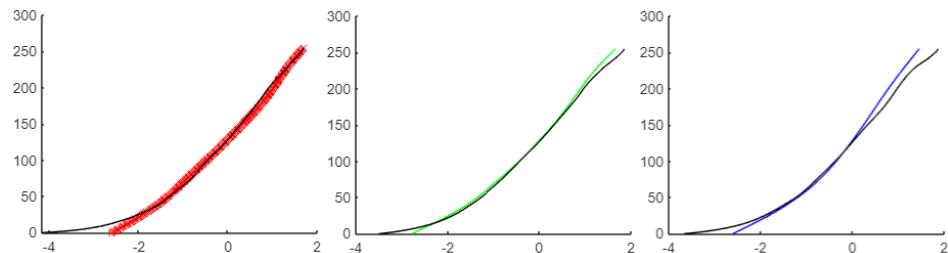
Bilaga A3



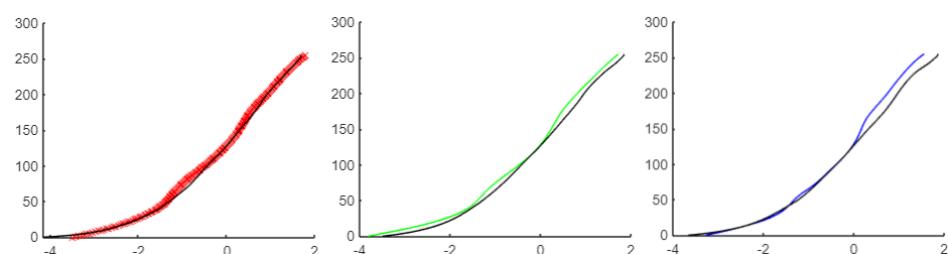
Bilaga A4



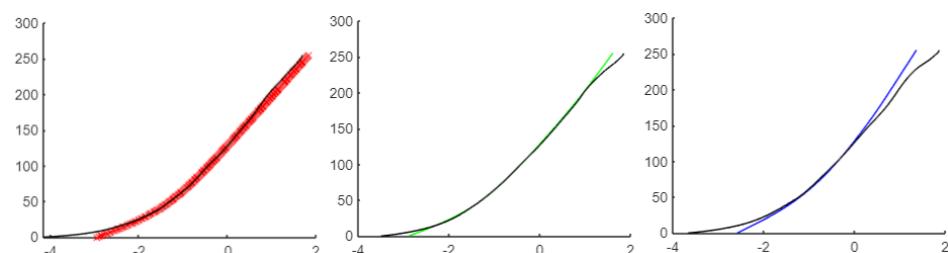
Bilaga B1



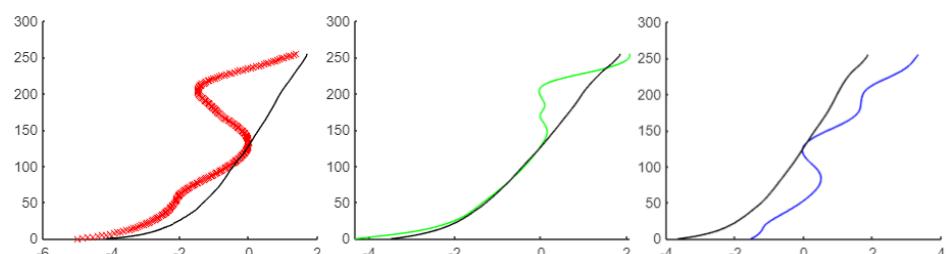
Bilaga B2



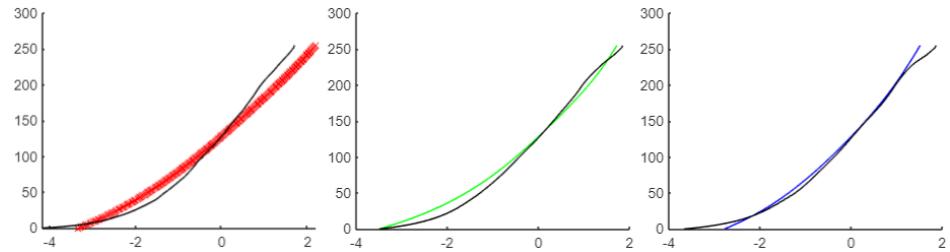
Bilaga B3



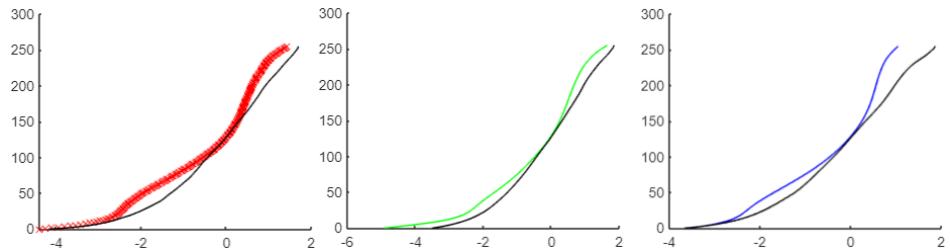
Bilaga B4



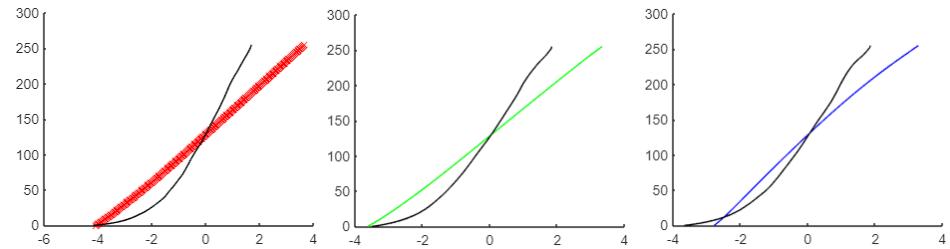
Bilaga C1



Bilaga C2



Bilaga C3



Bilaga C4

