

TNM059 – Grafisk teknik

Lab 2 – Digital Rastrering

DEL 2 Laboration

Digital Rastrering

Denna laboration innehåller ett antal uppgifter som ska lösas med hjälp av MATLAB. Era svar bör skrivas i ett dokument [Lab_2.2_Laboration_Svar.docx](#), där ni dessutom infogar erforderliga bilder.

OBS: Mycket viktigt: För att spara bilder, använd **bara** MATLAB funktionen ***imwrite***. Spara bilderna **enbart** i formatet **.tif** i denna laboration, med upplösningen 150 dpi. För att spara en bild, t.ex. ***b11***, till en **.tif** bild som heter ***b11.tif***, använd kommandot nedan. Kommandot sparar bilden med upplösningen 150 dots (eller pixlar)/tum, vilket gör att de får lämplig storlek när ni infogar dem i Word.

```
>> imwrite(b11, 'b11.tif', 'Resolution', 150);
```

OBS: Mycket viktigt: skala inte om bilderna efter att ni har infogat dem i Worddokumentet!

OBS: Spara svarsdokumentet som **.pdf** innan ni lämnar in det på Lisam.

För uppgifterna i detta dokument behöver ni inte lämna in några m-filer, men vi rekommenderar starkt att ni sparar era experiment i en m-fil, ifall ni behöver gå tillbaka och rätta till något senare. Ibland, kan ni återanvända era koder i senare uppgifter.

Alla bilder och funktioner som ni kommer att behöva finns på Lisam under: [Kursdokument/Labbar/Lab2/files](#) och kan även nås via fliken Laborationer på startsidans vänstermeny.

Börja med att läsa in bilden ***kvarn.tif*** i Matlab m.h.a. funktionen ***imread***. Alla uppgifter ska utföras på denna gråskalebild som är **512 × 512** pixlar stor. I dessa uppgifter antar vi att bilden är normerad så att dess pixelvärden ligger mellan 0 och 1. Börja med att **normera** bilden efter att ni har läst in den (**Se** avsnitt 5 i Lab1).

1) Tröskelrastrering

För att kunna göra dessa uppgifter krävs det att ni har läst avsnitt 3.2.1 i kompendiet eller/och motsvarande del från föreläsningsanteckningar.

1.1.

Börja med att rastrera bilden genom att tröskla den med en fix tröskel, t.ex. 0.5 (se avsnitt 4 i Lab1 för att se hur man enkelt kan göra det med ett enradigt MATLAB kommando). Döp resultatbilden till ***b11***. Hur många grånivåer representerar den? Infoga resultatbilden i svarsdokumentet!

1.2.

Ladda filen ***troskelmatriser.mat*** m.h.a. funktionen ***load***. Nu har ni fått tre tröskelmatriser, ***tr1***, ***tr2*** och ***tr3***. Ta först en titt på matriserna ***tr1*** och ***tr2***!

Hur många grånivåer representerar ***tr1*** resp. ***tr2***? Resonera!

Rastrera bilden med *tr1* och *tr2* m.h.a. funktionen *troskel* och döp resultaten till *b12_tr1*, resp. *b12_tr2*. Glöm inte att matriserna skall normeras så att alla tröskelvärdena ligger mellan 0 och 1. (Hur?) Infoga resultatbilderna i svarsdokumentet och förklara vad ni ser för skillnad mellan resultatbilderna.

1.3.

Ta en titt på tröskelmatriken *tr3*! Hur många grånivåer representerar denna matris?

Jämför denna matris med *tr2*. Vad är den största skillnaden mellan *tr2* och *tr3*?

Rastrera bilden med *tr3* m.h.a. funktionen *troskel* och döp resultat till *b13*. Glöm inte att tröskelmatriken skall normeras så att alla tröskelvärdena ligger mellan 0 och 1. Infoga resultatbilden i svarsdokumentet och förklara vad ni ser för skillnad mellan *b13* och *b12_tr2*.

1.4.

Tröskla bilden med två egna 4×4 tröskelmatriker. Den ena ska representera linjeraster och den andra spiraleraster. Skriv era egna tröskelmatriker och infoga resultatbilderna i svarsdokumentet.

Nu svara på följande frågor gällande tröskelrastrering:

1.5.

Förklara hur tröskelmatrikens storlek kan relateras till *lpi*, dvs. leder större tröskelmatrik till högre eller lägre *lpi*? Resonera!

Ledning: studera era resultatbilder i uppgift 1.2 (dvs *b12_tr1* och *b12_tr2*) för att se hur tröskelmatrikens storlek relateras till storleken av rasterpunkterna. Fundera sedan på hur rasterpunkternas storlek relateras till *lpi*. **Läs avsnitt 3.1.1** i kompendiet som tydligt förklarar vad *lpi* är. Titta speciellt på **Figur 3.4** på sidan 11.

1.6.

Givet en tröskelmatrik: Förklara hur man tar reda på antalet grånivåer som tröskelmatriken representerar. **Ledning:** läs sidan 16 (speciellt sista stycket) i kompendiet.

2) Tabellrastrering

För att kunna göra dessa uppgifter krävs det att ni har läst avsnitt 3.2.2 i kompendiet eller/och motsvarande del från föreläsninganteckningar.

2.1.

Använd funktionen *tabellrast* för att tabellrastrera bilden och döp resultatet till *b21*. Det är en mycket enkel funktion som använder medelvärdet av varje 4×4 omgivning i inbilden som index till en 4×4 rastertabell. Hur många grånivåer kan dessa representera? Infoga resultatbilden i svarsdokumentet.

2.2.

Använd funktionen *tabellrast2* för att tabellrastrera bilden och döp resultatet till *b22*. Även denna funktion använder medelvärdet av varje 4×4 omgivning i inbilden som index till en 4×4 rastertabell. Infoga resultatbilden i svarsdokumentet.

2.3.

Trots att båda tabellrastreringarna i uppgift 2.1 och 2.2 ger upphov till lika många grånivåer ser resultatbilderna annorlunda ut. Förklara vad som är skillnaden genom att titta på dessa två resultatbilder (dvs. *b21* och *b22*). **Ledning:** läs sidan 25 i kompendiet.

2.4.

Använd funktionen *tabellrast3* för att tabellrastrera bilden och döp resultatet till *b24*. Den här funktionen använder medelvärde av varje 2×2 omgivning i inbilden som index till en 4×4 rastertabell. Infoga resultatbilden i svarsdokumentet. Hur många grånivåer kan dessa representera? (**OBS:** titta noga på bilden samt fundera på rastercellernas storlek innan ni svarar på detta). Hur stor (pixel x pixel) blir resultatbilden? Förklara varför resultatet blir dubbelt så stor som inbilden i varje led?

3) Felspridning (Error Diffusion)

För att kunna göra dessa uppgifter krävs det att ni har läst avsnitt 3.2.3 i kompendiet eller/och motsvarande del från föreläsningsanteckningar.

Funktionen *errordif* kan användas för att rastrera en gråskalebild enligt felspridningsmetoden. Använd kommandot *help* för att ta reda på hur man använder funktionen *errordif*:

```
>> help errordif
```

3.1.

Använd filtret som Floyd och Steinberg har föreslagit (Figur 3.19a (sidan 29) i kompendiet) och rastrera originalbilden enligt felspridningsmetoden med funktionen *errordif*. Kalla resultatbilden för *b31* och infoga den i svarsdokumentet.

3.2.

Använd filtret som Jarvis, Judice och Ninke har föreslagit (Figur 3.19b (sidan 29) i kompendiet) och rastrera originalbilden enligt felspridningsmetoden med funktionen *errordif*. Kalla resultatbilden för *b32* och infoga den i svarsdokumentet.

3.3.

Diskutera skillnaderna ni ser mellan bilderna *b31* och *b32* (t.ex. vilken är skarpare, i vilken av dem ser ni tydligare störande strukturer i ljusa delar resp. mellantonsområden).

3.4.

Rastrera bilden med två egna felfilter. Det ena filtret ska endast innehålla en vikt (ett element) och det andra två vikter (två element). Skriv vilka filter ni använt och infoga resultatbilderna i svarsdokumentet. I *help*-kommandot för *errordif* står det tydligt hur man ska definiera sina filter. Skriv också åt vilket håll (eller vilka håll) era felfilter sprider felet.

3.5.

Det finns två artefakter associerade med den icke-modifierade felspridningsmetoden. Vilka är de och i vilka delar (skuggor, mellantoner, högdagrar) av en bild är var och en av dem mer synlig? (**Ledning:** läs sidan 32 i kompendiet)

3.6.

För att minska effekten av dessa två artefakter finns ett antal förslag på modifiering av felspridningsmetoden. Här ska vi göra en enkel modifiering genom att lägga till brus på inbilden, före rastering. Börja med att lägga till (**addera**) brus till inbilden. För att generera brus kan ni använda funktionen **rand**, men observera att bruset måste ha medelvärdet noll. Skapa en brusmatris som är lika stor som inbilden. Eftersom **rand** likformigt generar slumpstal mellan noll och ett, blir medelvärdet 0.5. Om man adderar denna matris till inbilden kommer den brusiga bilden att bli ljusare än originalet. För att undvika detta bör bruset ha medelvärdet 0, vilket enkelt kan göras genom att subtrahera 0.5 från matrisen. Denna matris kommer nu att innehålla slumpstal mellan -0.5 och 0.5 och eftersom originalbildens pixelvärden ligger mellan 0 och 1, kommer brusnivån vara för hög. Man kan sänka bruset genom att multiplicera brusmatrisen (som har medelvärdet 0) med en faktor, k , som är mindre än 1.

Addera nu en lämplig mängd brus till inbilden och rastera den resulterande brusbilden med samma filter som i uppgift 3.1 (dvs. Floyd-Steinberg) och kalla resultatbilden för **b36**. Om faktorn k är nära 1 då blir resultatbilden mycket brusigt, och om den är nära 0, kommer den likna bild **b31**, där artefakterna är tydligt synliga. Skriv vilken faktor k ni har använt för en tillfredställande resultatbild och infoga resultatet i svarsdokumentet.

4) Iterativ rastering

Använd funktionen **imcdp** för att rastera bilden med IMCDP metoden och döp resultatbilden till **b40**. Den här metoden går genom under en föreläsning och finns även kortfattat beskriven i avsnitt 4.1.1 i kompendiet. Detaljer om hur denna metod fungerar är utanför kursens ramar. Infoga resultatbilden i svarsdokumentet. Ser ni några skillnader mellan denna bild och **b31**? (T.ex: vilken är skarpare, vilken lider mindre av artefakter och tydliga strukturer?)

5) Objektiva kvalitetsmått

För att kunna göra dessa uppgifter krävs det att ni har läst kapitel 5 i kompendiet eller/och motsvarande del från föreläsningsanteckningar.

5.1.

Använd den fördefinierade funktionen **snr** i MATLAB som beräknar signal to noise (SNR) värdet. **OBS:** Första argumentet i funktionen ska vara signalen (originalbilden) och det andra argumentet ska vara "noise" (**OBS:** "noise" i detta fall är skillnaden/differensen mellan originalbilden och rasterbilden). Ett större värde på SNR betyder bättre rasterbild, eller? Testa nu denna funktion på rasterbilderna från uppgift 1.1 (bild **b11**), 3.1 (bild **b31**) och 4 (bild **b40**) och skriv SNR-värdena för dessa tre bilder. Vilken rasterbild liknar originalet mest resp. minst enligt måttet? Stämmer det överens med er bedömning av kvalitet? Varför? (**Ledning:** Läs noga avsnitt 5.1 på sidan 53 i kompendiet innan ni svarar)

5.2.

Studera hur funktionen **snr_filter** är skriven genom att öppna funktionen och läsa alla kommentarer. Efter att ni har förstått hur den fungerar, testa den på rasterbilderna från uppgift 1.1 (bild **b11**), 3.1 (bild **b31**) och 4 (bild **b40**) och skriv de modifierade SNR-värdena för dessa tre bilder. Vilken rasterbild liknar originalet mest resp. minst enligt måttet? Stämmer det bättre överens med er bedömning av kvalitet? Varför? (**Ledning:** Läs noga avsnitt 5.2 på sidan 54 i kompendiet innan ni svarar)

5.3.

Studera hur funktionen **QNS** är skriven genom att öppna funktionen och läsa alla kommentarer. **OBS:** Första argumentet i funktionen ska vara originalbilden och det andra argumentet ska vara rasterbilden. Efter att ni har förstått hur den fungerar testa den på rasterbilderna från uppgift 1.1 (bild **b11**), 3.1 (bild **b31**) och 4 (bild **b40**). Infoga alla tre QNS-bilderna (innan ni infogar bilderna, jämför era bilder med bilderna i Figur 5.3 i kurskompendiet för att säkerställa att ni inte använt funktionen på felsätt). Vilken rasterbild liknar originalet mest resp. minst enligt måttet? Varför? (**Ledning:** Läs noga avsnitt 5.3 på sidan 55 i kompendiet innan ni svarar)