Finn Fem Fel

- ett försök att hitta skillander i bilder

UPPSALA
UNIVERSITET

Signaler och System
Höstterminen -02
IT3

Rasha Alshammari
Andreas Nissemark
Zakai kass-Saliba
Pavel Carballo

Innehållsförteckning

Introduktion

”Finn fem fel” - det ultimata mandomsprovet.

Det är ingen hemlighet att vår tids mest spektakulära paradox är människans genialitet i förhållande till hennes inkompetens. I och med alla teknikdeterministiska idéernas födelse i mitten – slutet av 1980 talet, har vi kunnat ge tekniken, dess förtjänade betydelse för vår utveckling. Tekniken har frälsat oss från oss själva.
Nej, detta är inte ett utdrag ur någon sorts teknisk-deterministiskt manifest, vars sidor präglas av slangord som *Tekniker, förena eder!* Detta handlar istället om ett försök att uppmana läsaren till att inte förlora perspektivet när det handlar om teknisk utveckling och tekniska lösningar.
Vi har, med hjälp av ett superkraftfullt verktyg kallad Matlab och en dator som klarar att räkna flera tusen flyttalsoperationer per sekund, utvecklat ett program, som kan lösa ett problem som gäckat människorna sedan tidernas begynnelse, Finn fem fel i två bilder.

Sammanfattning

Det finns ett antal kända metoder och algoritmer för att hitta skillnader i bilder. Bortsett från vår något sarkastiska introduktionsförfattare, är vi alla överens om att vår applikation har oändliga tillämpningsområden. Från polisarbete, till enkla nöjen som Finn Fem Fel underlättas avsevärt av vårt mycket kraftfulla program.

Efter ett antal dagars teoretisk förberedelse bestämdes det i vår grupp att vi skulle använda oss av två kända algoritmer för den sortens bildbehandling som vi hade tänkt oss, nämligen att hitta den absoluta skillnaden mellan bilderna och att räkna korrelationen mellan dessa och jämföra det med ett bestämt tröskelvärde.
Ännu en fråga, som gruppen försökt besvara är filtrens möjliga påverkan på vara resultat. Kan vi förbättra resultaten genom att lågpass- eller högpassfiltrera bilderna innan beräkningarna skedde?

Implementering

Absoluta skillnaden

Vårt första steg vår att försöka räkna ut den absoluta skillnaden mellan två bilder.

$c = | a - b |$

Med ovanstående formel lyckades vi skapa en bild c, där man kunde se var skillnaderna låg, mellan bilderna a och b. Detta fungerade väldigt bra då man exempelvis ville upptäcka borttagandet av något i den ursprungliga bilden a.
Det är viktigt att ta upp att vår applikation inte hanterar bilder, som är olika stora. Det är alltså väsentligt att förstå att vårt programs styrka ligger i att upptäcka hur en bild a, har modifierats av olika faktorer och blivit en bild b. Funktionen fungerar dock inte särskilt väl när modifikationerna handlar om färg- eller kontrastbyten.

Korrelation

Ett sätt att få ett kvantitativt mått på hur ”lika” signalerna x(t) och y(t) är, skulle kunna vara att bilda tidsmedelvärde av skillnaden x(t) – y(t), som skulle kunna förväntas bli 0 om signalerna är identiska. Denna metod är dock mindre lyckad då stora positiva och negativa skillnader mellan signalerna kan ta ut varandra och resultera i att medelvärdet blir litet även om de båda signalerna är mycket olika. För att komma förbi denna begränsning räknar vi istället tidsmedelvärdet för skillnaden i kvadrat, d.v.s. effekten, som också måste vara 0 för identiska signaler.

$$P = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} (x(t) - y(t))^2 \, dt = \lim_{T\to\infty} \frac{1}{T} \left[\int_{-T/2}^{T/2} x^2(t) dt + \int_{-T/2}^{T/2} x^2(t) dt - 2 \int_{-T/2}^{T/2} x(t) y(t) dt \right]$$

Ingen av dessa storheter har någonting att göra med hur lika signalerna är, varför denna information måste återfinnas i följande integral, betecknad ρ_{xy}

$$\rho_{xy} = \lim_{T\to\infty} \frac{1}{T} \int_{-T/2}^{T/2} x(t) y(t) dt$$

Den ovanstående integralen kallas för korrelationen mellan x(t) och y(t) och dess värde är större, ju större likheterna är mellan x(t) och y(t). För tidsdiskreta sekvenser ges korrelationen av

$$\rho_{xy} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) y(n)$$

Det absoluta värdet beror dessutom på signalernas amplitudsnivåer, vilket inte säger någonting om likheten mellan signalerna varför vi subtraherar signalernas eventuella likspänningsnivåer och normaliserar med avseende på signalernas effektivvärde.

Det normaliserade värdet kallas korrelationskoefficienten och ges i det diskreta fallet av:

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \overline{A})^2)(\sum_m \sum_n (B_{mn} - \overline{B})^2)}}, \quad \overline{A} = mean(A), \overline{B} = mean(B)$$

Korrelationskoefficienten r kan endast ligga mellan –1 och 1. Om r = 1, brukar man säga att A och B har full korrelation. Vid den andra extrempukten i r = -1 säger man att matriserna har full omvänd korrelation. Om r = 0 så är inte matriserna korrelaterade. Magnituden av r, abs(r) representerar 'mängden' korrelation (positiv eller negativ) mellan A och B. Korrelationskoefficienten är ett mått på relationen två bilder emellan. Man får exempelvis att korrelationskoefficienten blir lika med 1 om två bilder är identiska. Om man istället inverterar färgerna i en av bilderna, kommer r att bli lika med –1.

Filter

Det filtret vi använde oss av för att kontrollera filterpåverkan på våra signaler var ett Butterworthfilter.

Butterworthfiltret är konstruerat för att ge ett så jämnt passband som möjligt. Ett lågpass Butterworthfilter karaktäriseras av att det saknar nollställen och att filtrets poler ligger jämnt fördelade på en halvcirkel i vänstra halvplanet. Filtrets gränsfrekvens ω_c svarar mot halvcirkelns radie. Man kan visa att frekvens gången för ett n:te ordningens lågpass Butterworthfilter med gränsfrekvens ω_c ges av:

$$|H(\omega)| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}}$$

Med Matlab-anropen `butter(n,k)` samt `butter(n,k,'high')`,där n = ordningen och k = Wn som är cutoff frekvensen delad med Nyquistfrekvensen (halva samplingsfrekvensen), kunde vi lågpass- och högpassfiltrera bilderna.

De enda slutsatserna vi kan dra av våra försök med olika n och k (se ovan) är att det INTE går att förbättra resultaten bara för att man högpass- eller lågpassfiltrerar bilderna.

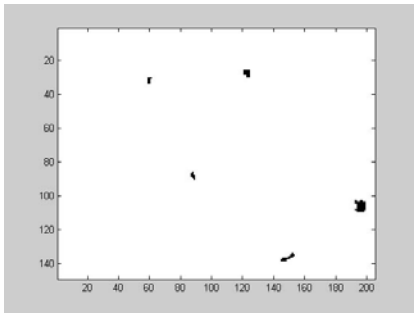
Resultat

Meningen med vår projekt var att ”Finna fem fel”. Nedan ser man att detta lyckades.

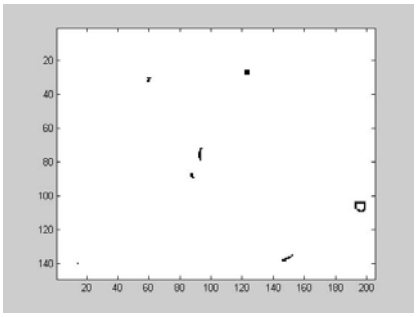
Bild 1

Bild2

Absoluta skillnaden

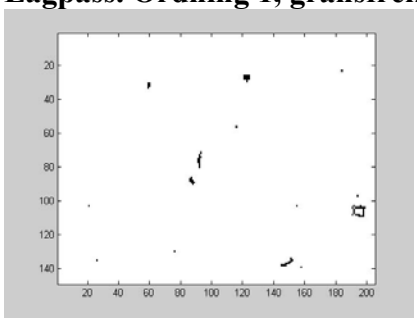


Korrelationen

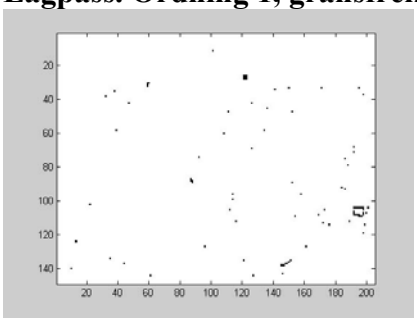


Vår slutsats när det gäller filter var att det inte hjälpte alls att låg-/högpassfiltrera bilderna. Nedan ser man beviset:

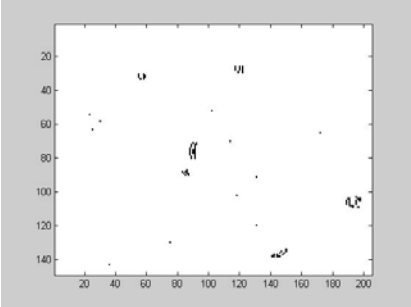
Lågpass. Ordning 1, gränsfrekv 0,1



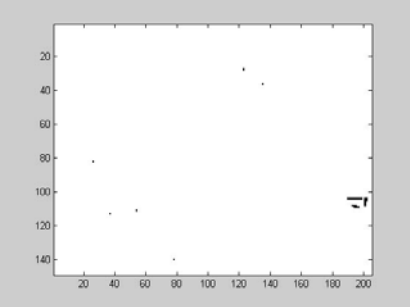
Lågpass. Ordning 1, gränsfrekv 0,5



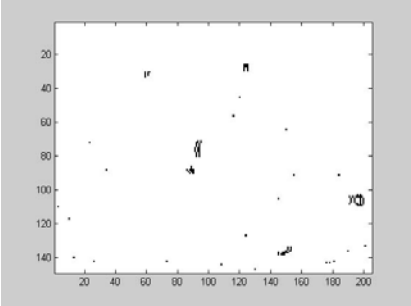
Lågpass. Ordning 3, gränsfrekv 0,1



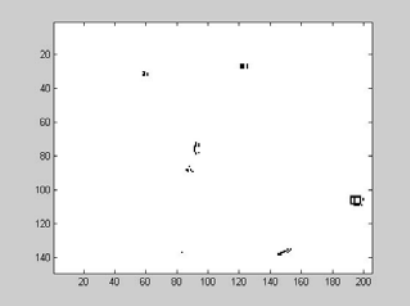
Lågpass. Ordning 3, gränsfrekv 0,5



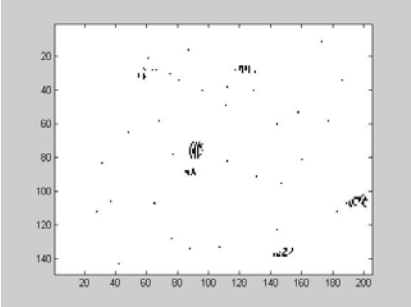
Högpass. Ordning 1, gränsfrekv 0,1



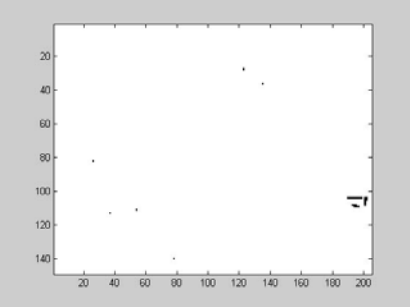
Högpass. Ordning 1, gränsfrekv 0,5



Högpass. Ordning 3, gränsfrekv 0,1



Högpass. Ordning 3, gränsfrekv 0,5

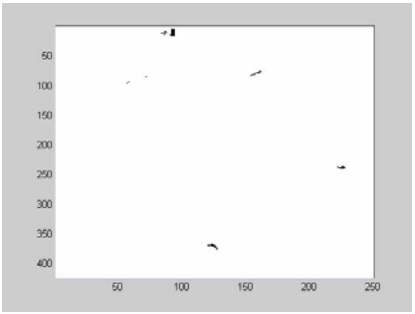


Vi testade ännu 2 bilder

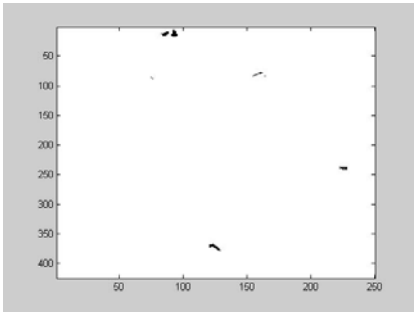
Bild 1

Bild 2

Absoluta skillnaden



Korrelationen



Slutsatser och funderingar

Det är förvånasvärt enkelt att implementera ett program som hittar skillnader i bilder, trots det känner vi i gruppen att det är iochmed detta projekt som vi för första gången skapar något som kan ha RIKTIGA tillämpningar i den RIKTIGA världen. Det faktumet att vi skapat något riktigt var en inspirations- och motivationskälla för vår projektgrupp. Att filtrering inte gav bra resultat var oerhört oväntat, men lärorikt.
Med tanke på de begränsningar vi har kan man utveckla vårt projekt väldigt mycket. Vi skulle kunna tänka oss att förbättra projektet genom att jämföra ”delar av bilderna”...om man exempelvis vill hitta ett ansikte i ett klassfoto. Sedan skulle man kunna implementera en funktion trim, som ser till att göra bilderna lika stora, på det sättet skulle vi kunna jämföra förstoringar eller förminskningar med den ursprungliga bilden utan att skillnader upptäcks.
Ännu en utveckling är att markera saker som lagts till i bilderna med en viss färg och saker som tagits bort med en annan, för att på så sätt få ett mer överskådligt intryck av förändringarna i bilderna. Det lämnar vi dock till de kommande generationerna. God jul!

Bibliografi

1) **Signaler och System. Anders Svärdström**

2) **ECE 197H – Multimedia Systems**
http://www-unix.oit.umass.edu/~keropie3/ece197h/project1homepage.htm

Appendix

Kod

```
function mycorr

/* I denna funktion räknar vi ut
korrelationen mellan 2 bilder och
jämför det med ett tröskelvärde, som
användaren bestämmer */

clear;

close all;

a = input('Orginalbild ','s');
b = input('Bild att testa ','s');
th = input('Threshold? ');

a=imread(a);
b=imread(b);

%FILTER
 %c = input('ord:');
 %d = input('frek:');
 %[X, map] = butter(c, d,'high');
 %a=imfilter(a,[X,map]);
 %b=imfilter(b,[X,map]);

if (length(size(a))) ~= 2
   a = rgb2gray(a);
end
if (length(size(b))) ~= 2
   b = rgb2gray(b);
end

a=double(a);
b=double(b);

a = a - mean2(a);
b = b - mean2(b);

c = (a.*b) ./ sqrt((a.^2) .* (b.^2));

sizeC = size(c);

for x = 1:sizeC(1)
   for y = 1:sizeC(2)
     if double(c(x,y)) >= th
         c(x,y) = 255;
     else
         c(x,y) = 0;
     end
end
end

figure(2);
colormap(gray(2));
image(c);
```

```
function absdiff

/* I denna function räknar vi ut den
absoluta skillnaden mellan 2 bilder */

clear;
close all;

a = input('Orginalbild: ','s');
b = input('Bild att testa: ','s');

a = imread(a);
b = imread(b);

if (length(size(a))) ~= 2
   a = rgb2gray(a);
end
if (length(size(b))) ~= 2
  b = rgb2gray(b);
end

a = double(a);
b = double(b);

c=abs(a-b);


figure(1);
colormap(gray(2));
imagesc(c);
```