

Övervakningssystem

-skillnader i bilder

Lärare:

Mathias Johansson

Gruppen:

Jakob Brundin

Gustav Björcke

Henrik Nilsson

Sammanfattning

Syftet med detta projekt är att fördjupa oss inom ett område inom kursen Signaler och System och i själva arbetet lyckas jämföra två bilder och finna skillnader i dessa. Detta implementeras sedan i ett enkelt övervakningssystem.

Programmet, som skrivits i Matlab, får in två bilder tagna inom ett kort tidsintervall. På dessa bilder räknas först korrelationen ut och denna jämförs mot ett gränsvärde. Detta konstaterar om det skett en nog stor förändring för att systemet ska reagera.

Vi har efter arbetet kunnat konstatera att den svåraste delen vad gäller jämförelse av två bilder är att hitta en bra metod för att bestämma vad som är en signifikant förändring och inte.

Innehållsförteckning

Inledning	4
Syfte	4
Problemformulering	4
Kravspecifikation.....	5
Lösning och genomförande.....	5
Testkörning	8
Resultat och slutsatser	9
Utökningsmöjligheter.....	9
Referenser	10
Appendix.....	11

Inledning

Bildbehandling är något som vi är intresserade av och vi tror att tillämpningar inom övervaknings- och bildhanteringsbranschen är något som bara kommer att öka.

Digitalkameror har snart konkurrerat ut vanliga kameror, datorer är snart var mans egendom. Detta leder till att vem som helst snart har den hårdvara som krävs för ett enkelt övervakningssystem och då kommer tillhörande mjukvara att efterfrågas på en helt annan nivå än idag. Här ser vi något som vi skulle kunna inrikta oss mot.

Syfte

Syftet med detta projekt är att fördjupa oss inom ett område inom kursen Signaler och System och i och med detta utöka våra kunskaper i ämnet. Målet med själva arbetet är att lyckas jämföra två bilder och finna skillnader i dessa.

Problemformulering

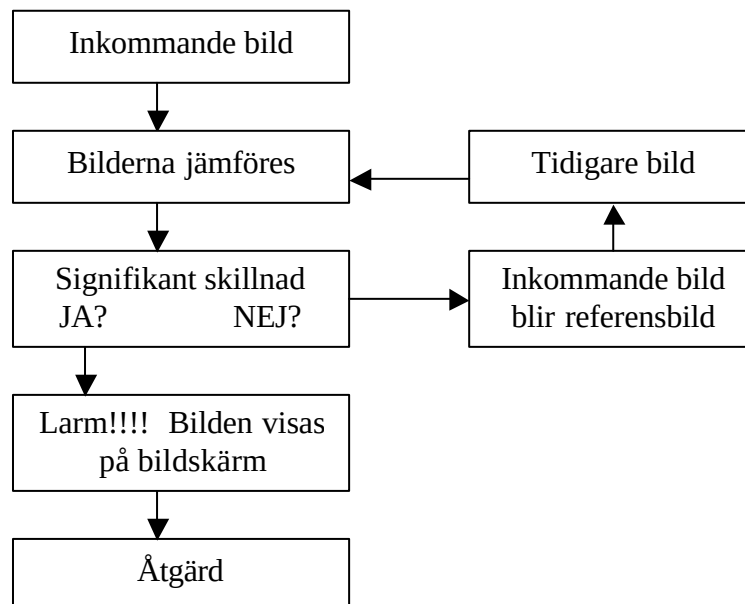
Hur kan metoder inom bildbehandlingsområdet implementeras till att skapa ett övervakningssystem?

Kravspecifikation

Övervakningssystemet består av en fast monterad digitalkamera ansluten till en dator där vår mjukvara finns installerad.

Med korta jämna tidsintervall tas bilder mot området som skall övervakas.

Dessa bilder skickas sedan vidare till mjukvaran som avgör om en förändring har skett och anses vara nog kritisk. Om så är fallet aktiveras ett larm där lämplig person blir uppmärksammas på att något har skett. Därefter kan personen få bildinformation presenterad på ett antal sätt och avgöra vad som tillkommit alternativt försvunnit på bilden.



Lösning och genomförande

Övervakningssystemets mjukvara har konstruerats i Matlab där våra idéer har konkretiserats till ett program. Programmet tar in två bilder vilka transformeras till matriser. Dessa matriser räknas sedan om till att representera färgernas intensitet med ett värde mellan 0-255, där låga värden motsvarar mörkt och höga värden motsvarar ljust. Därefter beräknas korrelationen mellan Bild1 och Bild2 enligt följande formel¹, vilken implementeras i metod2.m (se appendix).

$$C_{xy} = \frac{1}{N} \sum_{n=0}^{N-1} \frac{[x(n) - \overline{x(n)}][y(n) - \overline{y(n)}]}{\sigma_x \sigma_y}$$

$x(n)$ är intensiteten i varje pixel i bild1.

$y(n)$ är intensiteten i varje pixel i bild2.

σ är respektive bilds standardavvikelse.

Korrelationen används för att avgöra om någon relevant skillnad har skett mellan bilderna genom att värdet jämförs mot ett gränsvärde som vi valt.

Två identiska bilder resulterar i att korrelationen mellan bilderna är 1, dvs. bilderna är perfekt korrelerade. Två helt motsatta bilder, dvs. en bild och dess negativ ger korrelationen -1 . Två bilder utan någon som helst likhet ger korrelationen 0.

Om en bild anses ha förändrats så presenteras detta genom två bilder.

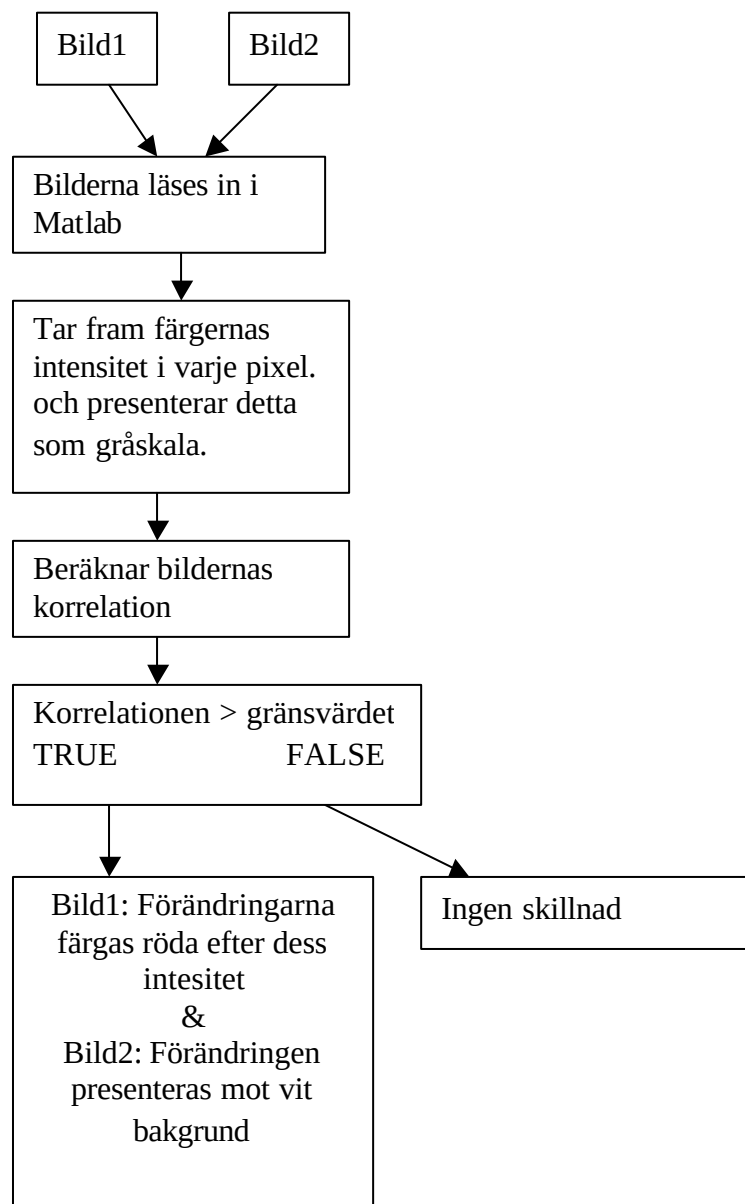
Den första visar bilden som förändrats, med förändringarna markerade med rött. En stor förändring i intensitet symboliseras av en stark röd färg och vice versa.

Först beräknas absolutbeloppet mellan bilderna och sparas i en matris. Sedan rekurerar vi genom matrisen vars värden jämförs mot de uppsatta gränserna för varje intensitetsnivå av rött. Resultatet sparas ned i en matris som slutligen representeras som en bild.

¹ Svärdström, Signaler och system, s.250

Den andra bilden lyfter endast fram förändringen i dess ursprungs intensitet. Övrig omgivning färgas vitt. Denna bilds pixel-intensitets absolutbelopp beräknas först på samma sätt som ovan. Efter detta så jämförs alla förändringar mot ett gränsvärde, alla pixlar som ej har förändrats nog mycket färgas vita.

Här presenteras hela förloppet schematiskt.

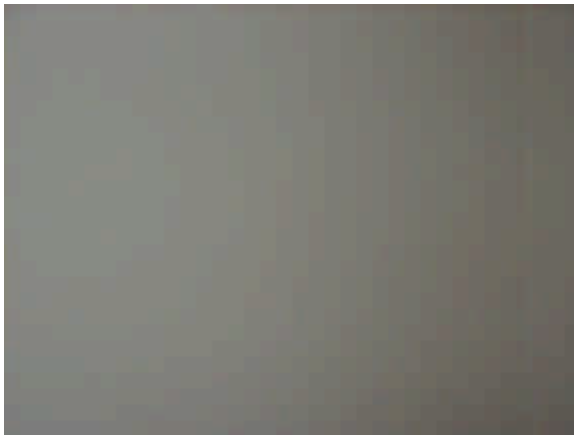


Testkörning

Här är resultatet av en testkörning av två bilder.

Ursprungsbild1

Här är bilden som visar ursprungsläget.



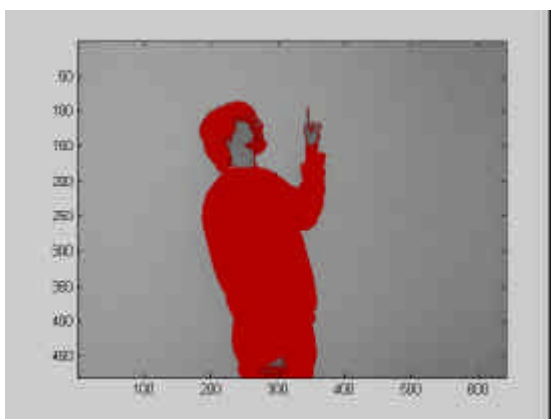
Ursprungsbild2

Här har någon mystisk person smygit sig in. Kanske har han något fuffens på gång?



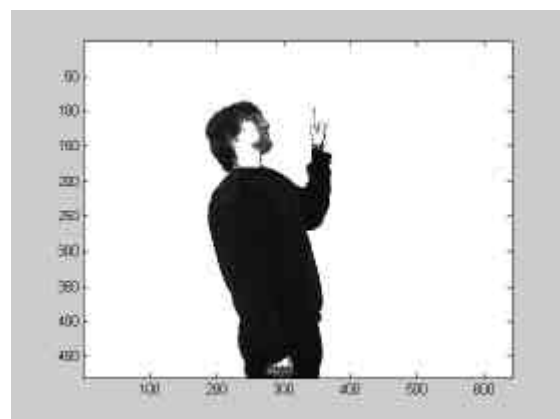
Resultatbild1

Här har förändringen markerats med rött i olika nyanser.



Resultatbild2

Här har förändringen framhävts i dess original intensitet medans det som ej har ändrats har fått färgen vitt.



Resultat och slutsatser

Med hjälp av Matlab har vi åstadkommit mjukvaran för ett enkelt bevakningssystem, vilket var vårt mål. Vi har under arbetets gång stött på flera problem som vi inte förutsåg vid starten av projektet. Det svåraste momentet var dock inte helt oväntat att bestämma om och hur en förändring har skett.

Vi använde oss av bildernas korrelation vilket är ett bra mått om en stor förändring har skett i begränsad del av bilden, tex om en människa kommit in i bilden.

Däremot fungerar måttet sämre för mindre förändringar i avgränsat område tex om en mus smyger in i bilden. Detta skulle kunna ge samma korrelationsvärde som en liten intensitetsförändring över hela bilden skulle generera, vilket i vår implementering inte skulle räknas som en förändring.

Ett annat problem vi har stött på under projektets gång var att om man t.ex. bara subtraherar två bilder utan att ta absolutbeloppet av varje enskild punkts förändring så kan stora förändringar åt olika håll ta ut varandra.

I det stora hela tycker vi att vi har lyckats få lite mer grepp om hur bildbehandling är en del av signalbehandlingsområdet och vi känner oss nöjda med att vi har lyckats uppnå huvudsyftet med projektet.

Utökningsmöjligheter

En utökning av projektet vore att vi fördjupa oss i andra metoder för att på ett mer precist sett avgöra huruvida en förändring av bilden har skett. En effektiv utökning hade varit att med hjälp av indata kunna identifiera vissa typer av förbestämda objekt. Med hjälp av detta hade man kommit från vissa begränsningar vad gällande verifieringen om att något signifikant har hänt i bilden.

Referenser

Svårdström.S, Signaler och System, 1999

Rice Universitets: Digital signal processing hemsida

<http://www.dsp.rice.edu/courses/elec301/>

Appendix

Matlab kod:

```
function overvakning(bild1, bild2)

% Projekt Signaler & System ht 2002, Gustav Björcke, Henrik Nilsson och Jakob
Brundin
%
% Huvudprogram som anropar delfunktionern.

%Läs in bilderna och spara dem i matriser
matris1 = imread(bild1);
matris2 = imread(bild2);

%omvandla bilderna till gråskala
matris1g = rgb2gray(matris1);
matris2g = rgb2gray(matris2);

%Rita upp resultat i figur.

if( metod2(matris1g,matris2g) < 0.99)
    figure(1)
    metod1(matris1g, matris2g);
    figure(2)
    metod3(matris1g, matris2g);
    'PIP PIP PIP'
else
    'Ingen skillnad...'
end
```

```

function metod1(bild1, bild2)

%Gör om varje tal i matrisen till en double
m1 = im2double(bild1);
m2 = im2double(bild2);

%Skapar en ny matris med den absoluta differensen mellan bild 1 och bild 2.
diff = (abs(m2 - m1));

%Sparar storleken på bilderna. (längd och bredd)
size1 = size(m1);
size2 = size(m2);

%Skapar en ny matris.
m3 = zeros(size1,size2,3);

%Sparar absolutbeloppen i diff.
m3 = diff;

%Loopa igeon bilden och kontrollera pixel för pixel om absolutbeloppen
%överstiger våra uppsatta gränser. Beroende på skillnaderna skrivs
%olika starka färger ut i bilden.

for i = 1:(size1(1))
    for j = 1:(size1(2))

        if ((diff(i,j) >= 0) & (diff(i,j) <= 0.13))
            m3(i,j,1)=m1(i,j);
            m3(i,j,2)=m1(i,j);
            m3(i,j,3)=m1(i,j);
        end;

        if ((diff(i,j) > 0.13) & (diff(i,j) <= 0.16))
            m3(i,j,1)=0.5;
        end;

        if ((diff(i,j) > 0.16) & (diff(i,j) <= 0.30))
            m3(i,j,1)=0.6;
        end;

        if ((diff(i,j) > 0.30) & (diff(i,j) <= 0.50))
            m3(i,j,1)=0.7;
        end;

        if ((diff(i,j) > 0.50) & (diff(i,j) <= 0.60))
            m3(i,j,1)=0.8;
        end;
    end;
end;

```

```
    if ((diff(i,j) > 0.60) & (diff(i,j) <= 0.70))
        m3(i,j,1)=0.9;
    end;

    if ((diff(i,j) > 0.70) & (diff(i,j) <= 1.0))
        m3(i,j,1)=1;
    end;

end
end

imagesc(m3);
```

```

function korrelation = metod2(bild1, bild2)

%Gör om varje tal i matrisen till en double
m1 =im2double(bild1);
m2 = im2double(bild2);

%Sparar storleken på bilderna. (längd och bredd)
size1 = size(m1);
size2 = size(m2);

%Initiera variabler som används i loopen.
kvadratsummam1 = 0;
kvadratsummam2 = 0;
raknare = 0;
summam1= 0;
summam2= 0;
summa = 0;

%Beräkna medelintensiteten för de två bilderna.
for i = 1:(size1(1))
    for j = 1:(size1(2))
        summam1 = m1(i,j) + summam1;
        summam2 = m2(i,j) + summam2;

        raknare = raknare + 1;
    end
end

meanm1 = summam1/raknare;
meanm2 = summam2/raknare;

%Beräknar korrelationen mellan de två bilderna
raknare = 0;
for i = 1:(size1(1))
    for j = 1:(size1(2))
        summa = ((m1(i,j)- meanm1)*(m2(i,j)- meanm2))+summa;

        kvadratsummam1 = (m1(i,j)- meanm1)^2 + kvadratsummam1;
        kvadratsummam2 = (m2(i,j)- meanm2)^2 + kvadratsummam2;
        raknare = raknare + 1;
    end
end

stdm1 = (kvadratsummam1/raknare)^0.5;
stdm2 = (kvadratsummam2/raknare)^0.5;

korrelation = (1/raknare) * (summa/(stdm1*stdm2))

```

```

function metod3(bild1, bild2)

%Gör om varje tal i matrisen till en double
m1 =im2double(bild1);
m2 = im2double(bild2);

%Skapar en ny matris med den absoluta differensen mellan bild 1 och bild 2.
diff = (abs(m2 - m1));

%Sparar storleken på bilderna. (längd och bredd)
size1 = size(m1);
size2 = size(m2);

%Skapar en ny matris.
m3 = zeros(size1,size2,3);

%Sparar absolutbeloppen i diff.
m3 = diff;

%Loopa igeon bilden och kontrollera pixel för pixel och beroende på om
absolutbeloppen
%överstiger respektive understiger uppsatta gränser så sätts pixlarna
%till ursprunglig intensitet respektive vitt.

for i = 1:(size1(1))
    for j = 1:(size1(2))

        if ((diff(i,j) >= 0) & (diff(i,j) <= 0.10))
            m3(i,j,1)=1;
            m3(i,j,2)=1;
            m3(i,j,3)=1;
        end;

        if ((diff(i,j) > 0.10) & (diff(i,j) <= 1))
            m3(i,j,1)=m1(i,j);
            m3(i,j,2)=m1(i,j);
            m3(i,j,3)=m1(i,j);
        end;

    end
end

imagesc(m3);

```