

Sammanfattning TNM030 - Bildbehandling och bildanalys

*Nathalie Ek (natek725), MT -07
2011, LIU Campus Norrköping*

Bildbehandling och bildanalys

- **Bildbehandling**

Kan kort sammanfattas som signalbehandling i två dimensioner.

- **Bildanalys**

Kan kort sammanfattas som reduktion av en bild till en beskrivning av bildennehållet.

Ögat

I det mänskliga ögat finns två typer av ljussensorer – *tappar* och *stavar*.

- **Tapparna**

Också kallade Photopic, sköter färgseendet och gör även så att människan kan urskilja detaljer. Det finns ca 6-7 miljoner tappar (per mm²) och de är till största del lokaliserade i fovean (gula fläcken).

- **Stavarna**

Även kallade Scotopic, sköter mörkerseendet och verkar endast vid låga ljusintensiteter. Det finns ca 75-150 miljoner stavar (per mm²) och de är utspridda längs hela retinan (näthinnan).

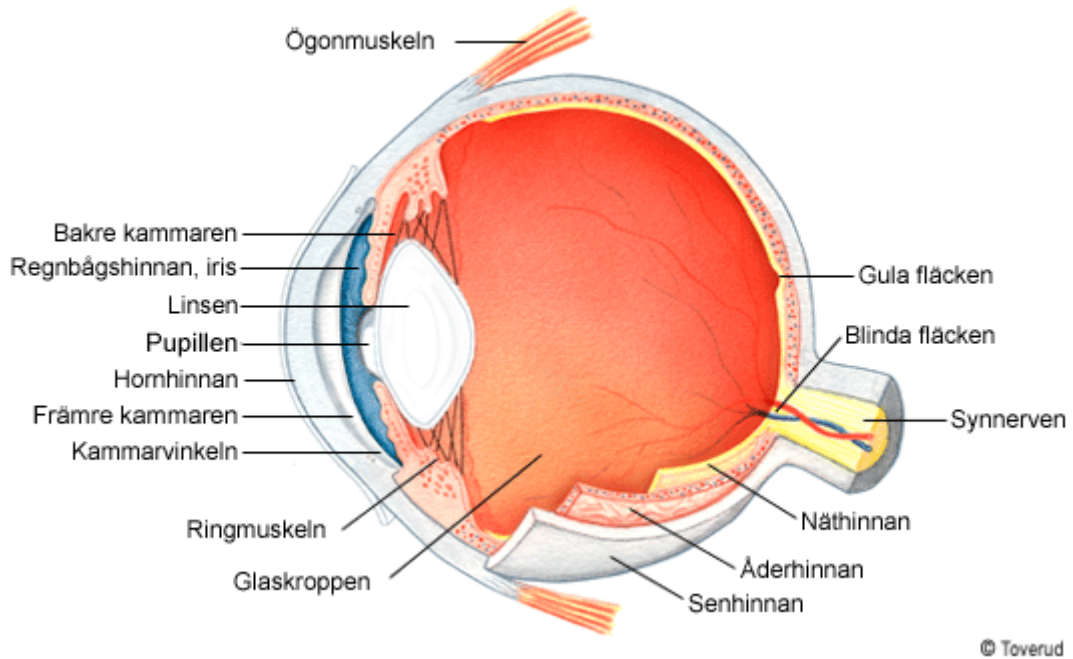
Synsinnet kan ge upphov till synvillor och två av dessa är "*Mach-band*" och "*Simultaneous contrast*".

- **Mach-band**

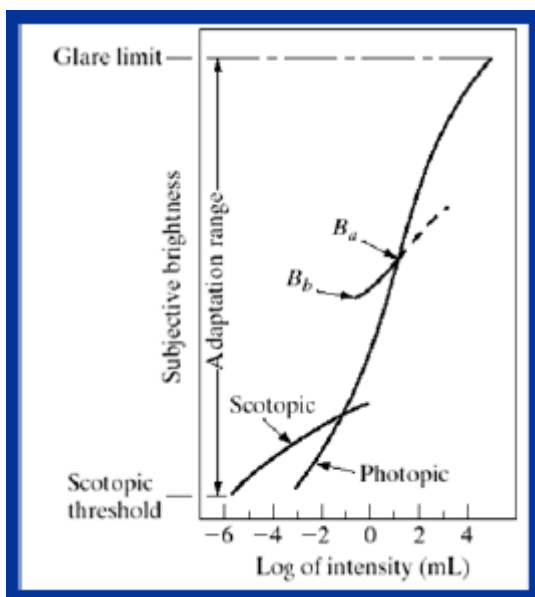
Mach-band gör så att vi upplever kontraster mellan två intensiteter större än vad den egentligen är.

- **Simultaneous contrast**

Är det fenomen som uppstår då en ruta med samma intensitet läggs på olika intensiteter. Alltså samma intensitet kan se olika ut beroende på bakgrunden.



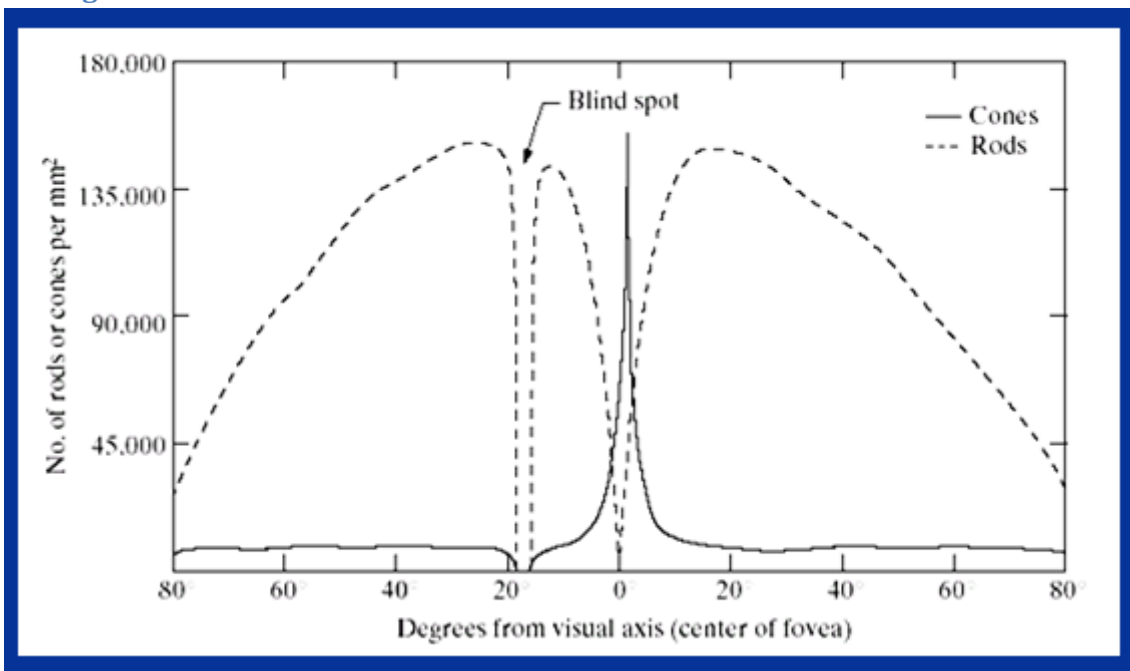
Vi har sämre färgseende i svagt ljus eftersom endast stavarna aktiveras, vilka är okänsliga för färg, de sköter mörkerseendet. I näthinnans ytterkanter dominerar stavarna vilket gör att färgseendet blir dåligt i synfältets periferi.



Följande diagram illustrerar det mänskliga ögats känslighet:

Diagrammet är en plot på ljusintensitet mot subjektiv ljusintensitet, hur intensiteten uppfattas av våra ögon. Från "*Scotopic threshold*" till "*Glare limit*" är området av olika nivåer av ljusintensitet som människan kan uppfatta. Den andra grenen på trädet visar övergången från *Scotopic* till *Photopic*.

- Ögats sensorer



Tonöverföringskurva

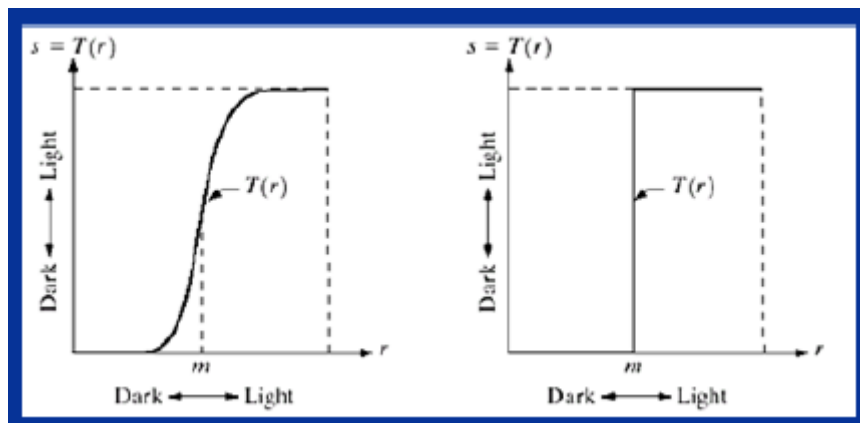
En bild med dålig kontrast (lågt dynamiskt omfång) kan korrigeras med en linjär tonöverföringskurva så att bilden får maximalt dynamiskt omfång. En sådan kurva ser ut enligt

$$Q = \frac{P - P_{min}}{P_{max} - P_{min}}$$

P = pixel i inbilden

Q = pixel i utbilden

Vid tröskling av en gråskalebild får vi tonöverföringskurvan nedan, där T är tröskelvärdet. (En tonöverföringskurva beskriver en transformation av pixelvärdena.)



Histogramutjämning

Vid histogramutjämning använder man sig av ett *kumulativt histogram*. Originalbilden normaliseras och därefter beräknas histogrammet för originalbilden, $H(P)$. Sedan bildar man ett kumulativt histogram genom att integrera originalbildens histogram. Det kumulativa histogrammet normaliseras, $H_k(P)$ sedan med antalet pixlar i bilden, N , och resultatet används som en ny tonvärdeskurva för bilden.

Målet med histogramutjämning är att försöka ge varje tonvärde lika många pixlar, d.v.s. modifiera bilden så att dess histogram blir jämt.

Vid histogramutjämning av en tonöverföringskurva ges av ekvationen nedan, där $H(P)$ är bildens histogram, $H_k(P)$ är det normaliserade kumulativa histogrammet och N är antalet pixlar.

$$H_k(P) = \sum \frac{H(\alpha)}{N}$$

En pedagogisk förklaring på vad som utförs, steg för steg, då en bild som kan ha gråskalevärden mellan 0 och 255 histogramutjämnas:

- Originalbilden normaliseras så att dess värden ligger mellan 0 och 1. Därefter plockar man fram originalbildens histogram och skapar sedan ett kumulativt histogram genom att integrera originalbildens histogram.
- Det kumulativa histogrammet normaliseras med antalet pixlar i bilden. Resultatet används sedan som en ny tonöverföringskurva för bilden. Målet är att vi vill ha ett jämnt antal pixlar i bilden fördelat på alla grånivåer = större dynamiskt omfång!

Avståndsmått

I digitala bilder kan man använda olika mått för att mäta avståndet mellan två pixlar. Det finns tre stycken vanliga metoder som används:

Euklidiskt avstånd, D_E

- Pythagoras sats \rightarrow kortaste vägen mellan två pixlar/punkter.
- $D_E = (\Delta x^2 + \Delta y^2)^{1/2}$

Cityblock distance, D_4

- Avståndet i antal pixlar i både x- och y-led.
- $D_4 = |\Delta x| + |\Delta y|$

Chessboard distance, D_8

- Det största avståndet för skillnaden i x-led och y-led mellan två pixlar (a, b) och (c, d) .
- $D_8 = \max(|\Delta x|, |\Delta y|)$

Fouriertransformen

- $F(u, v)$ är Fouriertransformen av en bild $f(x, y)$
- u och v är frekvensvariabler och mäts i **hertz**
- $F(u, v) = Re(u, v) + jIm(u, v)$
- Spektrum: $|F(u, v)| = \sqrt{e(u, v)^2 + Im(u, v)^2}$

Spektrumet är magnituden av frekvensvektorn och ändras vid multiplikation av en konstant men ändras inte vid translation:

$$|A * F(u, v)| = |A| * |F(u, v)|, |F_T(u, v)| = |F(u, v)|$$

Fasvinkeln är en enhet i förflyttning av sinusvågor med respekt åt dess ursprung. Den ändras inte vid multiplikation av en konstant men ändras vid translation.

Fasvinkel: $\Phi(u, v) = \tan^{-1} \left(\frac{Im(u, v)}{Re(u, v)} \right)$

Fasvinkel vid translation: $\tan^{-1} \left(\frac{A * Im(u, v)}{A * Re(u, v)} \right) = \tan^{-1} \left(\frac{Im(u, v)}{Re(u, v)} \right) = \Phi(u, v)$

Den **diskreta Fouriertransformen** är separerbar, vilket innebär att man kan räkna en **2D DFT** genom att göra en **1D DFT** längs bildens kolumner och sedan gör en **1D DFT** längs resultatets kolumner.

Faltningsteoremet:

$$F(u, v) \circ H(u, v) + N(u, v) = f(x, y)h(x, y) + n(x, y)$$

$$f(x, y) \circ h(x, y) + n(x, y) = F(u, v)H(u, v) + N(u, v)$$

Translation:

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0 + vy_0)}$$

$$f(u - u_0, v - v_0) \Leftrightarrow F(x, y)e^{-j2\pi(Xu_0 + Yv_0)}$$

Medelvärde:

Vid origo ligger bildens medelvärde, $F(0, 0)$. Det är summan av pixlarna delat på antalet pixlar.

"Hit or Miss"-transformation

Strukturelement kan nu, förutom objektvärden (1:or), även innehålla bakgrundsvärden (0:or) och s.k. "don't cares".

Ex
(tomma rutor = "don't care")

	1		
0	1	1	
0	0		

	1		
1	1	0	
	0	0	

	0	0	
1	1	0	
	1		

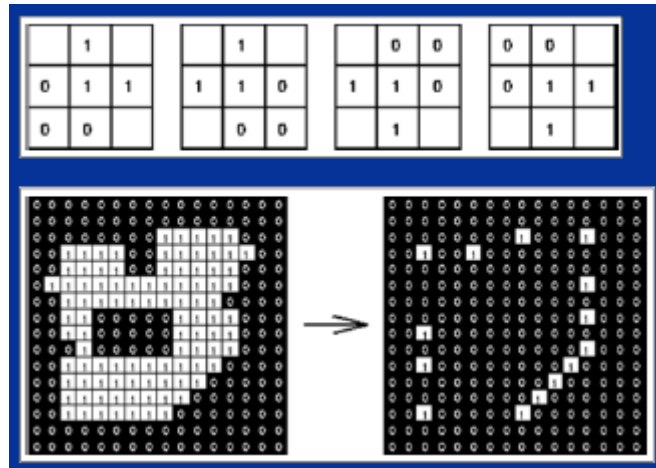
0	0		
0	1	1	
	1		

- Används för att detektera rektangulära hörn i en binär bild, innehållande rektangulära objekt.

- Anta att objekten (rektanglarna) är vita (1) och bakgrunden svart (0). Använd följande 3x3 strukturelement och dess 90 graders rotationer, d.v.s. totalt 4 strukturelement. Låt dessa i tur och ordning löpa genom bilden och sätt 1 i centripixelns position i utbilden vid träff.

Exempel – Detektering av hörn.

Låt dessa 4 strukturelement genomlöpa bilden var en gång.

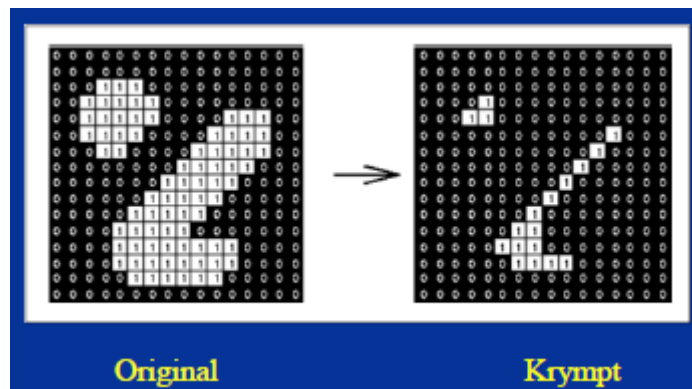


Morfologiska operationer

Morfologiska operationer används då man vill plocka bort skräp i en binär bild. Det finns fyra olika operationer – *krympning*, *dialation*, *öppning* och *stängning*.

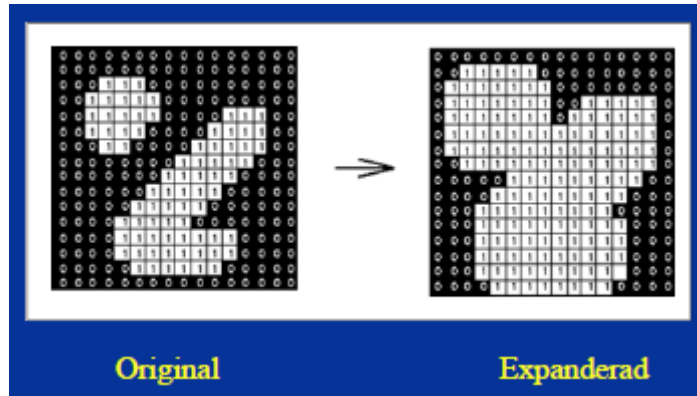
- Krympning (erosion)

Man låter ett strukturelement B gå över bilden (innehållande 1:or). Om hela B täcker ett objekt i bilden sätts centripixeln i bilden till 1, annars 0.



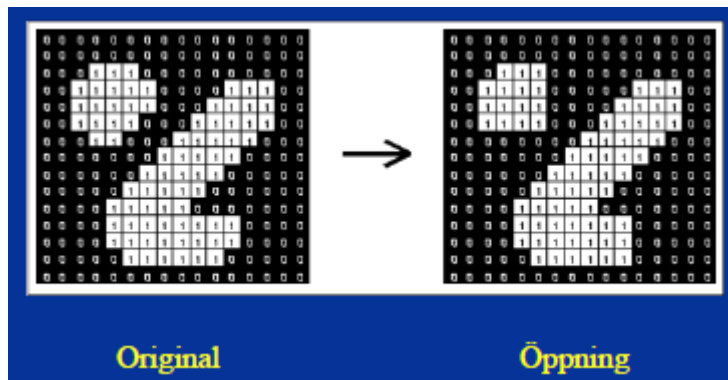
- Dialation (expansion)

Ett strukturelement B löper över bilden. Om någon del av B täcker en objektpixel i bilden sätts centripixeln i bilden till 1.



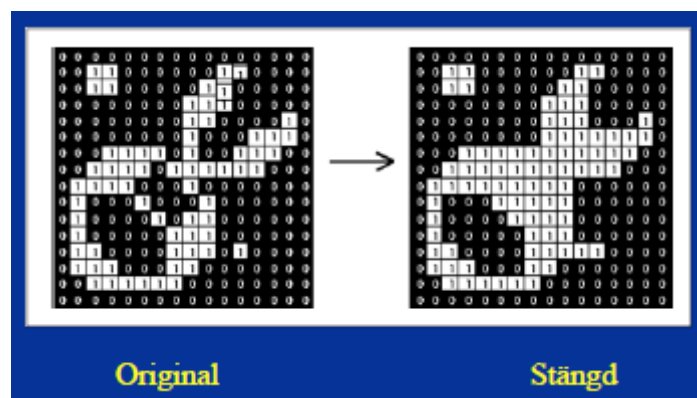
Öppning

Krympning följt av expansion. Öppning utjämnar konturer, bryter upp smala förbindelser mellan objekt och eliminerar små isolerade objekt. (Används då man vill bli av med "blobbar" i kanterna på en bild.)



Stängning

Expansion följt av krympning. Stängning utjämnar konturer, fyller i smala sprickor och eliminerar små hål.



Krympning och expansion kan beskrivas med *min- och maxfunktioner*. Strukturelementets storlek är n pixlar och värdet i centumpixelns position i utbilden är q . Värdena på de pixlar i inbilden som täcks av strukturelementet: p_1, p_2, \dots, p_n

Krympning

$$q = \min(p_1, p_2, \dots, p_n)$$

Expansion

$$q = \max(p_1, p_2, \dots, p_n)$$

Ex.1

En bild med cirkulära objekt (20px i diameter) samt skräp i form av tunna linjer på en eller ett par pixlar. För att få bort "skräp-linjerna" utnyttjas den morfologiska operationen "öppning", som består av en krympning följt av en expansion, med ett cirkulärt strukturelement vars diameter är större än linjernas tjocklek och mindre än eller lika med den minsta cirkelns diameter

Ex.2

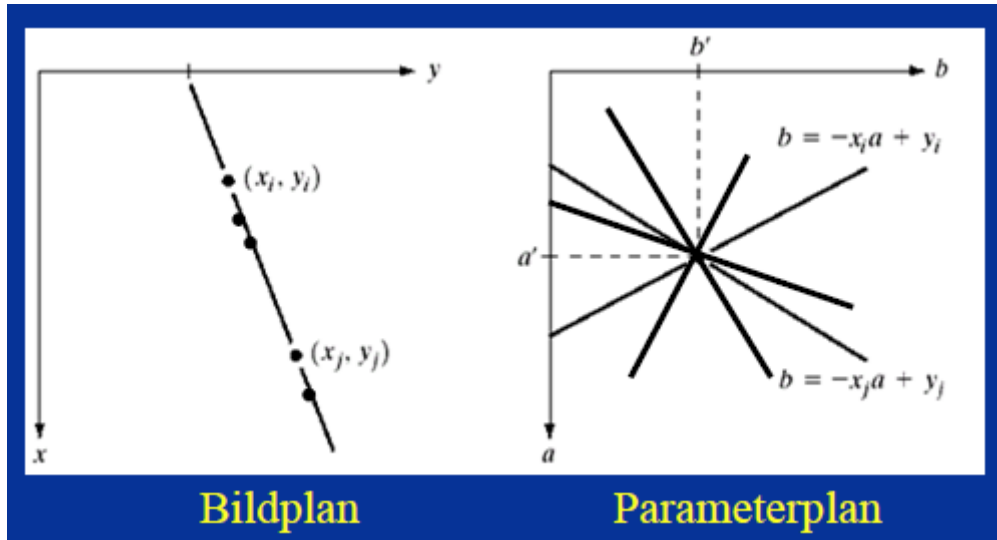
En binär bild innehåller konturen av ett objekt. Med hjälp av morfologiska operationer kan man fylla området innanför konturen. Man börjar med att skapa inversen A_c av den binära bilden som innehåller konturen. Man utgår därefter från en känd punkt innanför konturen. Denna punkt expanderar man och tar snittet mellan den och A_c . Vi utgår nu från en ny expanderande punkt och utför samma procedur till det att det inte är någon förändring av punkten

Hough-transformen

Denna transform används vid detektering av **linjer** i binära bilder. Transformen **verkar globalt** på bilden.

En punkt (x_k, y_k) i bilden kan ingå i ett ändligt antal linjer som alla uppfyller linjens ekvation $y_k = ax_k + b$, för olika stora a och b . Om man skriver om punkten till $b = x_k a + y_k$ får man en linje i ab -planet istället för xy -planet.

Alla punkter i bilden som nu ligger på samma linje (samma värde på a och b) i xy -planet kommer korsa varandra i samma punkt, (a', b') .



För varje linje som då korsar varandra i denna punkt, (a', b') kommer att sparas in i en ackumulatorcell.

För alla (a, b) som uppfyller $b = x_k a + y_k$ räknas motsvarande ackumulatorcell upp med 1. Ackumulatorceller med höga värden motsvarar verkliga linjer i bilden.

I exemplet ovan får skärningspunkten värde 2.

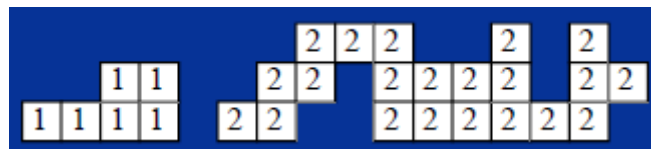
Dock kommer det bli problem då linjerna i bilden är vertikala och ej korsar axeln (vi får då ett oändligt värde). Man får istället skriva om linjens ekvation på normalform:

$$x \cos(\theta) + y \sin(\theta) = \rho, \quad (\text{där } \theta = \text{vinkel och } \rho = \text{avståndet})$$

En punkt/pixel i bilden kommer istället motsvaras av en sinuskurva i parameterplanet. Alltså, transformen mappar en punkt/pixel i bildplanet till en linje i parameterplanet.

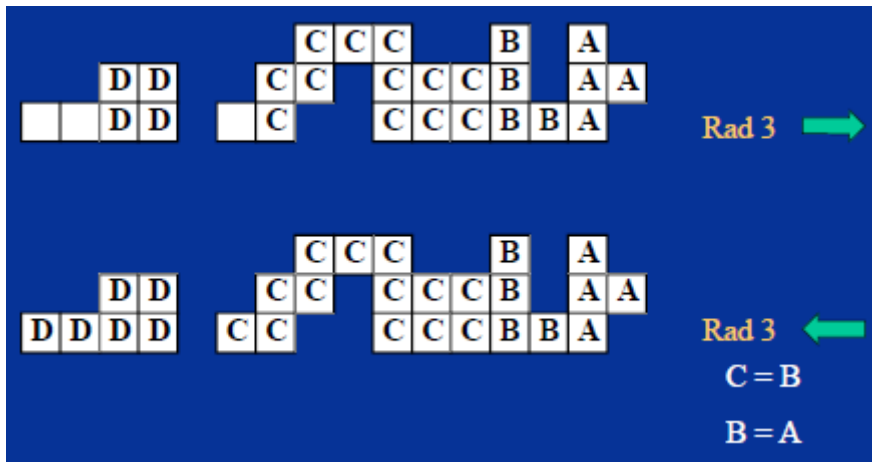
Etikettering (Labeling)

- Används för att räkna/särskilja antalet objekt i en binär bild.
- Man skannar rad för rad, en *högerscan* och en *vänsterscan*.
- Då bilden skannas sätts pixlarna i varje objekt till en (för objektet) unik kod, en s.k. etikett.



- **Högerscan**
Kopiera etikett ovanför, eller i andra hand från vänster om man är i objektpixel. Om bakgrundspixel eller etikett -> lämna pixelvärdet oförändrat.
- **Vänsterscan**
Om objektpixel, kopiera etikett från höger. Om ingen etikett finns, sätt ny etikett. Om bakgrundspixel eller etikett, lämna pixelvärdet oförändrat. Om objektpixel men ingen till

höger, sätt ny etikett. Om etikett existerar och pixeln till höger har annan etikett, notera detta.



Figur 1 - Bild med 3 rader som innehåller två 4-konnektiva objekt.

Signatur

Signatur används för att beskriva ett objekts kontur. En signatur är en 1d-representation av en 2D-kontur.

- Man ritar upp avståndet från objektets tyngdpunkt till konturen som en funktion av rotationsvinkeln.

Filter

- Filtrering i spatialdomänen

En mask (filterkärna) genomlöper originalbilden. Centrupixelns nya värde (filtersvaret) är en funktion av maskens värden och de pixlar som täcks av masken.

- Linjärt filter

Ett linjärt filter kan beskrivas som en faltung mellan 2D-funktioner - inbilden, f och filterkärnan, h .

Ett linjärt filter med filterkärna på $m*m$ appliceras på en bild med storleken $N*N$ pixlar. Filtreringen skulle kräva

- o $(m * m) * (N * N)$ st multiplikationer
- o $((m * m) - 1) * (N * N)$ st additioner

- Medianfilter

- o Är inte ett linjärt filter
- o Centrupixeln ersätts med **medianen** av de pixlar som masken täcker.
- o Utmärkt för eliminering av "spikigt" brus (salt and pepper noise).

- Ingen utslätning av skarpa kanter.
- **Deriverande filter**
 - Medelvärdesbildande (integrerande filter) ger utslätning (smoothing).
 - Deriverande filter ger skärpning (sharpning).
- **Laplace-filter**
 - Används ofta för skärpning av bilder.
 - Baserat på andraderivatan.
 - Är isotrop, d.v.s ger samma resultat oberoende av rotation (rotationsinvarians).
- **Högpasfilter**
 - Är ett deriverande filter
 - Finns idealt HP-filter
- **Lågpasfilter**
 - Är ett medelvärdes-filter
 - Finns idealt LP-filter
- **Ideella filter**
 - Get upphov till "ringing", vilket innebär att det uppstår vågor runt konturerna i bilden.
- **Gauss-filter**
 - Är ett linjärt filter vilket innebär att de beskrivs av en faltning mellan 2D-funktioner, f och h .
 - Filterfunktionen: $H(u, v) = e^{-\frac{D^2(u,v)}{2D_0^2}}$
 - Ger en jämnare överföring än ett idealt LP-filter och slipper oönskade effekter såsom "ringing".
 - Innehåller en variansparameter D_0 . Ju mindre D_0 desto suddigare blir bilden och filterkärnan blir då större i spatialdomänen.
 - Ett Gaussiskt HP-filter ger en jämnare övergång mellan höga och låga frekvenser. Ett sådant filter har en variansparameter, D_0 , som gör att filtret blir skarpare desto större värde på D_0 . Högt värde på D_0 ger ett litet filter i spatialdomänen.

■

Beskrivning: $1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$

- **LP-filter**

LP tar bort höga frekvenser och ger en oskarp bild. Ett lågt värde på D_0 ger en oskarp bild och ett större filter i spatialdomänen.

$$H(u, v) = e^{-\left(\frac{D(u,v) * D(u,v)}{2 * (D_0 * D_0)}\right)}$$

- **HP-filter**

HP tar bort lågfrekvenser och ger en skarpare bild. Ett lågt värde på D_0 ger en skarpare bild och ett större filter i spatialdomänen.

$$H(u, v) = 1 - e^{-\left(\frac{D(u,v) * D(u,v)}{2 * (D_0 * D_0)}\right)}$$

- Adaptiva filter

“Vanliga” filter verkar på samma sätt över hela bilden oberoende av bildens innehåll. Adaptiva filter å andra sidan, ändrar sin verkar efter bildinnehållet.

Ex.

Kantbevarande utslätning med medelvärdesbildande kärna. Adaptivitet: endast de pixlar som inte avviker alltför mycket från centumpixeln deltar i medelvärdesbildningen.

- Invers filtrering

Invers filtrering är en metod för bildrestaurering. Med metoden invers filtrering tar man fram en skattning av originalbildens transform genom att dela den degraderade bildens transform med degraderingsfunktionens transform:

$$F * (u, v) = \frac{G(u, v)}{H(u, v)} = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

Ett problem med invers filtrering är att $f(x, y)$ inte kan återskapas exakt för att $N(u, v)$ är okänt.

Om $H(u, v)$ är väldigt liten eller nära noll, kommer $\frac{N(u,v)}{H(u,v)}$ att dominera ekvationen.

- En bild $f(x,y)$ som degraderas av ett linjärt filter $h(x,y)$ och additivt brus $n(x,y)$ ger upphov till en degraderad bild $g(x,y)$ enligt:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y)$$

- Uttrycket för den degraderade bilden i frekvensdomänen:

$$G(u, v) = F(u, v)H(u, v) + N(u, v)$$

- Faltningsteoremet:

$$g(x, y) = f(x, y) * h(x, y) + n(x, y) \Leftrightarrow G(u, v) = F(u, v)H(u, v) + N(u, v)$$

$$(spatialdomänen \Leftrightarrow frekvensdomänen)$$

Gammakorrektion

Bilder som ska visas på monitor utsätts ofta för gammakorrektion. Anledningen är att olika skärmar förändrar signalen på olika sätt och därför krävs korrigerig av signalen innan en ”korrekt” utbild visas på skärmen. Denna korrektion är en global **tonmappning**.

Då en bild är mörkare än vad den bör vara på skärmen kan bilden gammakorrigeras för att upplevas/visas ljusare (mer korrekt) på skärmen.

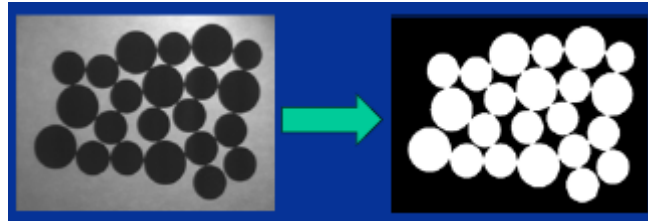
Med ett högt värde på gamma genereras en mörkare bild: $S = cr^\gamma$

Tröskelsättning

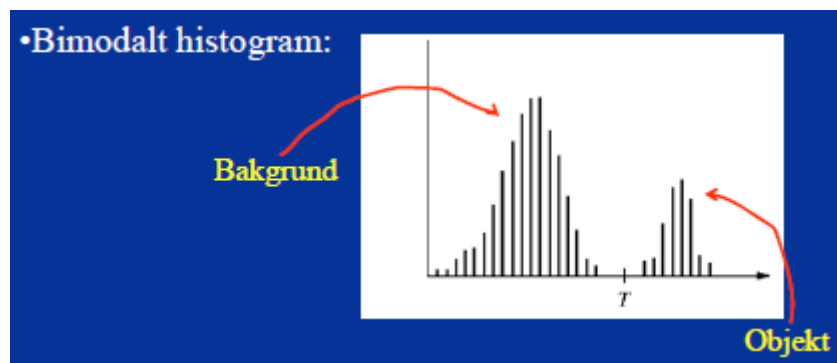
Den enklaste formen av segmentering ger en binär bild där objekten är separerade från bakgrunden.

- Tröskling

$$g(x,y) = \begin{cases} 1 & \text{om } f(x,y) > T \\ 0 & \text{om } f(x,y) \leq T \end{cases}$$

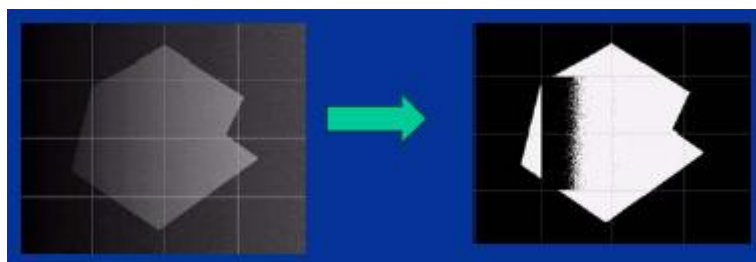


Sätt tröskeln, T , i dalen mellan topparna i det *bimodala histogrammet*. Ett *bimodalt Histogram* är ett histogram där man helt kan skilja bakgrund från objekt.



Tröskelsättning används då man automatiskt vill hitta ett lämpligt tröskelvärde av en bild där man har en bakgrund och ett objekt väl separerade m.a.p gråskala.

Ojämn belysning kan vara ett problem vid segmentering med tröskelsättning. Man kan förbättra situationen genom att bl.a. använda *adaptiv tröskelsättning*. Vid adaptiv tröskelsättning delas bilden in i mindre områden och man försöker hitta en tröskel/område. Om tröskelsättningen misslyckas i ett område, delas det in i ytterligare områden innan man på nytt försöker hitta en tröskel.



- **Vid tröskelsättning:**

1. Ansätt ett uppskattat tröskelvärde, T .
2. T delar då in histogrammet i två delar där vi beräknar medelvärdena μ_1 och μ_2 av intensiteterna för pixlarna med intensiteter över respektive under tröskeln T .
3. Beräkna ett nytt T :
$$T' = \frac{\mu_1 + \mu_2}{2}$$
4. Fortsätt tills ändringen i T mellan successiva iterationer blir liten. Alltså $T - T' \rightarrow 0!$

- **Optimal tröskling**

Om frekvensfunktionerna för bakgrunds- och objektpixlarna är kända eller väl uppskattade är det möjligt att hitta en optimal tröskel. Optimal i den meningen att sannolikheten för felklassificering av en pixel är minimal. Ofta antas att frekvensfunktionerna är Gaussiska (normalfördelning).

Sobelfilter/Gradientfilter

Gradientens riktning och magnitud kan beräknas med hjälp av ett gradientfilter. Ett exempel på ett sådant filter är Sobelfilter. Med ett Sobelfilter får man fram antingen gradienten i x-led eller i y-led, beroende på dess utseende.

- **Gradientens magnitud:**
$$M(x, y) = (g_x^2 + g_y^2)^{1/2}$$
- **Gradientens riktning:**
$$\tan^{-1}\left(\frac{g_x}{g_y}\right)$$

Gradienten beskriver riktningen hos förändringen i bilden, den tar fram kanter i bilden (från vitt \rightarrow svart och svart \rightarrow vitt).

Ett rotationsinvariant filter ger samma filtersvar oavsett hur bilden är roterad.

Vid förstaderiverande filter använder man magnituden av gradienten för att räkna skillnaderna i intensiteterna.

- **Definition av 1:a derivatan:**
$$\frac{df}{dx} = f(x + 1) - f(x)$$

Log-transform

För att expandera värdena hos de mörka pixlarna samt minska höga värden i en bild, använder man en log-transform på ett Fournier-spektrum.

Formeln på log-transformationen för att låga intensiteter in mappas som många olika intensiteter ut, samtidigt som höga intensiteter in inte har så många olika intensiteter ut att välja på.

- **Formel:** $s = c * \log(1 + r)$

s = intensitet (utvärde)

c = konstant

r = intensitet (utvärde)

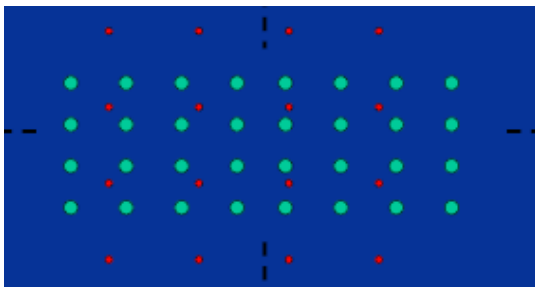
Interpolation

För att bestämma pixelvärdena i den förstorade/förminskade bilden krävs interpolation. Olika metoder är "*Nearest neighbour*", "*Bilinjär interpolation*" och "*Cubic spline*".

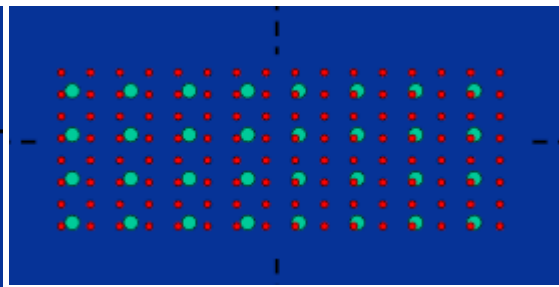
Bilinjär interpolation

Grundtanken är att ta de fyra närmaste grannarna till punkterna för att hitta den rätta intensiteten för just denna punkt/pixel.

- En bildbehandlingsoperation som **kräver interpolation är förminskning/förstoring**.
Gröna prickar = Originalets pixelpositioner.
Röda prickar = Omsamlade (förstorade/förminskade) bildens pixelpositioner.



Figur 2 – Förminskning



Figur 3 - Förstoring

- *Uttrycket ser ut på följande sätt:*

$$P_A = (1 - b)P_0 + b * P_1$$

$$P_B = (1 - b)P_3 + b * P_2$$

$$P = (1 - a)P_A + a * P_2$$

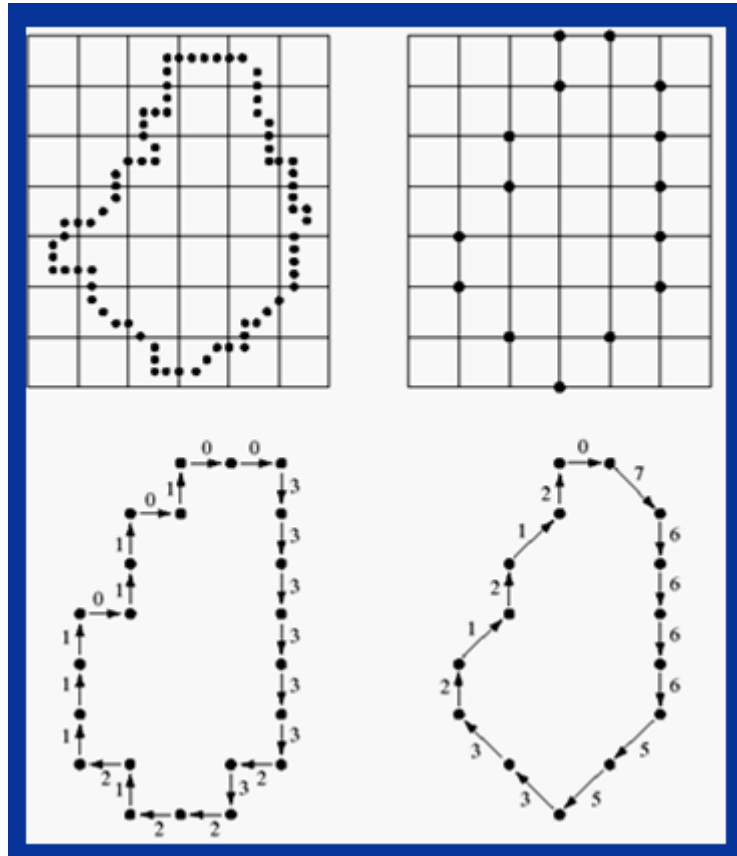
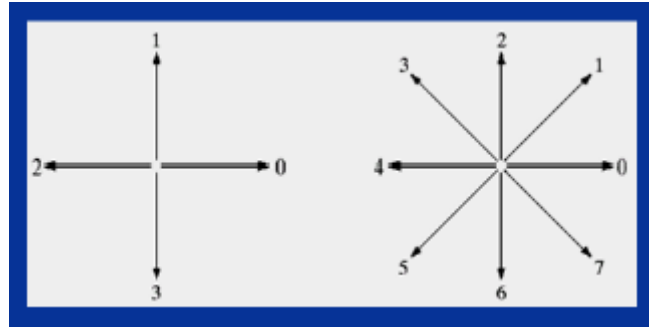
$$P = (1 - a) * ((P_0 + b * P_1) + a * ((1 - b)P_3 + b * P_2)$$

- **Nearest neighbour**
Sätt p till närmaste punkt/pixel.

Shape number

Ett sätt att beskriva ett objekts kontur är att använda s.k. "shape numbers".

Man börjar med att ta fram en kedjekod för objektets kontur genom att följa objektets kanter och beskriva riktningen med siffror. När man har fått fram en kedjekod kan man nu jämföra två objekt.



Dock skulle koden se annorlunda ut ifall objektet var roterat, därför tittar vi istället på skillnaden i riktningen, s.k. *"first difference"*. "First different" fungerar som så att man räknar antalet riktningsändringar moturs mellan två element.

När man tagit fram kodsekvensen för "first difference" måste man normalisera denna kod för att kunna jämföra objekten. Detta görs genom att se koden som en cirkulär talföljd och rotera den så att man får minsta möjliga heltal först.

Ex.1

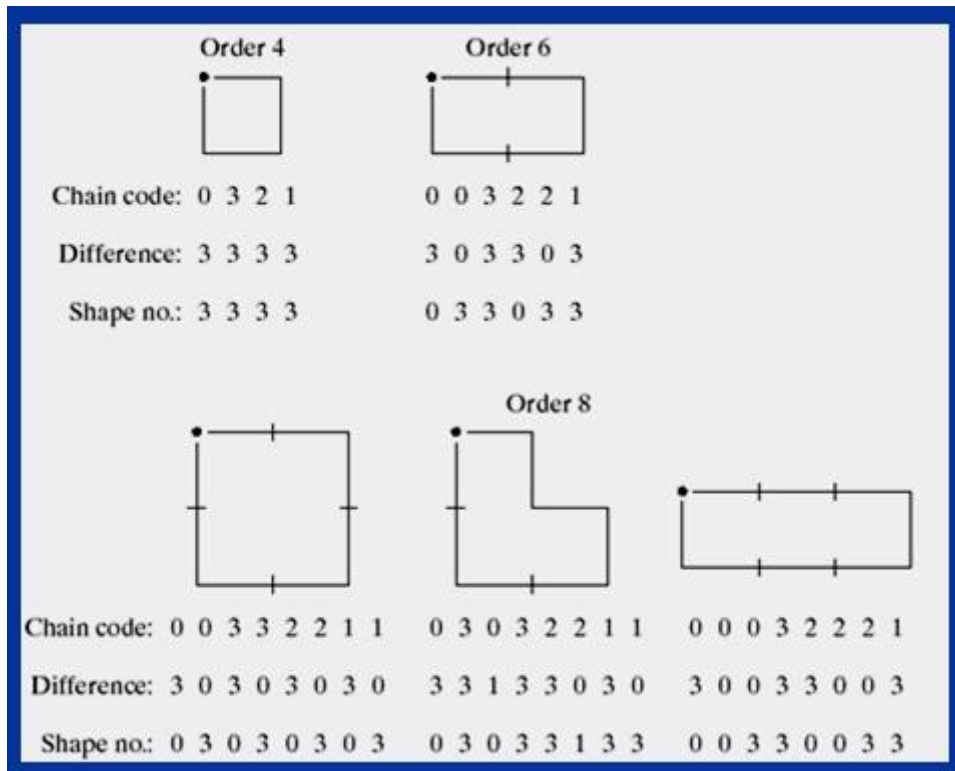
Kedjekoden: 1010303222

ger

First difference: 313313300

Rotation ger: 003133133

Det slutgiltiga "shape numbers" (som har roterats) kan jämföras med andra objekt för att se om de är lika, även om objekten skulle vara roterade eller ej.



Polygonapproximation

Polygonapproximation är en kompakt representation av objekt i binära bilder. Det finns tre typer av polygonapproximationer:

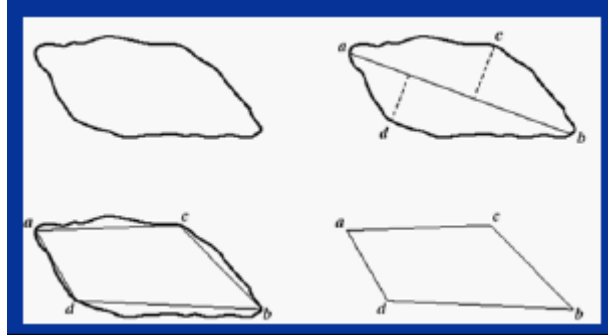
- Merging:

Starta i en konturpixel och testa med linjer tills de följande konturpixlarna. Fortsätt tills felet överstiger en viss tröskel, backa då ett steg bakåt och fixa linjen. Fortsätt sedan med nästa konturpixel som startpunkt. Felmåttet kan beräknas med "minsta kvadrat felet" bl.a.



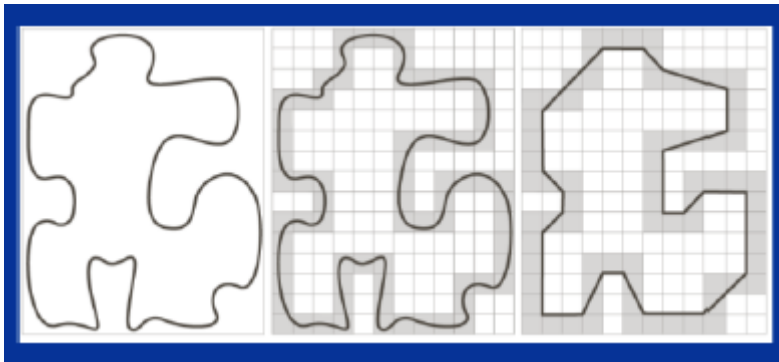
- Splitting:

Dra en linje mellan de två konturpunkterna som ligger längst bort från varandra. Nya hörnpunkter (vertex) bildas i de konturpunkter som har det största vinkelräta avståndet till linjen.



- Minimum perimeter polygon:

Man följer ett objekts kontur och kapslar in den i förbundna celler. Se cellernas yttre och inre konturer som väggar och "spänn" konturen som ett gummiband.

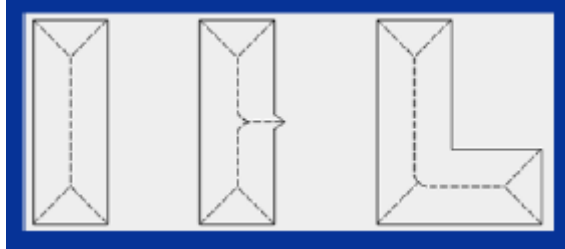


Medianfilter

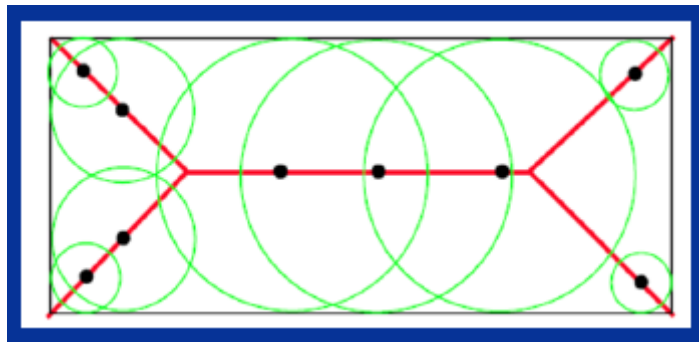
- Är ett olinjärt filter och fungerar så att centropixeln ersätts med medianen av de pixlar som masken täcker.
- Medianfilter används ofta för att undertrycka/ta bort "spikigt" brus i bilden. Ingen utslätning av kanterna sker.
- Fördelen med medianfilter (olinjära filter) jämfört med brusundertryckning med LP-filter är att man undviker utslätning av skarpa kanter.
- Medianfiltrets funktion kan inte utföras i frekvensdomänen, eftersom medianen ej är en multiplikation. För att kunna räkna ut medianen så måste du sortera siffrorna (minsta till största) och sedan plocka ut talet i mitten.

Skelett

- Definieras som de punkter i ett objekt som har mer än en konturpixel som närmsta konturpixel.



- En alternativ definition är att de är centrumpunkter för *maximala discar*. Maximala discar är de största möjliga discarna som ligger inom objektet och som berör objektets kontur i minst två punkter.



- Skelettet kan skapas med hjälp av någon form av tunningsoperationer som t.ex. *Hit-or Miss*, d.v.s en successiv eliminering av konturpixlar.

Minimum distance classifier

"Minimum distance classifier" är en vanlig metod för klassificering av olika områden i stillbilder, s.k. fjärranalys.

En satellit tar bilder i flera våglängdsband där de olika banden representerar t.ex. vatten, land m.m. E

Ett objekts egenskaper beskrivs av en uppsättning featurevektorer vilka innehåller area, färg m.m.

När man ska klassificera med "minimum distance classifier" börjar man först med en s.k. träningsfas. Under denna fas samlar man in ett stort antal feature vektorer från objekt vars klasstillhörighet är känd. Av dessa skapar man sedan en prototypvektor för varje klass, dessa utgörs av medelvärdena av feature vektorerna.

Varje prototypvektor utgör en punkt i det n-dimensionella feature spacet, där n är antalet deskriptorer (feature vektorer).

En feature vektor med okänd klasstillhörighet anses tillhöra den klass som representeras av den prototypvektor som är närmast i feature space.

Region growing

Region growing innebär att man försöker skapa sammanhängande områden av pixlar med liknande egenskaper.

Man undersöker rekursivt alla 8-konnektiva grannar till alla "seed points". En granne vars intensitet avviker mindre än en viss tröskel inkluderas i regionen.

Ett sätt att segmentera en bild är att använda "region growing".

Ex.1

Hitta sprickorna i en röntgenbild av svetsfog.



Ex.2

En bild med den fotometriska upplösningen 0-255 innehåller tre områden markerade med ABC. Område A är det som ska vara kvar i den binära bilden som blir resultatet av segmenteringen.

I A ligger pixelvärdena i intervallet 195-255, i B ligger pixelvärdena i intervallet 15-75 och i C i intervallet 150-215.

- Hur man går till väga:

Bilden trösklas med det högsta värdet i C plus 1 = $215+1 = 216$.

Vissa delar i A kommer att försvinna, eftersom A innehåller pixelvärden mellan 195-255. Men eftersom A innehåller minsta pixelvärdet 195 och då A ligger inuti B som har högsta intensiteten 75, kan vi jämföra trösklingens grannpixlar. Har de värde ≥ 195 tillhör de A och man kommer då få med hela A från originalbilden.

Bildrestaurering \Rightarrow Geometrisk distorsion

Bildåterskapande innebär att man skapar en bild som korrekt återger den scen som fanns framför kameran då bilden togs. Det kräver ofta en modell av kamerans egenskaper vad gäller fotometris och geometrisk distorsion.

Bildförbättring innebär att man skapar en bild som en mänsklig betraktare upplever som bättre genom t.ex. användning av Photoshop.

- **Geometrisk distorsion:**

Distorsion är den tekniska termen för förvrängning vid överföring av en signal (bild, ljus, ljud). Geometrisk distorsion innebär att bildens form förvrängs av objektivet vilket leder till att raka linjer bli krökta.

En bild som korrigeras av geometrisk distorsion kan beskrivas av:

$$X' = T_x(x, y)$$

$$Y' = T_y(x, y)$$

Om T_x och T_y är kända kan originalbilden återskapas med

$$X = T^{-1}(x', y')$$

$$Y = T^{-1}(x', y')$$

Det kan vara svårt att analytiskt lösa ut uttryck för T_x och T_y som beskriver varje pixel i bilden, då kan man använda polynomiska funktioner för att beskriva de spatiella transformationerna.

Målet är alltså att beskriva a_{rk} och b_{rk} för varje pixel i bilden. För att kunna lösa ut a_{rk} och b_{rk} behöver man positioner för referenspunkter i originalbilder (x, y) samt positioner för de distorderade referenspunkterna (u, v) .

Koefficienterna a_{rk} och b_{rk} kan då lösas ut genom att lösa flera linjära ekvationer, där varje ekvation representerar den speciella transformen för en referenspunkt.

V = Van der Monde matris

n = antalet referenspunkter

För att lösa inversen måste man använda **Moore-Peurose** då V-matrisen ej är kvadratisk. För att kunna återskapa en korrekt bild måste man hitta var varje pixel (x, y) har mappats i den distorderade bilden.

Unsharp - Masking

- Är en vanlig metod för skärpning av bilder.
- Man använder ett lågpasfilter på originalbilden för att skapa en oskarp bild. Denna oskarpa bild subtraheras från originalbilden och på så vis får man fram en bild med bara konturer, Denna "konturbild" appliceras sedan på originalet med en viss magnitud och därmed blir bilden skarpare (alla konturer i bilden blir skarpare) Resultatet är en bild med tydligare konturer -> en skarpare bild!