Prüfung „Grundlagen der digitalen Bildbearbeitung“
14.06.2005

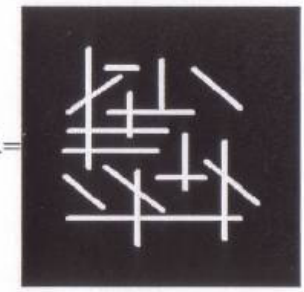
1) **Gegeben sind 24 Bilder, die als Eingabe als auch als Ergebnis einer der 10 Bildoperationen auftreten können**. (14 Punkte)

Grundlagen der Digitalen Bildverarbeitung LV 183.126	Datum: 14.6.2005
Mat.Nr.	Studium

2

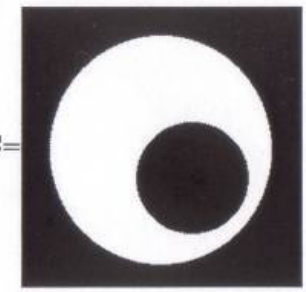
Binärbilder

A=



B=

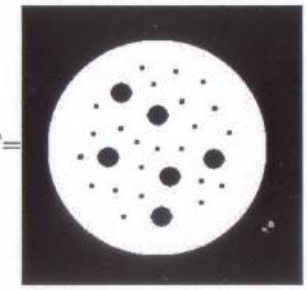
C=



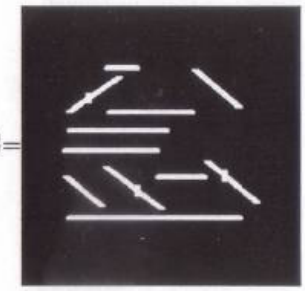
D=

E=

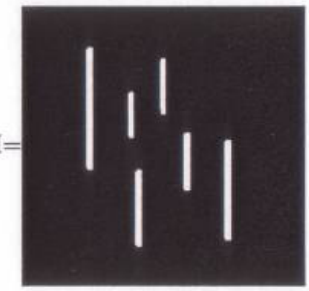
F=



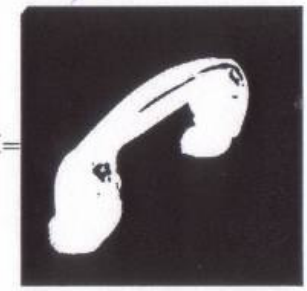
G=



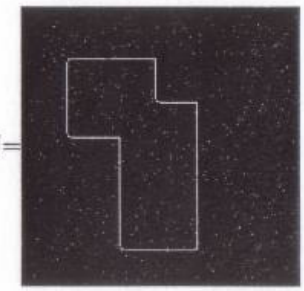
H=



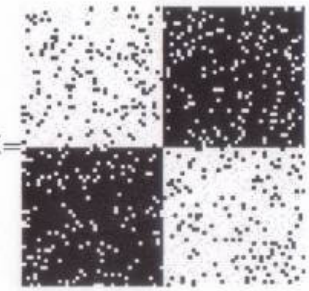
I=



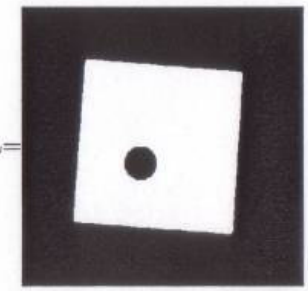
J=



K=



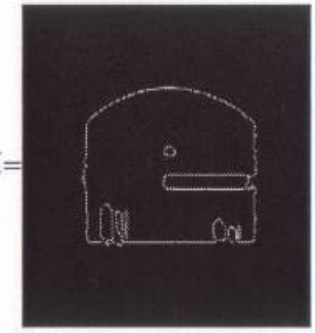
L=



Grundlagen der Digitalen Bildverarbeitung LV 183.126	Datum: 14.6.2005
Mat.Nr.	Studium

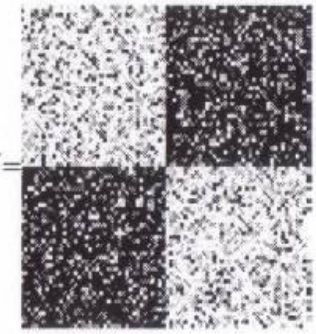
Binärbilder

M=

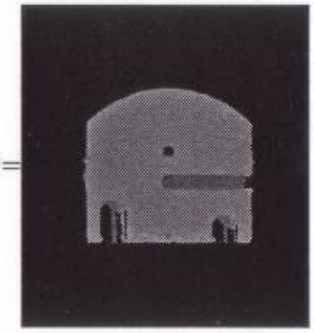


Grauwertbilder

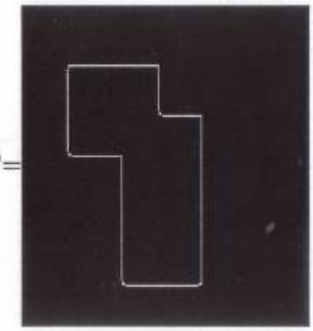
N=



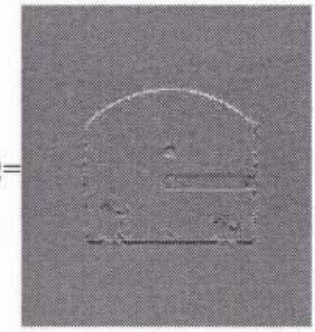
O=



P=

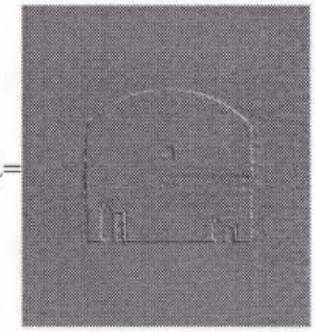


Q=



$g_{min} = -128, g_{max} = 127$

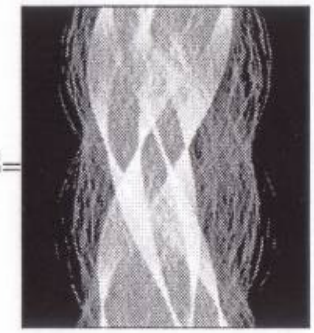
R=



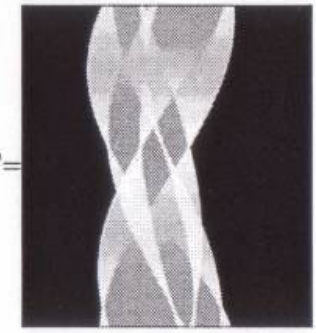
$g_{min} = -128, g_{max} = 127$

Grauwertbilder

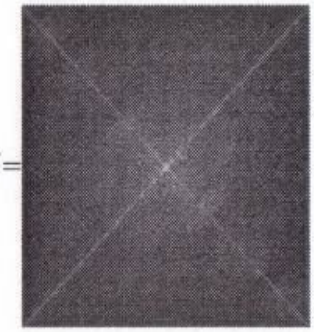
S=



T=



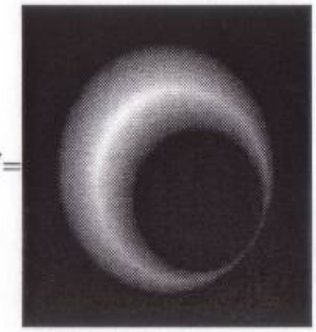
U=



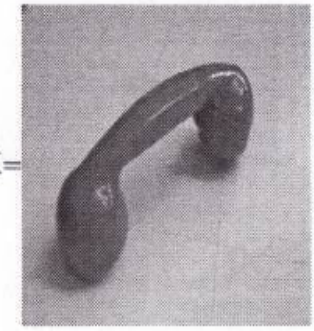
V=

Sonnet for Lena

W=



X=



10 folgenden Bildoperationen, die auf eines oder mehrere der Bilder A-X angewandt wurden und eines der Bilder A-X als Ergebnis haben. Rekonstruieren Sie:

0. __ = medfilt2 (__, [5 5]);
Begründung:

1. __ = conv2 (__, [1 -1 -1; 1 -2 -1; 111]);
Begründung:

2. __ = edge (__, `canny', [0.4 0.5], 1);
Begründung:

3. __ = 1 – im2bw (__, 100 (255));
Begründung:

4. __ = bwmorph (__, `skel`, Inf);
Begründung:

5. __ = imdilate (__, strel (`disk`,5));
Begründung:

6. __ = imopen (__, strel (`line`, 11, 90));
Begründung:

7. __ = lowdist(~__);
Begründung:

8. __ = `Hough Transformation (P);
Begründung:

9. __ = `Hough Transformation (J);
Begründung:

0. _B_ = medfilt2 (_K_, [5 5]);
Begründung: Der Medianfilter erzeugt keine neuen Farbwerte. Kleine Striche werden weggelassen (ich glaube (?) bei einem 5x5-Medianfilter werden Striche kleiner 3 weggelassen)
Beim Medianfilter werden die Grauwerte der Pixel in einer difinierten Umgebung eines Pixels aufgesammelt und der Größe nach sortiert. Nun ersetzt der Grauwert, der sich in der Mitte der sortierten Liste befindet, den Grauwert des aktuellen Pixels.

1. _R_ = conv2 (_O_, [1 -1 -1; 1 -2 -1; 111]);
siehe: http://www.informatik-forum.at/showthread.php?t=32343
Begründung:In diesem Fall kann das Ergebnis nur R sein, was man bei genauerer Betrachtung der Kanten im Bild R sieht.
Denn es handelt sich hier um eine Faltung mit G225°
+1-1-1
+1-2-1
+1+1+1
(siehe Skriptum Seite 41)

2. _M_ = edge (_O_, `canny', [0.4 0.5], 1); ?
Begründung: Der Canny-Operator geht wie folgt vor:
1)Glättung des ganzen Bildes mit einem Gaußfilter
2)ein 2D Gradientenoperator (1.Ableitung z.B. Roberts) hebt starke Grauwertschwankungen hervor,
Kanten werden Grate im Bild des Gradientenbetrags.
3)Verfolgen der Grate und Null setzten aller Pixel, die nicht am Grat liegen: -> dünne Linie (non-maxima suppression)
Canny erkennt so Intensitätssprünge (aber nicht jede Canny-Kante entspricht zwangsläufig einer Objektkante)

...ist aus dem Skriptum abgeschrieben.

3. _I_ = 1 – im2bw (_X_, 100 (255));
Begründung: Das Bild wird mit im2bw in ein Binärbild umgewandelt. "1-" erzeugt daraus (aus dem Binärbild) das "Negativ"
Bei der Umwandlung zum binärbild werden alle Grauwerte auf 1 oder 0 (weiß oder schwarz) gesetzt.
Hierbei gibt 100 den Treshold (=Grauwertschwelle) an, d.h. alle Werte bis inkl. 100 werden 0, alle Werte über 100 werden 1

4. _E_ = bwmorph (_C_, `skel`, Inf); ?
Begründung:Medialachsen-Transformation.
bwmorph ist eine morphologische Operation. Der zweite Parameter gibt an, um welche Op. es sich handelt (close, dilate, erode, ..). In diesem Fall wird 'skel angegeben.
'Skel' ist in Matlab eine Thinning-Funktion bzw. sogennannte Mittelachsen-/Medialachsen-Transformation. Es wird so bezeichnet, da Randpixel gelöscht werden und nur mehr das Skelett (die Mittellinie) überbleibt.
Beim Thinning werden also Kurven verdünt, ohne ihre Länge zu reduzieren und ohne den Zusammenhang zu zerstören.
(weiteres siehe S. 73f)

5. _F_ = imdilate (_D_, strel (`disk`,5));
Begründung: strel erstellt ein Strukturelement, in dem Fall eine Scheibe mit Radius 5 aus lauter 1ern (weiß). imdilate führt dann auf dem Bild D mithilfe dieses Strukturelements eine morphologische Dilatation durch.
Das SE vergrößert die weiße Fläche im Bild, da es in die fläche hineinpasst und die Region ausdehnt.
(wenn das aktuell betrachtete Pixel den GLEICHEN Wert (=weiß) hat wie der Bezugspunkt im Strukturelement, so wird die Nachbarschaft vom aktuell betrachteten Pixel mit dem SE erweitert)
Deshalb erscheint die weiße Fläche im Ausgabebild ausgedehnter.
(obwohl man auf den ersten blick meinen möchte, dass sich die schwarzen flächen verringert haben).

6. _H_ = imopen (_A_, strel (`line`, 11, 90));
Begründung: Das Strukturelement (das durch die Funktion strel erstellt wird) ist in diesem Fall eine 11 Pixel lange Linie mit Wert 1 (=weiß), die (lt. matlab-dokumentation) um 90° gegen den Uhrzeigersinn von der horizontalen Achse gedreht wird (d.h. das SE ist eine vertikale Linie)
imopen führt zuerst eine Erosion auf das Eingabebild aus, darauf dann eine Dilatation mit demselben SE.
Bei der Erosion wird das SE dann auf jeden Pixel gelegt, passt es nicht vollständig in die Region so wird der Referenzpixel im Bild entfernt.
wenn man die Linie auf die Pixel im Bild legt, so passt sie in die vertikalen weißen Striche, die dadurch stehen bleiben. -> die diagonalen und horizontalen Striche werden dadurch entfernt. Weiters verschwinden bei der Erosion die Enden der vertikalen Striche.
Bei der Dilatation geschieht das gleiche, nur werden die bestehenden vertikalen Striche wieder verlängert, da jeder weiße Pixel mit dem Strukturelement ersetzt wird.

7. _W_ = lowdist(~_C_);
Begründung: Für jeden Pixel wird die Distanz zum nächsten Pixel, der != 0 ist ausgegeben.
Wo das Bild weiß ist (wert=1) ist die distanz zum nächsten pixel das non-zero (=nicht schwarz=weiß) ist =0 (viele weiße pixel liegen nebeneinander)
d.h. dieser Teil wird dann schwarz (0) und umgekehrt.
Durch die Distanz-Transformation invertiert sich das Bild.
Um dieser Invertierung entgegen zu wirken (bzw. sie umzukehren) nimmt man den operator '~' der im matlab die NOT funktion darstellt (wiederum eine Invertierung des ganzen)

8. _T_ = `Hough Transformation (P);
Begründung: Hough Transformation=Detektion von Geraden (-Segmenten)
Darstellung im Parameterraum,Houghraum: Geradenbüschel -> Schar von Sinuskurven
Schnittpunkte der n Sinuskurven=Gerade
Hier sehen wir im Bild T klare Schnittpunkte der Sinuskurven => das ursprüngliche Bild muss P sein

9. _S_ = `Hough Transformation (J)
Begründung: Hough Transformation=Detektion von Geraden (-Segmenten)
Darstellung im Parameterraum,Houghraum: Geradenbüschel -> Schar von Sinuskurven
Schnittpunkte der n Sinuskurven=Gerade

Hier sehen wir im Bild S klare Schnittpunkte der Sinuskurven und unzähligen dünnen Sinuskurven=> das ursprüngliche Bild muss J sein, auf Grund der darin enthaltenen Störungen die die unzähligen dünnen Sinuskurven hervorrufen.

Danke an R2D2, xote, chrono, sentencedX und alle anderen aus dem Forum