Einführung in Visual Computing II: Digitale Bildverarbeitung Beispielsammlung zur Vorlesung

VL 186.822: Walter G. Kropatsch, Nicole Artner, Max Göbel
Institute of Computer Aided Automation 183/2 Vienna University of Technology
Pattern Recognition and Image Processing Group

26. März 2012

Einführung in Visual Computing
Teil 2: Grundlagen der Digitalen Bildverarbeitung
Beispielsammlung zur Vorlesung

Autoren: Walter G. Kropatsch, Nicole Artner, Max Göbel, Ines Janusch, Roxane Licandro
Auflage vom 26. März 2012

Diese Beispielsammlung beinhaltet zahlreiche Übungsbeispiele und Prüfungsbeispiele, um das Verständnis des Inhalts der Vorlesung Einführung in Visual Computing, Teil 2: Grundlagen der digitalen Bildverarbeitung zu vertiefen. können aus mehreren kapitelübergreifenden Teilen bestehen, die behandelten Kapitel sind an den zugehörigen Stellen gekennzeichnet. Die Vermerke beziehen sich auf die Auflage SS2012 des Skriptums. Die Beispiele sind (wie auch im Rahmen der Prüfung) durch Einsetzen bestimmter Stellen der Matrikelnummer individualisiert. Um die Beispiele zu variieren und mehrmals durchrechnen zu können, können zu Übungszwecken auch beliebige (entsprechend der Angabe) Zahlen eingesetzt werden.

Inhaltsverzeichnis

1 Bildoperationen 1
1.1 Filter, Threshold, Histogramm, Morphologie, Distanztransformation 2
1.2 Filter, Histogramm, Threshold, Morphologie, logische Verknüpfung 5
1.3 Filter, Histogramm, Threshold, Morphologie 8
1.4 Matlab Referenz 11

2 Mathematisches Nachvollziehen 13
2.1 Rekonstruktion aus Kantenbildern 13
2.2 Hough-Transformation (siehe Kapitel E) 15
2.3 Morphologie 18
2.4 Distanztransformation 21
2.5 Pyramide 24
2.6 Pfade mit Würfeln 27
2.7 Freeman Chain Code 29
2.8 Array Grammatiken 31
2.9 Lauflängen, Zusammenhang, Coocurrence 33
2.10 Integral Image und Predictive Coding 35

1 Bildoperationen

In diesem Kapitel soll das Verständnis für Bildoperationen anhand von 3 Beispielen vertieft werden. Die Bildoperationen werden auf Bilder angewendet, die im Anschluss jedes Beispiels zu finden sind. Beispiele können auch Bildoperationen enthalten, die in Form von MATLAB-Befehlen ausgedrückt werden. Die zugehörigen MATLAB Referenzen sind am Ende dieses Kapitels aufgelistet. Zusätzliche Referenzen können in der MATLAB Hilfe oder online unter http://www.mathworks.de/help/toolbox/images/ nachgeschlagen werden.

1.1 Filter, Threshold, Histogramm, Morphologie, Distanztransformation

Im ersten Beispiel sollen Sie Ergebnisbilder über vorgegebene Operationen mit den gegebenen Eingabebildern in Beziehung setzen. Im Anschluss an dieses Beispiel finden Sie 24 Bilder die als Eingabe als auch als Ergebnis einer Bildoperation auftreten können. Beachten Sie, dass nicht ALLE Bilder verwendet werden, es kann Bilder geben, die nicht als Eingabe- oder Ergebnisbilder aufscheinen.Die angegebenen Bilder haben eine Größe von 350x350 Pixeln. Grauwertbilder haben einen Wertebereich von 0 bis 255 (falls nicht anders angegeben). Logische Operationen werden nur auf Binärbilder (Schwarz-Weiss-Bilder) angewendet `true` wird durch den Wert 1 (=weiss) repräsentiert, `false` durch den Wert 0 (=schwarz).

Folgende Liste enthält 10 Bildoperationen, die auf eines oder mehrere (z.B. $Y+Z$) der Bilder A-X angewandt wurden und eines der Bilder A-X als Ergebnis haben. Ihre Aufgabe ist die Rekonstruktion dieser 10 Bildoperationen. Tragen Sie bitte die Bildnamen (`A-X`) in die Kästchen ☐ der jeweiligen Operation ein.

0. ☐ = `medfilt2(` ☐`);`

   Begründung: ..... (siehe Kapitel D)

1. ☐ = `conv2(` ☐`, fspecial('gaussian',[20 20],10);`

   Begründung: ..... (siehe Kapitel C, D)

2. ☐ = `conv2(` ☐`,[1 0 -1; 2 0 -2; 1 0 -1]);`

   Begründung: ..... (siehe Kapitel E)

3. ☐ = `conv2(` ☐`,[1 2 1; 0 0 0; -1 -2 -1]);`

   Begründung: ..... (siehe Kapitel E)

4. ☐ = `1-im2bw(` ☐`,200/255);`

   Begründung: ..... (siehe Kapitel C)

5. ☐ = `imhist(M);`

   Begründung: ..... (siehe Kapitel C)

6. ☐ = `imdilate(` ☐`,strel('disk',15));`

   Begründung: ..... (siehe Kapitel G)

7. ☐ = `imerode(` ☐`,strel('disk',15));`

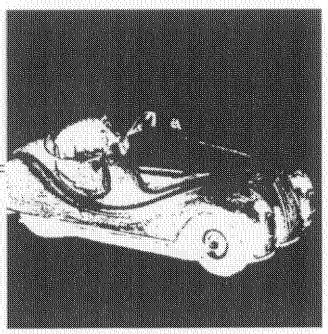
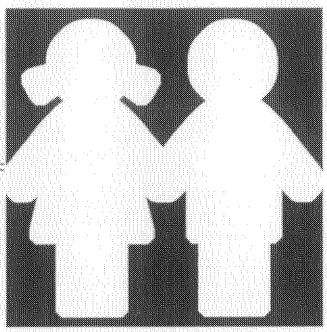
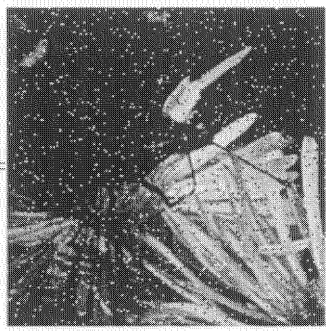
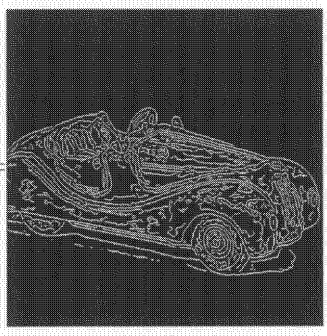
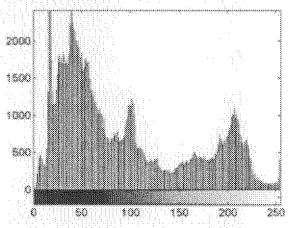
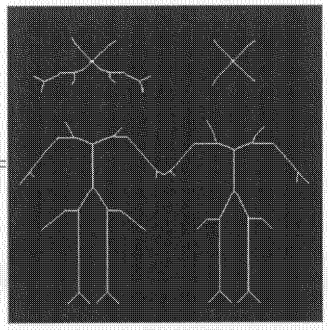
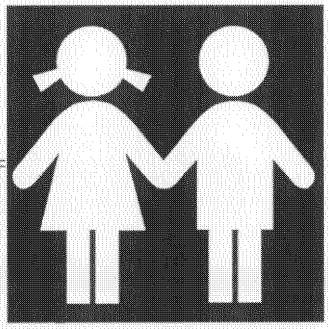
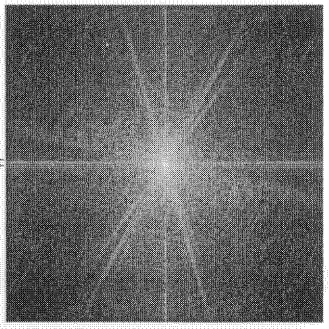
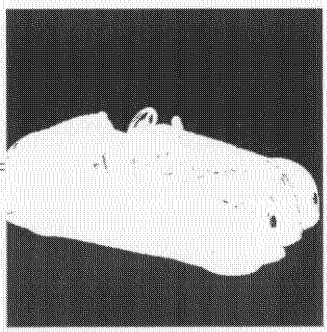
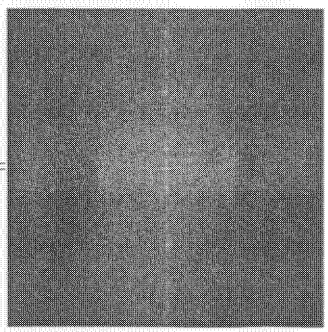
   Begründung: ..... (siehe Kapitel G)

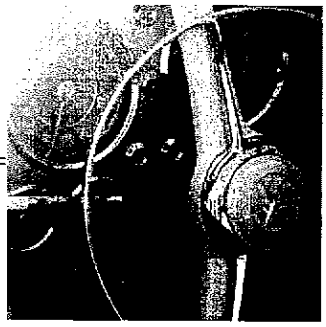
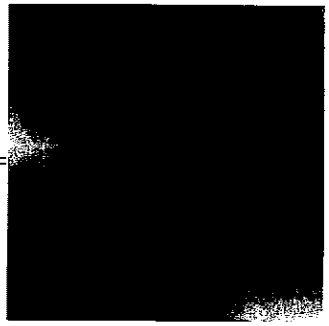
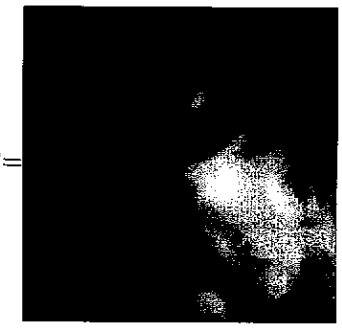
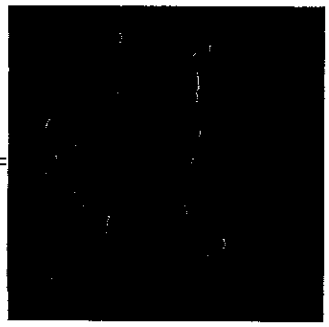
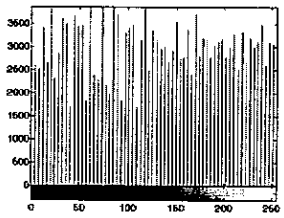
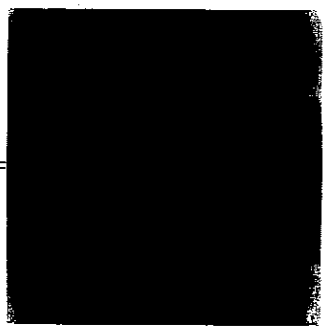
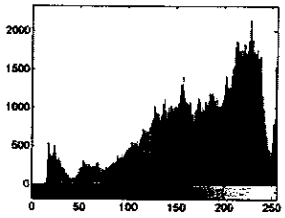
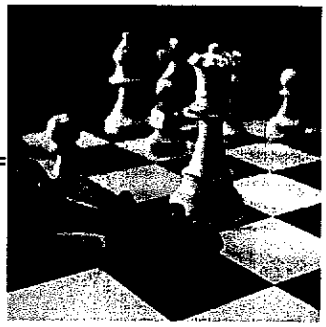
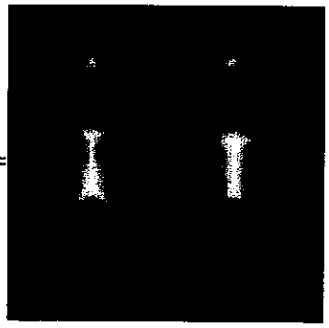
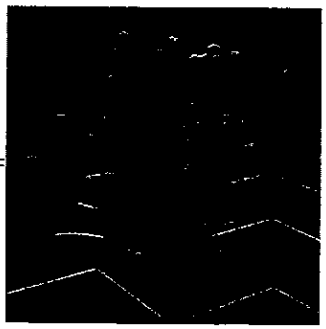
8. ☐ = `bwdist(~`☐`);`

   Begründung: ..... (siehe Kapitel G)

9. ☐ = `log1p(abs(fftshift(fft2(V))));`

   Begründung: ..... (siehe Kapitel E)

Binärbilder		Grauwertbilder
A= 	B= 	C= 
D=	E= 	F= 
G= 	H= 	I= 
J=	K= 	L= 

Grauwertbilder		
M= 	N= 	O=
P= 	Q= 	R= 
S= 	T=	U= 
V= 	W= 	X= 

1.2 Filter, Histogramm, Threshold, Morphologie, logische Verknüpfung

In diesem Beispiel sollen Sie Ergebnisbilder über vorgegebene Operationen mit den gegebenen Eingabebildern in Beziehung setzen. Im Anschluss an dieses Beispiel sind 24 Bilder aufgelistet, die als Eingabe als auch als Ergebnis einer Bildoperation auftreten können. Beachten Sie, dass nicht ALLE Bilder verwendet werden, es kann Bilder geben, die nicht als Eingabe- oder Ergebnisbilder aufscheinen.

Folgende Liste enthält 10 Bildoperationen, die auf eines oder mehrere (z.B. $Y + Z$) der Bilder A-X angewandt wurden und eines der Bilder A-X als Ergebnis haben. Ihre Aufgabe ist die Rekonstruktion dieser 10 Bildoperationen. Tragen Sie bitte die Bildnamen (`A-X`) in die Kästchen ☐ der jeweiligen Operation ein.

0. ☐ = `conv2(☐,fspecial('gaussian',[7 7],1));`

   Begründung: ..... (siehe Kapitel C, D)

1. ☐ = `medfilt2(☐);`

   Begründung: ..... (siehe Kapitel D)

2. ☐ = `imhist(S);`

   Begründung: ..... (siehe Kapitel C)

3. ☐ = `imhist(N);`

   Begründung: ..... (siehe Kapitel C)

4. ☐ = `imcomplement(☐);`

   Begründung: ..... (siehe Kapitel C)

5. ☐ = `imerode(☐,strel('disk',20));`

   Begründung: ..... (siehe Kapitel G)

6. ☐ = `im2bw(☐,100/255);`

   Begründung: ..... (siehe Kapitel C)

7. ☐ = `edge(☐,'canny',[0.4,0.5],1);`

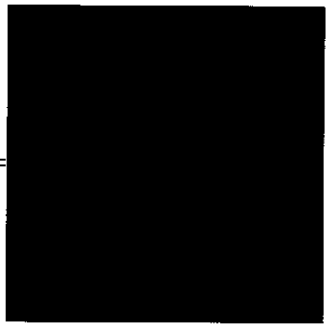
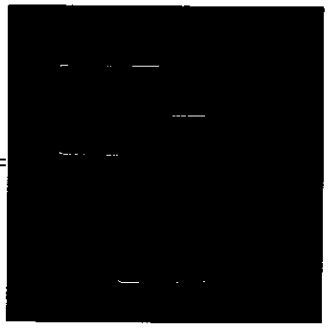
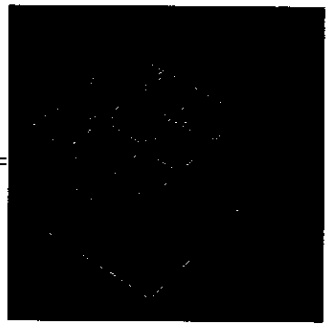
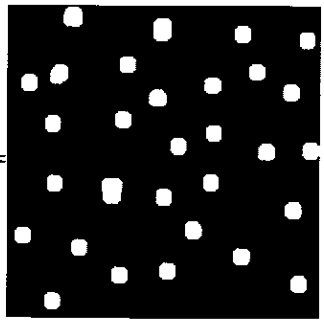
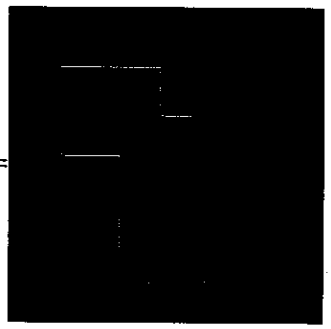
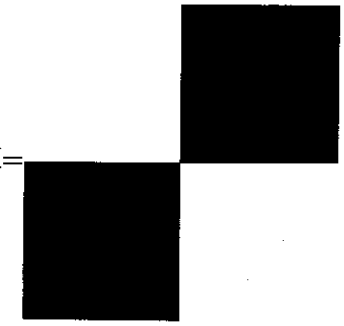
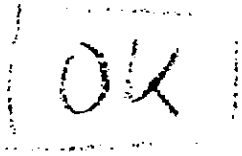
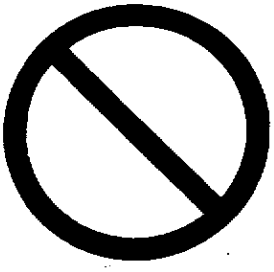
   Begründung: ..... (siehe Kapitel E)

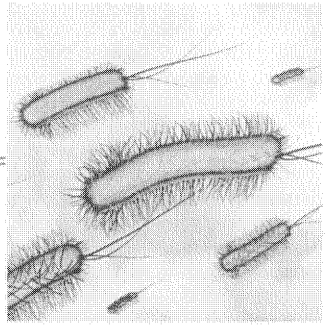
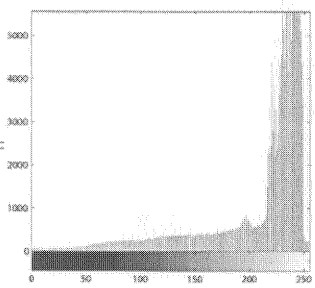
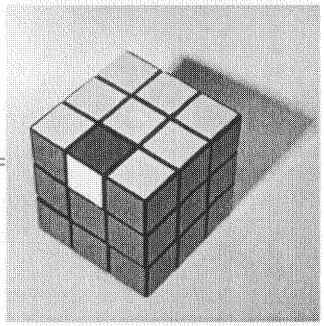
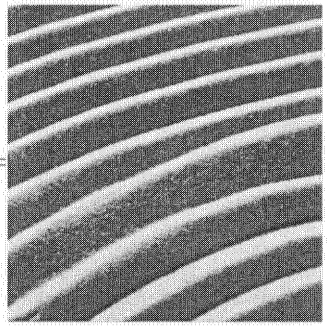
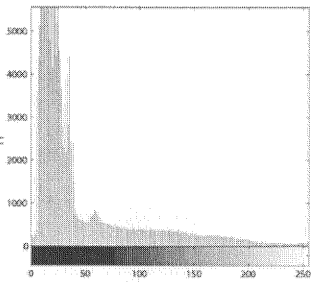
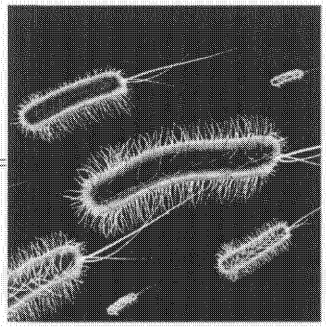
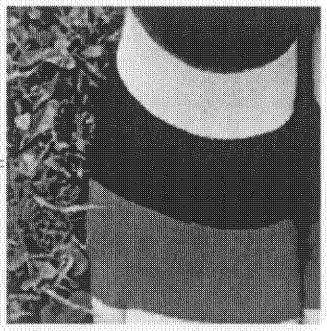
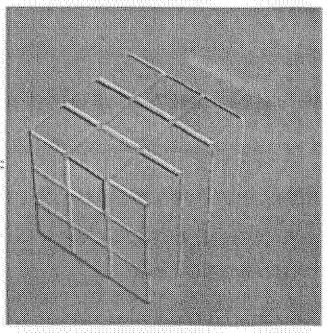
8. ☐ = `or(☐,☐); % Verwenden Sie 3 unterschiedliche Bilder!`

   Begründung: ..... (siehe Kapitel G)

9. ☐ = `and(☐,not(☐)); % Verwenden Sie 3 unterschiedliche Bilder!`

   Begründung: ..... (siehe Kapitel G)

Binärbilder		
A=	B= 	C= 
D= 	E=	F= 
G= 	H=	I= 
J= 	K=	L= 

Grauwertbilder		
M=	N= 	O= 
P= 	Q= 	R= 
S= 	T=	U=
V=	W= 	X= 

1.3 Filter, Histogramm, Threshold, Morphologie

Setzen Sie Ergebnisbilder über vorgegebene Operationen mit den gegebenen Eingabebildern in Beziehung. Auf den folgenden 2 Seiten finden Sie 24 Bilder die als Eingabe als auch als Ergebnis einer Bildoperation auftreten können. Beachten Sie, dass nicht ALLE Bilder verwendet werden, es kann Bilder geben, die nicht als Eingabe- oder Ergebnisbilder aufscheinen.

Folgende Liste enthält 10 Bildoperationen, die auf eines oder mehrere (z.B. $Y + Z$) der Bilder A-X angewandt wurden und eines der Bilder A-X als Ergebnis haben. Ihre Aufgabe ist die Rekonstruktion dieser 10 Bildoperationen. Tragen Sie bitte die Bildnamen (`A-X`) in die Kästchen ☐ der jeweiligen Operation ein.

0. ☐ `= conv2(` ☐ `,fspecial('laplacian'));`

   Begründung: ..... (siehe Kapitel C, D)

1. ☐ `= conv2(` ☐ `,[1 2 1; 0 0 0; -1 -2 -1]);`

   Begründung: ..... (siehe Kapitel C, D)

2. ☐ `= conv2(` ☐ `,[1 0 -1; 2 0 -2; 1 0 -1]);`

   Begründung: ..... (siehe Kapitel C, D)

3. ☐ `= imhist(F);`

   Begründung: ..... (siehe Kapitel C)

4. ☐ `= imhist(L);`

   Begründung: ..... (siehe Kapitel C)

5. ☐ `= imdilate(` ☐ `,strel('disk',5));`

   Begründung: ..... (siehe Kapitel G)

6. ☐ `= imerode(` ☐ `,strel('disk',5));`

   Begründung: ..... (siehe Kapitel G)

7. ☐ `= imopen(` ☐ `,strel('line',7,0));`

   Begründung: ..... (siehe Kapitel G)

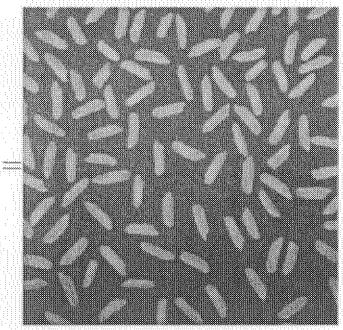
8. ☐ `= im2bw(` ☐ `,100/255);`

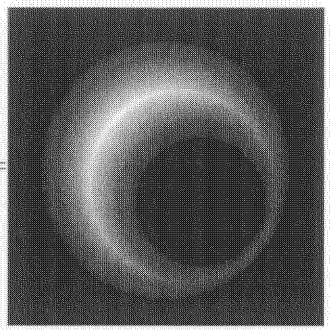
   Begründung: ..... (siehe Kapitel C)

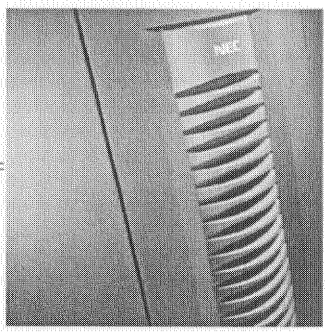
9. ☐ `= im2bw(` ☐ `,200/255);`

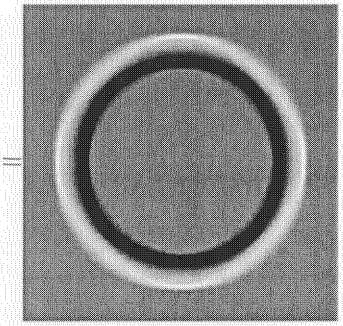
   Begründung: ..... (siehe Kapitel C)

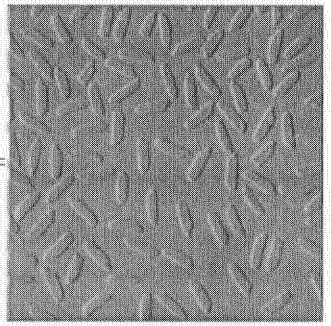
Grauwertbilder

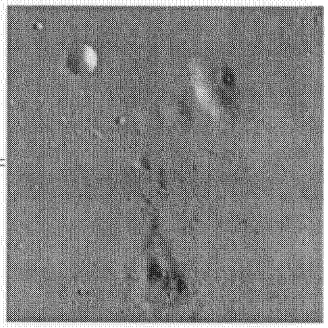
A= 

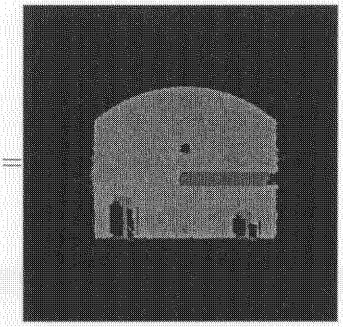
B= 

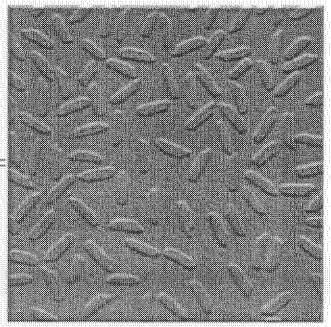
C= 

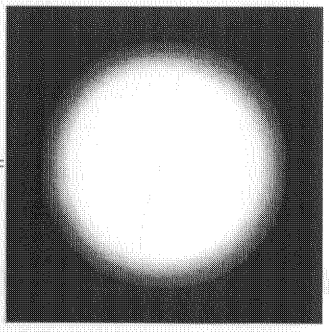
D= 

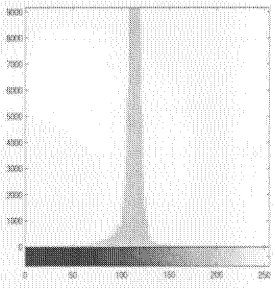
E= 

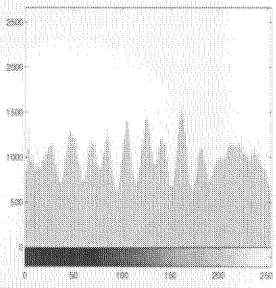
F= 

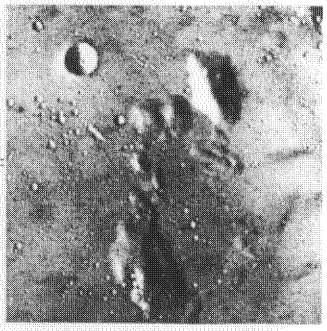
G= 

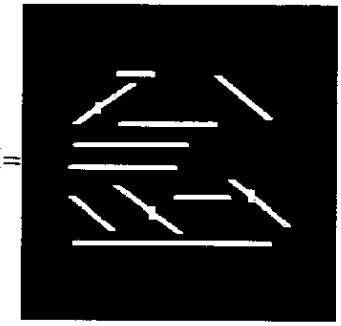
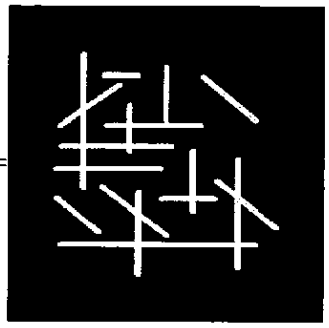
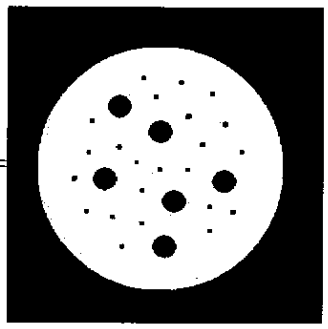
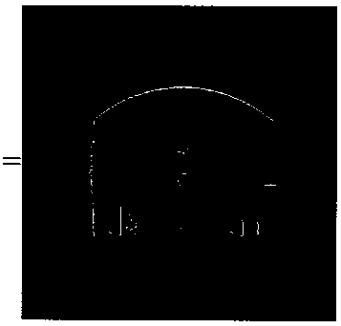
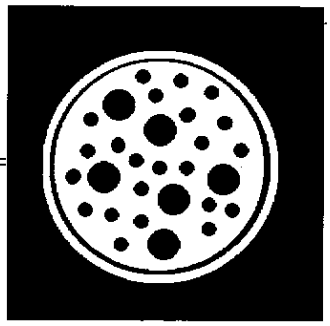
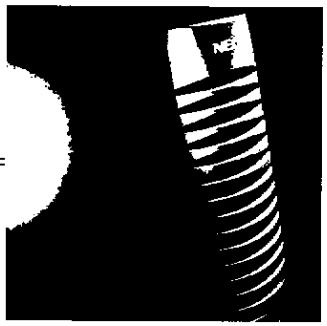
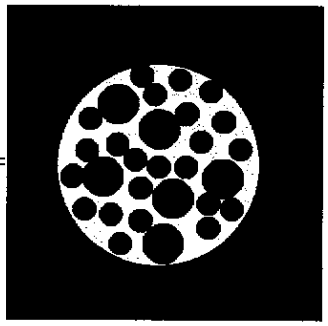
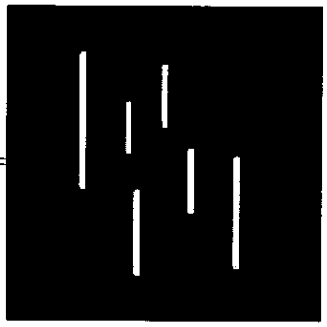
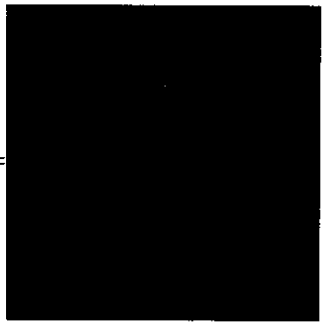
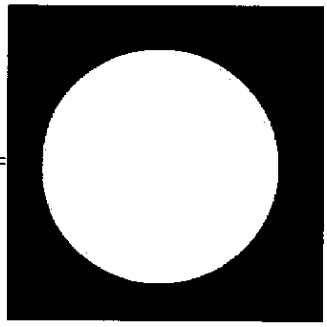
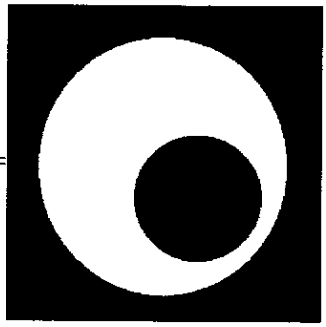
H= 

I= 

J= 

K= 

L= 

Binärbilder		
M= 	N= 	O= 
P= 	Q= 	R= 
S=	T= 	U= 
V= 	W= 	X= 

1.4 Matlab Referenz

Notationen

Matrix	$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$	`A=[a b; c d];`	*Spaltenvektor*	$x = \begin{pmatrix} y \\ z \end{pmatrix}$	`x=[y;z]`
Zeilenvektor	$e = \begin{pmatrix} f & g \end{pmatrix}$	`e=[f g]`			

Command Reference

`Y = abs(X)`
returns an array Y such that each element of Y is the absolute value of the corresponding element of X.

`D = bwdist(BW)`
computes the Euclidean distance transform of the binary image BW. For each pixel in BW, the distance transform assigns a number that is the **distance between that pixel and the nearest nonzero pixel** of BW. bwdist uses the Euclidean distance metric by default. D is the same size as BW.

`BW2 = bwmorph(BW,'skel',n)`
with n = Inf, removes pixels on the boundaries of objects but does not allow objects to break apart. The pixels remaining make up the image skeleton. This option preserves the Euler number.

`C=conv2(A,B)`
computes the two-dimensional convolution of matrices A and B.

`BW = edge(I,'canny',thresh,sigma)`
specifies sensitivity thresholds for the Canny method. thresh is a two-element vector in which the first element is the low threshold, and the second element is the high threshold. sigma is the standard deviation of the Gaussian filter.

`Y = fft2(X)`
returns the two-dimensional discrete Fourier transform (DFT) of X, computed with a fast Fourier transform (FFT) algorithm.

`Y = fftshift(X)`
rearranges the outputs of fft, fft2, and fftn by moving the zero-frequency component to the center of the array.

`h = fspecial('gaussian',hsize,sigma)`
returns a rotationally symmetric Gaussian lowpass filter of size hsize with standard deviation sigma (positive). hsize can be a vector specifying the number of rows and columns in h

`h = fspecial('laplacian')`
returns a 3-by-3 filter approximating the shape of the two-dimensional Laplacian operator.

`J = histeq(I,n)`
transforms the intensity image I, returning in J an intensity image with n discrete gray levels. A roughly equal number of pixels is mapped to each of the n levels in J, so that the histogram of J is approximately flat. (The histogram of J is flatter when n is much smaller than the number of discrete levels in I.) The default value for n is 64

`BW = im2bw(I,level)`
converts the intensity image I to black and white. The output binary image BW has values of 0 for all pixels in the input image with luminance g less than $level$ and 1 for all other pixels:

$$bw = 0 \Leftrightarrow \frac{g - g_{min}}{g_{max} - g_{min}} < level$$

`J=imhist(I)`
displays histogram of image data I.

`IM2 = imclose(IM,SE)`
performs morphological closing on the grayscale or binary image IM, returning the closed image, IM2. The structuring element, SE, must be a single structuring element object.

`IM2 = imdilate(IM,SE)`
dilates the grayscale, binary, or packed binary image IM, returning the dilated image, IM2. The argument SE is a structuring element object.

`IM2 = imerode(IM,SE)`
erodes the grayscale, binary, or packed binary image IM, returning the eroded image IM2. The structuring element, SE, must be a single structuring element object.

`J = imnoise(I,'gaussian',m,v)`
adds Gaussian white noise of mean m and variance v to the image I.

`J = imnoise(I,'salt & pepper',d)`
adds salt and pepper noise to the image I, where d is the noise density.

`IM2 = imopen(IM,SE)`
performs morphological opening on the grayscale or binary image IM with the structuring element SE. The argument SE must be a single structuring element object.

`y = log1p(x)`
computes log(1+x), compensating for the roundoff in 1+x.

`B = medfilt2(A)`
performs median filtering of the matrix A using the default 3-by-3 neighborhood.

`SE = strel('disk',R)`
creates a flat, disk-shaped structuring element, where R specifies the radius.

`SE = strel('line',LEN,DEG)`
creates a flat, linear structuring element, where LEN specifies the length, and DEG specifies the angle (in degrees) of the line, as measured in a counterclockwise direction from the horizontal axis.

`~A`
equals not(A);

`B = imcomplement(A)`
computes the complement of the image A.

`Y = and(A,B)`
is a matrix whose elements are 1's where both A and B have non-zero elements, and 0's where either has a zero element.

`Y = or(A,B)`
is a matrix whose elements are 1's where either A or B have non-zero elements, and 0's where both have a zero element.

2 Mathematisches Nachvollziehen

In diesem Kapitel sollen Sie einfache Bildverarbeitungsoperationen numerisch nachvollziehen. Bezeichne $M_1, M_2, \ldots M_7$ die 7 Ziffern Ihrer Matrikelnummer M.

2.1 Rekonstruktion aus Kantenbildern

2.1.1 Beispiel 1 - Grauwertbild aus zwei teilweise gegebenen Kantenbildern

1. Ein 4×4 Grauwertbild I wurde mit einem horizontalen und einem vertikalen Kantenfilter gefaltet (siehe Kapitel E): $H = (\boxed{-1} \quad 1) * I$, $V = \begin{pmatrix} 1 \\ \boxed{-1} \end{pmatrix} * I$. Für die Ränder wurde in beiden Fällen der zyklische Abschluß angenommen (siehe Kapitel D). Rekonstruieren Sie aus den 16 bekannten Werten alle restlichen 32 Werte von H, V, I.

$$H = \begin{pmatrix} -1 & 0 & 2 & \\ 0 & -2 & 5 & \\ 2 & 3 & -1 & \\ & -1 & 0 & 7 \end{pmatrix}, \quad V = \begin{pmatrix} & & & \\ -2 & & & \\ 4 & & & \\ -3 & & & \end{pmatrix}, \quad I = \begin{pmatrix} & & & 8 \\ & & & \\ & & & \\ & & & \end{pmatrix}.$$

2. Tip: Der zyklische Abschluss (siehe Kapitel D) stellt eine Beziehung zwischen den vier Werten der Zeilen von H und der Spalten von V her. Welche?

   .....

   .....

3. Tip: diagonal benachbarte Pixel von I sind über zwei 4-Wege (siehe Kapitel G) verbunden: zuerst horizontal und dann vertikal; oder zuerst vertikal und dann horizontal. Das ergibt eine weitere Beziehung zwischen zwei benachbarten Werten von H und von V:

   .....

   .....

4. Welche Formeln werden für die Rekonstruktion von I verwendet?

   .....

2.1.2 Beispiel 2 - Grauwertbild aus zwei teilweise gegebenen Kantenbildern

1. Ein 4×4 Grauwertbild I wurde mit einem horizontalen und einem vertikalen Kantenfilter gefaltet (siehe Kapitel E): $H = (\boxed{-1} \quad 1) * I$, $V = \begin{pmatrix} 1 \\ \boxed{-1} \end{pmatrix} * I$. Für die Ränder wurde in beiden Fällen der zyklische Abschluß angenommen (siehe Kapitel D). Rekonstruieren Sie aus den 16 bekannten Werten alle restlichen 32 Werte von H, V, I.

2. In die mit ○ gekennzeichneten Pixel tragen Sie die 7 Ziffern Ihrer Matrikelnummer ein (beliebig):

$$H = \begin{pmatrix} -1 & 0 & 2 & \\ 0 & -2 & \bigcirc & \\ 2 & \bigcirc & -1 & \\ & -1 & 0 & \bigcirc \end{pmatrix}, \quad V = \begin{pmatrix} & & & \\ \bigcirc & & & \\ \bigcirc & & & \\ \bigcirc & & & \end{pmatrix}, \quad I = \begin{pmatrix} & & & \bigcirc \\ & & & \\ & & & \\ & & & \end{pmatrix}.$$

3. Tip: Der zyklische Abschluss (siehe Kapitel D) stellt eine Beziehung zwischen den vier Werten der Zeilen von H und der Spalten von V her. Welche?

   .....

   .....

4. Tip: diagonal benachbarte Pixel von I sind über zwei 4-Wege (siehe Kapitel G) verbunden: zuerst horizontal und dann vertikal; oder zuerst vertikal und dann horizontal. Das ergibt eine weitere Beziehung zwischen zwei benachbarten Werten von H und von V:

   .....

   .....

5. Welche Formeln werden für die Rekonstruktion von I verwendet?

   .....

   .....

2.2 Hough-Transformation (siehe Kapitel E)

2.2.1 Beispiel 1 - Diskrete Tomographie

1. Zwei Parameter bestimmen den Inhalt des Hough-Akkumulators:
$a = \max\{M_i | i = 1, \ldots, 7\} = \square \quad b = \begin{cases} a-1 & \text{if median } \{M_i | i = 1, \ldots, 7\} = \square < 5 \\ a-2 & \text{sonst} \end{cases} = \square$.

2. Im Hough-Akkumulator werden die drei Winkel $\theta \in \{0^0, 90^0, 135^0\}$ und pro Winkel drei Abstände $r =$ und die 4 Intervalle dazwischen gezählt:

	$r < r_1$	r_1	$r_1 < r < r_2$	r_2	$r_2 < r < r_3$	r_3	$r_3 < r$
$\theta = 0^0$		$r_1 = 1$		$r_2 = a$		$r_3 = a + 3$	
$H(\theta, r) =$	0	b	3	$b + 1$	3	b	0
$\theta = 90^0$		$r_1 = 1$		$r_2 = b$		$r_3 = b + 3$	
$H(\theta, r) =$	0	a	3	$a + 1$	3	a	0
$\theta = 135^0$		$r_1 = \frac{1-a}{\sqrt{2}}$		$r_2 = \frac{b-a}{\sqrt{2}}$		$r_3 = \frac{b-1}{\sqrt{2}}$	
$H(\theta, r) =$	0	4	3	5	3	4	0

3. Die Hesse'schen Normalform $\boxed{r = \qquad\qquad}$ lässt sich für die drei Winkel[1] in ganzzahligen Koordinaten vereinfachen:

θ	$f(x, y)$	$=$	$g(r)$
0^0		$=$	
90^0		$=$	
135^0		$=$	

4. Jedes Element $H(\theta, r)$ entspricht der Anzahl von Pixel auf der durch θ und r bestimmten Geraden, kann also als 'Projektion' entlang der Gerade gedeutet werden. Ihre Aufgabe ist die Umkehr dieser Projektion (auch diskrete Tomographie genannt), wobei als Ergebnis das Binärbild gesucht ist, dessen Houghtransformation oben spezifiziert ist. Tipp: Ist $H(\theta, r) = 0$ so sind alle Pixel dieser Gerade 0. Gibt es genau n Elemente einer Geraden, die nicht Null sind und $H(\theta, r) = n$, dann sind alle fehlenden Pixel dieser Gerade 1.

$y = 12$														
$y = 11$														
$y = 10$														
$y = 9$														
$y = 8$														
$y = 7$														
$y = 6$														
$y = 5$														
$y = 4$														
$y = 3$														
$y = 2$														
$y = 1$														
$y = 0$														
$x =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13

[1] $\sin 135^o = 1/\sqrt{2}$

2.2.2 Beispiel 2 - Gesichtserkennung (siehe Kapitel E)

Im Technischen Report MSU-CSE-01-7 [16] ist diese Lagebestimmung des Gesichtes wie folgt beschrieben: The Hough transform is used to fit an elliptical shape to the face boundary map. An ellipse in a plane has five parameters: an orientation angle, two coordinates of the center, and lengths of major and minor axes. Since we know the locations of eyes and mouth, the orientation of the ellipse can be estimated from the direction of a vector that starts from the midpoint between the eyes towards the mouth. The location of the ellipse center is estimated from the face boundary map. Hence, we need only a two-dimensional accumulator for estimating the ellipse for bounding the face. The accumulator is updated by perturbing the estimated center by a few pixels for a more accurate localization of the ellipse.

Das Gesicht wird durch einen Kreis approximiert, der mit der Hough-Transformation bestimmt wird. Skizziere die Berechnungsschritte und fülle die fehlenden Formeln ein, wobei die jeweils vorher definierten Variablen verwendet werden können.

Rotation in vertikale Lage: bezeichne A_l, A_r, M die Positionen von linkem und rechtem Auge und Mund. Dann ist die Hauptachsenrichtung H (Vektornotation) folgendermaßen gegeben:

$$H = \boxed{} \tag{1}$$

Damit sind die Komponenten $\binom{p}{q} = \frac{H}{|H|}$ der normierten Hauptachsenrichtung bestimmt. Die Rotationsmatrix N muß folgender Bedingung genügen und berechnet sich wie folgt (Formeln in leere Kästchen eintragen):

$$N\begin{pmatrix}p\\q\end{pmatrix} = \begin{pmatrix}\boxed{} & \boxed{}\\ \boxed{} & \boxed{}\end{pmatrix}\begin{pmatrix}p\\q\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix} \tag{2}$$

Akkumulation der Kreisparameter: Zur Vereinfachung verwenden wir statt der Ellipse einen Kreis. Seien Z das geschätzte Kreiszentrum und (B, V) ein Kantenpunkt B mit seiner Orientierungsrichtung V ($|V| = 1$!). Nach Verschiebung des Zentrums in den Ursprung und der Rotation in die vertikale Lage erhält die Kante neue Werte (B', V') (formelmäßige Angabe in Vektornotation):

$$B' = \begin{pmatrix}x\\y\end{pmatrix} = \boxed{} \tag{3}$$

$$V' = \begin{pmatrix}s\\t\end{pmatrix} = \boxed{} \tag{4}$$

Aus der Kreisgleichung läßt sich der Radius aus der Kante (B', V') mit Position $B' = \binom{x}{y}$ (3) und der Richtung der Tangente $V' = \binom{s}{t}$ (4) bestimmen (Anm.: in Parameterdarstellung $x = x(t), y = y(t)$ ist die Tangentialrichtung durch $(dx/dt, dy/dt)$ bestimmt.):

$$r^2(x, y, s, t) = \boxed{} \tag{5}$$

2.2.3 Beispiel 3 - Parallele Linien (siehe Kapitel E)

1. Sei $S = \text{mod}(M, 3) + 5 = \square$ der Abstand einer Schar von vertikalen Linien.

2. Die Linienschar sei wie folgt definiert:

$$B(x,y) = \begin{cases} 1 & \text{wenn } \text{mod}(x-2, S) = 0, x \in [0,10], y \in [0,5] \\ 0 & \text{sonst} \end{cases}$$

3. Zeichnen Sie alle Punkte $B(x,y) = 1$ als x im folgenden Bild zusätzlich zu den zwei Störpixeln "*" ein:

$y = 5$											
$y = 4$											
$y = 3$						*					
$y = 2$					*						
$y = 1$											
$y = 0$											
$x =$	0	1	2	3	4	5	6	7	8	9	10

4. Bestimme die Hough-Transformation für alle Punkte ($B(x,y) = 1$) mit der Hesse'schen Normalform $r =$ und $\theta \in \{0^0, 45^0, 90^0, 135^0\}$. Zuerst werden die $r-$Werte für die vier Winkel[2] benötigt:

θ	$r(x,y)$	Werte für r bzw. $r\sqrt{2}$ für alle (x,y) mit $B(x,y) = 1$
0^0	$r =$	
45^0	$r = (\qquad)/\sqrt{2}$	
90^0	$r =$	
135^0	$r = (\qquad)/\sqrt{2}$	

5. Trage die Anzahl der Punkte in den folgenden quantisierten Hough-Akkumulator $H(\theta, r)$ ein:

$r \in$	...-5,-6	-4	-3	-2,-1	0	1,2	3	4	5,6	7	8,9	10	11...
$r\sqrt{2} \in$	...-4	-3	-2	-1	0	1	2	3	4	5	6	7	8...
$\theta = 0^0$													
$\theta = 45^0$													
$\theta = 90^0$													
$\theta = 135^0$													

[2] $\cos 45^\circ = \sin 45^\circ = \sin 135^\circ = 1/\sqrt{2}$

2.3 Morphologie

2.3.1 Beispiel 1 - Diskrete Geraden und Morphologie (siehe Kapitel G)

1. Diskrete Geraden entstehen durch Abtastung und werden mit Freeman Chain Codes (FCC) (siehe Kapitel H) beschrieben. Dieser FCC besteht aus maximal zwei verschiedenen und aufeinanderfolgenden Codes $c, d, |d - c| = 1$. Nur der Code c kann wiederholt auftreten: $c^n d c^n d \ldots$.

2. Wählen Sie c, d als jene zwei aufeinanderfolgenden Ziffern Ihrer Matrikelnummer M, die für kleinstes k die folgende Bedingung erfüllt: $M_i < 7$ und $|M_{i+1} - M_i| = k, k = 1, 2 \ldots$.
   $c = M_i = \square$, $d = M_i + (M_{i+1} - M_i)/k = \square$.

3. Den Startpixel tragen Sie im Bild so ein (•), dass die Gerade das Bildfenster quert. Markieren Sie dann jene Pixel der diskreten Geraden $c^3 d c^3 d \ldots$ im Bild durch •, die nach dem Code d erreicht werden.

4. Mit welchem morphologischen Operator und welchem Strukturelement (SE_1) werden die Lücken der Gerade geschlossen? $B_1 = \boxed{}\,(B, SE_1)$, markiere Pixel mit ∘.

5. Mit welchem morphologischen Operator und welchem SE_2 werden danach falsche Linienstücke[3] eliminiert? $B_2 = \boxed{}\,(B_1, SE_2)$, markiere gelöschte Pixel mit X.

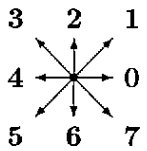
$B =$

$y = 12$												
$y = 11$												
$y = 10$												
$y = 9$												
$y = 8$												
$y = 7$												
$y = 6$												
$y = 5$												
$y = 4$										•		
$y = 3$												
$y = 2$												
$y = 1$												
$y = 0$												
$x =$	0	1	2	3	4	5	6	7	8	9	10	11	12

$SE_1 =$

$SE_2 =$

FCC



[3] andere Richtungen, Noise

2.3.2 Beispiel 2 - Morphologie und Zusammenhang (siehe Kapitel G)

Ein Binärbild der Grösse 8×10 wird aus dem Histogramm der Ziffern Ihrer Matrikelnummer bestimmt. Füllen Sie in die Kästchen der folgenden Matrix die Häufigkeiten der Ziffern Ihrer Matrikelnummer ein:

Ziffer	0	1	2	3	4	5	6	7	8	9
h(.)										

Sei h_{max} der größte Wert in $h(.)$. Dann ergibt der gerundete Wert $\lambda = [\frac{6}{h_{max}}] = \square$ den Vergrösserungsfaktor, mit dem die Häufigkeit multipliziert wird, bevor sie als Balkendiagramm in der unten stehende Bildmatrix mit $\bullet$ eingetragen wird. $B(x, y) = \bullet$ für Ziffer $x \in [0, 9]$ und $y = 1, \ldots \lambda * h(x)$, wenn $h(x) > 0$.

Diese Binärmatrix öffnen Sie mit folgendem Strukturelement: $\begin{array}{|c|c|c|}\hline & o & \\ \hline o & \bullet & o \\ \hline \end{array}$ (OPEN) und markieren die neuen 1-Pixel (nach der Dilatation) mit einem o. Jedes Element der Matrix, das nach der Erosion verschwindet, wird mit x gekennzeichnet. Im Ergebnis scheinen danach folgende Zeichen auf: $\square$, $o, \bullet$. Die mit x gelöschten Pixel zählen als 0. Für den Bildrand verwenden Sie bitte den zyklischen Abschluss.

$y \uparrow$											
7											
6											
5											
4											
3											
2											
1											
0	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	
	0	1	2	3	4	5	6	7	8	9	$\rightarrow x$

$y \uparrow$											
7											
6											
5											
4											
3											
2											
1											
0	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	
	0	1	2	3	4	5	6	7	8	9	$\rightarrow x$

Wieviele 4-Zusammenhangskomponenten (OHNE zyklischen Abschluss) hat das geöffnete Binärbild (nur o und $\bullet$)? $CC_4 = \square$.

Bestimmen Sie die Co-Occurence Matrix der 3 auftretenden Symbole $\square$, $\bullet, o$ für die Verschiebung $\delta = (1, 1)$ (wieder bei zyklischem Abschluss, $|N_\delta| = 80$!):

$P_\delta =$

	$\square$	$\bullet$	o	Σ
$\square$	/80	/80	/80	/80
$\bullet$	/80	/80	/80	/80
o	/80	/80	/80	/80
Σ	/80	/80	/80	/80

2.3.3 Beispiel 3 - Morphologie und Zusammenhang (siehe Kapitel G)

Bezeichne $M_1, M_2, \ldots M_7$ die Ziffern Ihrer Matrikelnummer. Füllen Sie in die Kästchen der folgenden Matrix die Ziffern Ihrer Matrikelnummer ein:

y	1	2	3	4	5	6	7
x							

1. Diese 7 Pixel kennzeichnen Sie bitte in der unten stehende Bildmatrix mit •.
2. Diese Binärmatrix dilatieren Sie mit folgendem Strukturelement: $\begin{array}{|c|c|}\hline x & x \\ \hline \bullet & x \\ \hline\end{array}$ und markieren die neuen 1-Pixel mit einem x. Jedes Element der Matrix hat danach eines der Zeichen □ , $X, \bullet$ Für den Bildrand verwenden Sie bitte den zyklischen Abschluss.

$y \uparrow$										
7										
6										
5										
4										
3										
2										
1										
	0	1	2	3	4	5	6	7	8	9 $\rightarrow x$

3. Wieviele 4-Zusammenhangskomponenten (OHNE zyklischen Abschluss) hat das dilatierte Binärbild (nur x und •)? $CC_4 =$ □ .
4. Bestimmen Sie die Co-Occurence Matrix der 3 auftretenden Symbole □ , •, x für die Verschiebung $\delta = (1, 1)$ (wieder bei zyklischem Abschluss, $|N_\delta| = 70!$):

$P_\delta =$

	□	•	x	Σ
□	/70	/70	/70	/70
•	/70	/70	/70	/70
x	/70	/70	/70	/70
Σ	/70	/70	/70	/70

2.4 Distanztransformation

2.4.1 Beispiel 1 - Distanztranformation auswerten (siehe Kapitel G)

In diesem Abschnitt sollen Sie einfache Bildverarbeitungsoperationen numerisch nachvollziehen.

1. Tragen Sie bitte in die folgende 8×10 Matrix d_{23} in Zeile 0, Spalte M (M ist die letzte Ziffer Ihrer Matrikelnummer) $d_{23}(0, M) = 0$ ein.

2. Füllen Sie diese Matrix mit den Distanzwerten der 2/3-Metrik d_{23} vollständig auf.

d_{23}	**diskrete 2/3 Distanzen**									
7										
6										
5										
4										
3										
2										
1										
Start → 0										
	0	1	2	3	4	5	6	7	8	9

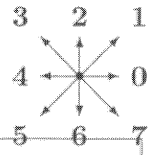
3. Bestimmen Sie dann das **Histogramm** (d_{23}) dieser Matrix (siehe Kapitel C):

1	0													
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

15	16	17	18	19	20	21	22	23	24	25

4. Daraus läßt sich der Median bestimmen: ☐

5. Zuletzt bestimme den Freeman chain code des diskreten Kreises mit Radius = Median. Um Löcher zu schliessen, werden auch um 1 größere Distanzwerte verwendet (siehe Kapitel H):



☐

2.4.2 Beispiel 2 - 4-Distanzen mit Hindernis (siehe Kapitel G)

In diesem Teil sollen Sie einfache Bildverarbeitungsoperationen numerisch nachvollziehen.

1. Tragen Sie bitte in der folgenden Matrix in der linken oberen Ecke die letzte Ziffer M_7 Ihrer Matrikelnummer ein. Der Rest der Distanzmatrix ausser dem Pixel mit dem Wert 0 sei mit Unendlich (∞) initialisiert.

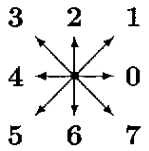
2. Füllen Sie die restliche Matrix mit den Distanzwerten der 4-Nachbarschaft auf, wobei über die 7 mit $\times$ markierten Pixel keine Distanzen propagiert werden.

				×				
				×				
				×				
×	×	×	×	×				
0								

3. Bestimmen Sie dann das **Histogramm** dieser Matrix (siehe Kapitel C):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

4. Zuletzt bestimme jenen 8-Weg (siehe Kapitel G) vom linken oberen Eckpunkt (wo die letzte Ziffer der Matrikelnummer steht,) bis zum gegenüberliegenden rechten unteren Eckpunkt, sodass die Summe der Distanzwerte minimal ist. Beschreibe den resultierenden Weg mit dem Freeman chain code (siehe Kapitel H).



5. Die Summe der Distanzen des optimalen Weges betragen

2.4.3 Beispiel 3 - Minimaler 8-Pfad durch 4-Distanzen mit Hindernis (siehe Kapitel G)

1. Tragen Sie bitte in der folgenden Matrix in der linken oberen Ecke die letzte Ziffer M_7 Ihrer Matrikelnummer ein. Der Rest der Distanzmatrix ausser dem Pixel mit dem Wert 0 sei mit Unendlich (∞) initialisiert.

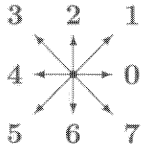
2. Füllen Sie die restliche Matrix mit den Distanzwerten der 4-Nachbarschaft auf, wobei über die 7 mit $\times$ markierten Pixel keine Distanzen propagiert werden.

								0
				$\times$				
				$\times$				
				$\times$				
$\times$	$\times$	$\times$	$\times$	$\times$				

3. Bestimmen Sie dann das **Histogramm** dieser Matrix (siehe Kapitel C):

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

4. Zuletzt bestimme jenen 8-Weg (siehe Kapitel G) vom linken oberen Eckpunkt (wo die letzte Ziffer der Matrikelnummer steht,) bis zum gegenüberliegenden rechten unteren Eckpunkt, sodass die Summe der Distanzwerte minimal ist. Beschreibe den resultierenden Weg mit dem Freeman chain code (siehe Kapitel H).



5. Die Summe der Distanzen des optimalen Weges betragen

2.5 Pyramide

2.5.1 Beispiel 1 - Medianpyramide (siehe Kapitel I)

1. Die Häufigkeit der Grauwerte 1 bis 7 der Grundebene Med_0 wird durch Ziffern Ihrer Matrikelnummer bestimmt: $H_0(g) = M_g, g = 1, 2, \ldots 7$. Der Rest der 7×7-Grundebene ist mit 0 aufgefüllt. (Histogramm siehe Kapitel C)

$g =$	0	1	2	3	4	5	6	7
$H_0(g) =$								
$K_0(g) =$								49
$H_1(g) =$								
$CC(g+0.5) =$								x

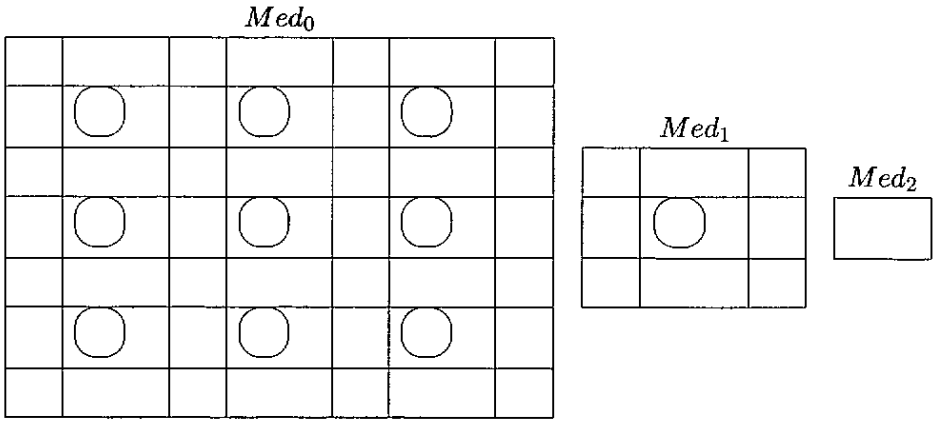
2. Zur Bestimmung des Median dient das kumulative Histogramm $K_0(g) = \sum_{i=0}^{g} H_0(i)$.

3. Das zentrale Element von Med_0, Med_1, Med_2 wird mit dem Median aller 49 Pixel der Grundebene Med_0 gefüllt.

4. Das Histogramm der Ebene 1 der $3 \times 3/4$ Medianpyramide wird durch $H_1(g) \approx [H_0(g)/5]$ ermittelt. Beachte, dass 'ausgewogen' gerundet wird und die Gesamtpixelanzahl der Ebene 1 genau 9 sein und der Median vorkommen muss!

5. Als Reduktionsfunktion soll der Median des 3×3 Reduktionsfensters dienen.

6. Verteile die Grauwerte der Histogramme H_0, H_1 so auf die Ebenen der Medianpyramide, dass
   - der Median jedes 3×3 Reduktionsfensters genau dem Elternpixel entspricht;
   - gleiche Grauwerte innerhalb jeder Ebene möglichst eine einfach 4-zusammenhängende Region bilden (siehe Kapitel G).



7. Bestimme die Anzahl der 4-Zusammenhangskomponenten $CC(t) = |\{Med_0(x,y) < t\}|$ für $t = 0.5, 1.5, \ldots 6.5$ (in obiger Tabelle 1. in Zeile $CC(g+0.5)$ eintragen) (siehe Kapitel G).

2.5.2 Beispiel 2 - Mittelwertpyramide (siehe Kapitel I)

1. Tragen Sie in die durch ○ gekennzeichneten Felder der Pyramide P spaltenweise die Ziffern Ihrer Matrikelnummer ein.

2. Ergänzen Sie die restlichen Felder der $2 \times 2/4$ Mittelwertpyramide und der Reduce-Expand-Laplace-Pyramide mit ganzen Zahlen so, dass für alle Elemente von P die Reduktionsfunktion erfüllt ist und in L betragsmässig möglichst kleine Werte stehen. Als Expand-Funktion verwenden Sie dabei eine einfache Projektion .

$P =$

M_1	M_2	M_3	M_4
		○	
○			

M_5	M_6
○	

M_7
○

$L = E(P) - P =$

	○		
			○

	○

Begründung: .....

.....

.....

.....

.....

.....

.....

2.5.3 Beispiel 3 - Laplacepyramide (siehe Kapitel I)

1. Die Grundebene L_0 einer $2 \times 2/4$ Laplacepyramide ist gegeben.
2. Die Ebenen $L_1 = R(L_0)$ und $L_2 = R(L_1)$ werden aus der jeweilig darunter liegenden Ebene mit folgender Reduktionsfunktion bestimmt: $R\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \lfloor \frac{a+b+c+d}{3} \rfloor$. Rationale Quotienten werden abgerundet. In der Spitze L_3 der Pyramide steht der Wert 6.

$L_0 =$

0	0	0	0	1	1	1	1	0
1	1	1	1	1	1	2	1	0
1	2	2	1	1	2	2	1	0
1	2	2	1	1	1	1	1	0
1	1	1	3	3	1	1	1	0
1	1	1	1	1	2	2	1	0
1	2	2	1	1	2	2	1	0
1	2	1	1	1	1	1	1	0
1	1	1	1	0	0	0	0	0

$L_1 =$

$L_2 =$

$L_3 = \boxed{6}$

3. Diese 4 Ebenen bilden eine Laplace Pyramide, aus der Sie die zugrundeliegende Grauwertpyramide mit folgender Expansionsfunktion rekonstruieren: $E(x) = \begin{pmatrix} x-2 & x-2 \\ x-2 & x-2 \end{pmatrix}$.

$G_0 =$

$G_1 =$

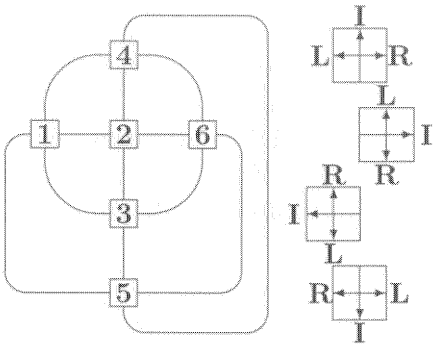
$G_2 =$

$G_3 = \boxed{}$

2.6 Pfade mit Würfeln

2.6.1 Beispiel 1 - Minimaler Pfad (siehe Kapitel H)

Ein Würfel wird im linken unteren Eck eines 3×10 Bildrasters gelegt. Die oben aufscheinende Augenzahl wird in dem darunterliegende Pixel eingetragen. Durch Kippen des Würfels über eine seiner 4 Kanten kommt der Würfel auf dem jeweilig benachbarten Pixel zu liegen. Auch hier wird die nun oben liegende Augenzahl im Pixel eingetragen. Durch wiederholtes Kippen wird ein Pfad von 4-benachbarten Pixeln mit den Augenzahlen des Würfels (1,2, ..., 6) gefüllt.



Die Struktur des Würfels ist durch seinen Graphen festgelegt, wobei nur die jeweils oben liegende Augenzahl sichtbar ist und ein einmaliges Kippen durch eine Kante ausgedrückt wird. Eine Folge von Kippbewegungen (z.B. 3,2,4,5,6,3) wird durch eine RULI-chain beschrieben. Das Beispiel startet mit 3 ostwärts: I^3RL korrespondiert

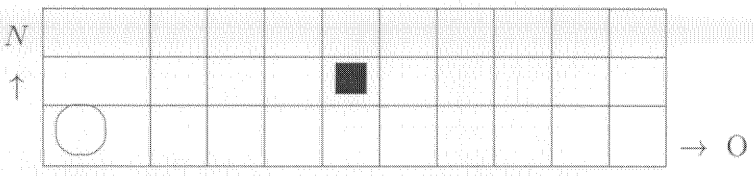
3	2	4	5	
			6	3

. Entsprechende Bewegungen (L ...links, R...rechts, I...in dieselbe Richtung) erfolgen auch auf dem Pixelraster.

1. Platziere einen Würfel mit der Augenzahl $\text{argmin}\{M_i | i = 1, 2, 3, ...6\} =$ ○ (bei Mehrdeutigkeiten die vordere Ziffer) im linken unteren Eck des 3×10 Feldes und trage die Augenzahl im markierten Feld ein. Von hier starten Pfade in zwei Richtungen N und O.

2. Sie haben die Aufgabe, jede der 29 restlichen Positionen durch schrittweises Kippen des Würfels und Eintragen der oben liegenden Augenzahl zu füllen. Alle Pfade beginnen im linken unteren Feld und führen jeweils zu Pixeln, die weiter rechts (O) oder weiter oben (N) liegen. Der durch ■ gekennzeichnete Pixel wird ausgelassen. Jeder Pixel oberhalb und rechts des Startpixels hat in seinen Pfaden ein oder zwei Vorgänger!

3. Im Falle von zwei Vorgängern wird nur jener Pfad fortgesetzt, der im Pixel die **kleinere** Augenzahl speichert. Die Lösung wird erleichtert, wenn die Kante des unterbrochenen Pfades markiert wird.



4. Auf welchem Pfad wird der Würfel vom Startpunkt zum rechten oberen Pixel bewegt?

Folge der Augenzahlen: .....

korrespondierende RLI-chain: .....

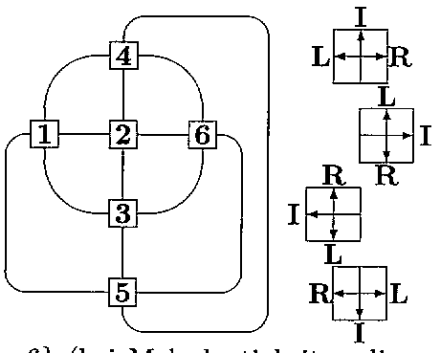
.....

.....

2.6.2 Beispiel 2 - Maximaler Pfad (siehe Kapitel H)

Ein Würfel wird im linken unteren Eck eines 5×5 Bildrasters gelegt. Die oben aufscheinende Augenzahl wird in dem darunterliegende Pixel eingetragen. Durch Kippen des Würfels über eine seiner 4 Kanten kommt der Würfel auf dem jeweilig benachbarten Pixel zu liegen. Auch hier wird die nun oben liegende Augenzahl im Pixel eingetragen. Durch wiederholtes Kippen wird ein Pfad von 4-benachbarten Pixeln mit den Augenzahlen des Würfels (1,2, ..., 6) gefüllt.

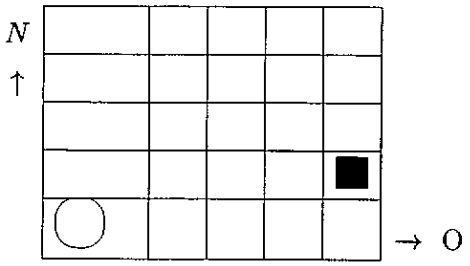
Die Struktur des Würfels ist durch seinen Graphen festgelegt, wobei nur die jeweils oben liegende Augenzahl sichtbar ist und ein einmaliges Kippen durch eine Kante ausgedrückt wird. Eine Folge von Kippbewegungen (z.B. 3,2,4,5,6,3) wird durch eine RULI-chain beschrieben: N 3: I^3RL. Entsprechende Bewegungen (L ...links, R...rechts, I...in dieselbe Richtung) erfolgen auch auf dem Pixelraster.



1. Platziere einen Würfel mit der Augenzahl $\text{argmax}\{M_i | i = 1, 2, 3, ...6\}$ (bei Mehrdeutigkeiten die vordere Ziffer) im linken unteren Eck des 5×5 Feldes und trage die Augenzahl im markierten Feld ein. Von hier starten Pfade in zwei Richtungen N und O.

2. Sie haben die Aufgabe, jede der 23 restlichen Positionen durch schrittweises Kippen des Würfels und Eintragen der oben liegenden Augenzahl zu füllen. Alle Pfade beginnen im linken unteren Feld und führen jeweils zu Pixeln, die weiter rechts (O) oder weiter oben (N) liegen. Der durch ■ gekennzeichnete Pixel wird ausgelassen. Jeder Pixel oberhalb und rechts des Startpixels hat in seinen Pfaden ein oder zwei Vorgänger!

3. Im Falle von zwei Vorgängern wird nur jener Pfad fortgesetzt, der im Pixel die grössere Augenzahl speichert.



4. Auf welchem Pfad wird der Würfel vom Startpunkt zum rechten oberen Pixel bewegt?

Folge der Augenzahlen: .....

RLI-chain: .....

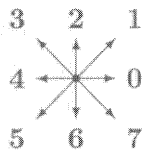
.....

.....

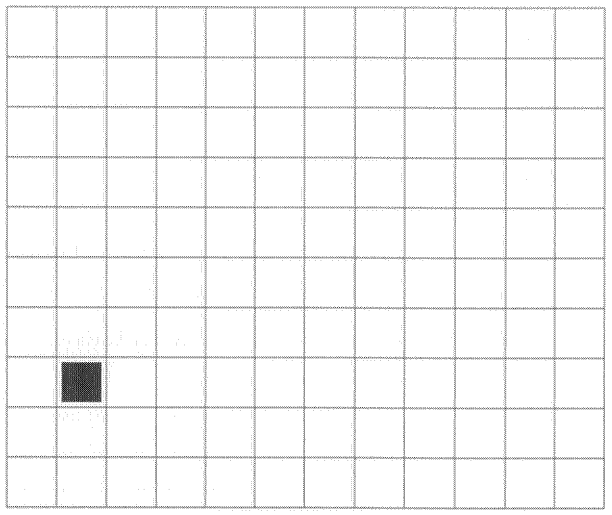
2.7 Freeman Chain Code

2.7.1 Beispiel 1 - Digitale Kurven abbilden

Eine digitale geschlossene Kurve ist durch den Freeman Chaincode (siehe Kapitel H) gegeben. Wählen Sie die Kurve (1.), wenn die mittlere Ziffer Ihrer Matrikelnummer grösser als 5 ist, sonst die zweite Kurve (2.):



1. 2220000000222444444477777774444442
2. 222000000022 444444 777 666 44442
3. Markieren Sie bitte in der folgenden Matrix ausgehend vom Pixel ■ die Pixel der entsprechenden digitalen Kurve.



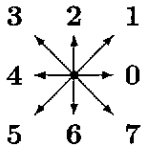
4. Ist das binäre Bild wohlgeformt (siehe Kapitel G)? Ja / Nein
5. Markieren Sie im obigen Bild die Zusammenhangskomponenten $Z_4(\overline{\text{8-Kurve}})$, die nicht Hintergrund sind, mit den Marken $1, 2, \ldots$. (siehe Kapitel G)
6. Bestimmen Sie die Grösse der Zusammenhangskomponenten (siehe Kapitel G):

Marke	Anzahl Pixel
1	
2	
3	
4	

2.7.2 Beispiel 2 - Durch FCC gegebene Region erodieren

Bezeichne $M_1, M_2, \ldots M_7$ die Ziffern Ihrer Matrikelnummer. Füllen Sie die Kästchen der folgenden 4 Zeilen entsprechend der jeweils rechts angegebenen Rechenvorschriften. In der ersten Zeile fügen Sie den Wert 0 ein, wenn die Ziffern 1, 3, 5 und 7 Ihrer Matrikelnummer gerade sind, den Wert 1, wenn sie ungerade sind. In die jeweils darunter liegende Zeile wird der um 2 erhöhte Wert der darüberliegenden Zeile eingefügt:

	M_1		M_3		M_5		M_7	
Zeile 1	□	→	□	→	□	→	□	$= mod(M_i, 2)$
							↓	
Zeile 2	□	←	□	←	□	←	□	= Zeile 1 + 2
	↓							
Zeile 3	□	→	□	→	□	→	□	= Zeile 2 + 2
							↓	
Zeile 4	□	←	□	←	□	←	□	= Zeile 3 + 2



1. Diese 16 Zahlen ergeben in der Reihenfolge der Pfeile einen Freeman Chaincode (siehe Kapitel H), den Sie grafisch im folgenden Raster eintragen.

2. Verbinden Sie ausgehend vom gekennzeichneten Quadrat die durch den Freeman Chaincode definierten +Pixel:

+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	+	+	+	+	+
+	+	+	+	$\boxed{+}$	+	+	+	+

3. Erodieren Sie die vom Freeman Chaincode umgebene Region mit folgendem Strukturelement (siehe Kapitel G): $\boxed{\begin{matrix} + & + & + \\ + & \oplus & + \\ + & + & + \end{matrix}}$ und markieren Sie die verbleibenden Pixel mit einem Kreis.

4. Wieviele 4-Zusammenhangskomponenten hat das erodierte Binärbild (siehe Kapitel G)?
$CC_4 =$ □ .

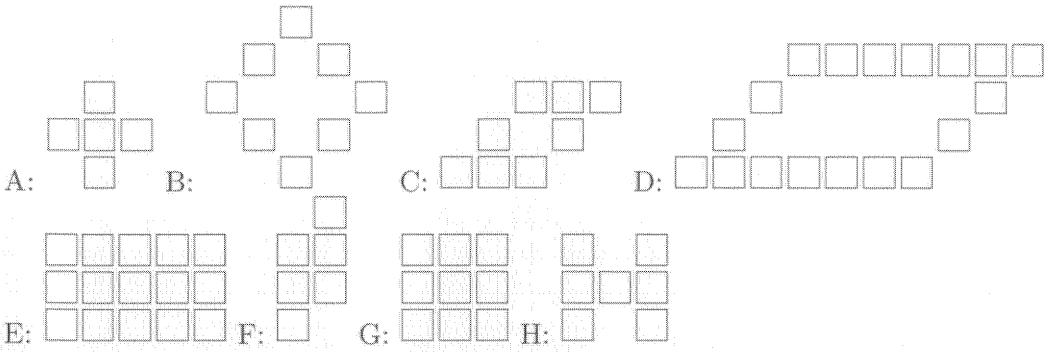
2.8 Array Grammatiken

2.8.1 Beispiel 1 - Formen von Klassen von FCC erkennen (siehe Kapitel H)

Die folgende Grammatik $G = (\{0,1,2,3,4,5,6,7\}, \{A,B,C,D,E,R,S,T\}, S, P)$ erzeugt eine Klasse von Freeman Chain Codes (FCC). Für die Einsetzung der Terminalsymbole (=FCC) gibt es drei Sätze von Regeln: I, II und III, von denen je eine gewählt werden kann. ϵ steht für den Nullstring: $\epsilon x = x\epsilon = x$.

P:	$S \to TRE$,	$R \to RR$,	$TE \to \epsilon$,	$R \to ABCD$,
	$BA \to AB$,	$CA \to AC$,	$DA \to AD$,	
	$CB \to BC$,	$DB \to BD$,	$DC \to CD$,	
I:	$AE \to E44$,	$BE \to E5$,	$CE \to E00$,	$DE \to E1$.
II:	$AE \to E5$,	$BE \to E7$,	$CE \to E1$,	$DE \to E3$.
III:	$AE \to E2$,	$BE \to E7$,	$CE \to E2$,	$DE \to E5$.

Welche der folgenden Figuren wird von der gewählten Grammatik erkannt?

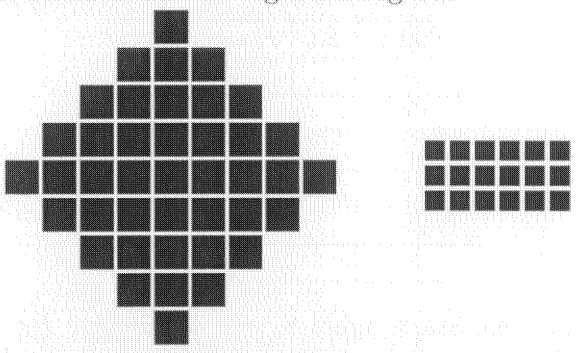


Welche Folge von Chaincodes wird von der Grammatik erzeugt (formale Sprache, z.B. $2^n4^n6^n0^n$)?

Markieren Sie den Startpunkt des erzeugten FCC.

2.8.2 Beispiel 2 - Grammatik für Klasse von Binärbildern bestimmen (siehe Kapitel H)

Bestimmen Sie eine Grammatik für die folgenden Figuren:



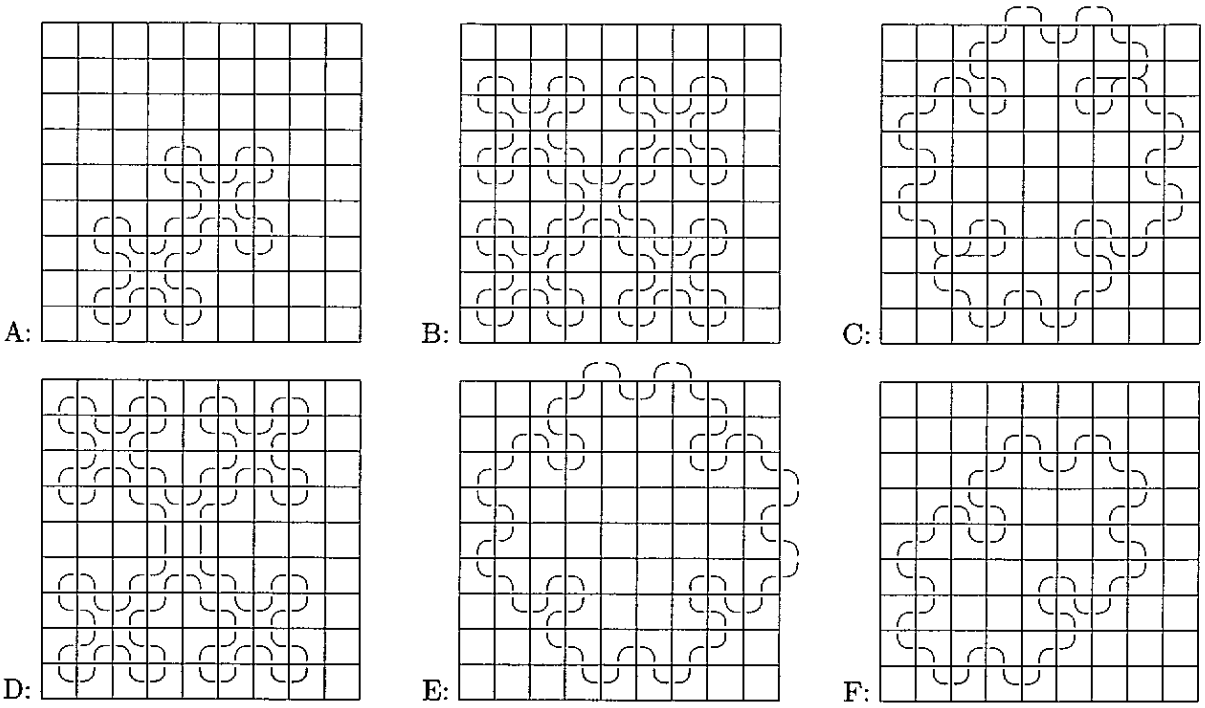
1. ein auf der Spitze stehendes Quadrat der Seitenlänge n, z.B. $n = 5$ (siehe Grafik links)
2. den Rand des vorigen Quadrats im Freeman Chaincode $1^n3^n5^n7^n$ mit einer Grammatik.
3. ein achsparalleles Rechteck mit Seitenverhältnis $1:2$ (Höhe in y-Richtung n, Breite in x-Richtung $2n$; z.B. $n = 3$ (siehe Grafik rechts)

2.8.3 Beispiel 3 - Klasse von RULI Chain Codes erkennen (siehe Kapitel H)

1. Die folgenden Grammatiken $G_m = (\{R_n, L_n, I_n\}, \{S_m, R_i, L_i, I_i | 0 \leq i < n\}, S, P), m = 0, 1, 2.$ erzeugen je eine Klasse von RULI Chain Codes (RCC) nach einer vorgegebenen Anzahl von Substitutionen n. Mit den drei Startsymbolen S_0, S_1, S_2 werden von jedem Satz von Produktionsregeln 3 Klassen von RULI chains verschiedener Länge erzeugt.

$P:$	$S_0 \to (R_0^3 L_0)^2$	$S_1 \to R_0^4$	$S_2 \to (R_0^2 I_0)^2$
$P_3:$	$R_{i-1} \to L_i R_i R_i;$	$P_4:$	$R_{i-1} \to L_i R_i R_i R_i L_i;$
	$L_{i-1} \to L_i;$		$L_{i-1} \to L_i R_i L_i;$
	$I_{i-1} \to I_i;$		$I_{i-1} \to I_i.$

2. Welche der folgenden Figuren wird von welchem Startsymbol und von welcher Grammatik in wieviel Ableitungsschritten erkannt?



Figur	S_m	P_k $n=?$
A		
B		
C		
D		
E		
F		

3. Ableitung der RULI-Chain mit dem Startsymbol S_0 und den Regeln P_3: $S_0 \Rightarrow$ ..... $\Rightarrow$

$\Rightarrow$ .....

$\Rightarrow$ .....

$\Rightarrow$ .....

2.9 Lauflängen, Zusammenhang, Coocurrence

2.9.1 Beispiel 1 - Lauflängenencodiertes Bild auswerten (siehe Kapitel H)

1. Bestimmen Sie bitte zu den lauflängencodierten Zeilen passende Grauwertbild. Als Wiederholungsfaktoren setzen Sie den Wert ein, der sich aus den Ziffern M_i Ihrer Matrikelnummer ergibt (siehe Kapitel H).

	RLC			Grauwertbild								
i	$(9-M_i)$	$\times g_i$,	M_i $\times g_i'$	1	2	3	4	5	6	7	8	9
1		$\times 3$,	$\times 2$									
2		$\times 2$,	$\times 3$									
3		$\times 1$,	$\times 0$									
4		$\times 0$,	$\times 1$									
5		$\times 3$,	$\times 2$									
6		$\times 2$,	$\times 3$									
7		$\times 1$,	$\times 0$									

2. Wieviel Zusammenhangskomponenten der 4-Nachbarschaft sind für die vier Grauwerte entstanden (siehe Kapitel G)?

$CC_4(0)$	$CC_4(1)$	$CC_4(2)$	$CC_4(3)$

3. Zuletzt bestimmen Sie bitte die Coocurrence Matrix zweier Pixel p, q des Grauwertbildes in folgender Lage (siehe Kapitel J):

q	
	p

:

		q				
		0	1	2	3	$\sum$
p	0					
	1					
	2					
	3					
	$\sum$					

Dabei ist ein zyklischer Abschluß von Zeilen und Spalten zu berücksichtigen.

Begründung: .....

.....

.....

.....

.....

2.9.2 Beispiel 2 - Lauflängenencodiertes Bild auswerten (siehe Kapitel H)

1. Bestimmen Sie bitte zu den lauflängencodierten Zeilen passende Grauwertbild. Als Wiederholungsfaktoren setzen Sie den Wert ein, der sich aus den Ziffern M_i Ihrer Matrikelnummer ergibt (siehe Kapitel H).

	RLC		Grauwertbild								
i	$(9-M_i)$ $\times\lvert i-4\rvert$,	M_i $\times(3-\lvert i-4\rvert)$	1	2	3	4	5	6	7	8	9
1	$\times 3$,	$\times 0$									
2	$\times 2$,	$\times 1$									
3	$\times 1$,	$\times 2$									
4	$\times 0$,	$\times 3$									
5	$\times 1$,	$\times 2$									
6	$\times 2$,	$\times 1$									
7	$\times 3$,	$\times 0$									

2. Wieviel Zusammenhangskomponenten der 4-Nachbarschaft sind für die vier Grauwerte entstanden (siehe Kapitel G)?

$CC_4(0)$	$CC_4(1)$	$CC_4(2)$	$CC_4(3)$

3. Zuletzt bestimmen Sie bitte die Coocurrence Matrix zweier Pixel p, q des Grauwertbildes in folgender Lage (siehe Kapitel J):

		q
p		

:

		q				
		0	1	2	3	$\sum$
p	0					
	1					
	2					
	3					
	$\sum$					

Dabei ist ein zyklischer Abschluß von Zeilen und Spalten zu berücksichtigen.

Begründung: .....

.....

.....

.....

.....

2.10 Integral Image und Predictive Coding

2.10.1 Beispiel 1 - Übertragen eines Integral Images mit Predictive Coding

Bezeichne $M_1, M_2, \ldots M_7$ die Ziffern Ihrer Matrikelnummer.

1. Markiere jene Pixel des Binärbildes mit 1, die durch die Ziffern Ihrer Matrikelnummer wie folgt bestimmt sind:
$$B(i, M_i) = B(M_i, i) = 1 \quad \text{für } i = 1, 2, \ldots, 7.$$
Die restlichen Pixel sind 0 und können leer gelassen werden.

2. Füllen Sie nun in die Kästchen die Werte des Intergral Images I(B) des Binärbildes B (siehe Kapitel J).

$B =$

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

$I(B) =$

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

3. Das Integral Image I soll mit einem linearen Prädiktor der Ordnung 3 mit folgenden Koeffizienten

$a_1 = -1$	$a_2 = +1$
$a_3 = +1$	$\tilde{B}(i,j)$

übertragen werden (siehe Kapitel J) (Array $\tilde{B}$ dient der Hilfestellung und muß nicht ausgefüllt werden!). Die Differenz $D(i,j) = I(i,j) - \tilde{B}(i,j)$ wird dann mit Lauflängenkodierung (Run lenght code) (siehe Kapitel H) spaltenweise übertragen. Geben Sie den Code jeder Spalte an:

$\tilde{B} =$

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

Spalte 0 : .....
Spalte 1 : .....
Spalte 2 : .....
Spalte 3 : .....
Spalte 4 : .....
Spalte 5 : .....
Spalte 6 : .....
Spalte 7 : .....
Spalte 8 : .....
Spalte 9 : .....

2.10.2 Beispiel 2 - Übertragen eines Integral Images mit Predictive Coding

1. Tragen Sie die Ziffern Ihrer Matrikelnummer $M_1, M_2, \ldots M_7$ in die folgende Tabelle und berechnen dann die Werte von S_i nach folgender Formel: $S_1 = M_1; S_{i+1} = S_i + M_{i+1} + 5$ für $i = 2, 3, 4, 5, 6, 7$.

i	1	2	3	4	5	6	7
M_i							
S_i							
$x_i = \lfloor S_i/10 \rfloor$							
$y_i = \mathrm{mod}(S_i, 10)$							

2. Daraus ergeben sich dann die Koordinaten jener Pixel (x_i, y_i), die Sie in folgender Matrix B mit 1 markieren. Die restlichen Pixel sind 0 und können leer gelassen werden.

3. Füllen Sie dann in die Kästchen die Werte des Intergral Images $I(B)$ des Binärbildes B (siehe Kapitel J).

$B =$

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

$I(B) =$

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

4. Das Integral Image I soll mit einem linearen Prädiktor der Ordnung 3 mit folgenden Koeffizienten

$a_1 = -1$	$a_2 = +1$
$a_3 = +1$	$\tilde{B}(i,j)$

übertragen werden (siehe Kapitel J) (Array $\tilde{B}$ dient der Hilfestellung und muß nicht ausgefüllt werden!). Die Differenz $D(i,j) = I(i,j) - \tilde{B}(i,j)$ wird dann mit Lauflängenkodierung (Run lenght code) (siehe Kapitel H) zeilenweise übertragen. Geben Sie den Code jeder Zeile an:

$\tilde{B} =$

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

Zeile 0 : .....
Zeile 1 : .....
Zeile 2 : .....
Zeile 3 : .....
Zeile 4 : .....
Zeile 5 : .....
Zeile 6 : .....
Zeile 7 : .....
Zeile 8 : .....
Zeile 9 : .....