

# Bildverarbeitung

## Morphologische Operationen

### Dilation (OR)

Das Bild wird auf alle true Elemente des strukturierenden Elementes verschoben (die Mitte des strukturierenden Elementes bildet dabei die Mittel selbst bzw. gerade links oberhalb).

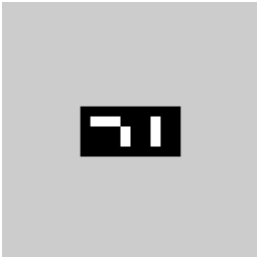
Danach werden all diese erzeugten Bilder mit dem OR Operator verknüpft. Der Rand wird immer mit einer 0 fortgesetzt!

### Erosion (AND)

Das Bild wird auf alle true Elemente des strukturierenden Elementes verschoben (die Mitte des strukturierenden Elementes bildet dabei die Mittel selbst bzw. gerade links oberhalb).

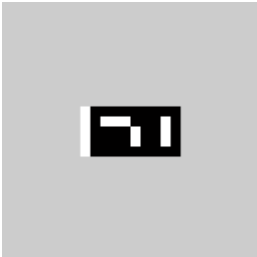
Danach werden all diese erzeugten Bilder mit dem OR Operator verknüpft. Der Rand wird immer mit einer 1 fortgesetzt!

Suche die links-unten Eckberührungen des folgenden Bildes:



Dazu nehmen wir folgendes strukturierendes Element:  $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

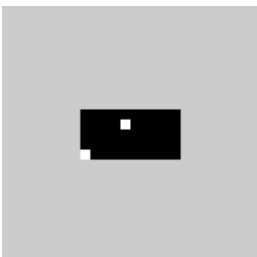
Also bilden wir folgende Verschiebungsbilder:



und



Wenn wir nun ein AND daraus machen ergibt sich folgendes Bild:



Da haben wir nun aber einen Fehler am Rand. Diesen dürfen wir einfach nicht berücksichtigen!

### Schliessen

Zuerst dilatieren, dann erodieren! Also zuerst OR, dann AND!

Es werden alle Löcher des Vordergrundes gestopft, die keinen innern Punkte haben, der „äussere Rand“ des Objektes wird dabei nicht wesentlich vergrössert. Es besteht aber die Gefahr, dass verschiedene Komponenten des Objektes beim Schliessen verklumpen.

## Öffnen

Zuerst erodieren, dann dilatieren! Also zuerst AND, dann OR!

Durch das Öffnen des Bildes werden insbesondere alle Komponenten aus dem Hintergrund entfernt, die keine inneren Punkte haben. Es besteht aber die Gefahr, dass das Objekt an einer engen Stelle eingeschnürt wird oder in zwei Komponenten zerfällt.

## Matlab

### Anwendungen

Anwendung	Beschreibung
<code>s=b(365/3:365/3*2,500/3:500/3*2,:);</code>	Wählt den mittleren Bildausschnitt eines in neun Teile aufgeteiltes Bild.
<code>s2=b(1:3:365,1:3:500,:);</code>	Verkleinert das Bild mit Unterabtastung
<code>imshow(b(:,:,x));</code>	x=1: rot / x=2: grün / x=3: blau
<code>zeros(365,500,3);</code>	Erzeugt eine Matrize aus lauter 0 mit der Dimension 365x500x3!
<code>cat(3,r,g,b)</code>	Erzeugt die dritte Dimension und legt dort die drei Übergabe Matrizen an. [r,g,b]
<code>ones(365,500,3)</code>	Erzeugt eine Matrize aus lauter 1 mit der Dimension 365x500x3!
<code>white=im2uint8(ones(365,500,3)*255); n=imsubtract(white,b);</code>	Erzeugt das Negativ Farbbild!
<code>imshow(gray,8);</code>	Graubild mit reduzierter Anzahl Graustufen (hier 8)!
<code>imshow(b); map=COLORMAP; imshow(gray,map);</code>	Die Colormap kann verändert werden! So kann diese ausgelesen werden!
<code>g1=bitand(gray,bin2dec('00000001'));</code>	So wird aus einem Graubild das erste Bit ausgelesen!
<code>gm1=bitand(gray,bin2dec('11111110')); gm2=bitand(gm1,bin2dec('11111100'));</code>	So kann eine Bildebene gelöscht werden!
<code>gn=imcomplement(gray); oder gn=bitxor(gray,bin2dec('11111111'));</code>	Negativbild eines Grauwert Bildes
<code>bitshift(g7,1);</code>	Shiftet die Bits 1 nach links
<code>imadd(gray,x)</code>	Hellet ein Bild auf (x=0..255)
<code>matrix=repmat((0:255),256,1); matrix=uint8(matrix);</code>	Erzeugt eine Matrix, in der das Bild von links nach rechts an Intensivität gewinnt.
<code>g1=rot90(g)</code>	Rotiert ein Bild 3x Mal um 90° und füge die

Anwendung	Beschreibung
<pre>g2=rot90(g1); g3=rot90(g2); gall=[g g1];[g3 g2]]</pre>	Bilder dann wie ein Puzzle zusammen!
<pre>whitematrix=ones(256,256,3)*255; workingmatrix= repmat((0:255),256,1); redmatrix=whitematrix; redmatrix(:,:,3)=workingmatrix; redmatrix(:,:,2)=workingmatrix; imshow(uint8(redmatrix));</pre>	Macht einen farbigen Bildverlauf!
<pre>min(min(g))</pre>	Minimalwert eine Grauwertbildes
<pre>im2uint8(imadjust(im2double(bnew(:,:,1) ),[0.3 0.7],[1]));</pre>	Alle Werte werden Proportional zwischen 0.3 und 0.7 eingeschoben!
<pre>x = [0 500]; y = [182 182]; improfile(b,x,y)</pre>	Farbwertprofil von Spalte 0-500 auf der Zeile 182!
<pre>function y=whiten(b,h,height)     tmp = b;     h=repmat(h,height,1);     y=imadd(tmp,uint8(h));</pre>	Diese Funktion hellt den Wert einer Matrize von links nach rechts auf!
<pre>tmp = b; x=255; z=uint8(rand(365,500)*x)</pre>	Rauschen!
<pre>m1=repmat((( [1:365] )/4) ',1,500); m2=repmat((( [1:500] )/4) ',365,1); m=m1+m2; tmp=b; tmp(:,:,2)=imadd(tmp(:,:,1),uint8(m)); imshow(tmp);</pre>	Hellet den Grünwert von links oben nach rechts unten auf!
<pre>b=imread('uhr.jpg'); improfile(b,[0 800], [300 300] ); g= b(:,:,1)&gt;190 &amp; b(:,:,2)&gt;190 &amp; b(:,:,3)&gt;190; imshow(g); sum(sum(g));</pre>	Erzeugt ein Binärbild und bestimmt die Fläche!
<pre>[y,x]=find(g&gt;0); dy = mean(y) dx = mean(x)</pre>	Findet den Schwerpunkt eines Bildes.
<pre>gou=abs(diff(im2double(g))); grl=abs(diff(im2double(g),1,2)); gou=gou(1:599,1:799); grl=grl(1:599,1:799); gourl=or(gou,grl);</pre>	Extrahiert die oberen und unteren Randpixel
<pre>mx=repmat(1:500,[500 1]); my=repmat((1:500)',[1 500]); mx=mx-250; my=my-250; tmp=mx.^2+my.^2&lt;10000;</pre>	Erzeugt eine Kreisscheibe!

## Befehle

Matlab Befehl	Beschreibung/Beispiel
imread	<code>b=imread('fct_stgallen1195.jpg');</code>
imshow	<code>imshow(b)</code>
subplot(spalten,zeilen,index)	<code>subplot(2,2,1), imshow(b(:,:,1));</code>

<b>Matlab Befehl</b>	<b>Beschreibung/Beispiel</b>
rgb2gray	rgb2gray(b)
imhist(b)	Histogramm
histseq(b)	Grauwertausgleich
imadjust	Farbwertausgleich