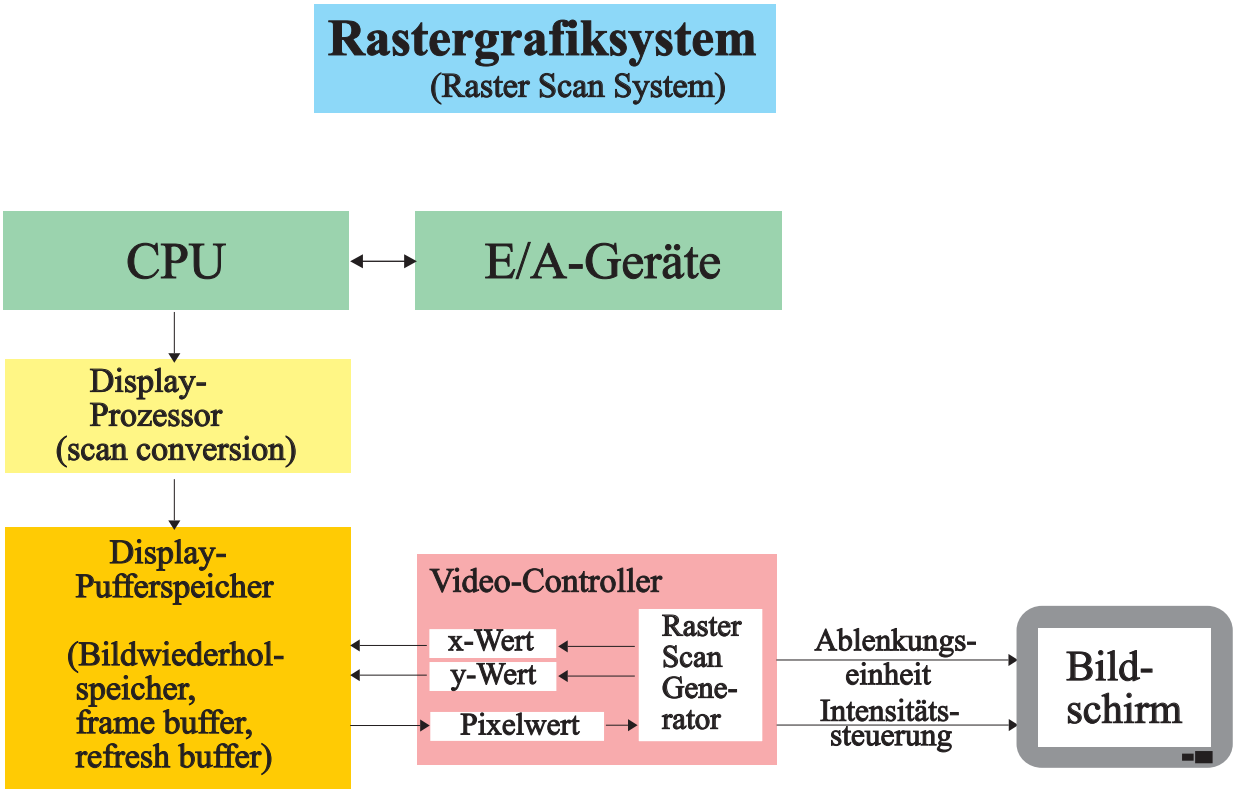
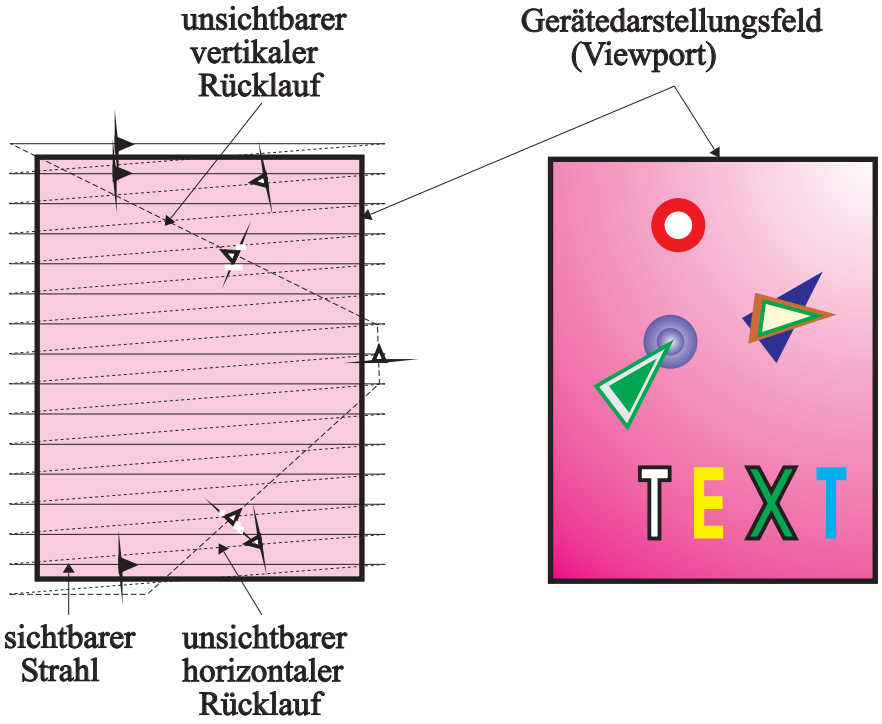
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Rastergrafik-Algorithmen

1. Darstellung von Linien, Polygonen und Kurven
2. Clippen
3. Füllen von Polygonen und geschlossenen Kurven
4. Farbmodelle, Farbtabellen, Halbtonverfahren
5. Minderung von Aliasing-Effekten

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder





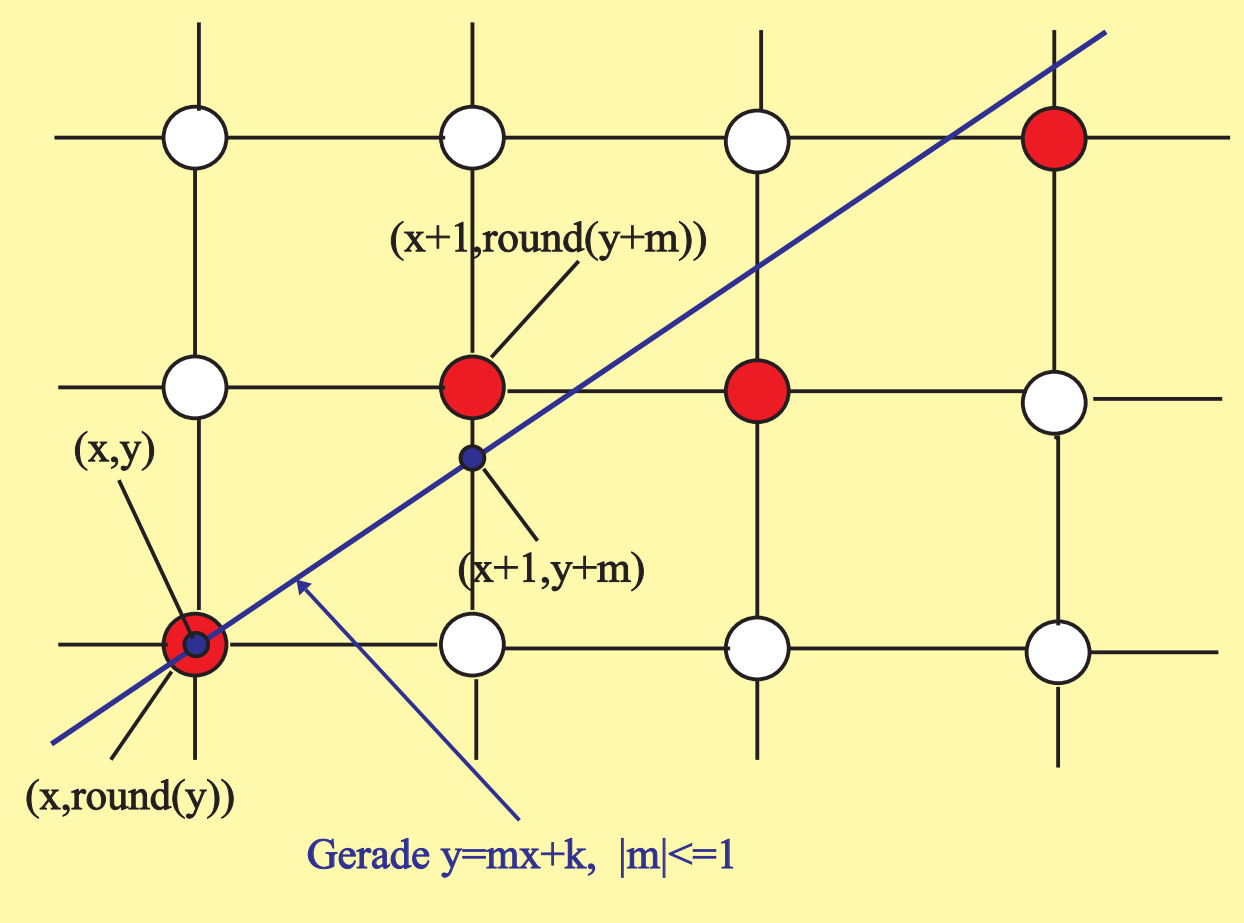
Rastergrafik-Algorithmen

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

1. Darstellung von Linien, Polygonen und Kurven

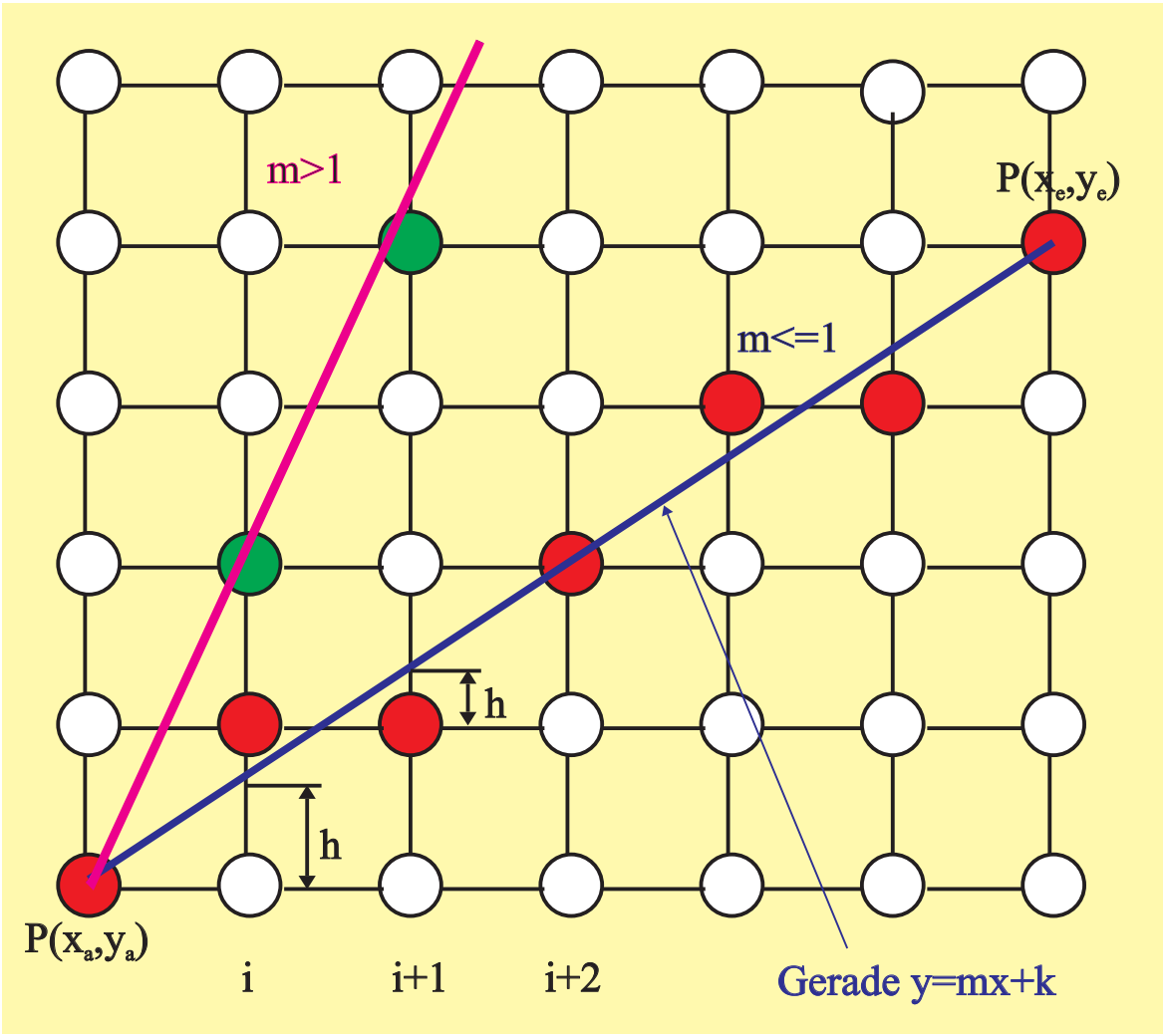
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Liniendarstellung
Inkrementeller Algorithmus



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Bresenham-Algorithmus (1)



Bresenham-Algorithmus (2)

```
procedure Bresenham(xA,yA,xB,yB,Color: Word);
  var dx,dy,R,C,G,K1,K2: integer;
begin
  dx:=xB-xA;   dy:=yB-yA;
  K1:=2*dy;    K2:=2*(dy-dx);

  G:=2*dy-dx;  C:=xA;  R:=yA;
  while C<=xB do
  begin
    PutPixel(C,R,Color);
    C:=C+1;
    if G>0 then begin R:=R+1; G:=G+K2 end
           else G:=G+K1
  end
end {Bresenham};
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Bresenham-Algorithmus (3)

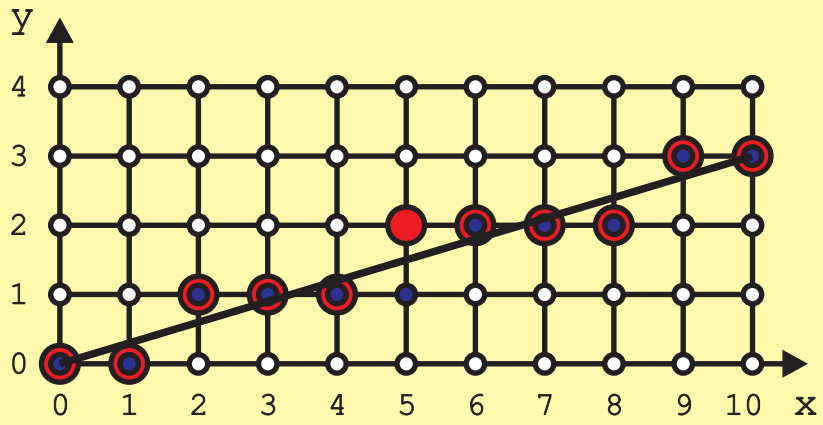
$x_A = 0 \quad y_A = 0$
$x_B = 10 \quad y_B = 3$

Test für den Zeilenwechsel: G>0

C(=x_i)	0	1	2	3	4	5	6	7	8	9	10
R(=y_i)	0	0	1	1	1	1	2	2	2	3	3
G m	-4	2	-12	-6	0	6	-8	-2	4	-10	-4

Test für den Zeilenwechsel: G>=0

C(=x_i)	0	1	2	3	4	5	6	7	8	9	10
R(=y_i)	0	0	1	1	1	2	2	2	2	3	3
G m	-4	2	-12	-6	0	-14	-8	-2	4	-10	-4

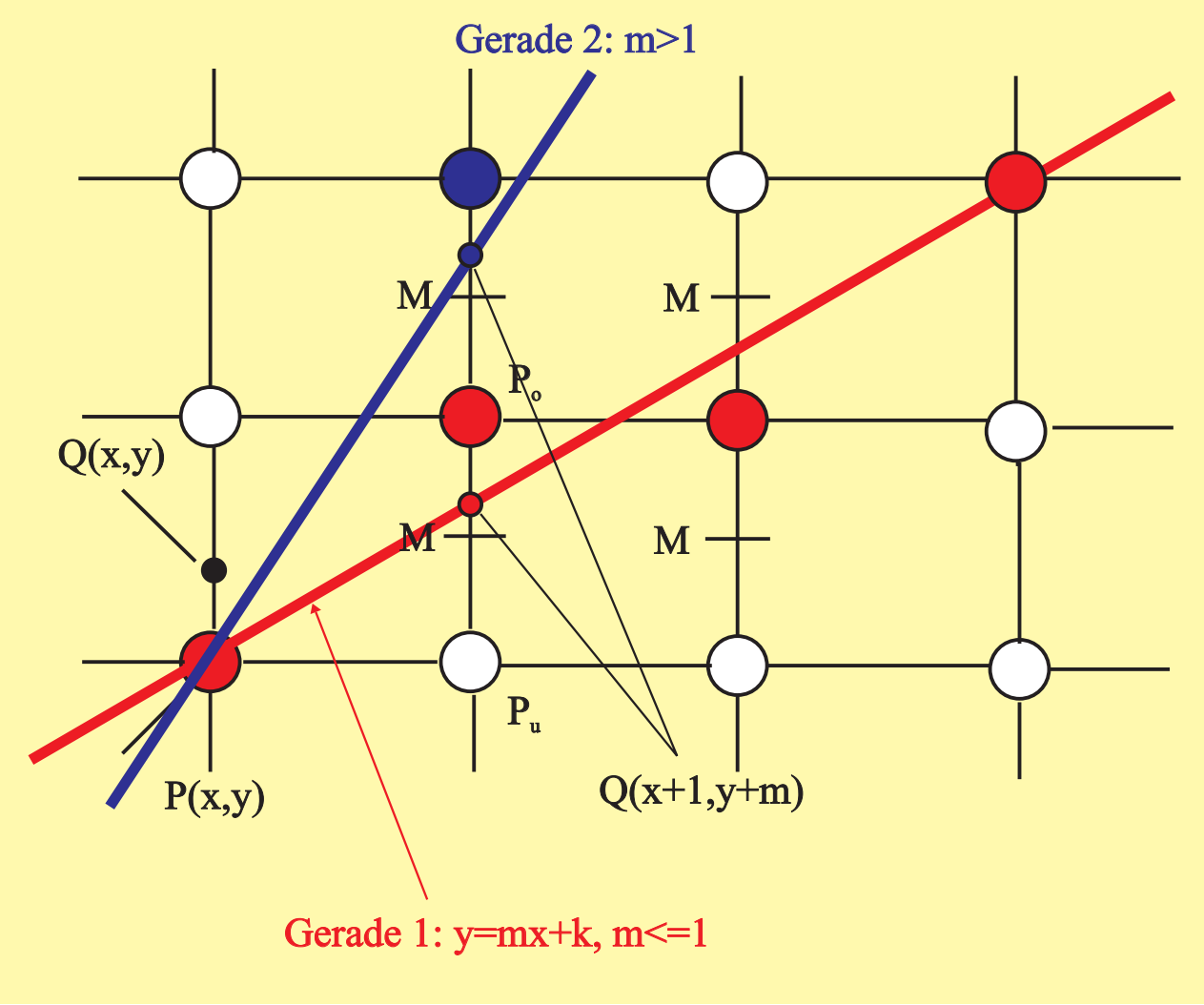


Punktauswahl beim Test G>0

Punktauswahl beim Test G>=0

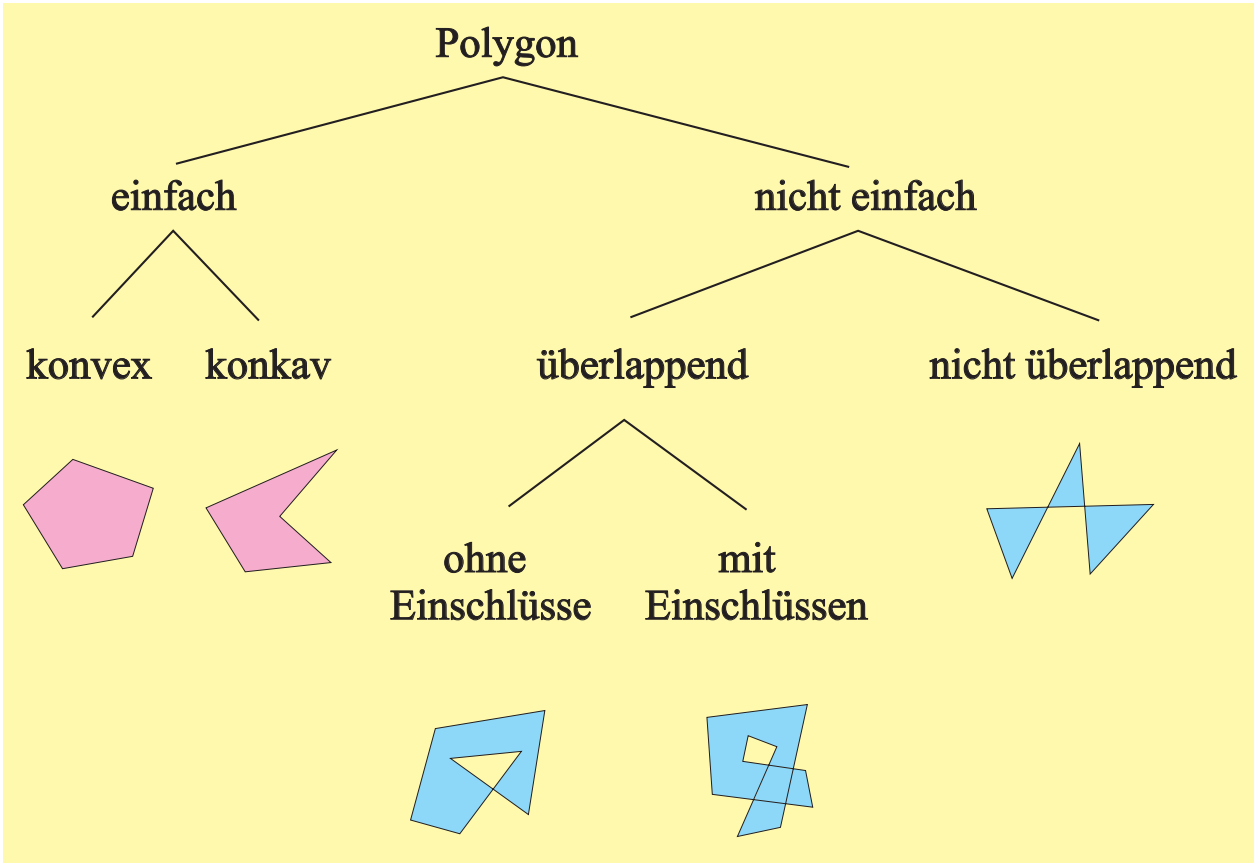
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Midpoint Line-Algorithmus



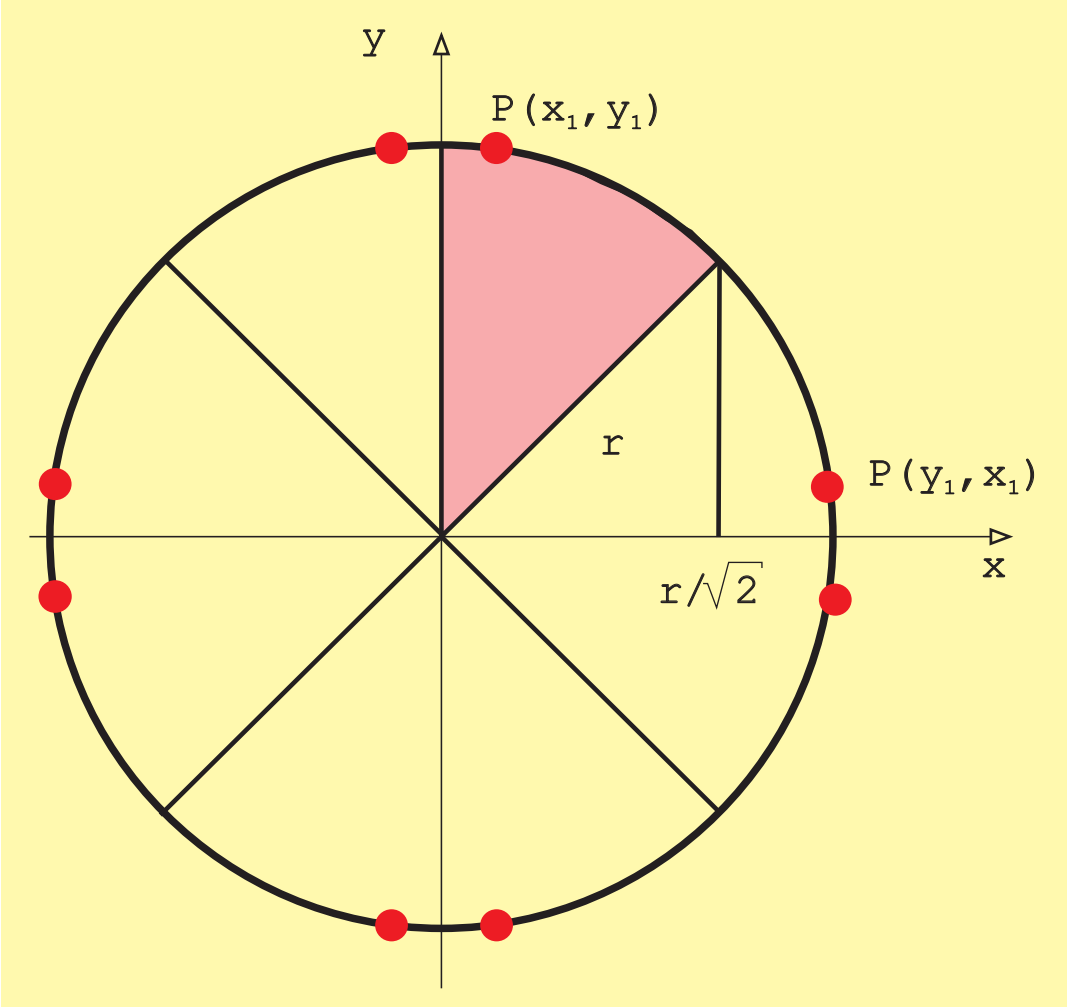
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Klassifizierung von Polygonen



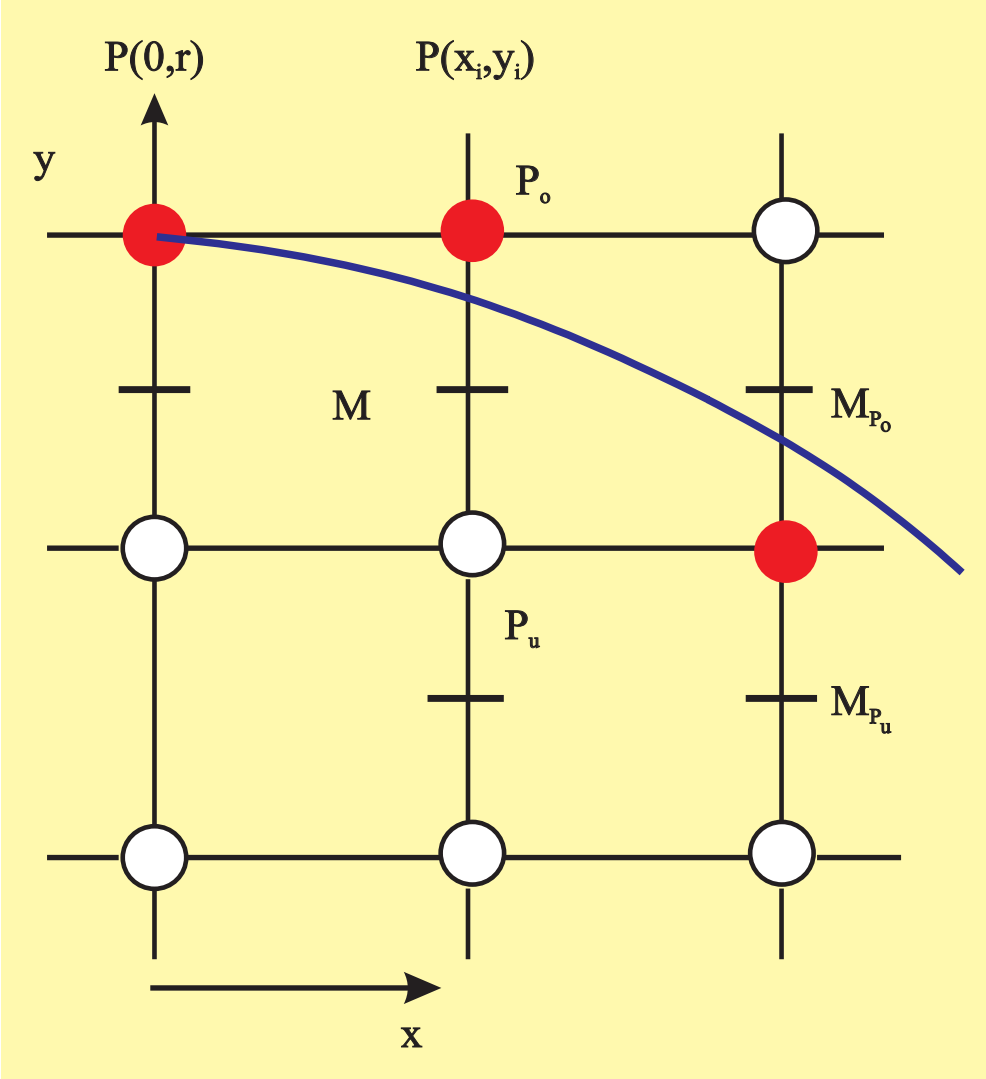
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

8-Punkt-Symmetrie des Kreises



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Midpoint Circle-Algorithmus



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

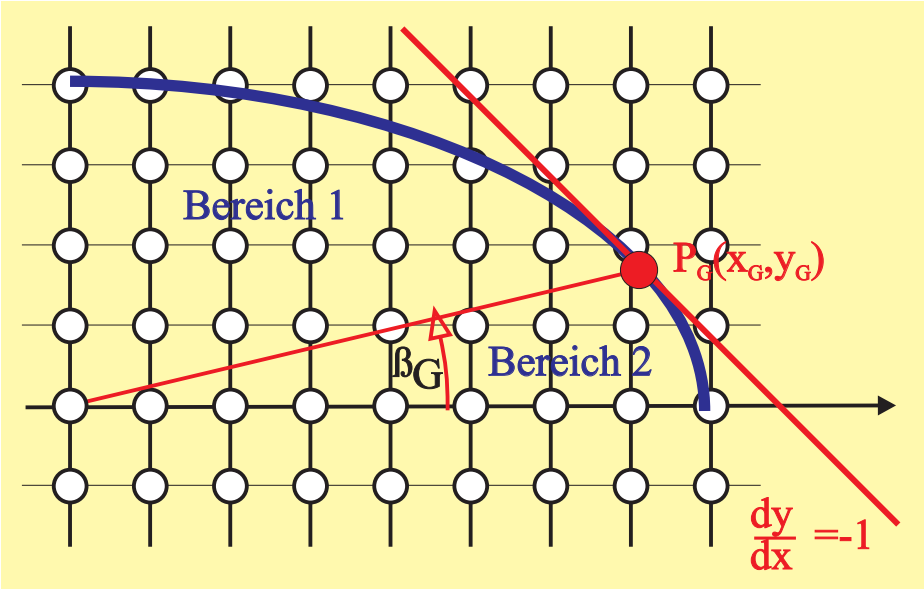
Midpoint Circle-Algorithmus

```
procedure CirclePoints(XM,YM,X,Y,Color: Word);
begin
  PutPixel(XM+X,YM+Y,Color);
  PutPixel(XM+Y,YM+X,Color);
  PutPixel(XM+X,YM-Y,Color);
  PutPixel(XM+Y,YM-X,Color);
  PutPixel(XM-X,YM-Y,Color);
  PutPixel(XM-Y,YM-X,Color);
  PutPixel(XM-X,YM+Y,Color);
  PutPixel(XM-Y,YM+X,Color);
end {CirclePoints};


procedure MidpointCircle1{2}(XM,YM,R,Color: Word);
  var x,y: integer; T: real;  {var x,y,T: integer;}
begin
  x:=0;  y:=R;  T:=1.25-R;    {T:=1-R;}
  CirclePoints(XM,YM,x,y,Color);
  while y>x do
  begin
    if T<0 then       T:=T+2*x+3
          else begin T:=T+2*(x-y)+5; y:=y-1 end;
    x:=x+1;
    CirclePoints(XM,YM,x,y,Color);
  end
end {MidpointCircle12};
```

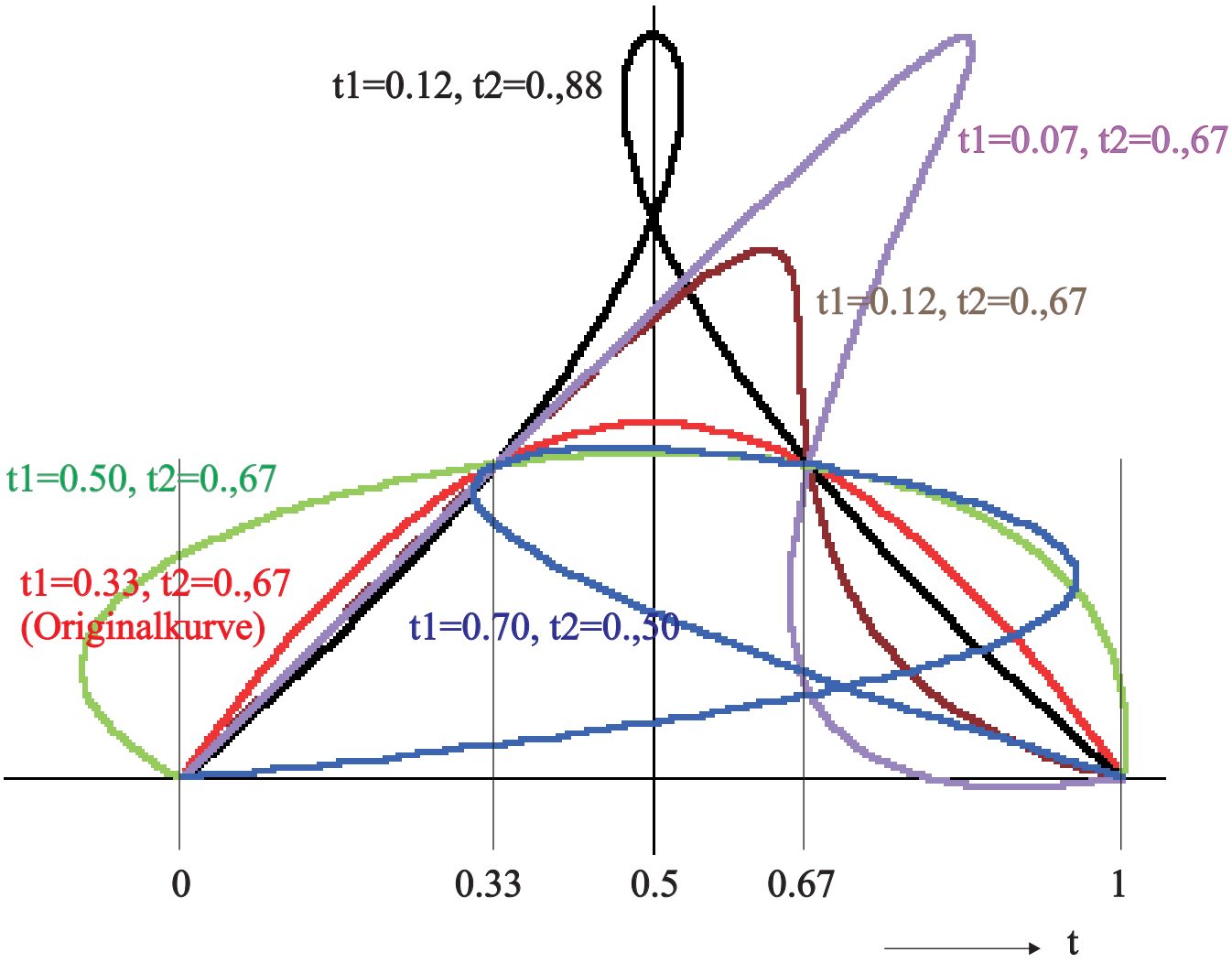
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Midpoint Ellipse-Algorithmus



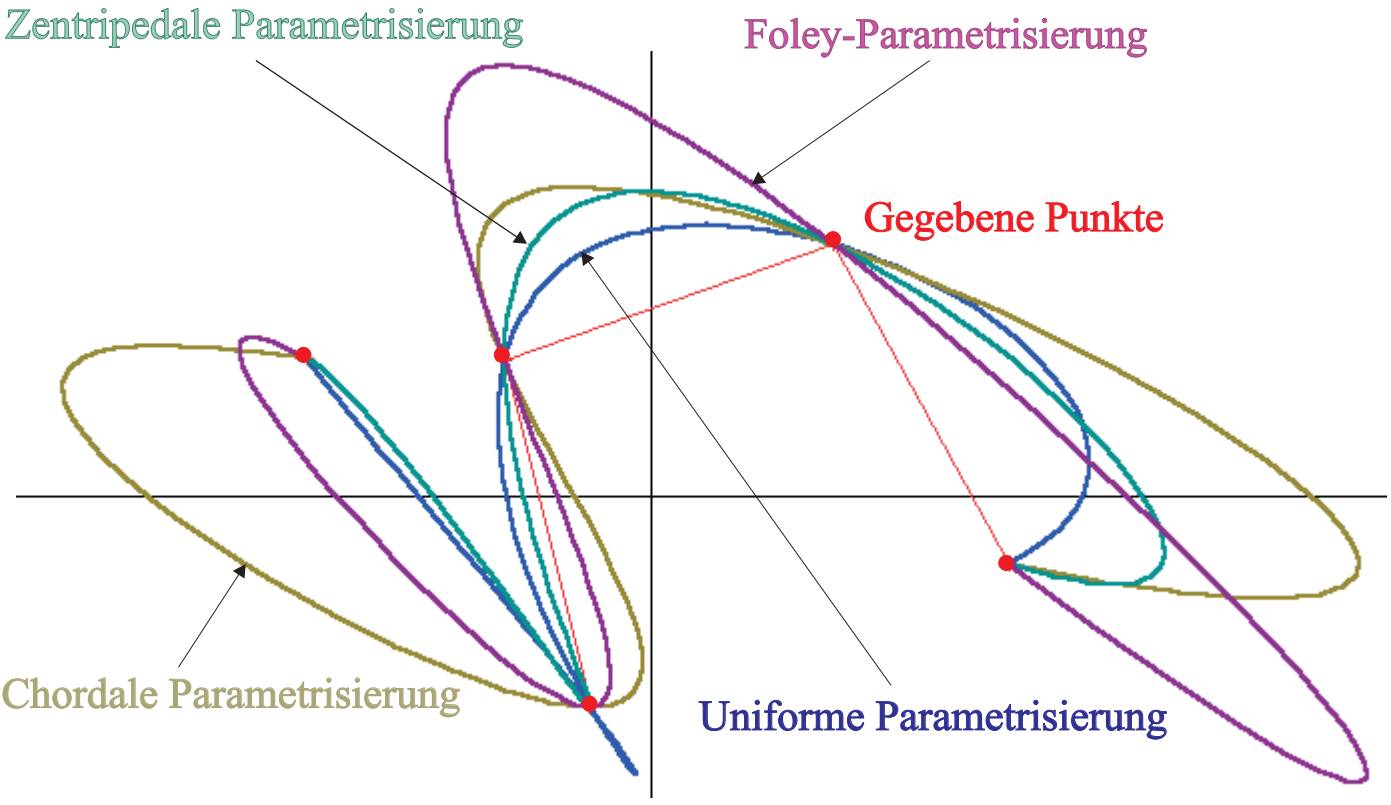
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Parametrisierung einer kubischen Kurve



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Parametrisierung von Kurven mit verschiedenen Verfahren



Rastergrafik-
Algorithmen

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

B-Splines (1)

Definition

$$\overline{P}(t) = \sum_{i=0}^{n} B_{i,k}(t)\,\overline{P}_i \qquad (1)$$

Rekusionsvorschrift für die Basisfunktionen

$$B_{i,1}(t) = \begin{cases} 1 & \text{für} \quad t_i \le t \le t_{i+1} \\ 0 & \text{sonst} \end{cases} \qquad (2)$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t) \qquad \begin{array}{l} k>1 \\ 0 \le i \le n \end{array}$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

B-Splines (2)

Basisfunktionen für kubische B-Splines

$$B_{i,1}(t) = \begin{cases} 1 & \text{für} \quad t_i \le t \le t_{i+1} \\ 0 & \text{sonst} \end{cases}$$

$$B_{i,2}(t) = \frac{t - t_i}{t_{i+1} - t_i} B_{i,1}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} B_{i+1,1}(t)$$

(3)

$$B_{i,3}(t) = \frac{t - t_i}{t_{i+2} - t_i} B_{i,2}(t) + \frac{t_{i+3} - t}{t_{i+3} - t_{i+1}} B_{i+1,2}(t)$$

$$B_{i,4}(t) = \frac{t - t_i}{t_{i+3} - t_i} B_{i,3}(t) + \frac{t_{i+4} - t}{t_{i+4} - t_{i+1}} B_{i+1,3}(t)$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

B-Splines (3)

Basisfunktionen für kubische B-Splines

$$B_{i,4}(t) = \begin{cases} \dfrac{(t - t_i)^3}{(t_{i+3} - t_i)(t_{i+2} - t_i)(t_{i+1} - t_i)} & \text{für } t_i \le t \le t_{i+1} \\ \dfrac{(t - t_i)^2(t_{i+2} - t)}{(t_{i+3} - t_i)(t_{i+2} - t_i)(t_{i+2} - t_{i+1})} + \dfrac{(t_{i+3} - t)(t - t_{i+1})(t - t_i)}{(t_{i+3} - t_i)(t_{i+3} - t_{i+1})(t_{i+2} - t_{i+1})} + \dfrac{(t_{i+4} - t)(t - t_{i+1})^2}{(t_{i+4} - t_{i+1})(t_{i+3} - t_{i+1})(t_{i+2} - t_{i+1})} & \text{für } t_{i+1} \le t \le t_{i+2} \\ \dfrac{(t - t_i)(t_{i+3} - t)^2}{(t_{i+3} - t_i)(t_{i+3} - t_{i+1})(t_{i+3} - t_{i+2})} + \dfrac{(t_{i+4} - t)(t - t_{i+1})(t_{i+3} - t)}{(t_{i+4} - t_{i+1})(t_{i+3} - t_{i+1})(t_{i+3} - t_{i+2})} + \dfrac{(t_{i+4} - t)^2(t - t_{i+2})}{(t_{i+4} - t_{i+1})(t_{i+4} - t_{i+2})(t_{i+3} - t_{i+2})} & \text{für } t_{i+2} \le t \le t_{i+3} \\ \dfrac{(t_{i+4} - t)^3}{(t_{i+4} - t_{i+1})(t_{i+4} - t_{i+2})(t_{i+4} - t_{i+3})} & \text{für } t_{i+3} \le t \le t_{i+4} \\ 0 & \text{sonst} \end{cases} \quad (4)$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

B-Splines (4)

Für ein B-Spline mit
kubischen Basisfunktionen (k=4)
fünf (aktiven) Kontrollpunkten P_0 ... P_4 (n=4)
lautet der Knotenvektor nach der weiter vorn angegebenen Vorschrift:

$$T = (t_0\ t_1\ t_2\ t_3\ t_4\ t_5\ t_6\ t_7\ t_8)$$

$$T = (\underbrace{0\ 0\ 0\ 0}_{\text{k-mal}}\ 1\ \underbrace{2\ 2\ 2\ 2}_{\text{k-mal}})$$

Mit diesem Knotenvektor ergeben sich nach einiger Zwischenrechnung die kubischen Basisfunktionen $B_{i,4}(t)$ mit $0 \le i \le n$, d. h. $0 \le i \le 4$, zu:

$$B_{0,4}(t) = \begin{cases} (1-t)^3 & \text{für } t_3 \le t \le t_4 \rightarrow 0 \le t \le 1 \\ 0 & \text{sonst} \end{cases}$$

$$B_{1,4}(t) = \begin{cases} t(7t^2 - 18t + 12)/4 & \text{für } 0 \le t \le 1 \\ (2-t)^3/4 & \text{für } 1 \le t \le 2 \\ 0 & \text{sonst} \end{cases}$$

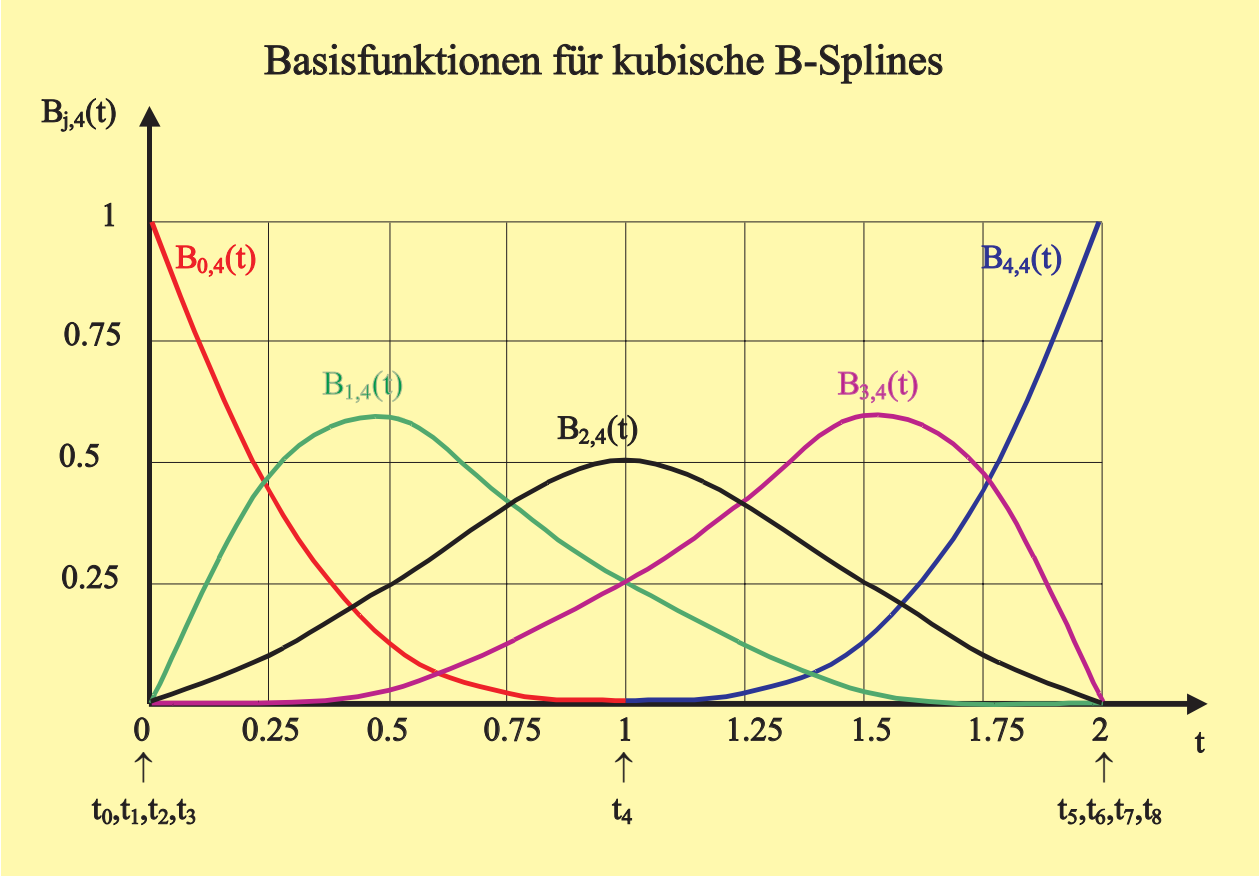
$$B_{2,4}(t) = \begin{cases} t^2(3-2t)/2 & \text{für } 0 \le t \le 1 \\ (2-t)^2(2t-1)/2 & \text{für } 1 \le t \le 2 \\ 0 & \text{sonst} \end{cases}$$

$$B_{3,4}(t) = \begin{cases} t^3/4 & \text{für } 0 \le t \le 1 \\ (2-t)(7t^2-10t+4)/4 & \text{für } 1 \le t \le 2 \\ 0 & \text{sonst} \end{cases}$$

$$B_{4,4}(t) = \begin{cases} (t-1)^3 & \text{für } 1 \le t \le 2 \\ 0 & \text{sonst} \end{cases}$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

B-Splines (5)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

NURBS (1)

Definition

$$\bar{P}(t) = \sum_{i=0}^{n} R_{i,k}(t)\,\bar{P}_i \qquad (1)$$

$$R_{i,k}(t) = \frac{w_i B_{i,k}(t)}{\sum_{j=0}^{n} w_j B_{j,k}(t)}$$

Rekusionsvorschrift für die Basisfunktionen

$$B_{i,1}(t) = \begin{cases} 1 & \text{für} \quad t_i \le t < t_{i+1} \\ 0 & \text{sonst} \end{cases} \qquad (2)$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t) \qquad \begin{array}{l} k>1 \\ 0 \le i \le n \end{array}$$

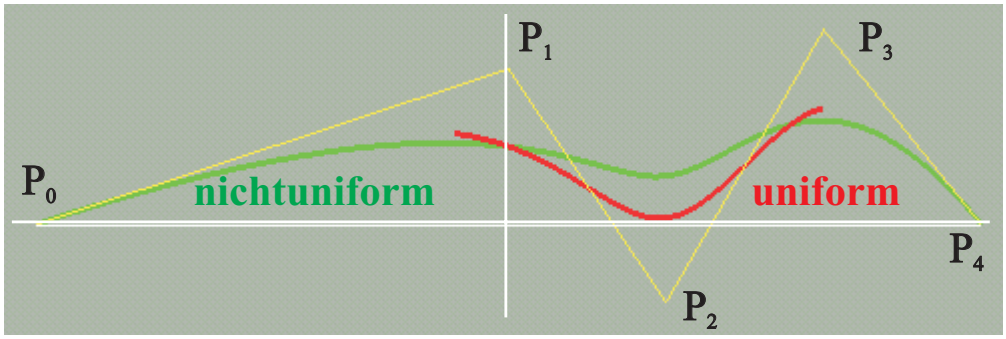
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

B-Splines (6)

Beispiel zu uniformen und nichtuniformen B-Splines

Kontrollpunkte:
$P_0 = (-30, 0)$
$P_1 = (0, 20)$
$P_2 = (10, -10)$
$P_3 = (20, 25)$
$P_4 = (30, 0)$

	uniformes B-Spline		nichtuniformes B-Spline	
t	x	y	x	y
0	-3.33	11.66	-30.00	0.00
0.33	2.35	7.93	-7.41	10.23
0.66	6.54	2.90	4.07	8.52
1	10.00	0.83	10.00	6.25
1	10.00	0.83	10.00	6.25
1.33	13.33	4.51	15.18	10.74
1.66	16.66	10.77	21.48	13.01
2	20.00	15.00	30.00	0.00

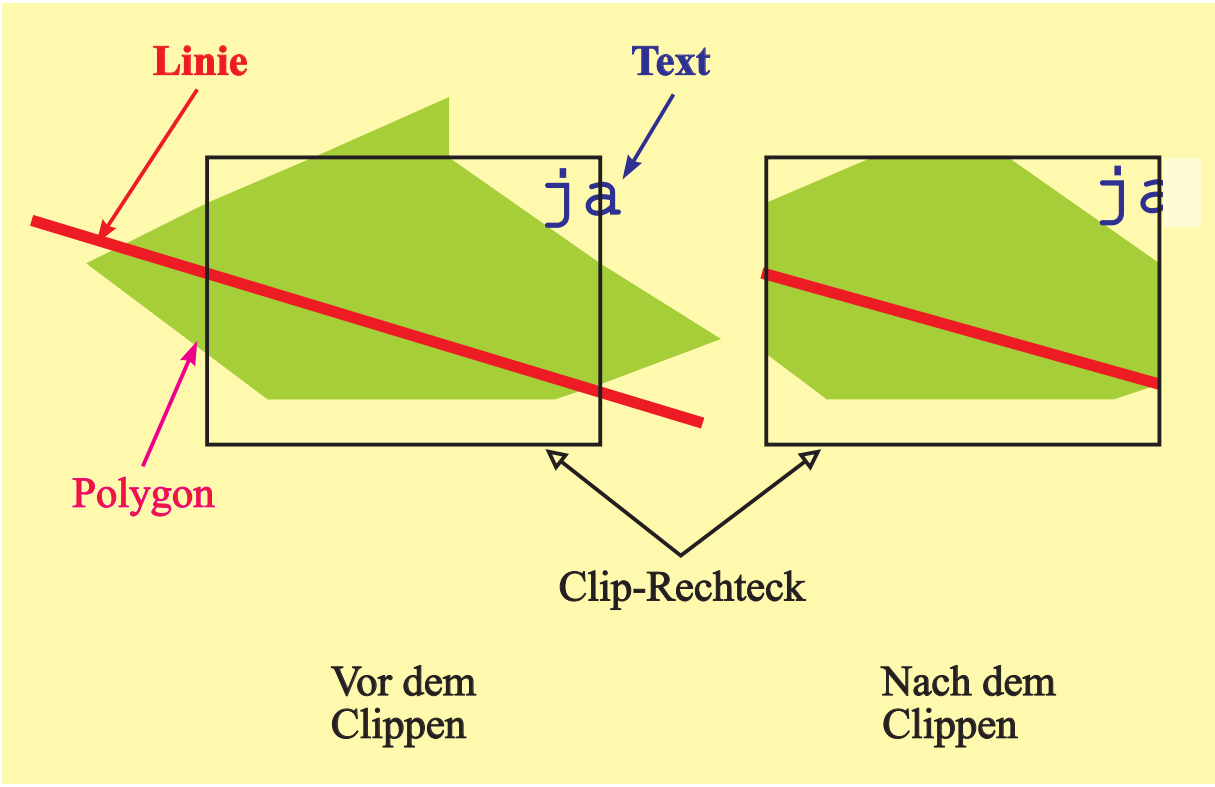


Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

2. Clippen

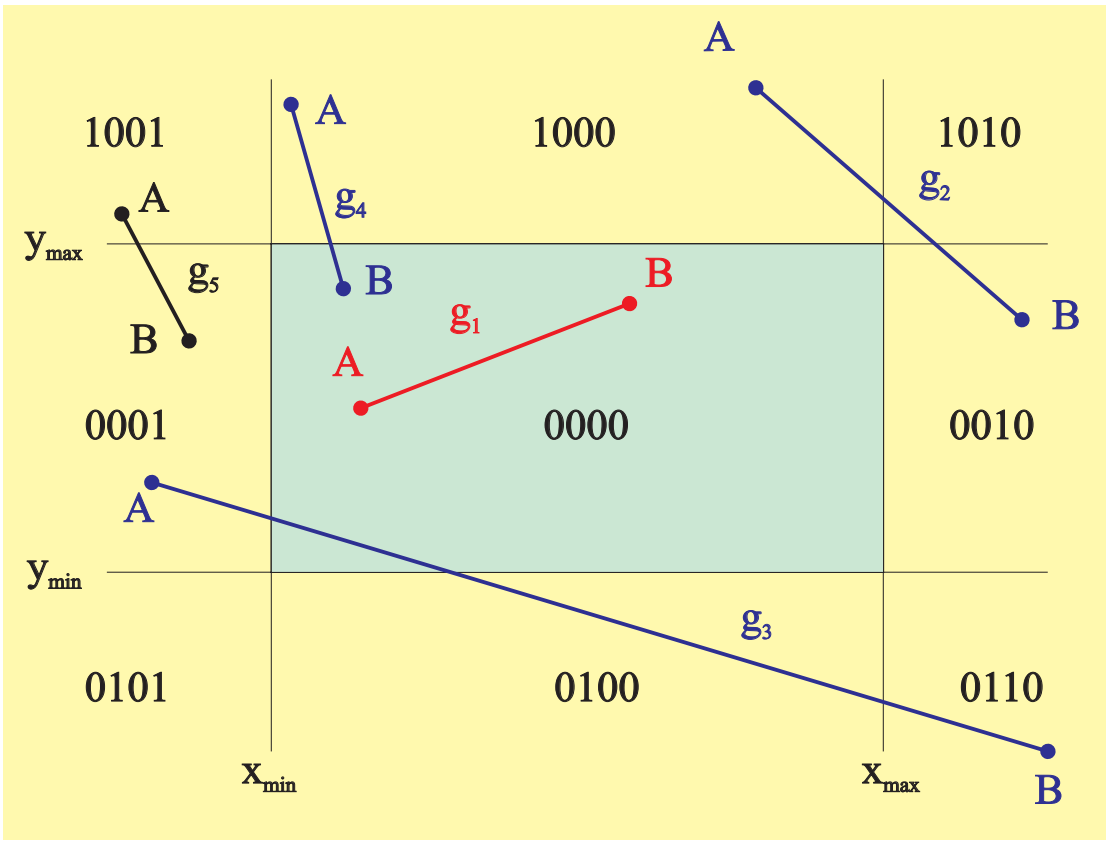
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Clippen von Darstellungselementen



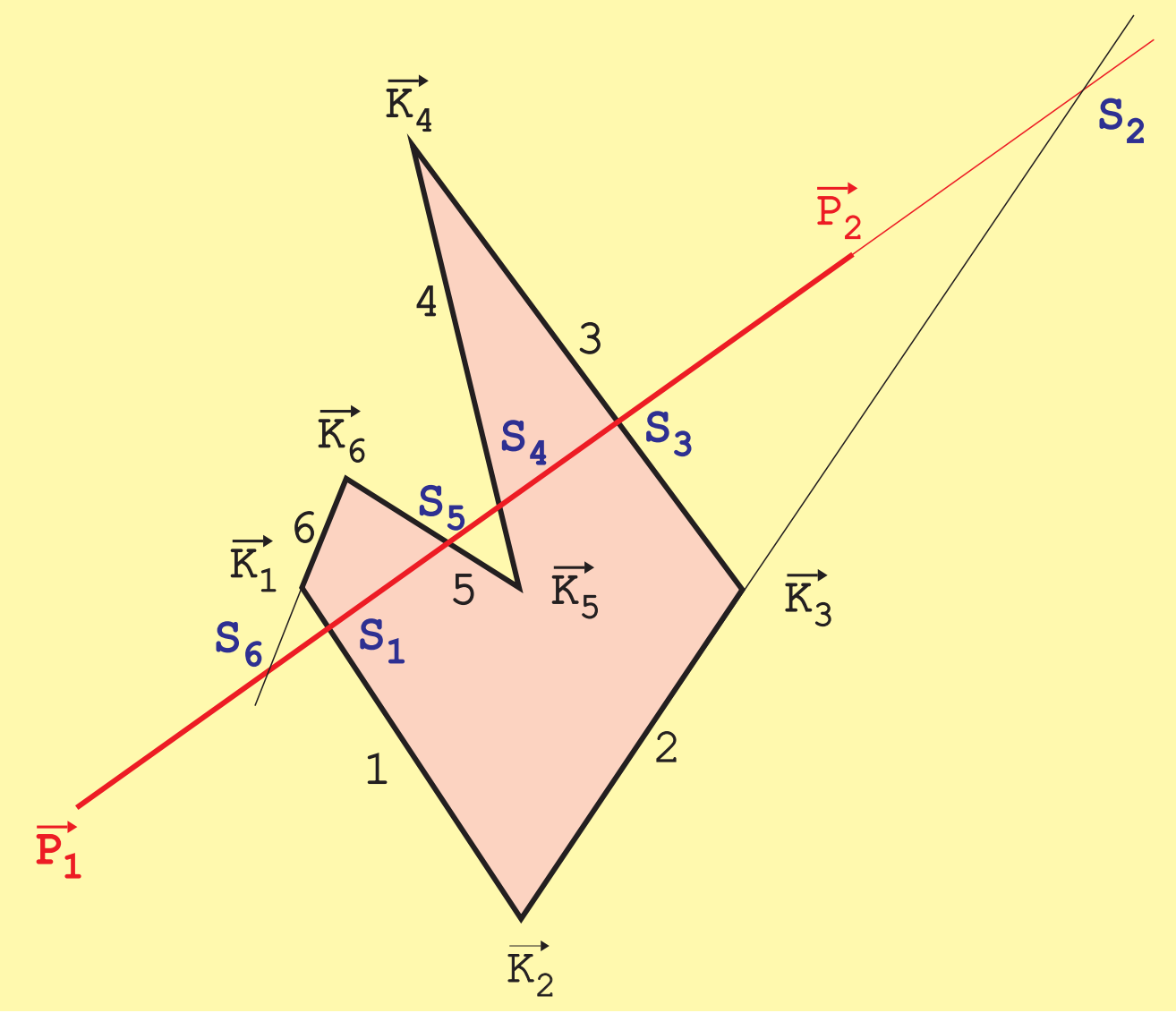
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Cohen-Sutherland-Algorithmus



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

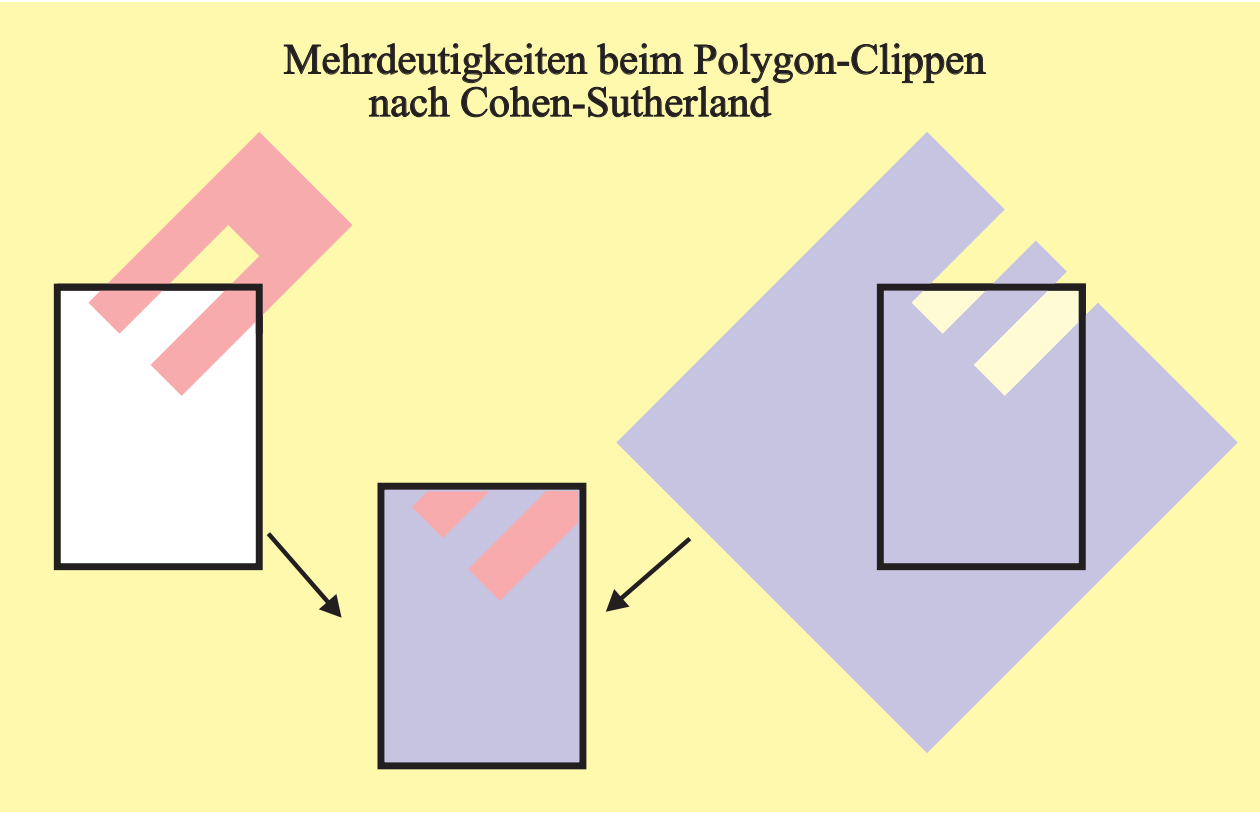
Clippen an Polygonen



Rastergrafik-
Algorithmen

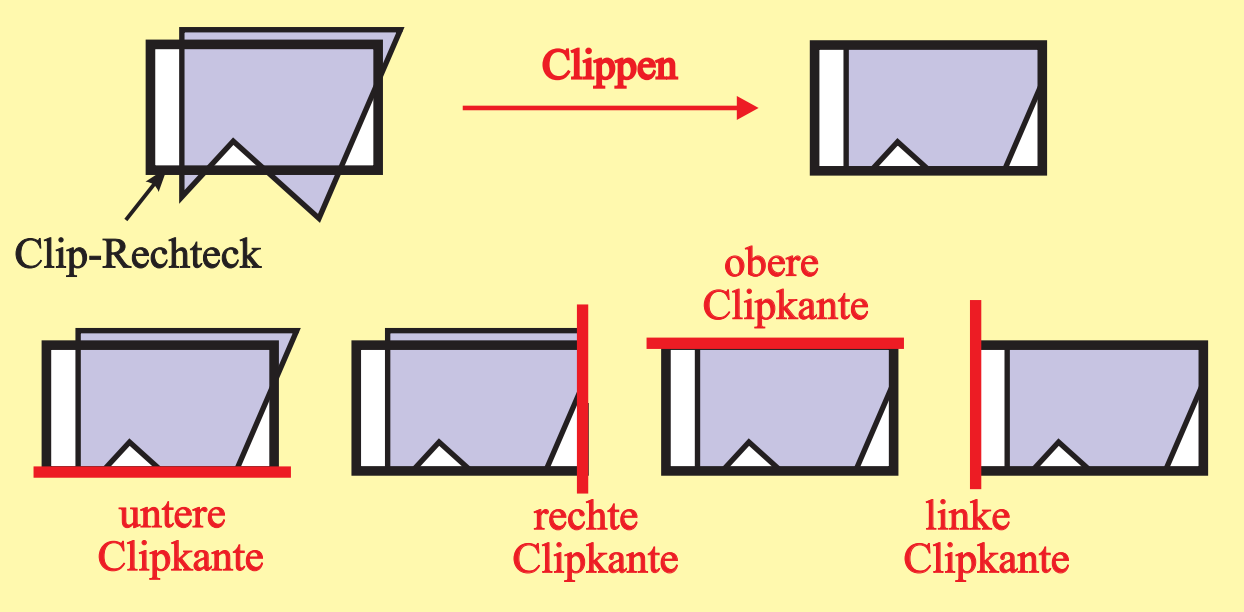
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Polygon-Clippen nach Cohen-Sutherland



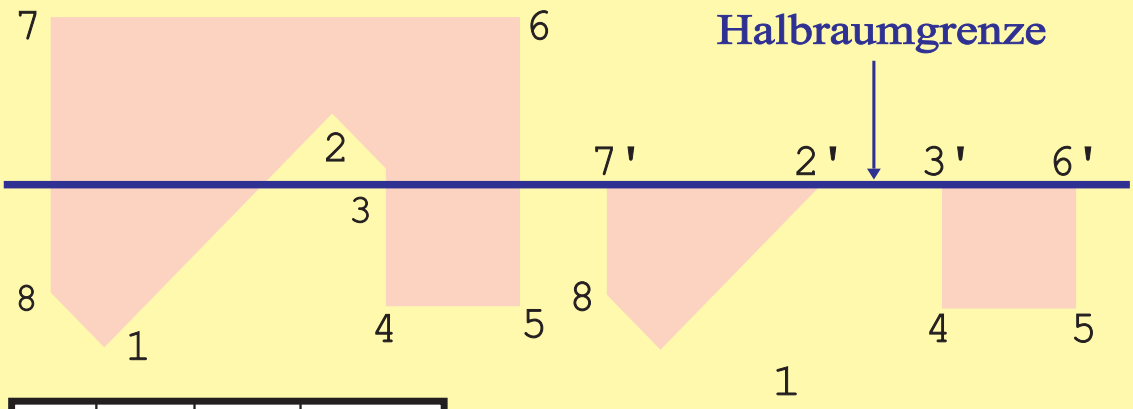
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Polygon-Clippen nach Sutherland-Hodgman (Prinzip)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Polygon-Clippen nach Sutherland-Hodgman

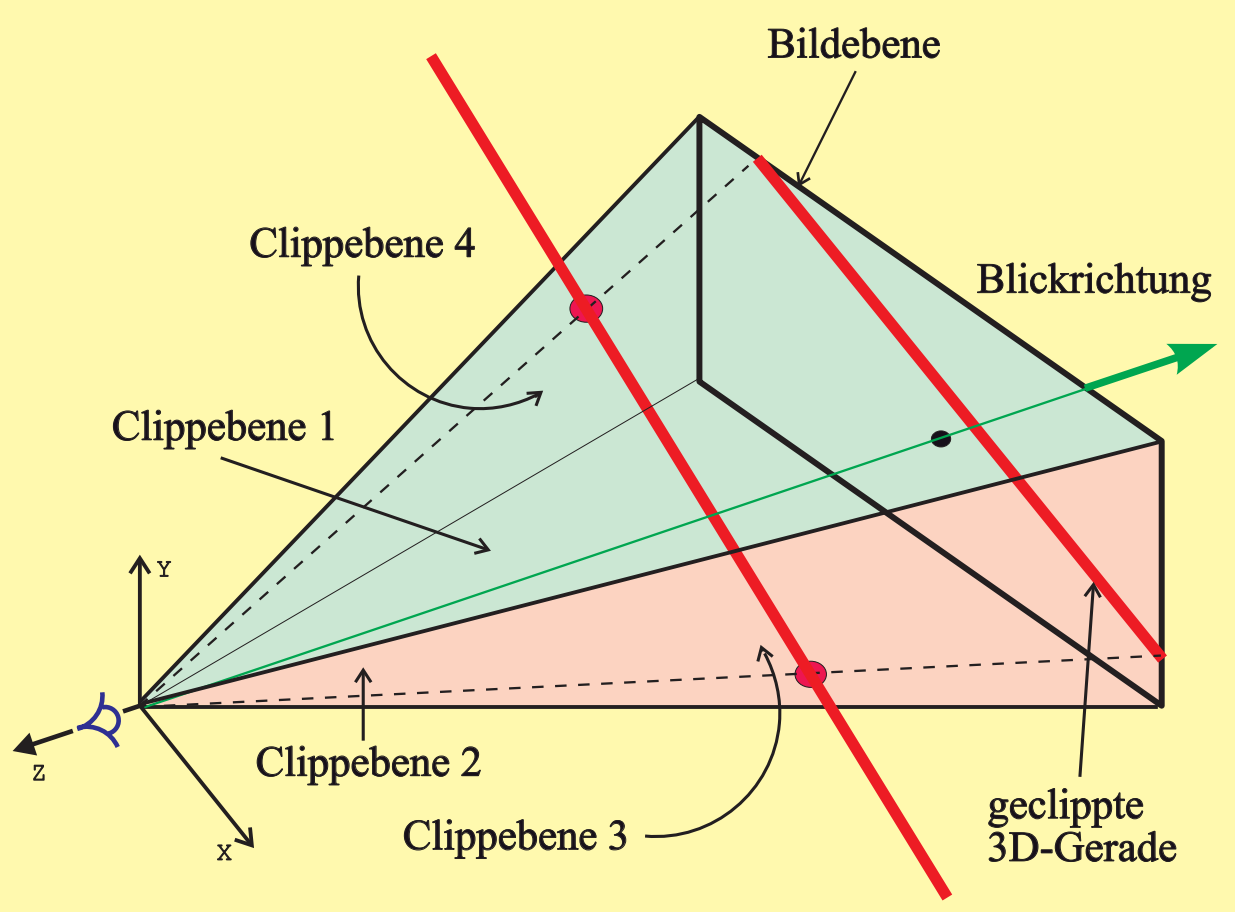


A	B	S	P
1	2	2'	1 ,2'
2	3		
3	4	3'	3',4
4	5		4 ,5
5	6	6'	5 ,6'
6	7		
7	8	7'	6',7'
			7',8
8	1		8 ,1

Legende:
A Startpunkt der Polygonkante
B Endpunkt der Polygonkante
S Schnittpunkt mit der Halbraumgrenze
P Punkte des geclippten Polygons

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

3D-Clippen von Geraden



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

3. Füllen von Polygonen und geschlossenen Kurven

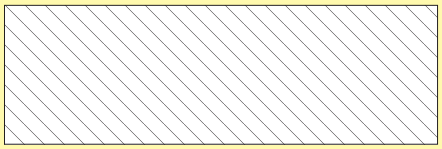
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Füllarten

hollow (leer)

solid (voll)

pattern (Muster)



hatch (Schraffur)

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Algorithmen zum Flächenfüllen

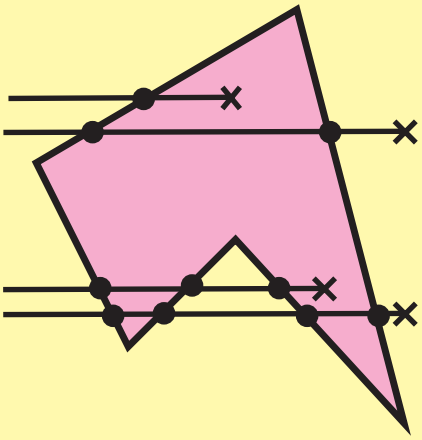
Arten von Füllalgorithmen

Fläche ist explizit geometrisch beschrieben	Fläche ist implizit im Bildspeicher beschrieben
- Gerade-Ungerade-Methode - Windungszahl-Methode - XOR-Methode - Scan Line-Algorithmus	- Saatfüll- oder Flood Fill-Algorithmus - Pixellauf- Algorithmus

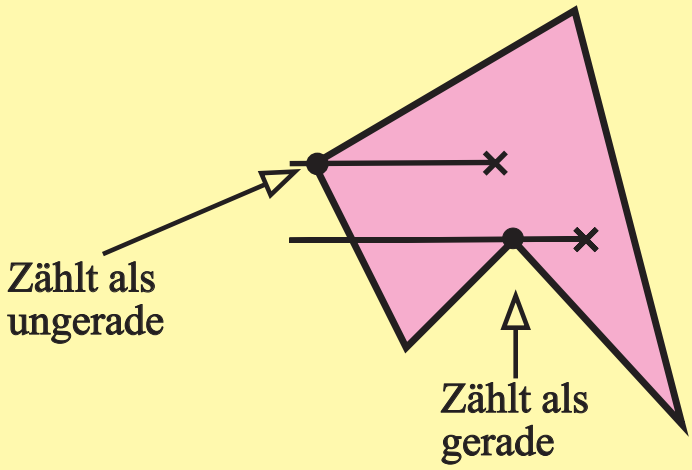
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Innenpunkt-Test (1)

Gerade-Ungerade-Methode

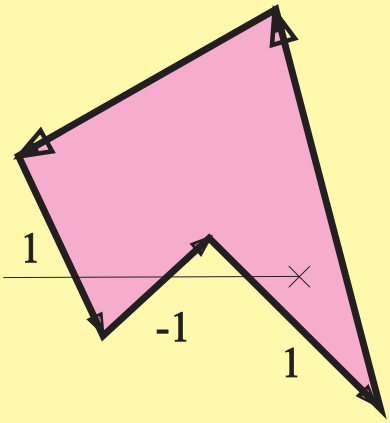


Gerade-Ungerade-Methode zum Testen von Innenpunkten



Zählmethode der Eckpunkte als Schnittpunkte

Windungszahl-Methode

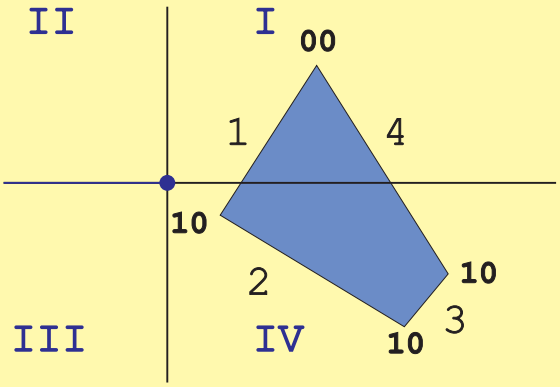


Windungszahl als Summe der Richtungszahlen

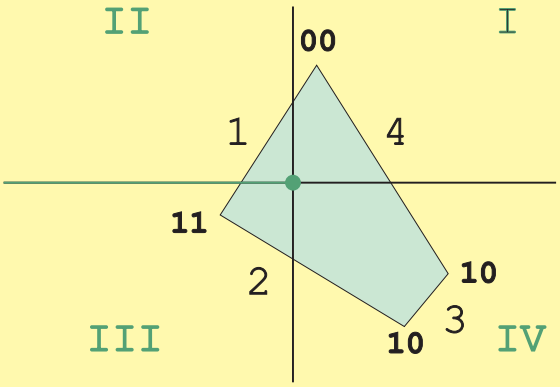
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Innenpunkt-Test (2)

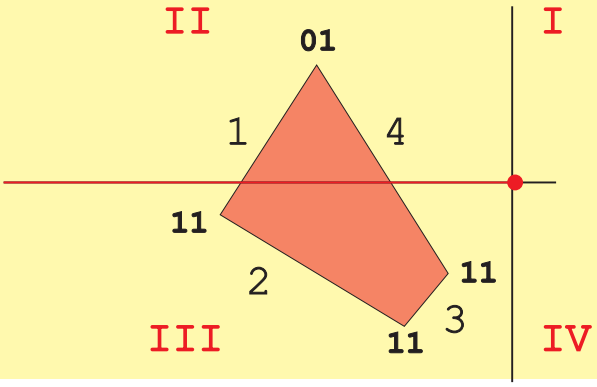
XOR-Methode



Kante	A XOR B
1	10
2	00
3	00
4	10



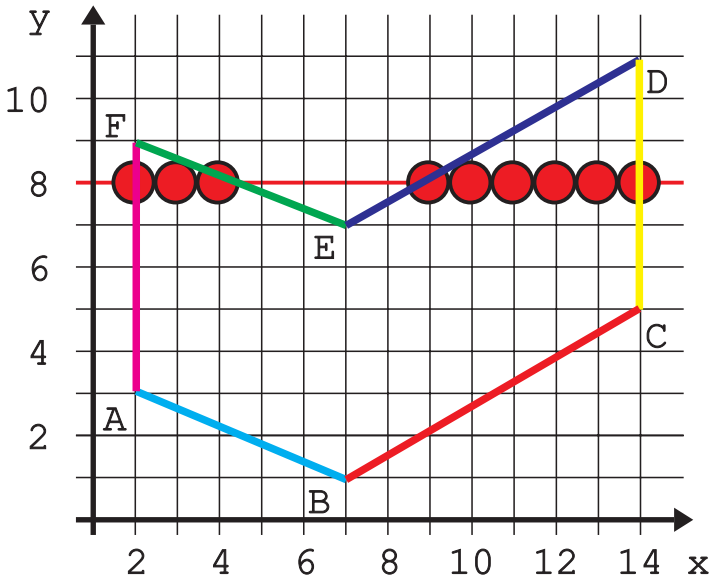
Kante	A XOR B
1	11
2	01
3	00
4	10



Kante	A XOR B
1	10
2	00
3	00
4	10

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Scan Line-Algorithmus



Globale Kantentabelle ET (edge table)

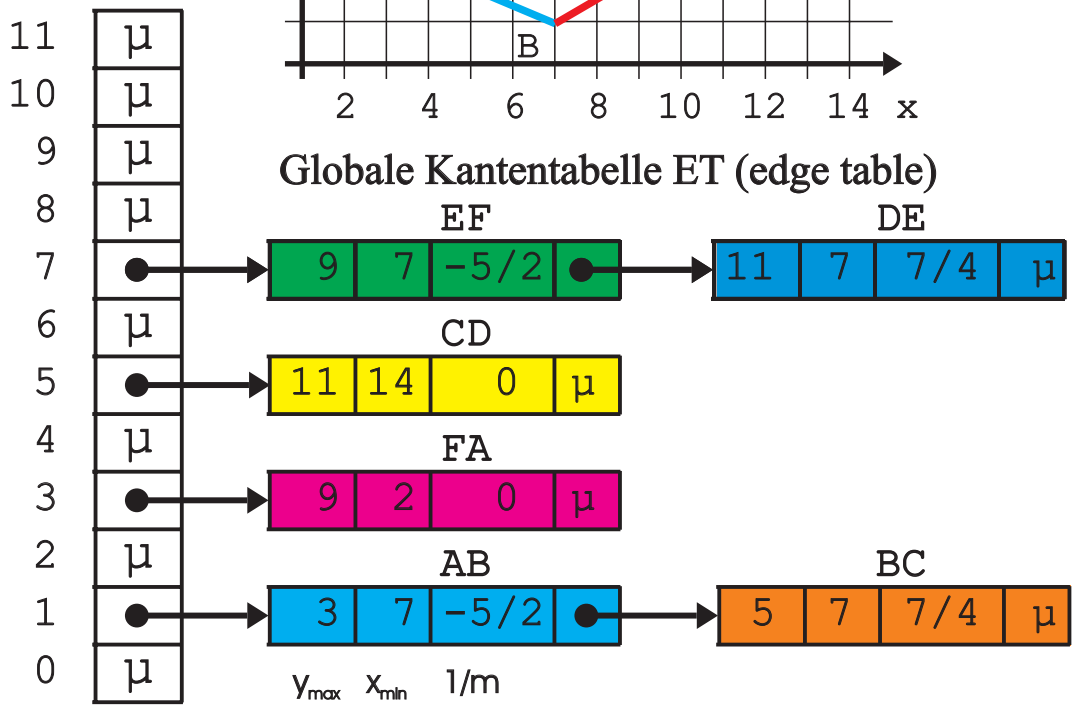
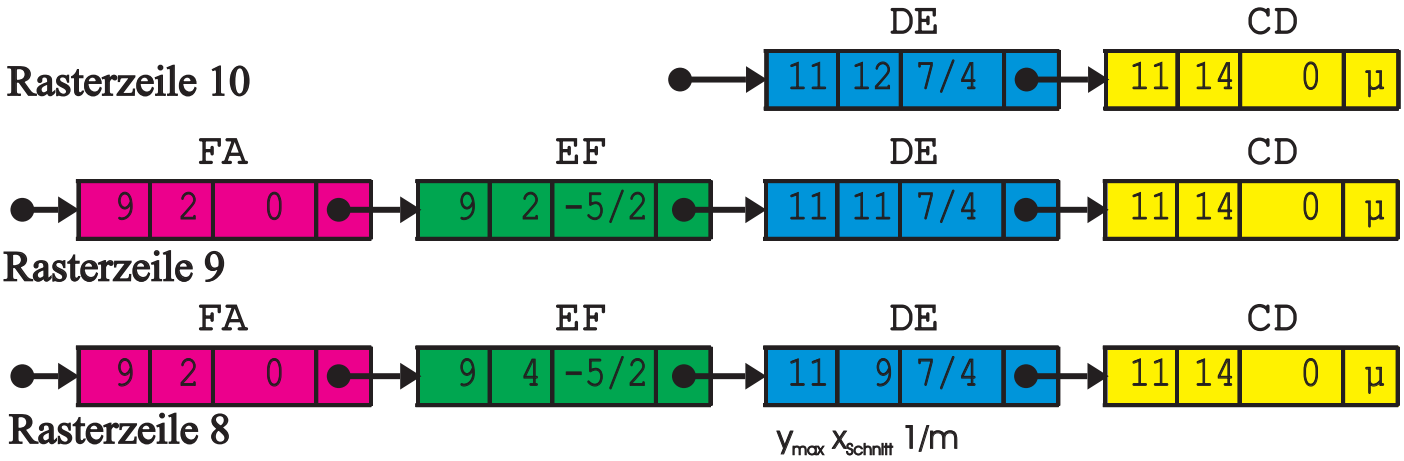
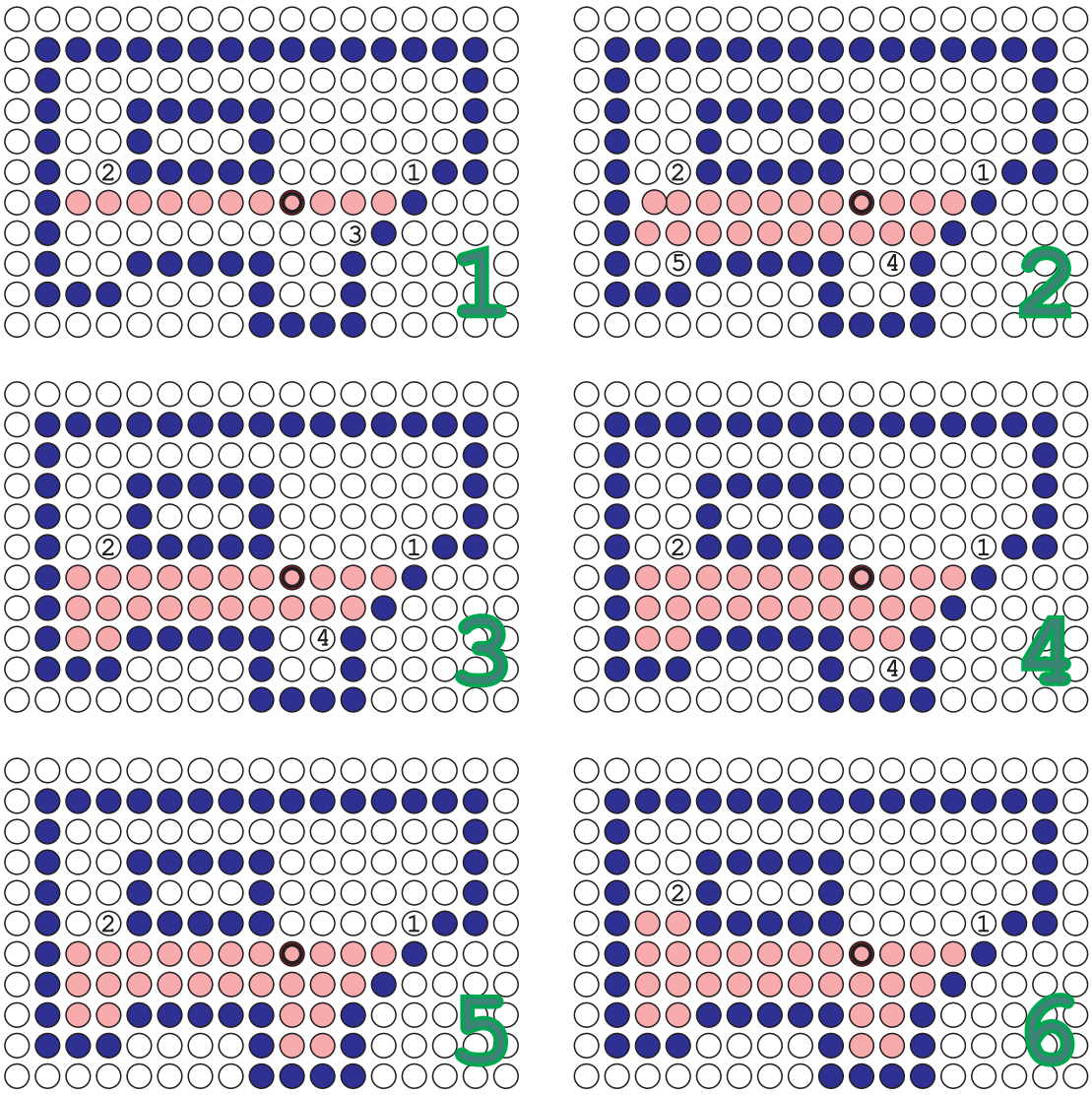


Tabelle der aktiven Kanten AET (active edge table)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

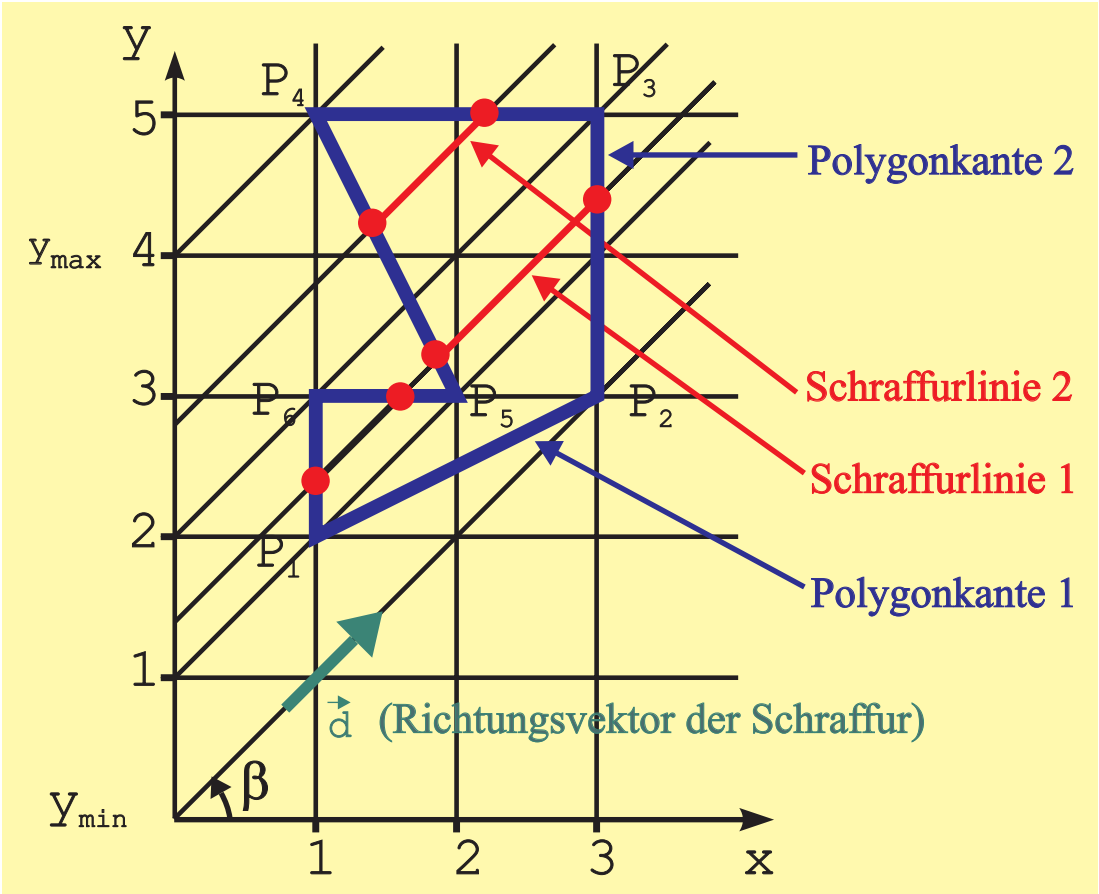
Flächenfüllen mit Pixelläufen



Saatkorn (Startpixel für den ersten Pixellauf)
Randpixel
gefüllte Pixel

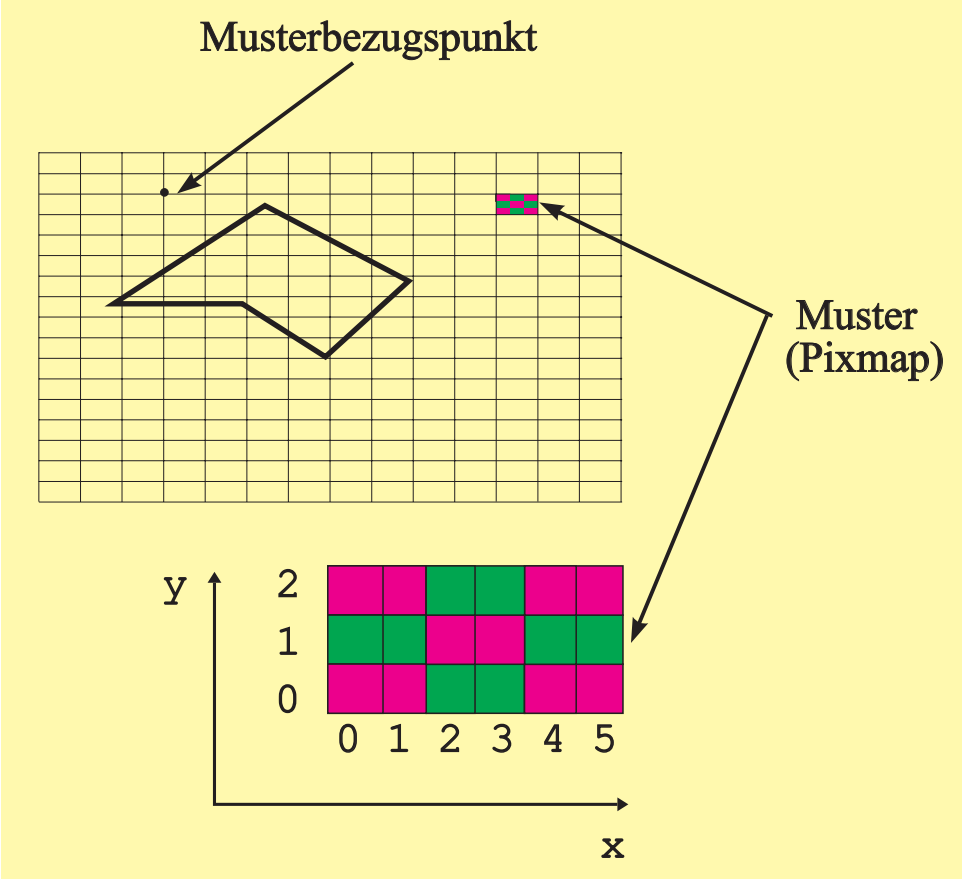
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Polygon-Schraffur



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Füllen mit Mustern



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Füllen mit Mustern (Prinzip)

```
program Pattern;
  :
const zeile  =29;        spalte =19;
type  zeilen = 0..zeile;  spalten= 0..spalte;

var   PixMap: array [zeilen,spalten] of byte;
      x,y,i,j,driver,mode: integer;

procedure Muster(n: byte);
  var x,y: word;
begin
  case n of
  1: begin
       for x:=0 to zeile do
       for y:=0 to spalte do
       if odd(x div (y+1))
          then PixMap[x,y]:=yellow
          else PixMap[x,y]:=blue;
     end {1};
  2: begin .. end {2};
       :
  end {case}
end {Muster};

begin
   :
  Muster(1);
  for x:=10 to 300 do
  for y:=10 to 120 do
  PutPixel(x,y,PixMap[x mod zeile,y mod spalte]);
  readln;
   :
end {Pattern}.
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

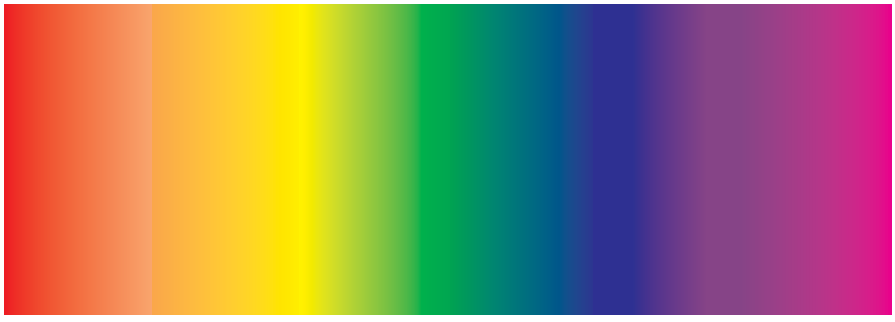
4. Farbmodelle, Farbtabellen, Halbtonverfahren

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

A new Theory about Light and Colours

“Im Jahre 1666 versah ich mich mit einem dreikantigen Glasprisma, um damit das berühmte Farbphänomen hervorzurufen. Zu diesem Zweck verdunkelte ich mein Zimmer und bohrte ein kleines Loch in die Jalousien, un eine hinreichende Menge Sonnenlicht hereinzulassen. Dann hielt ich mein Prisma in den hereinfallenden Lichtstrahl, so dass er gebrochen und an die gegenüberliegende Wand geworfen wurde. Ich fand es überaus ergötzlich, die dabei entstehenden lebhaften und kräftigen Farben eine Weile zu beobachten.” (Isaac Newton, 1672)

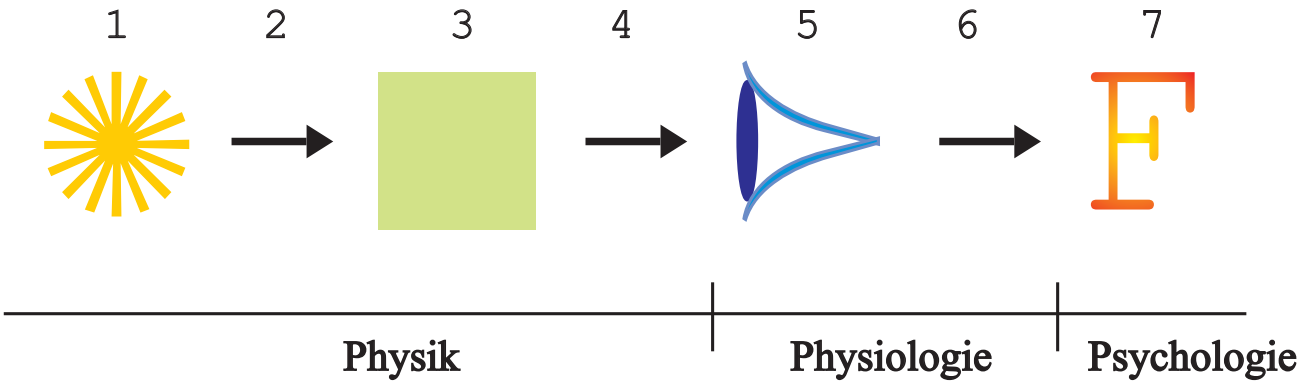
Spektralfarben



Rastergrafik-
Algorithmen

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Wirkungskette zwischen Licht und Farbempfindung



1 Lichtquelle

2 Lichtstrahlen

3 opakes oder transparentes Material

4 "Restlicht" (Farbreiz)

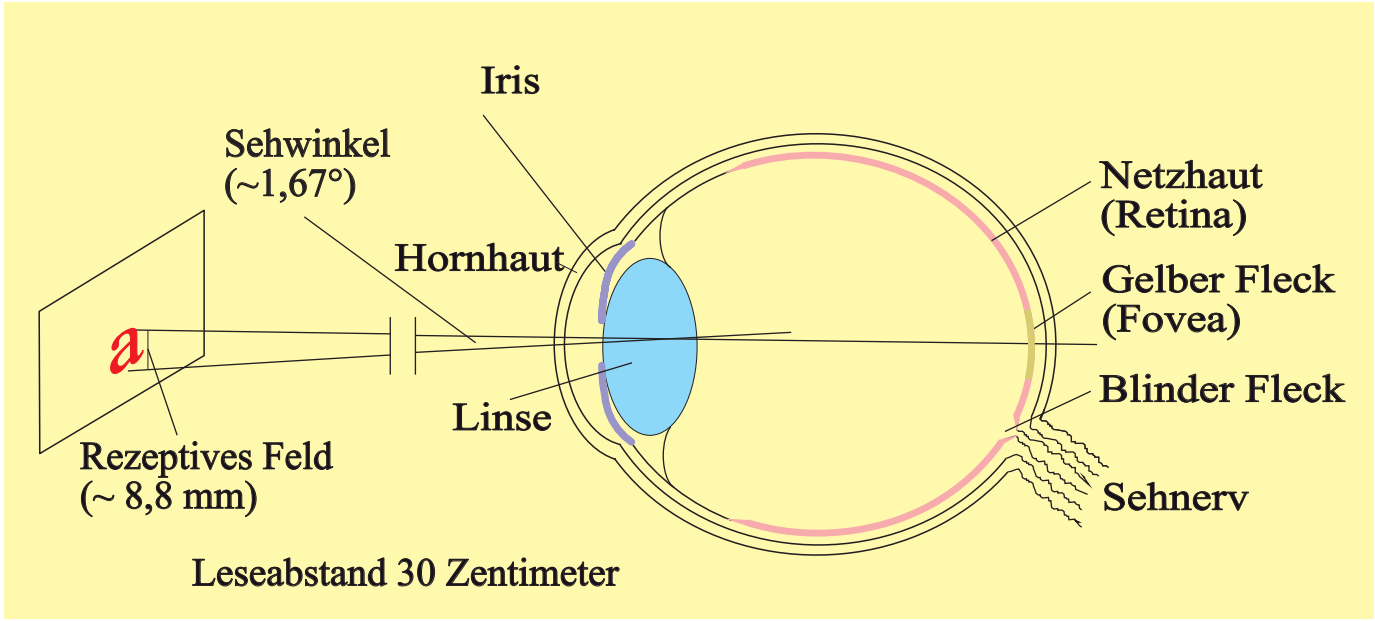
5 Auge des Betrachters

6 "Code" für das Gehirn

7 Farbempfindung des Betrachters

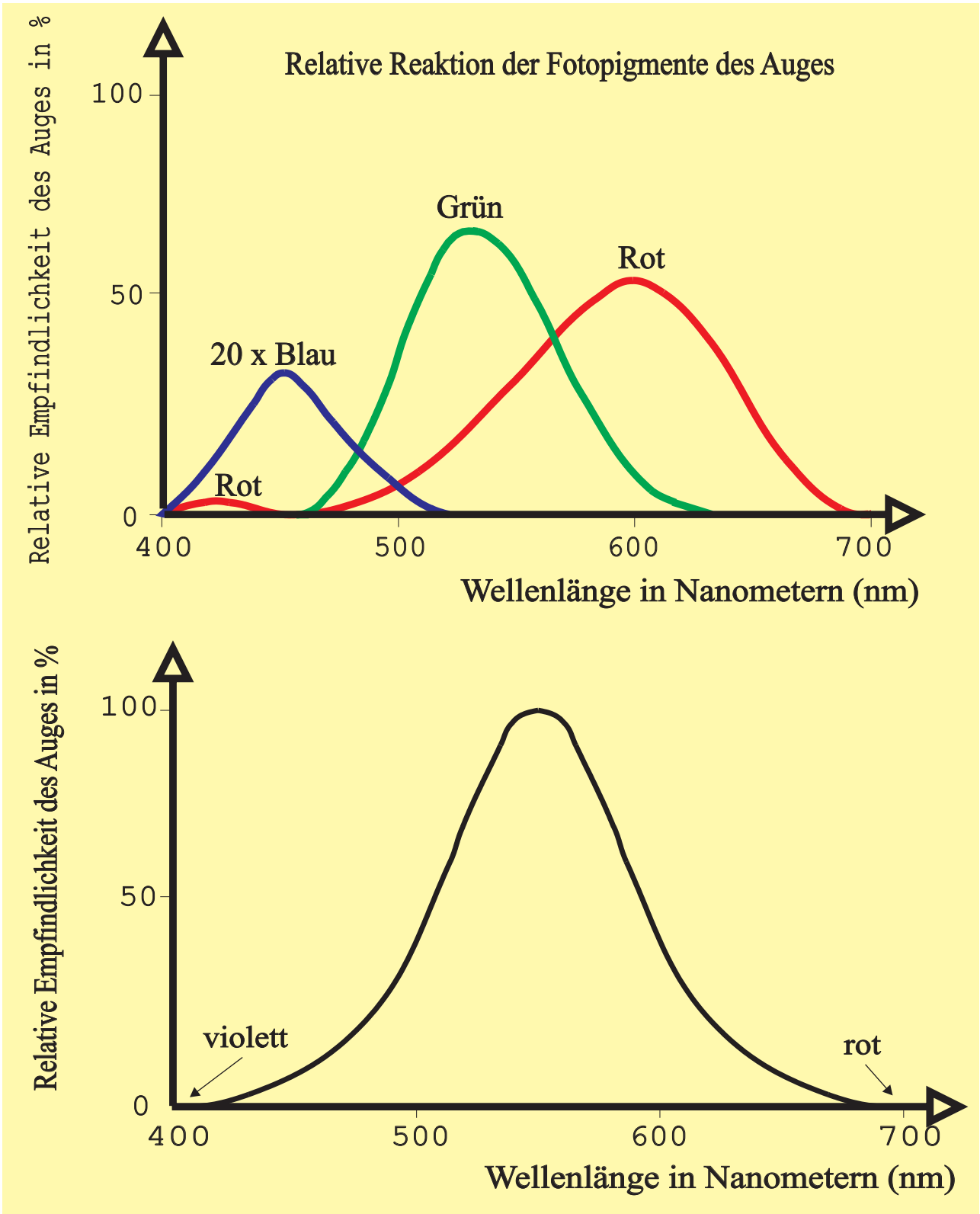
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Aufbau des menschlichen Auges



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

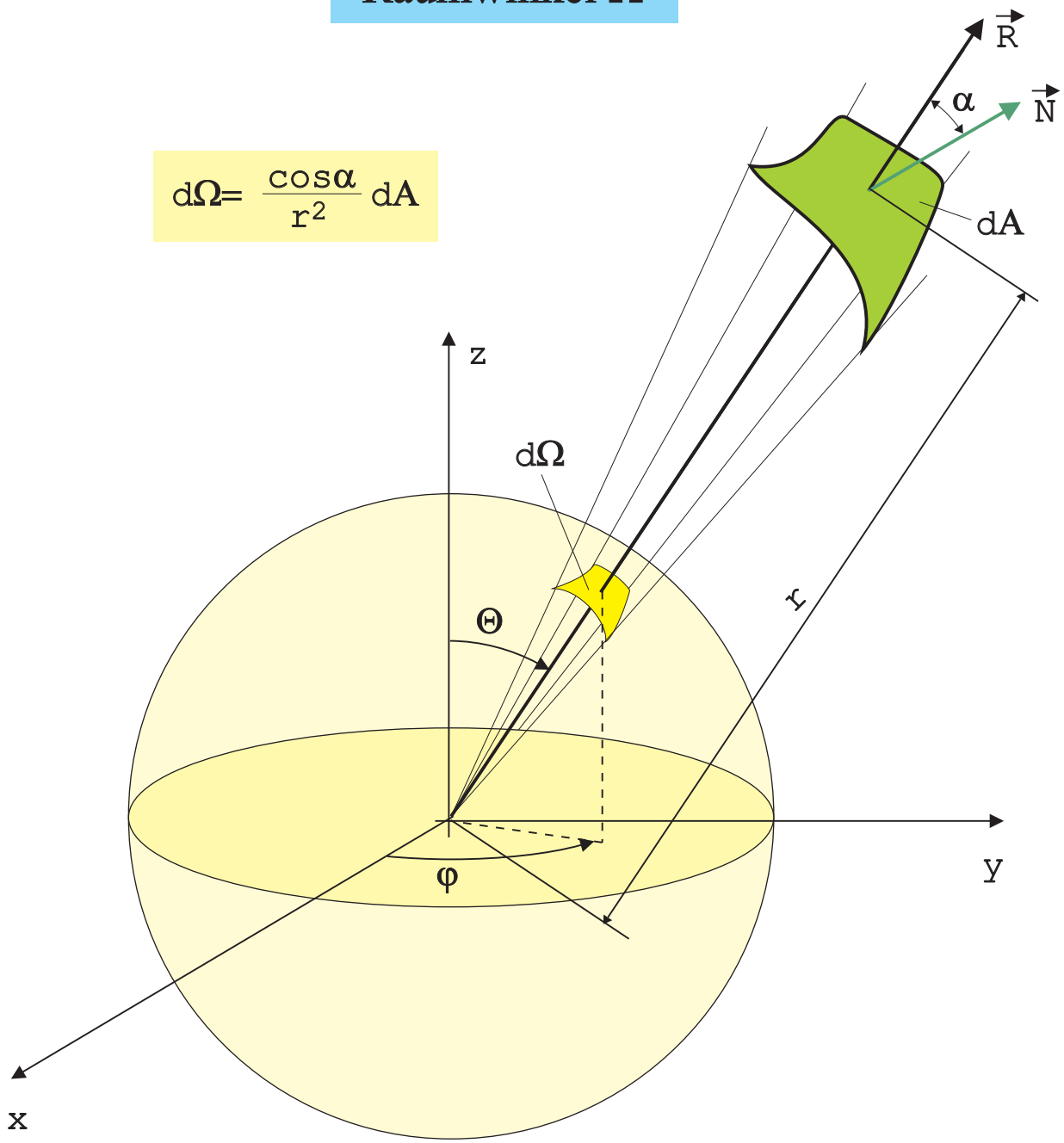
Lichtstärkenreaktion des Auges



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Raumwinkel Ω

$$d\Omega = \frac{\cos\alpha}{r^2} dA$$



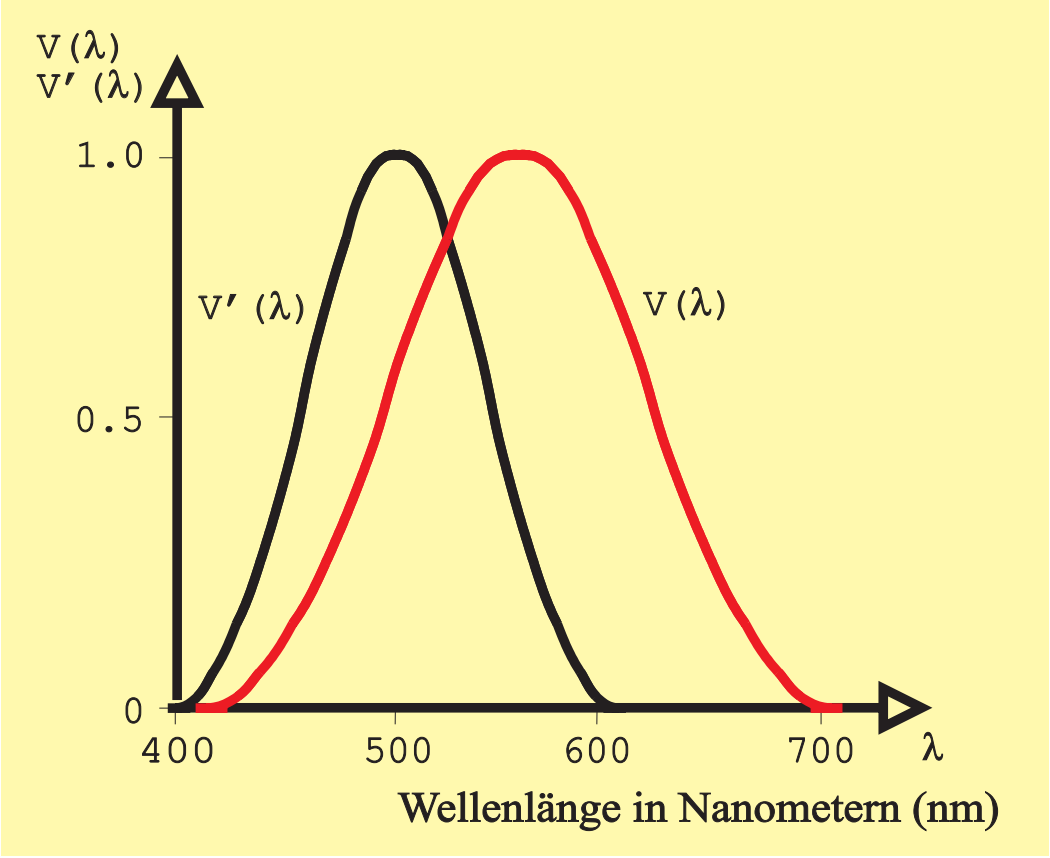
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Radio- und photometrische Größen

Definition	Radiometrie	Photometrie
Q	Strahlungsenergie [J],[W*s]	Lichtmenge [lm*s]
F=dQ/dt	Strahlungsleistung [W]	Lichtstrom [lm]
I=dF/dW	Strahlungsstärke [W/sr]	Lichtstärke [cd]
M=dF/dA	spezifische Ausstrahlung [W/m²]	spezifische Lichtausstrahlung [lm/m²]
E=cosQ*I/r²	Bestrahlungsstärke [W/(sr*m²)]	Beleuchtungsstärke [lm/m²],[lx]
L=(dI/dA)/cosQ	Strahldichte [W/(sr*m²)]	Leuchtdichte [cd/m²]

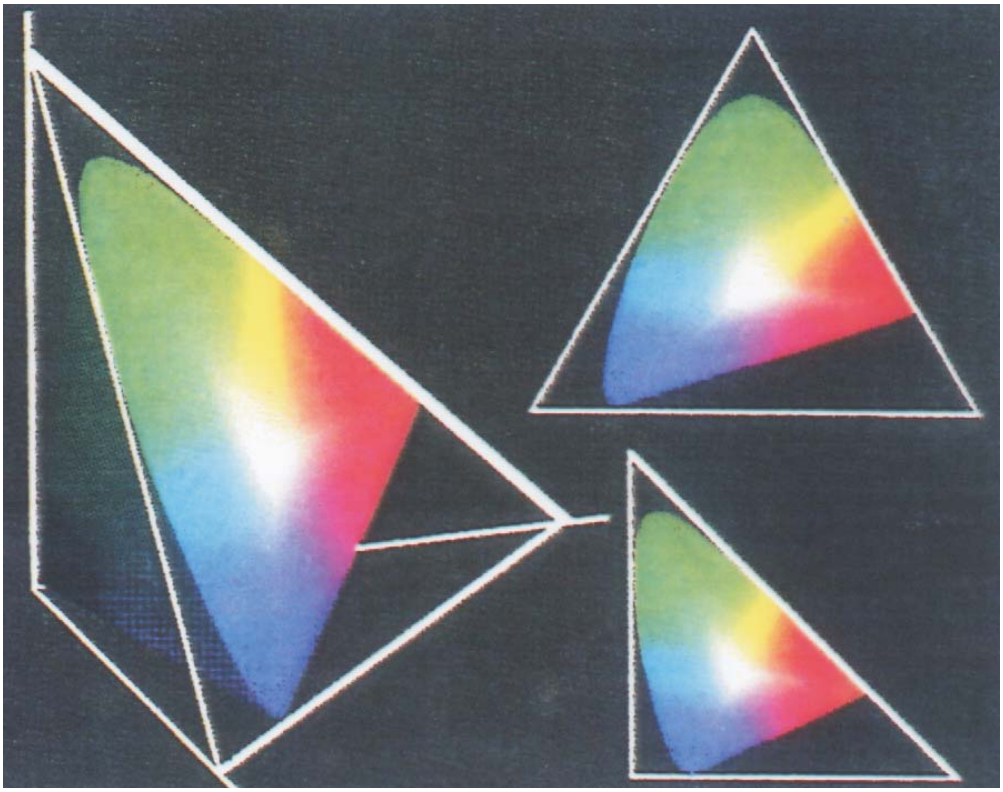
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Hellempfindlichkeitskurven
(V für das Tag- und V' für das Nachtsehen)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

CIEXYZ-Diagramme

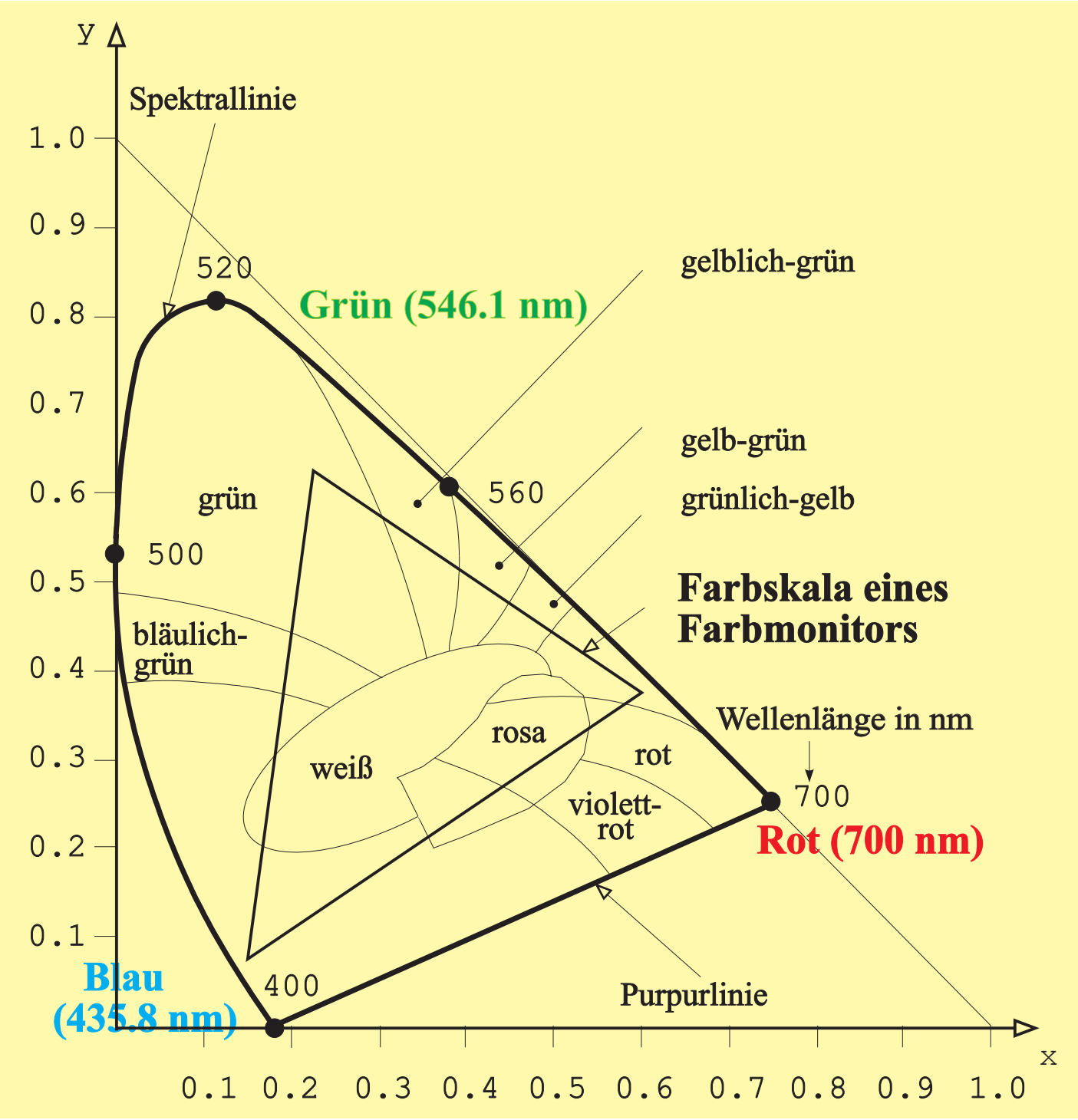


Rastergrafik-
Algorithmen

raster408

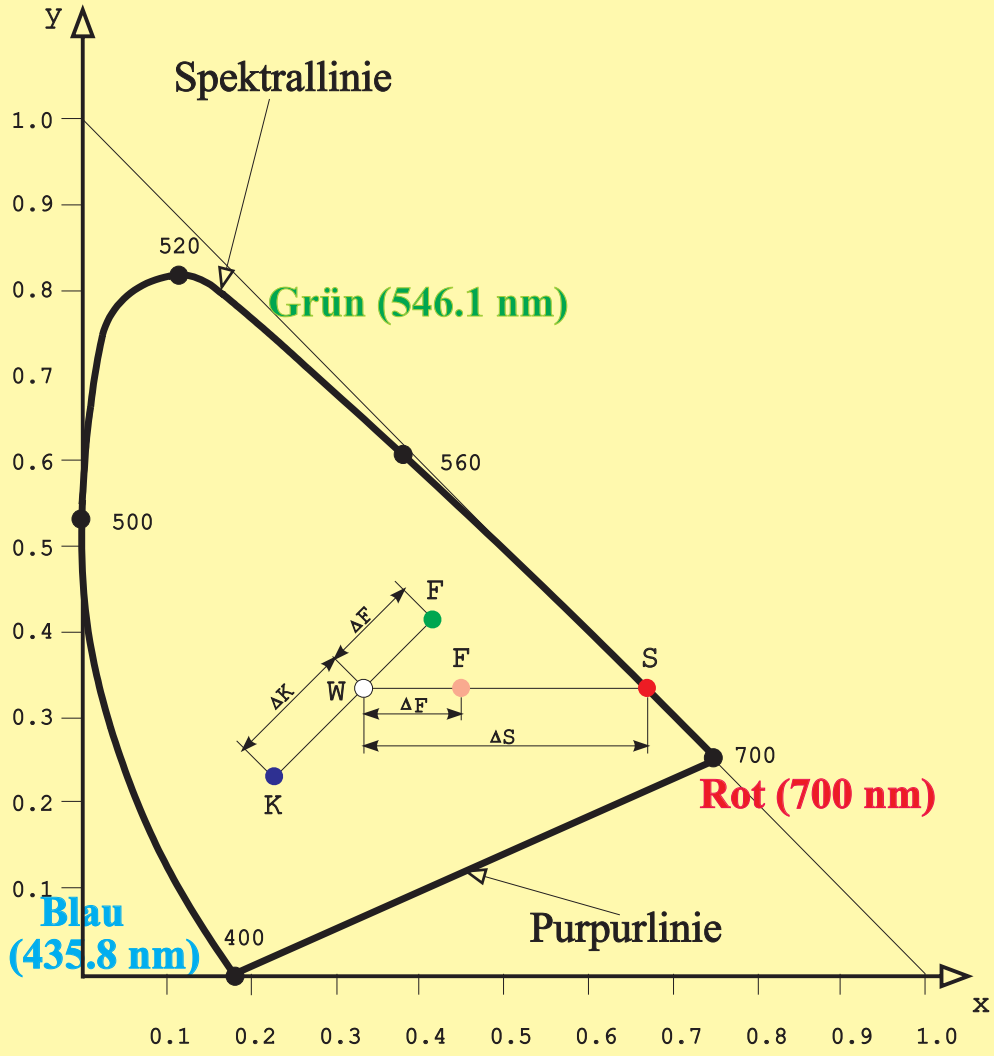
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

CIEXYZ-System (1931)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Reinheit, Farbton und Komplementärfarbe im CIEXYZ-System



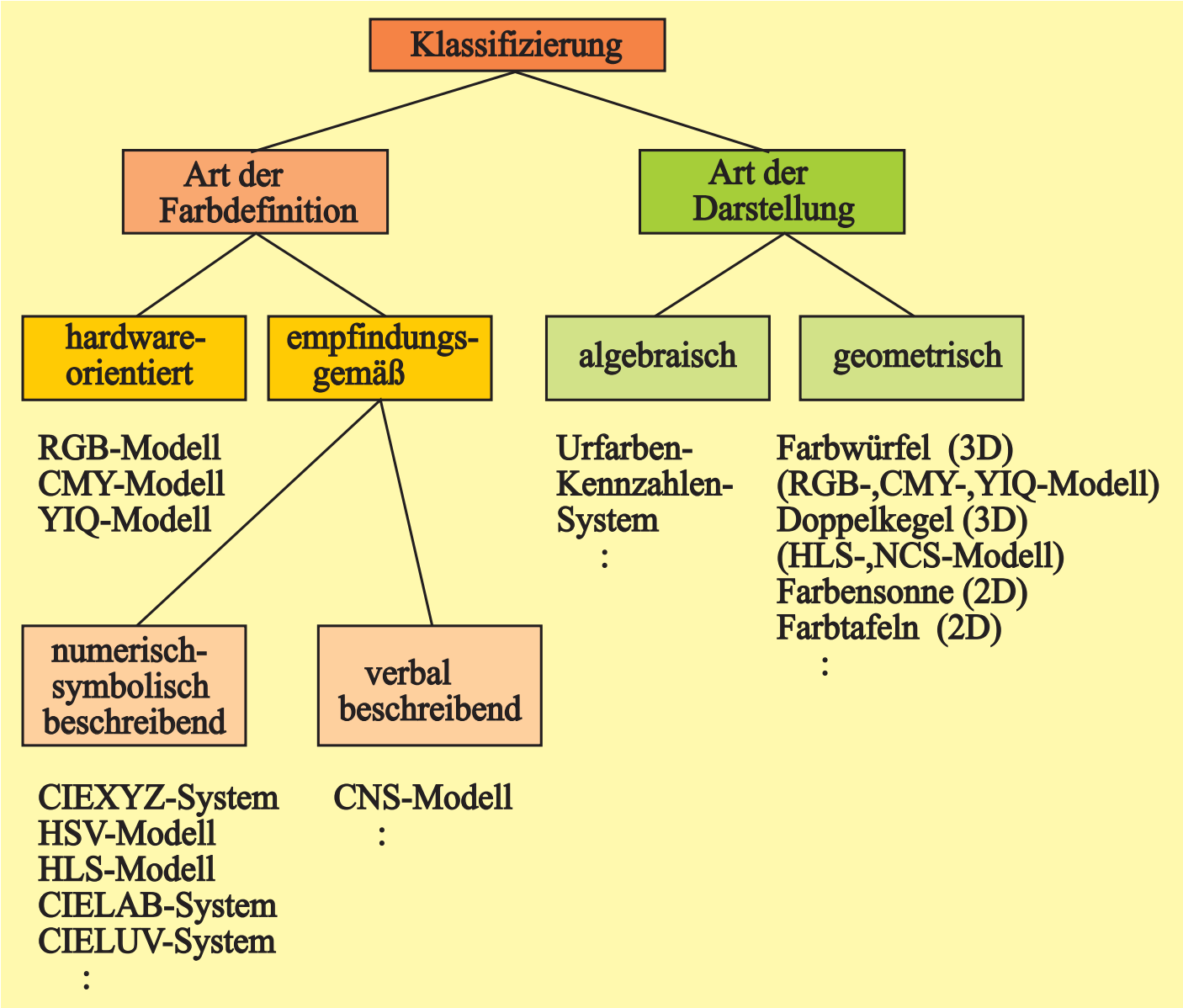
$$\text{Reinheit} = \frac{\Delta F}{\Delta S}$$

Farbton = Wellenlänge/Frequenz an der Stelle S

Komplementärfarbe K von F liegt auf der Verlängerung von $\overrightarrow{FW}$

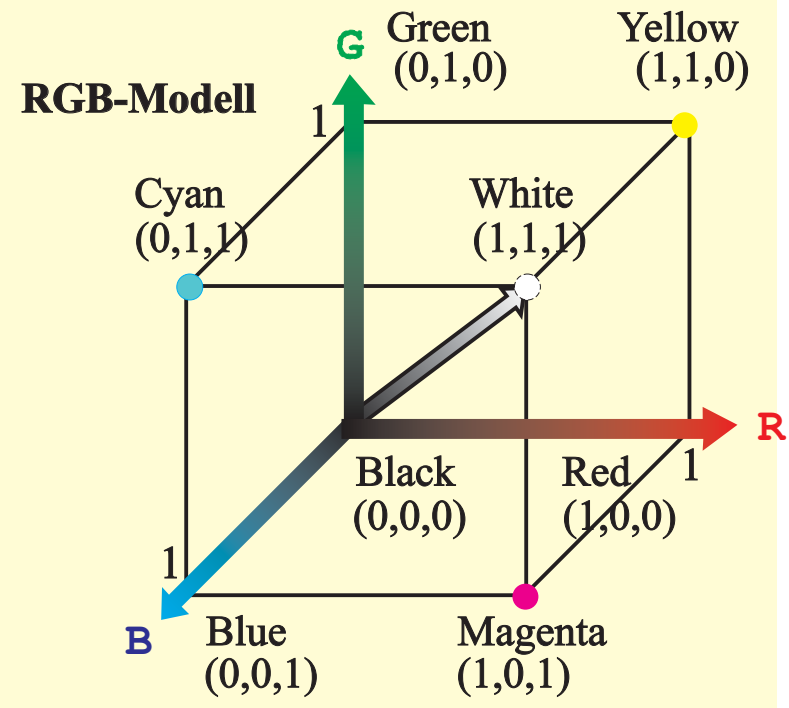
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

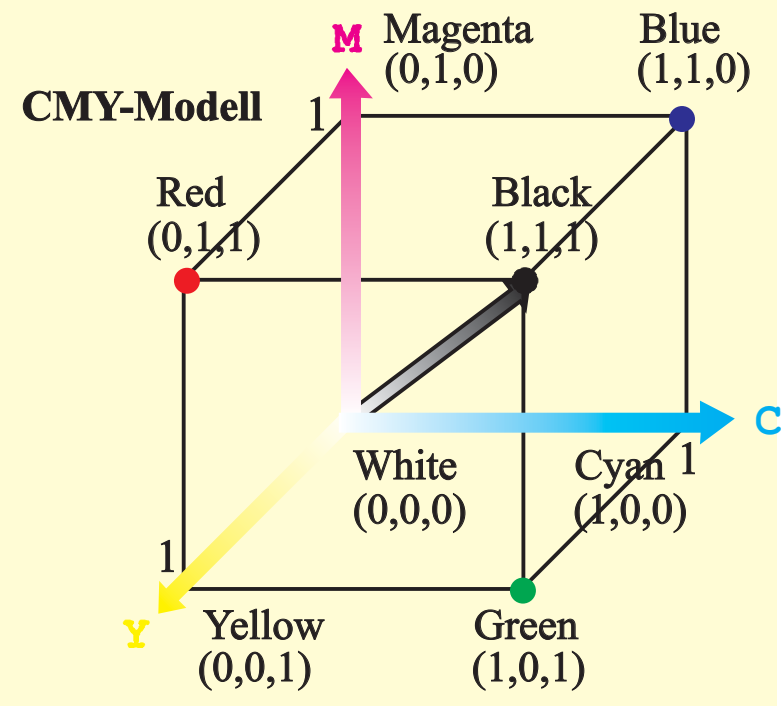
Klassifizierung von Farbmodellen



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

RGB- und CMY-Farbmodelle

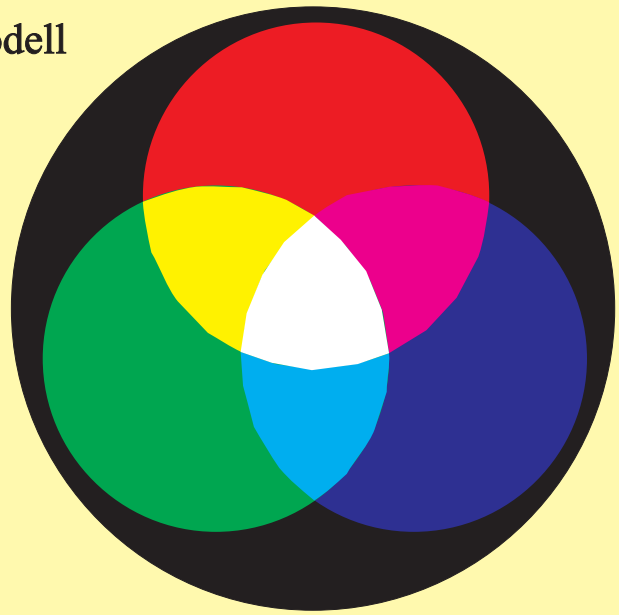




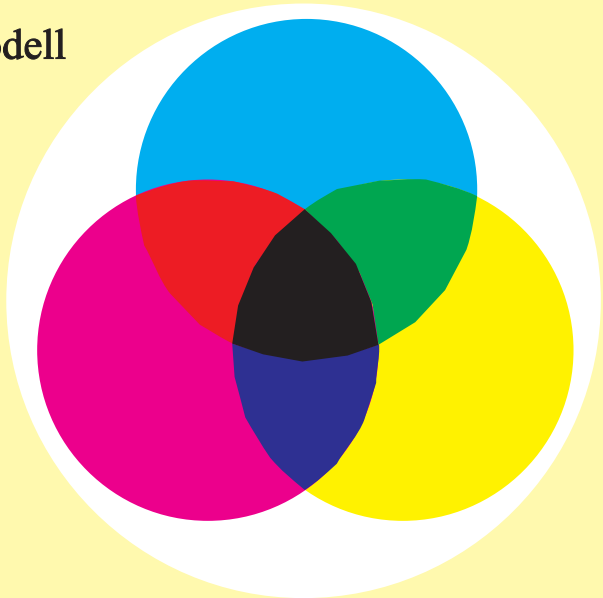
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

RGB- und CMY-Farbmodell

RGB-Modell



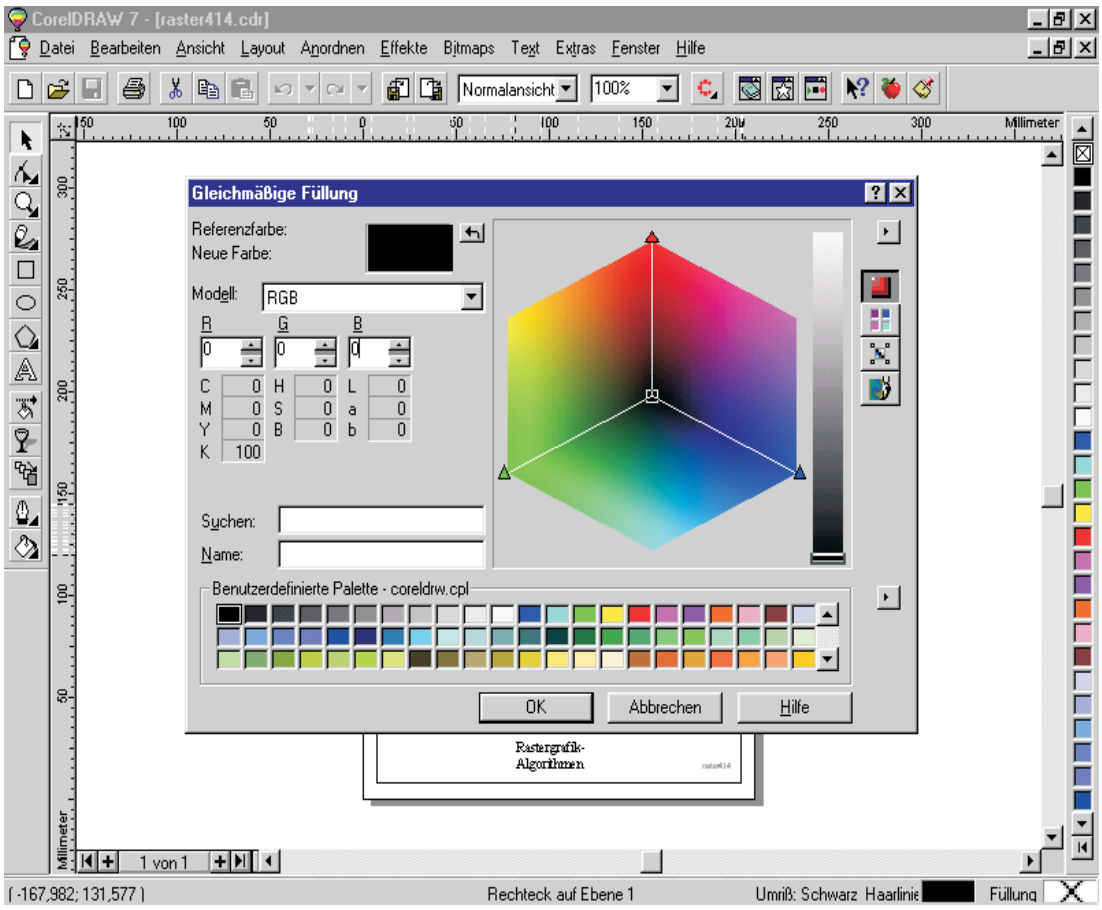
CMY-Modell



Rastergrafik-
Algorithmen

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

RGB-Modell (CorelDRAW)

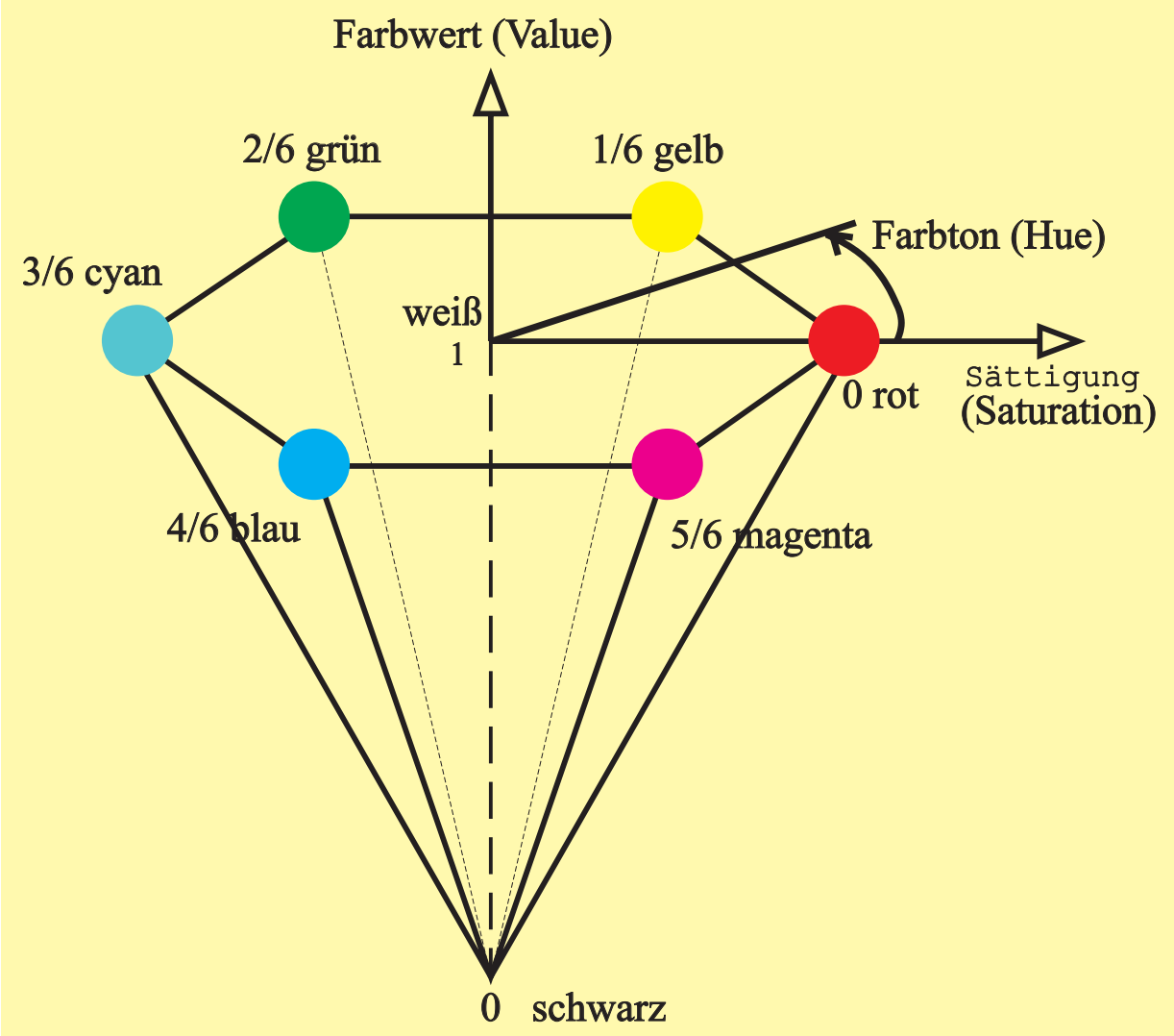


Rastergrafik-
Algorithmen

raster414

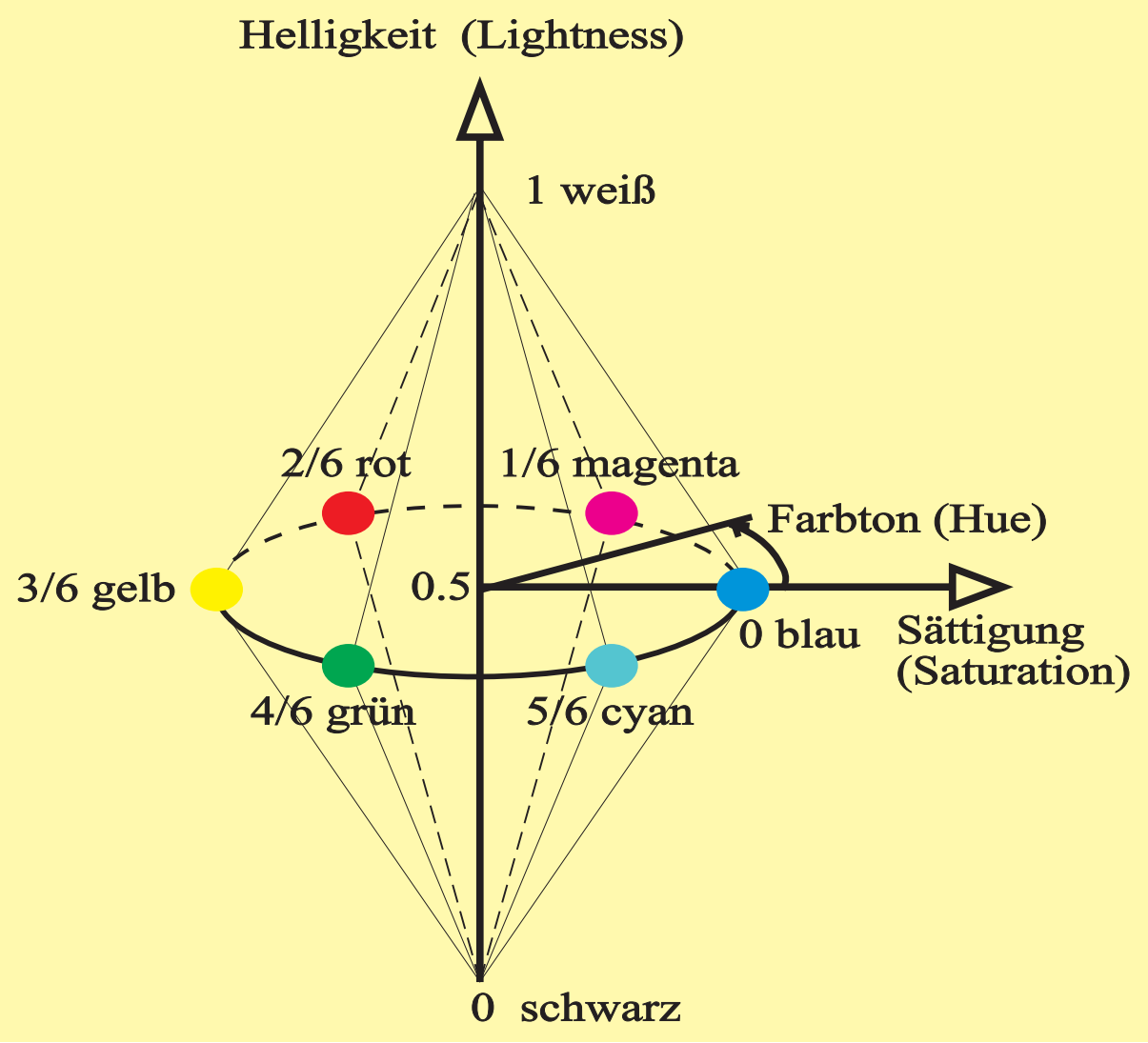
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

HSV-Farbmodell



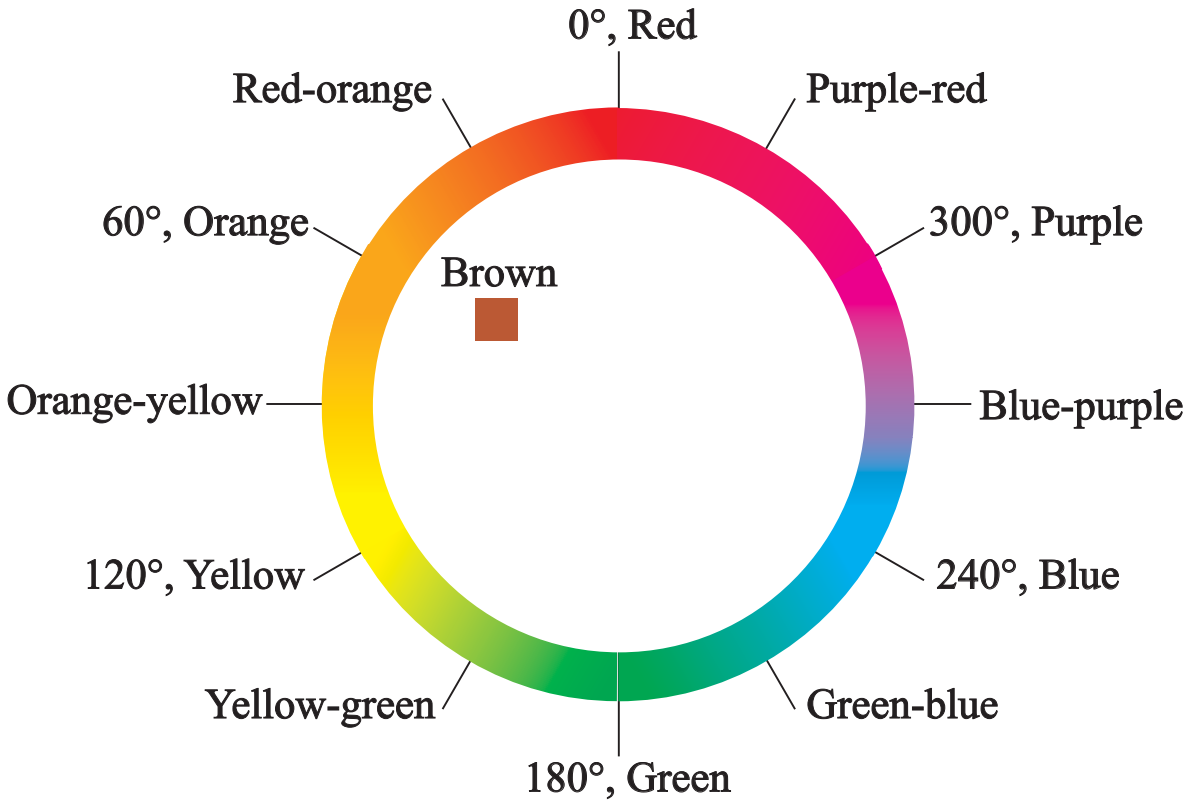
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

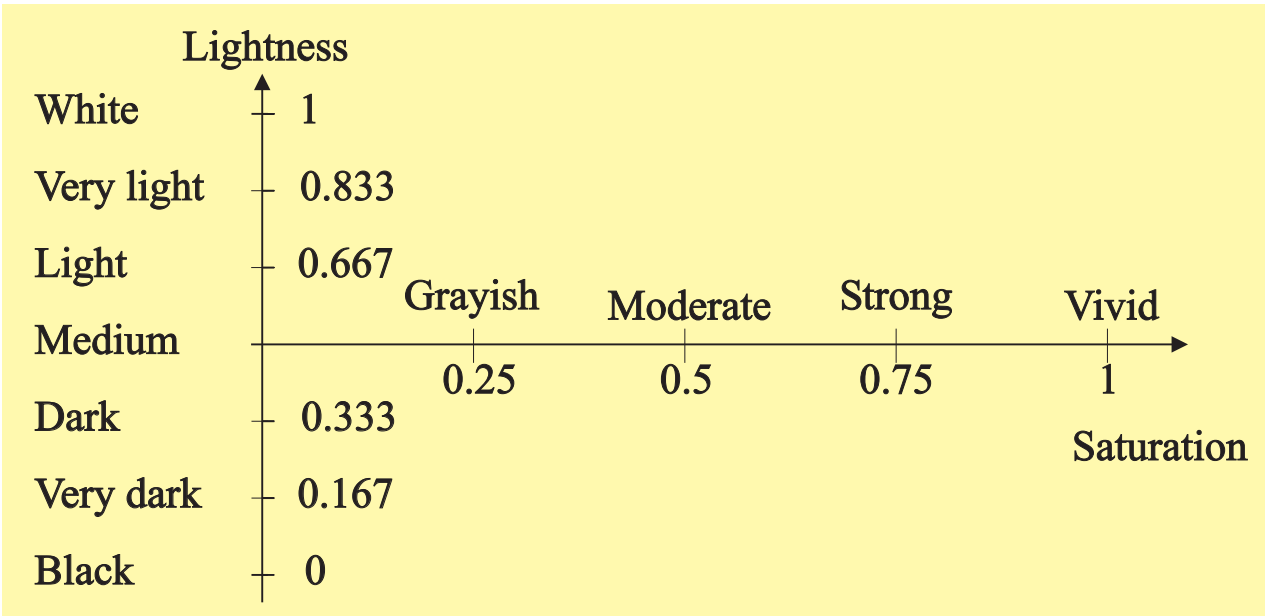
HLS-Farbmodell



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

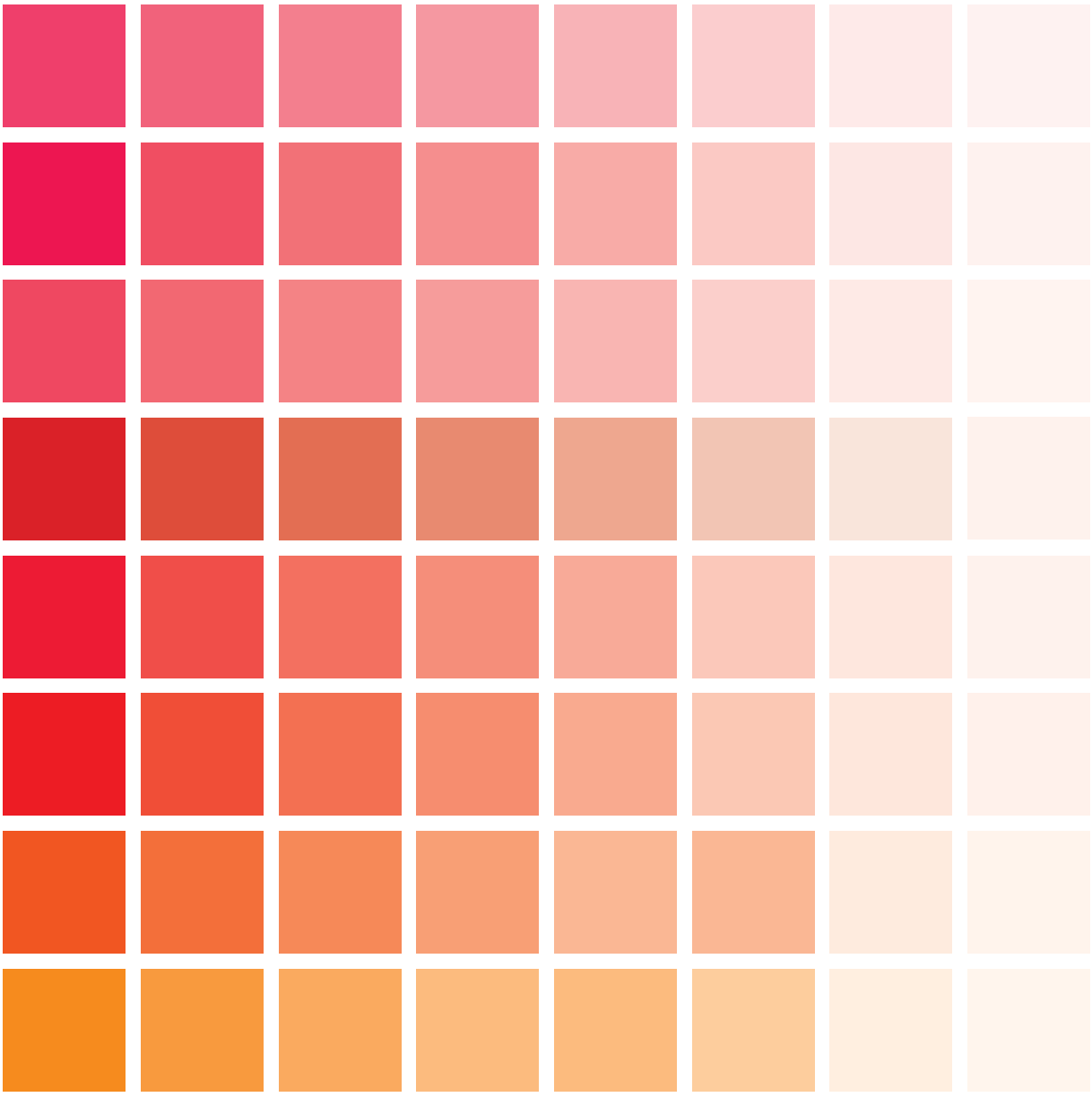
CNS-Farbmodell





Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

PANTONE Matching System

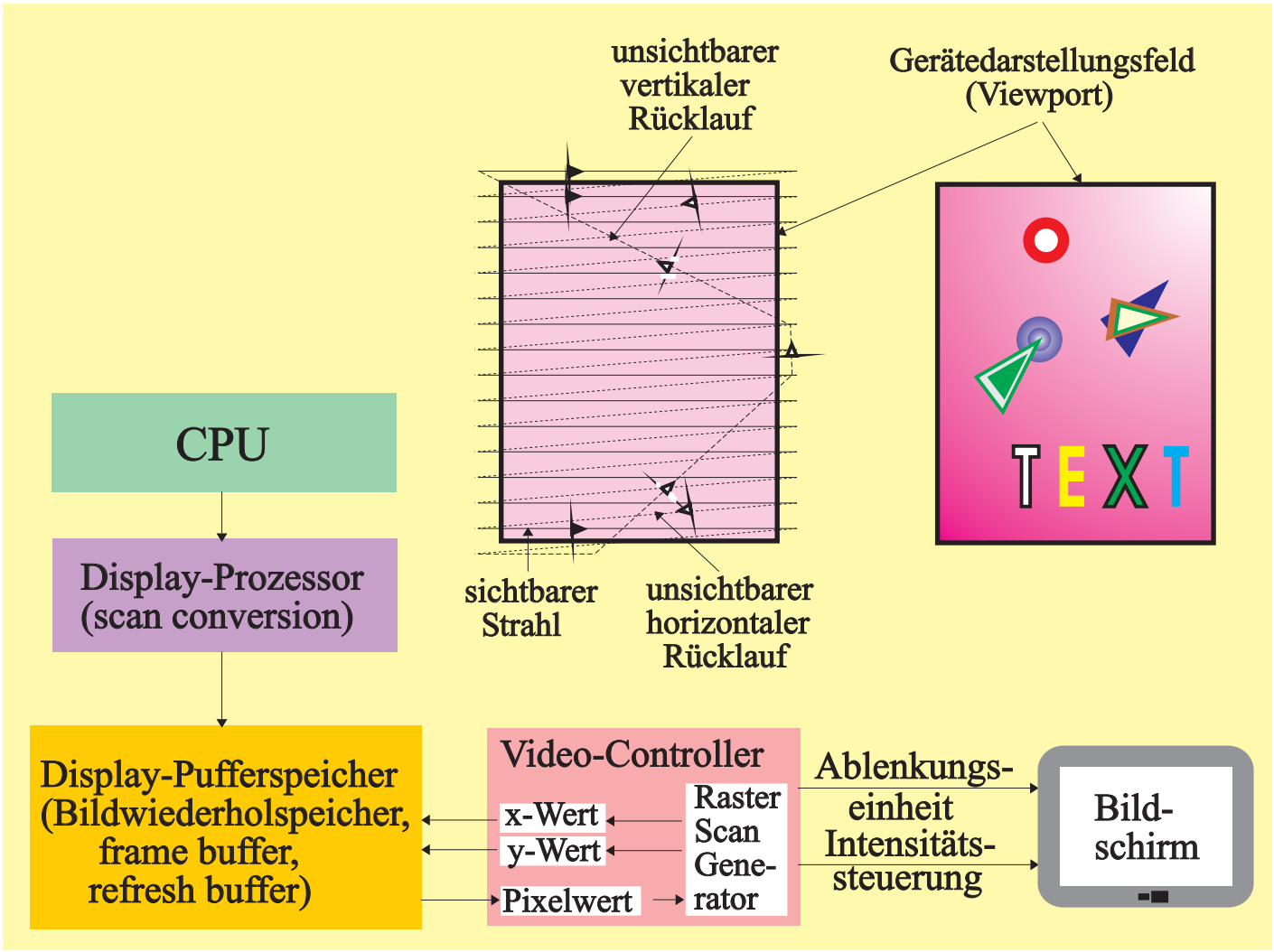


Rastergrafik-
Algorithmen

raster418

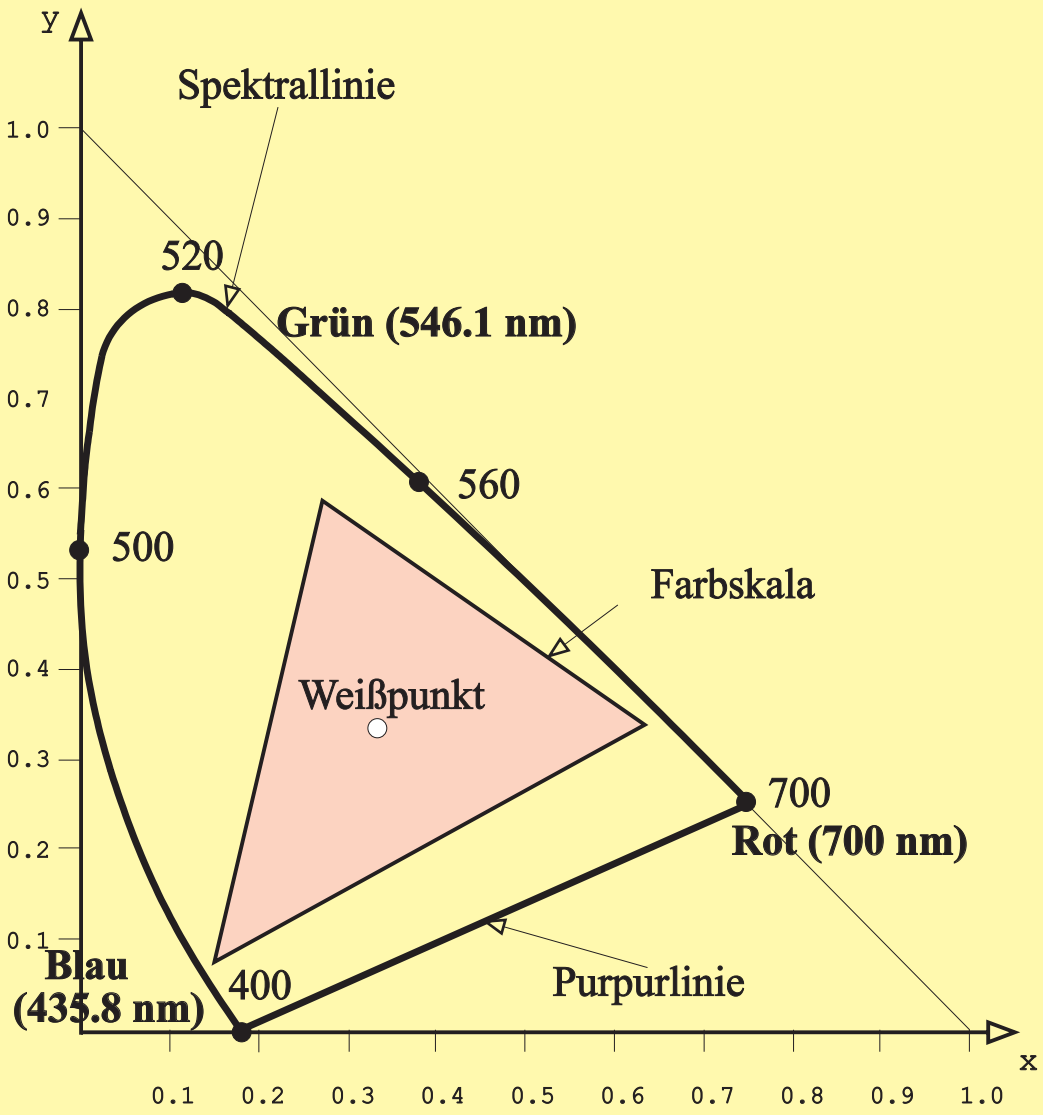
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Rastergrafiksystem
(Raster Scan System)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Farbskala eines Farbbildschirms

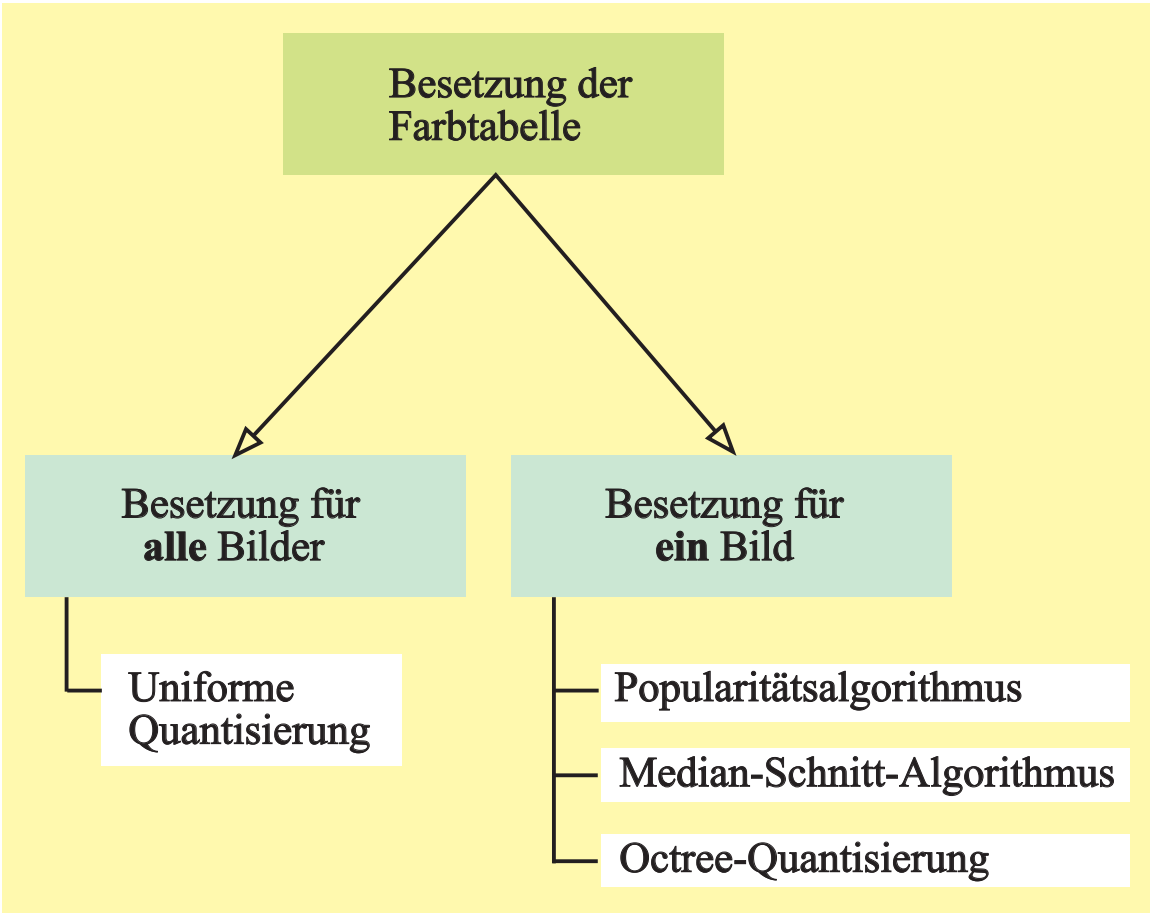


x-y-Koordinaten eines Farbbildschirms:

Rot = (0.628, 0.346)
Grün = (0.268, 0.588)
Blau = (0.150, 0.070)
Weiß = (0.333, 0.333)

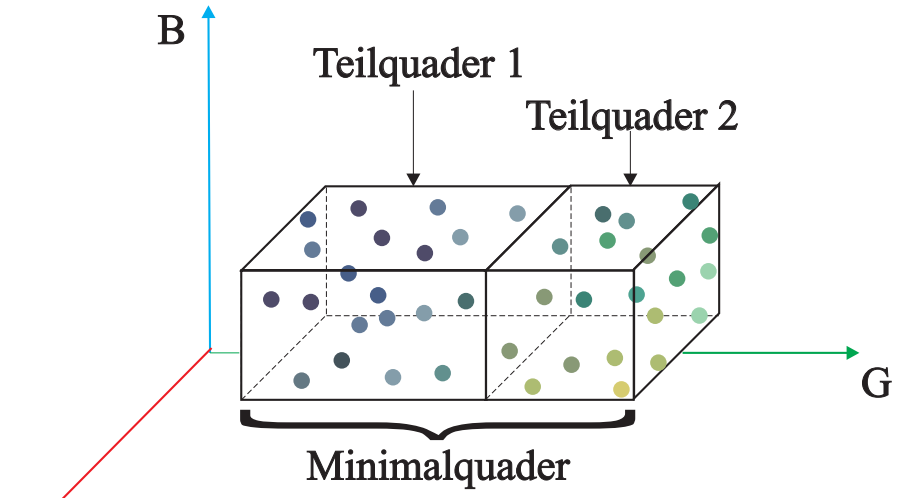
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Besetzung der Farbtabelle



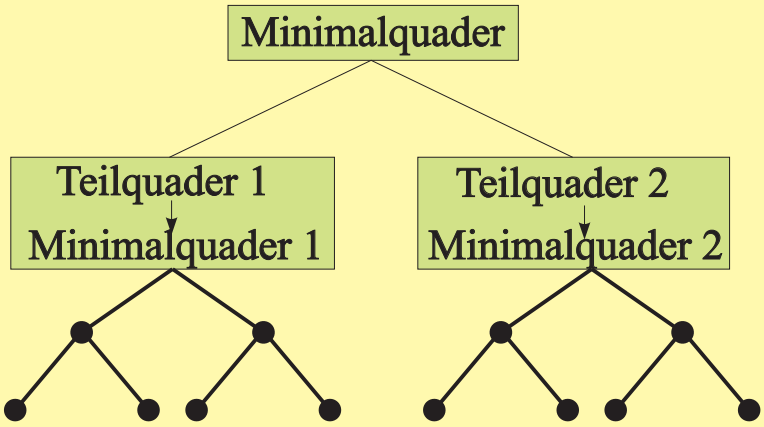
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Median-Schnitt-Algorithmus



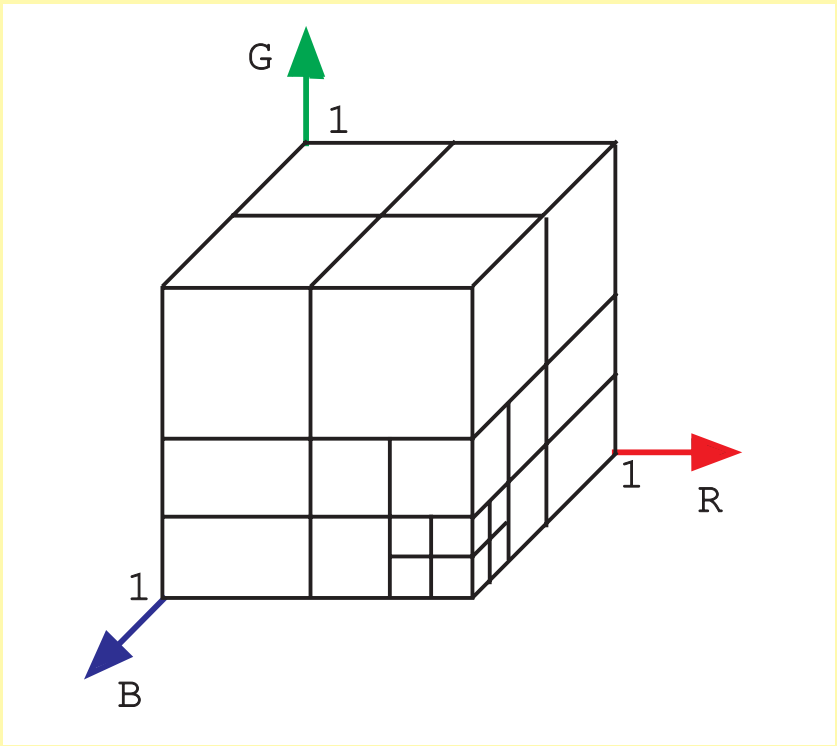
Farbwert des Teilquaders: $F = \frac{1}{n} \sum_{i=1}^{n} \bar{F}_i$

Rekursive Unterteilung mit dem binären Baum



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Octree-Quantisierung

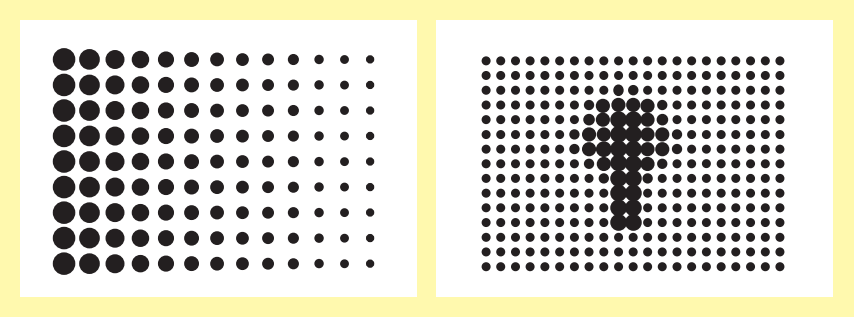


Wiederholte Unterteilung der Farbwürfel in jeweils acht gleichgroße Teilwürfel bis jeder Teilwürfel nur noch **ein** Pixel enthält.

Parallel dazu Reduzierung auf die maximal mögliche Farbanzahl (z.B. 256) durch Mittelwertbildung der entfernten Farben (Blätter).

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Halbton-Verfahren (Prinzip)



Halbtonverfahren in der Computergrafik

- Halbton-Simulation
- Dither-Verfahren
- Fehlerverteilungs-Verfahren
- Fehlerdiffusions-Verfahren

Rastergrafik-
Algorithmen

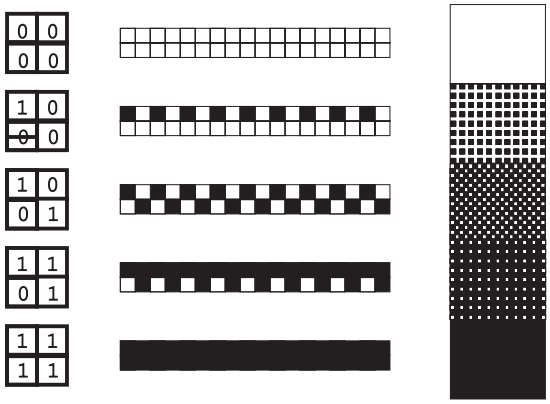
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Halbton-Simulation

Prinzip: Unterteilung des Bildschirms in (meist quadratische) Bereiche mit m Zeilen und n Spalten.
Damit optische (integrative) Wirkung durch Setzen von Pixeln in jedem Bereich erzielbar (m*n+1 Möglichkeiten).

Beispiel für m = n = 2:

Bildungsvorschrift: $\begin{pmatrix} 1 & 3 \\ 4 & 2 \end{pmatrix}$



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Dither-Matrizen

2nx2n-Dither-Matrix nach Bayer (1973)

$$D_{(2n,2n)} = \begin{pmatrix} 4D_{(n,n)} & 4D_{(n,n)}+2E_{(n,n)} \\ 4D_{(n,n)}+3E_{(n,n)} & 4D_{(n,n)}+E_{(n,n)} \end{pmatrix}$$

$E_{(n,n)}$ ist eine quadratische n-reihige Matrix, deren Komponenten alle den Wert 1 besitzen.

2x2-Dither-Matrix

$$D_{(2,2)} = \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}$$

4x4-Dither-Matrix

$$D_{(4,4)} = \begin{pmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}$$

8x8-Dither-Matrix

$$D_{(8,8)} = \begin{pmatrix} 0 & 32 & 8 & 40 & 2 & 34 & 10 & 42 \\ 48 & 16 & 56 & 24 & 50 & 18 & 58 & 26 \\ 12 & 44 & 4 & 36 & 14 & 46 & 6 & 38 \\ 60 & 28 & 52 & 20 & 62 & 30 & 54 & 22 \\ 3 & 35 & 11 & 43 & 1 & 33 & 9 & 41 \\ 51 & 19 & 59 & 27 & 49 & 17 & 57 & 25 \\ 15 & 47 & 7 & 39 & 13 & 45 & 5 & 37 \\ 63 & 31 & 55 & 23 & 61 & 29 & 53 & 21 \end{pmatrix}$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

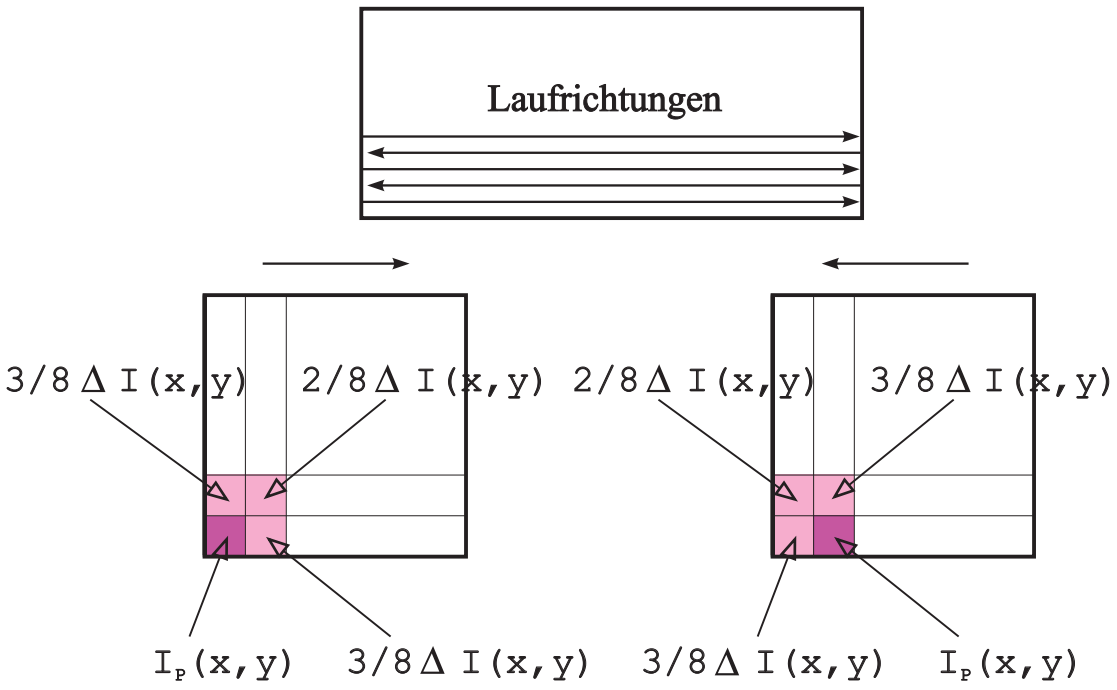
Fehlerverteilungsverfahren nach Floyd-Steinberg

Prinzip: Verteilung eines bei der Farbdarstellung gemachten Fehlers

$$\Delta I(x,y)=I_P(x,y)-I(x,y)$$

auf mehrere Pixel.

$I(x,y)$ ist der exakte, $I_P(x,y)$ der dargestellte Wert.



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Fehlerdiffusions-Verfahren (Teil 1)

Jedes Pixel P(x,y) gehört zu einer Klasse

$$m_{x \bmod n, y \bmod n}$$

Dabei sind $m_{i,j}$, $1 \le i,j \le n$, die Komponenten der Diffusionsmatrix $M_{(n,n)}$.

Diffusionsmatrix $\overline{M}_{(4,4)}$ (Beispiel)

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

Baron, kann den Intensitätsfehler des Pixels nicht verteilen

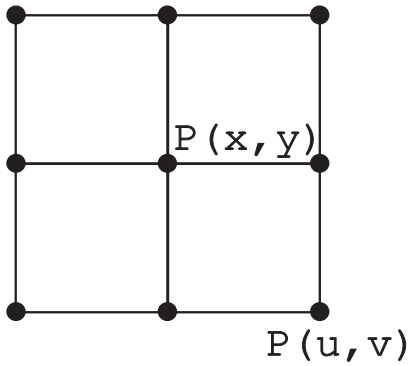
Fast-Baron, kann den Intensitäts-fehler des Pixels nur auf ein Nach-barpixel verteilen

Prinzip: Berechnung aller zu einer der möglichen n*n Klassen gehörenden Pixel.
Verteilung des bei der Darstellung ver-ursachten Fehlers auf die noch nicht berechneten Nachbarpixel entsprechend ihres Abstandes.

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Fehlerdiffusions-Verfahren (Teil 2)

Pixelabstände



Rechtwinklig benachbart:

$(x-u)^2+(y-v)^2=1$

Diagonal benachbart:

$(x-u)^2+(y-v)^2=2$

Den rechtwinklig benachbarten Pixeln wird gewöhnlich der doppelte Anteil des Fehlers zugewiesen wie den diagonal benachbarten.

Verbesserung der Bildqualität

$$I'(x,y) = \frac{I(x,y) - \alpha \bar{I}(x,y)}{1-\alpha}$$

$$\bar{I}(x,y) = \frac{1}{9} \sum_{u=x-1}^{x+1} \sum_{v=y-1}^{y+1} I(u,v)$$

Für $\alpha=0.9$ ergibt sich

$$I'(x,y) = 9I(x,y) - \sum_{\substack{u,v \\ 0<(x-u)^2+(y-v)^2<3}} I(u,v)$$

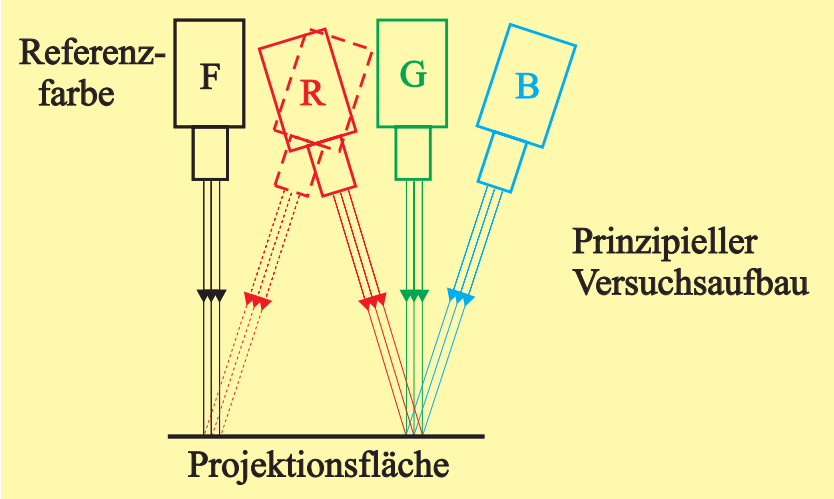
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

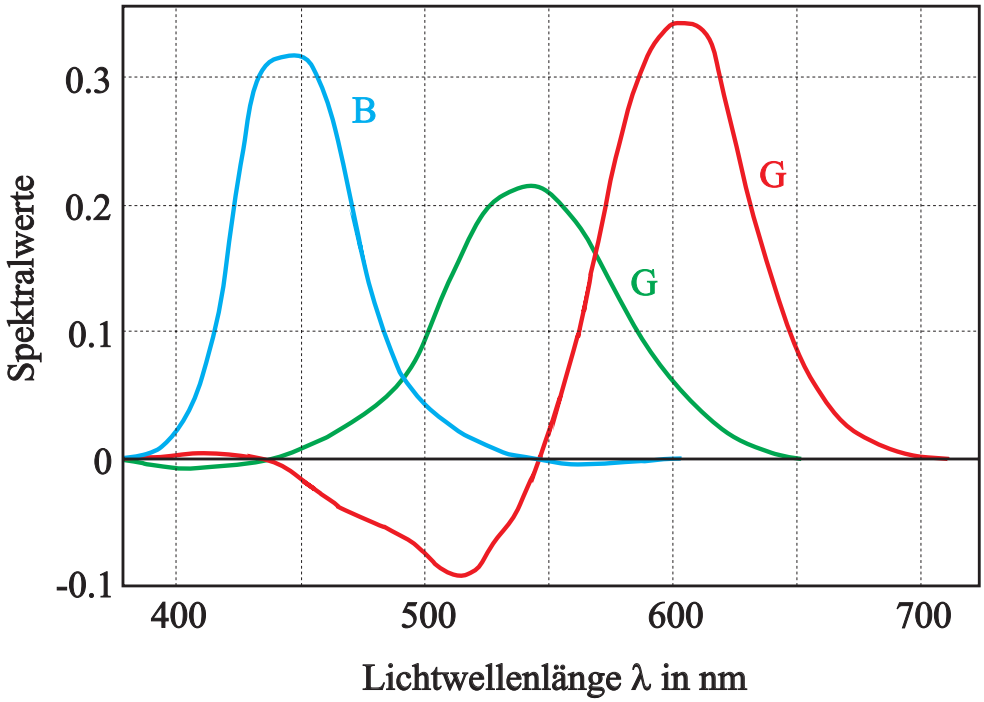
Farbnamen und RGB-Werte in CORELDRAW

Farbname	Rot	Grün	Blau
Eisblau	0.6	1.0	1.0
Himmelblau	0.0	0.8	1.0
Babyblau	0.4	0.6	1.0
Dämmerblau	0.4	0.4	0.8
Blau	0.0	0.0	1.0
Rosa	1.0	0.6	0.8
Sanftrosa	1.0	0.6	0.6
Tropischrosa	1.0	0.4	0.4
Ziegelrot	0.8	0.2	0.0
Rot	1.0	0.0	0.0
Mondgrün	0.8	1.0	0.4
Leuchtendgrün	0.6	1.0	0.0
Grün	0.0	1.0	0.0
Hellviolett	1.0	0.6	1.0
Violett	0.8	0.4	0.8
Magenta	1.0	0.0	1.0
Sand	1.0	0.8	0.6
Pfirsisch	1.0	0.6	0.4

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

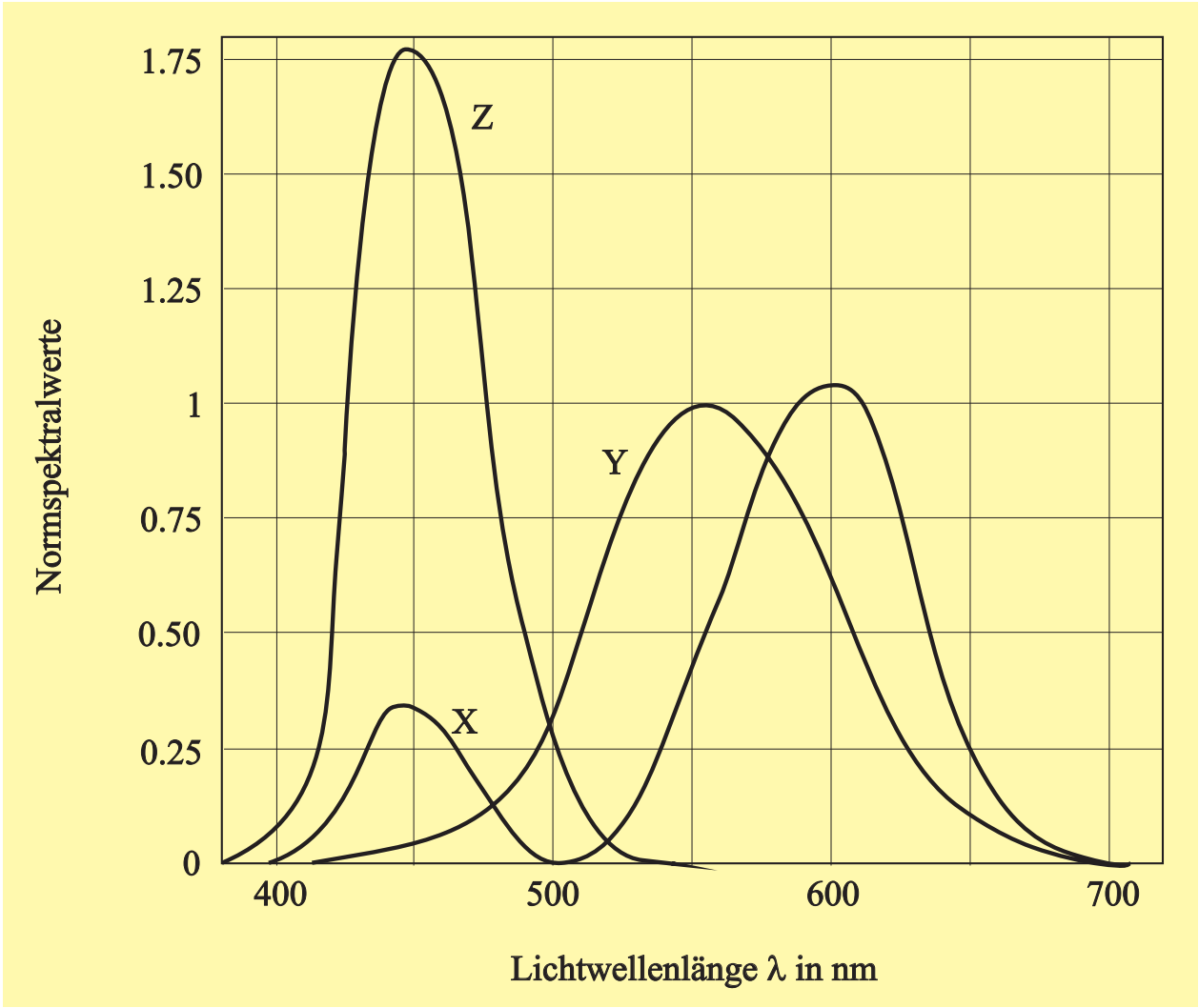
Farbmischkurven





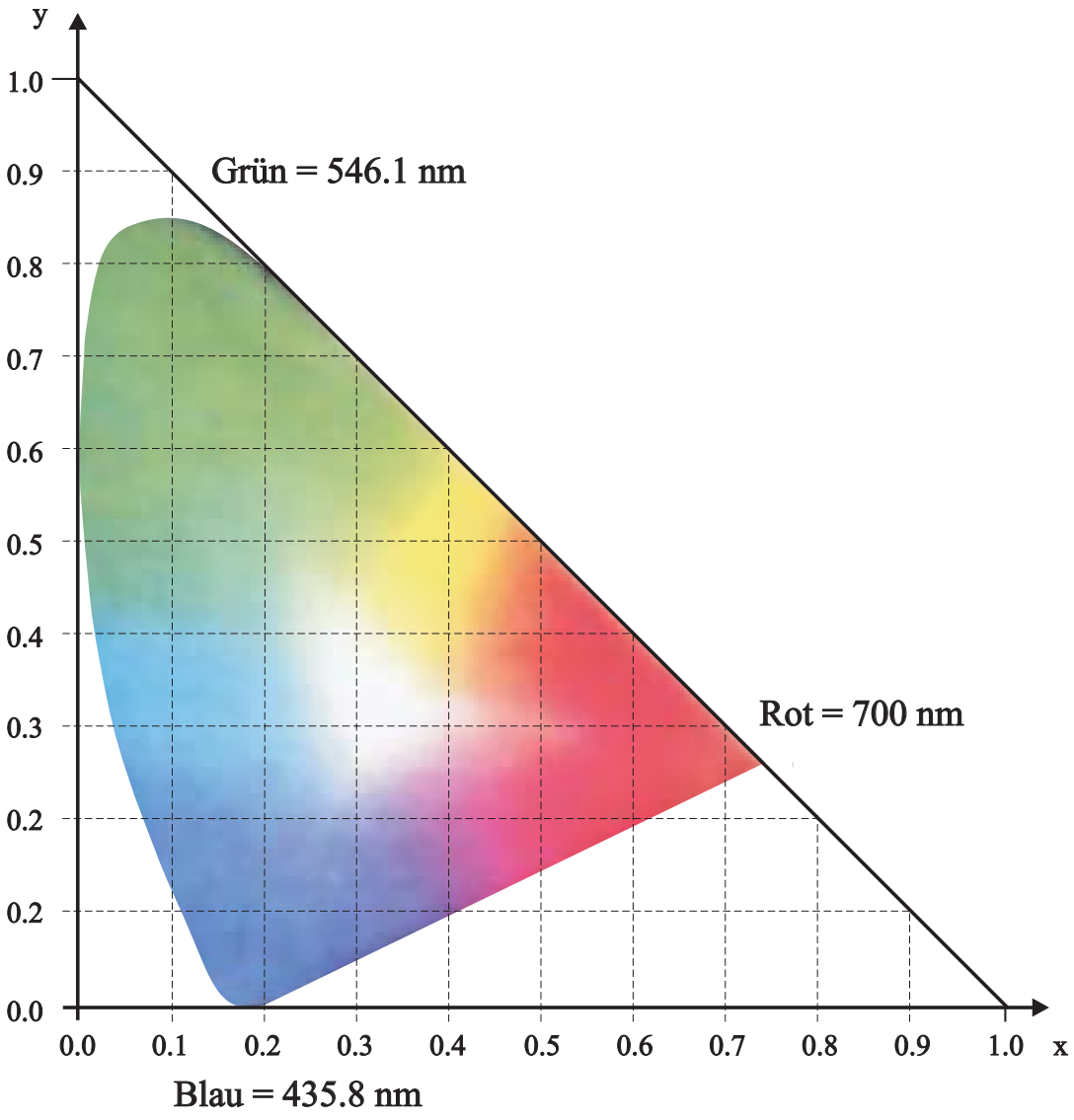
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

CIE-Normfarbmischkurven



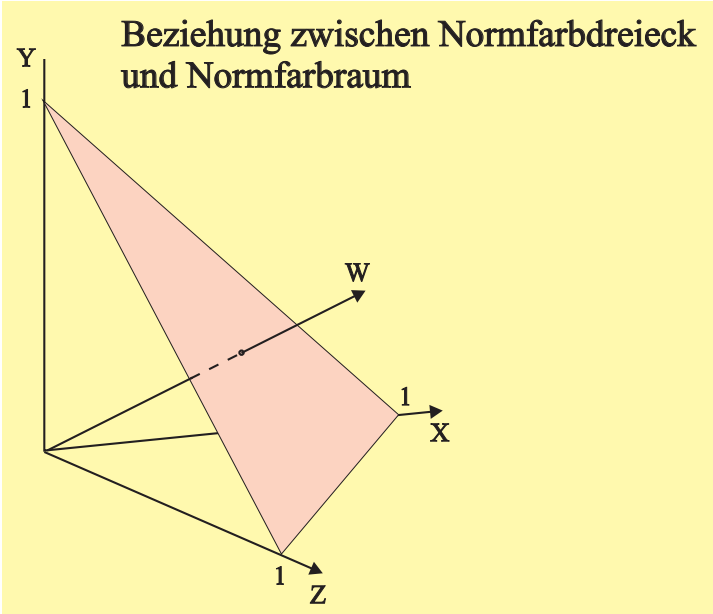
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

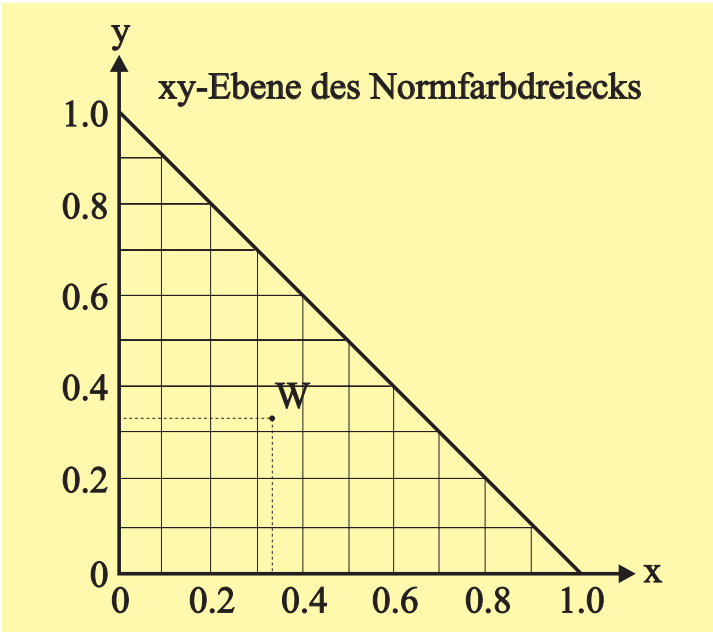
CIEXYZ-Diagramm



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

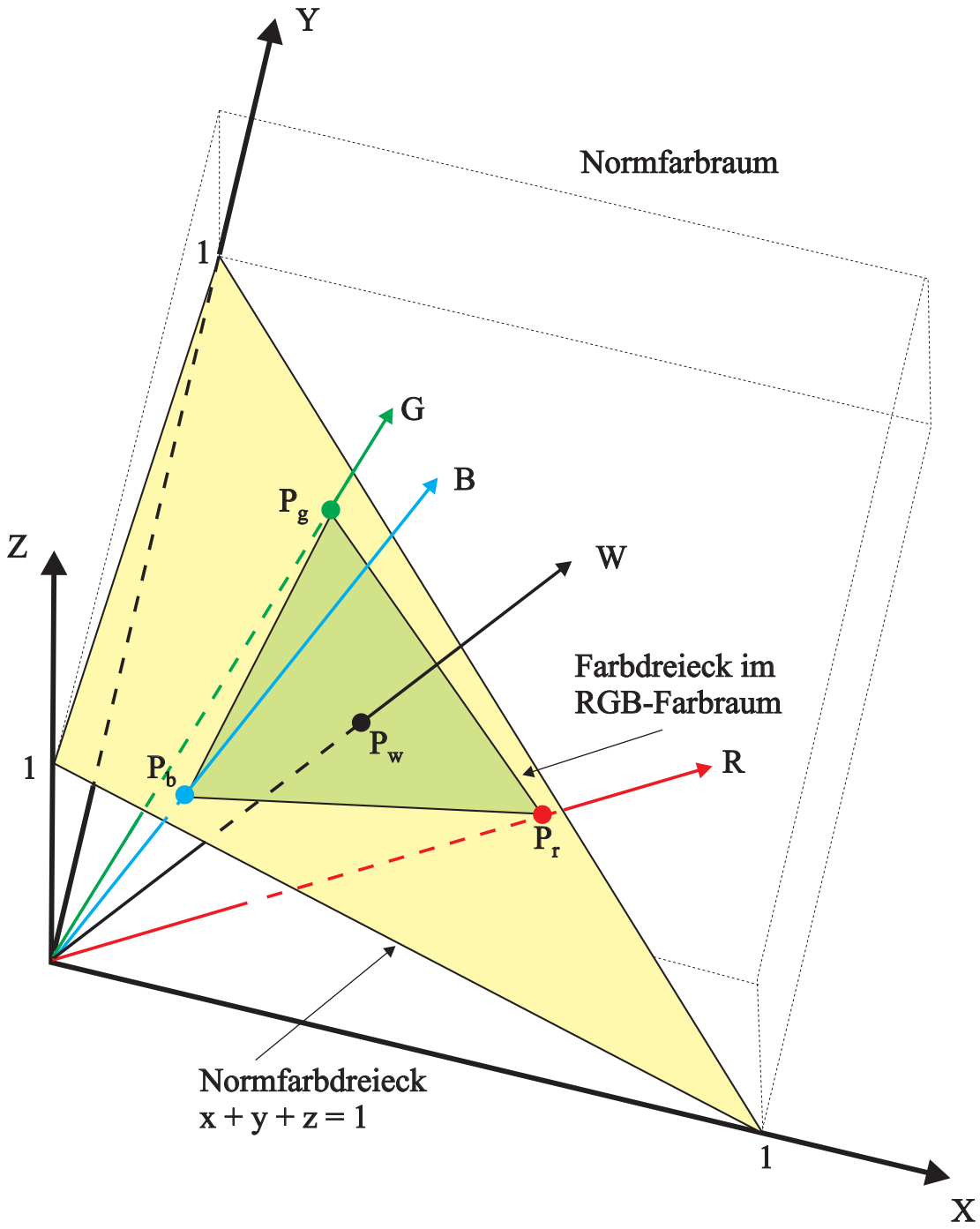
Normfarbdreieck





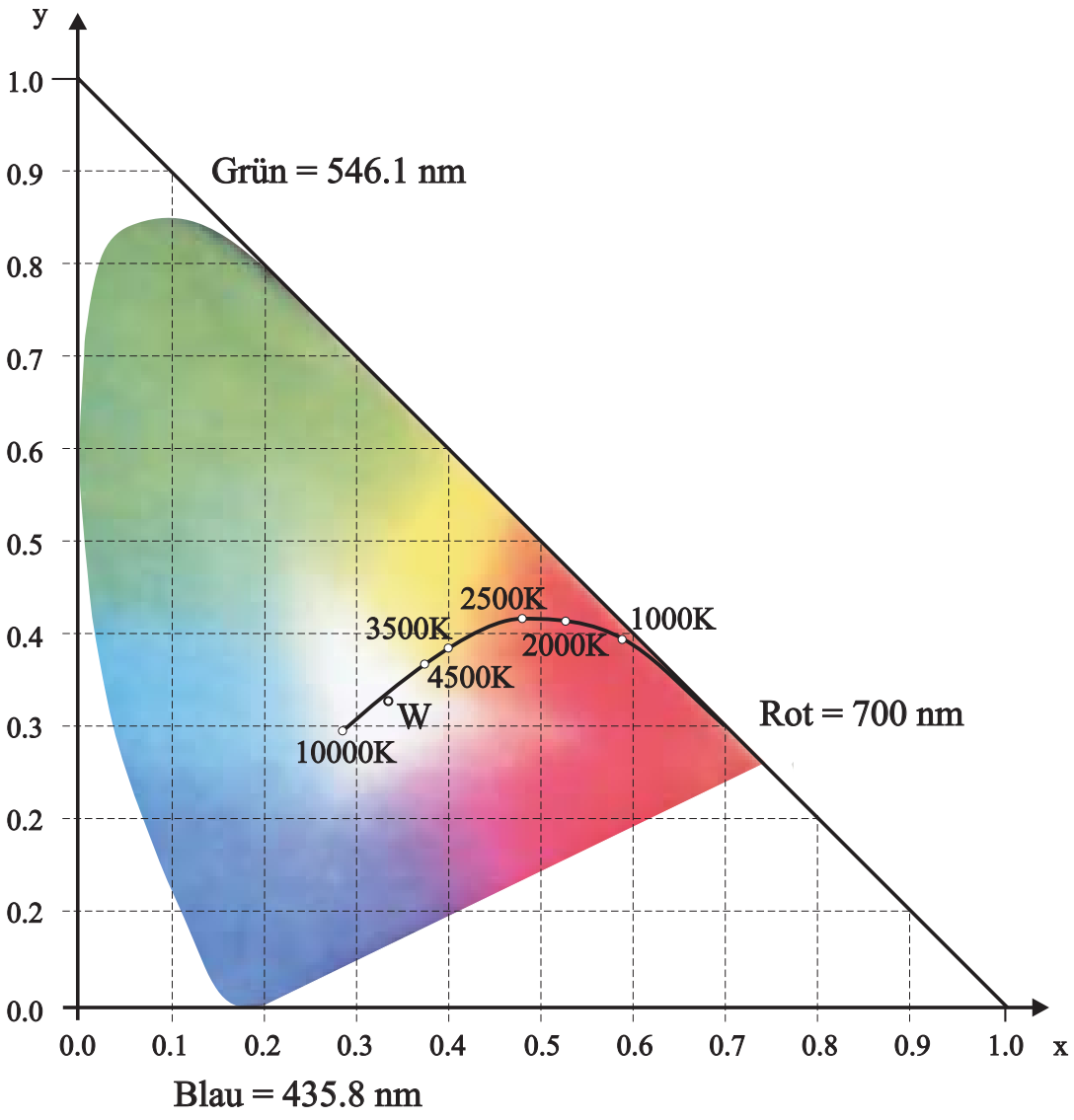
Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

RGB- und Normfarbraum



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Farbtemperatur



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

5. Minderung von Aliasing-Effekten (Anti-Aliasing)

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Aliasing

Ursache

Entstehen falscher Muster durch zu geringe Abtastung bei der Digitalisierung (Rasterung) analoger Daten

Effekte

Stairstepping:	stufenförmige Intensitätsänderung entlang einer Linie (Zacken, jaggies)
Linebreakup:	Unterbrechung von dünnen Linien
Crawling:	Erscheinen/Nichterscheinen von dünnen Linien
Scintillation:	Größenänderung bei der Bewegung von Objekten

Abtasttheorem von Shannon

Bei der Digitalisierung muß die Abtastfrequenz mindestens doppelt so hoch sein wie die höchste in der Quelle vorkommende Frequenz

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Anti-Aliasing-Techniken (Ansätze)

Modifizierter Grundalgorithmus zum Erzeugen der Ausgabeprimitive (modifizierter Bresenham- oder Midpoint-Algorithmus)

Nachbehandlung von erzeugten Bildern mit einem Postprocessing- Algorithmus (Nachfiltern mit oder ohne Supersampling)

Anwendung einer Filterfunktion auf das zu erzeugende Bild vor dessen Ausgabe (Prefiltering)

Nutzung der Hardware-Fähigkeiten moderner Bildschirme (Pixel Phasing)

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Anti-Aliasing-Techniken
Modifizierter Grundalgorithmus

Modelle für die Berechnung der Pixelintensitäten

ungewichtete Flächenberücksichtigung (Bresenham)

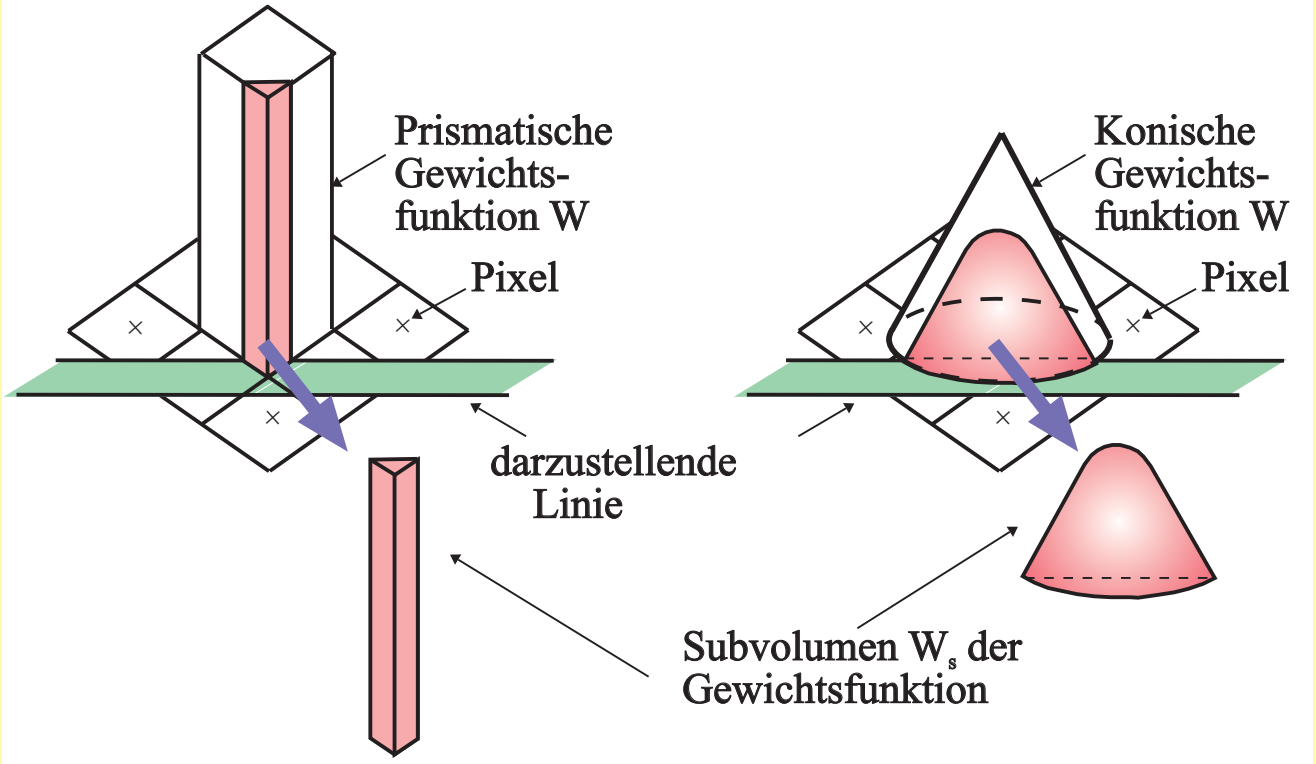
Intensität des Pixels ist proportional zum Grad ΔA seiner Überdeckung

$I=f(\Delta A)$

gewichtete Flächenberücksichtigung (Gupta-Sproull)

Intensität des Pixels ist proportional zum Grad ΔA seiner Überdeckung und zum Abstand h des überdeckten Bereiches ΔA vom Pixelzentrum

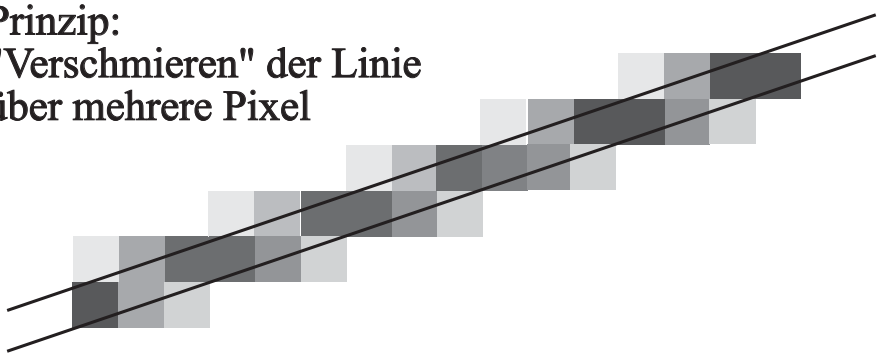
$I=f(\Delta A,h)$



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Anti-Aliasing-Techniken
Modifizierter Grundalgorithmus (Bresenham)

Prinzip:
"Verschmieren" der Linie
über mehrere Pixel

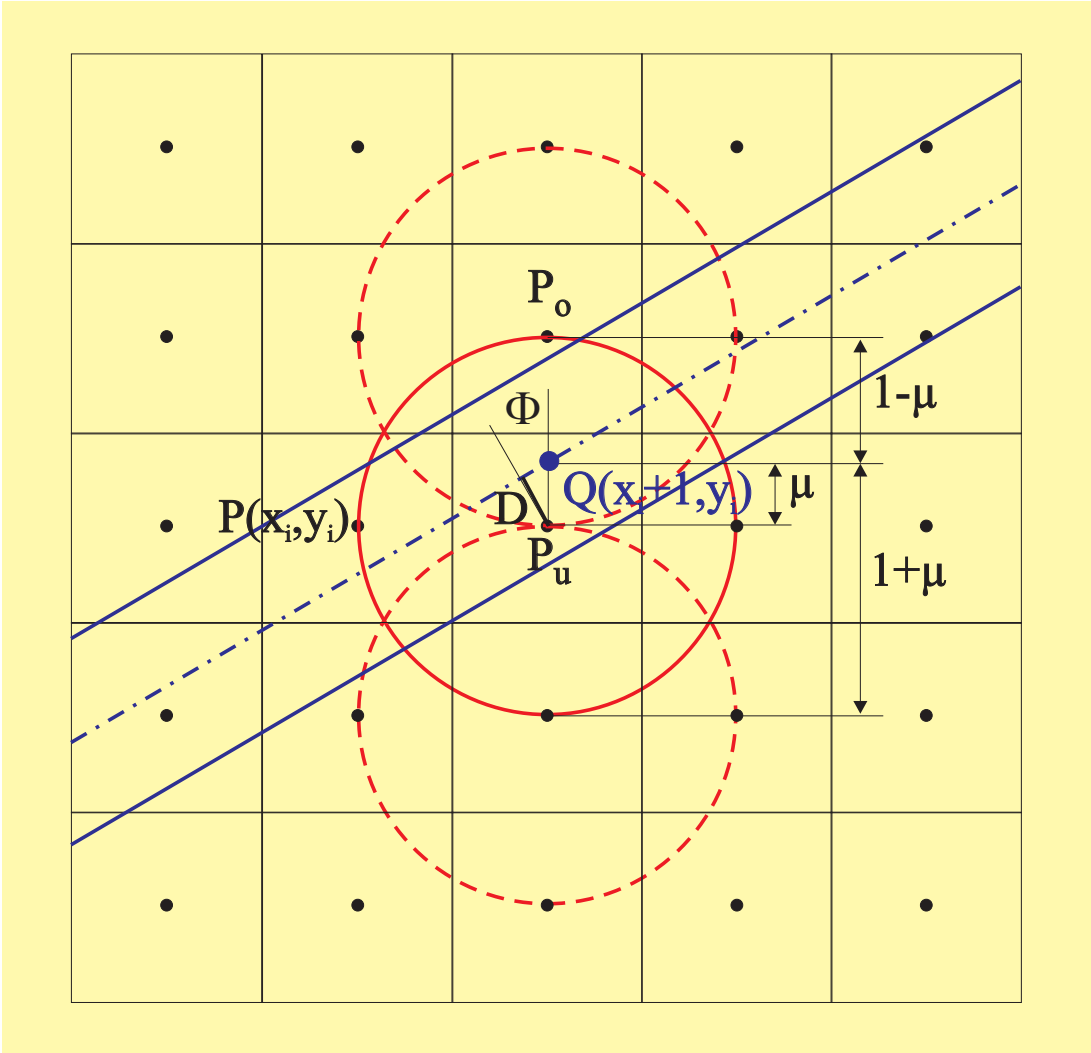


Modifikation der Pixelintensität in Abhängigkeit vom Abstand zur Ideallinie. Die berechnete Intensität des zu setzenden Pixels wird um einen abstandsabhängigen Korrekturwert ΔI erniedrigt, das benachbarte Pixel erhält diesen Wert ΔI.

```
procedure BresenhamAliasing(x1,y1,x2,y2,Intmax:
word);
  var error,x,y,dx,dy,IntPix: integer;
Begin
  dx:=x2-x1;  dy:=y2-y1;  error:=-dx div 2;
  y:=y1;      IntPix:=0;
  for x:=x1 to x2 do
  Begin
    Error:=error+dy;
    if error>=0 then
    Begin
      Y:=y+1;
      Error:=error-dx;
    End;
    PutPixel(x,y,Intmax-IntPix);
    PutPixel(x,y+1,IntPix);
    IntPix:=error*Intmax div dx;
  End;
end {BresenhamAliasing};
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Gupta-Sproull-Algorithmus (1)



Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Gupta-Sproull-Algorithmus (2)

Kegel-Filterfunktion f(D) für D=0 .. 1.5

D	f(D)	D	f(D)
0	0.789	13/16	0.228
1/16	0.775	14/16	0.184
2/16	0.760	15/16	0.145
3/16	0.736	16/16	0.110
4/16	0.703	17/16	0.080
5/16	0.662	18/16	0.056
6/16	0.613	19/16	0.036
7/16	0.558	20/16	0.021
8/16	0.500	21/16	0.010
9/16	0.441	22/16	0.004
10/16	0.383	23/16	0.001
11/16	0.328	24/16	0
12/16	0.276		

Ansatz und Bedingungen für die Approximation der Kegel-Filterfunktion:

$$f(D)=f_0 + a_2D^2 + a_3D^3 + a_4D^4 + a_5D^5$$

$f(0) = f_0 = 0.789$ $\quad f(1/2) = f_1 = 0.500$
$f(1) = f_2 = 0.110$ $\quad f(3/2) = f'(3/2) = 0$

Berechnete Koeffizienten der approximierten Filterfunktion:

$$a_2 = [-333*f_0 + 486*f_1 - 243*f_2]/27$$
$$a_3 = [\ 646*f_0 - 1134*f_1 + 810*f_2]/27$$
$$a_4 = [-444*f_0 + 864*f_1 - 756*f_2]/27$$
$$a_5 = [\ 104*f_0 - 216*f_1 - 216*f_2]/27$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Gupta-Sproull-Algorithmus (3)

```
function PixelIntensitaet_Gupta(D: double): longint;
{ Filterfunktion nach Gupta-Sproull }
begin
  D:=(1-abs(0.789+(((-0.08089*D-0.05467)*D+1.17756)*D-1.721)*D*D));
  PixelIntensitaet_Gupta:=RGBColor(D,D,D);
end {PixelIntensitaet_Gupta};

procedure AntiAliasingMidpointLine_Gupta(Projekt: TForm; XA,YA,XB,YB: Word);
  var DX,DY,R,C,G,INC1,INC2,Zwei_v_DX: integer;
      Nenner,Zwei_DX_Nenner: double;
begin
  DX:=XB-XA;   DY:=YB-YA;
  C:=XA;       R:=YA;
  G:=2*DY-DX;
  INC1:=2*DY;  INC2:=2*(DY-DX);
  Zwei_v_DX:=0;
  Nenner:=1/(2*sqrt(1.0*DX*DX+1.0*DY*DY));
  Zwei_DX_Nenner:=2*DX*Nenner;
  Projekt.Canvas.Pixels[C,R  ]:=PixelIntensitaet_Gupta(0);
  Projekt.Canvas.Pixels[C,R+1]:=PixelIntensitaet_Gupta(Zwei_DX_Nenner);
  Projekt.Canvas.Pixels[C,R-1]:=PixelIntensitaet_Gupta(Zwei_DX_Nenner);
  while C<XB do
  begin
    if G<0 then begin Zwei_v_DX:=G+DX; G:=G+INC1; C:=C+1 end
           else begin Zwei_v_DX:=G-DX; G:=G+INC2; C:=C+1; R:=R+1 end;
    Projekt.Canvas.Pixels[C,R  ]:=PixelIntensitaet_Gupta(Zwei_v_DX*Nenner);
    Projekt.Canvas.Pixels[C,R+1]:=PixelIntensitaet_Gupta(Zwei_DX_Nenner
                                                         -Zwei_v_DX*Nenner);
    Projekt.Canvas.Pixels[C,R-1]:=PixelIntensitaet_Gupta(Zwei_DX_Nenner
                                                         +Zwei_v_DX*Nenner);
  end
end {AntiAliasingMidpointLine_Gupta};
```

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Anti-Aliasing-Techniken
Filtermethode

Charakteristik

Das darzustellende Bild wird mit der Auflösung des Bildschirms als Rasterbild erzeugt.

Aus diesem Rasterbild wird das auf dem Bildschirm sichtbare Bild durch Mittelung der Intensitäten auch der Nachbarpixel mittels eines Filters berechnet.

Dadurch werden die Intensitäten der Pixel des Rasterbildes über mehrere Pixel des Bildschirmbildes "verschmiert".

Filter (Beispiel):

1/36	4/36	1/36
4/36	16/36	4/36
1/36	4/36	1/36

Gesamtintensität $I_P(x,y)$ des Pixels P(x,y)

$$I_P(x,y)=[\ I(x-1,y-1) + 4I(x,y-1) + I(x+1,y-1) + 4I(x-1,y) + 16I(x,y) + 4I(x+1,y) + I(x-1,y+1) + 4I(x,y+1) + I(x+1,y+1)]/36$$

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Anti-Aliasing-Techniken
Supersampling-Methoden

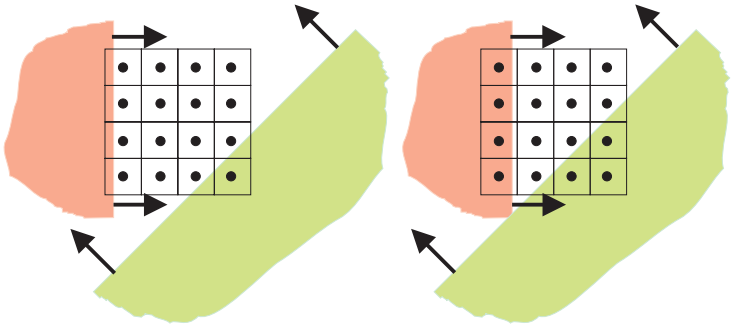
Supersampling:
Pro Pixel werden mehrere Abtastpunkte verwendet. Das auszugebende Bild wird für eine höhere Auflösung gerastert. Üblich ist in beiden Richtungen eine 2- bis 4-mal größere Auflösung. Der Intensitätswert des Bildschirmpixels ist dann der Mittelwert der zugehörigen Rasterpixel.

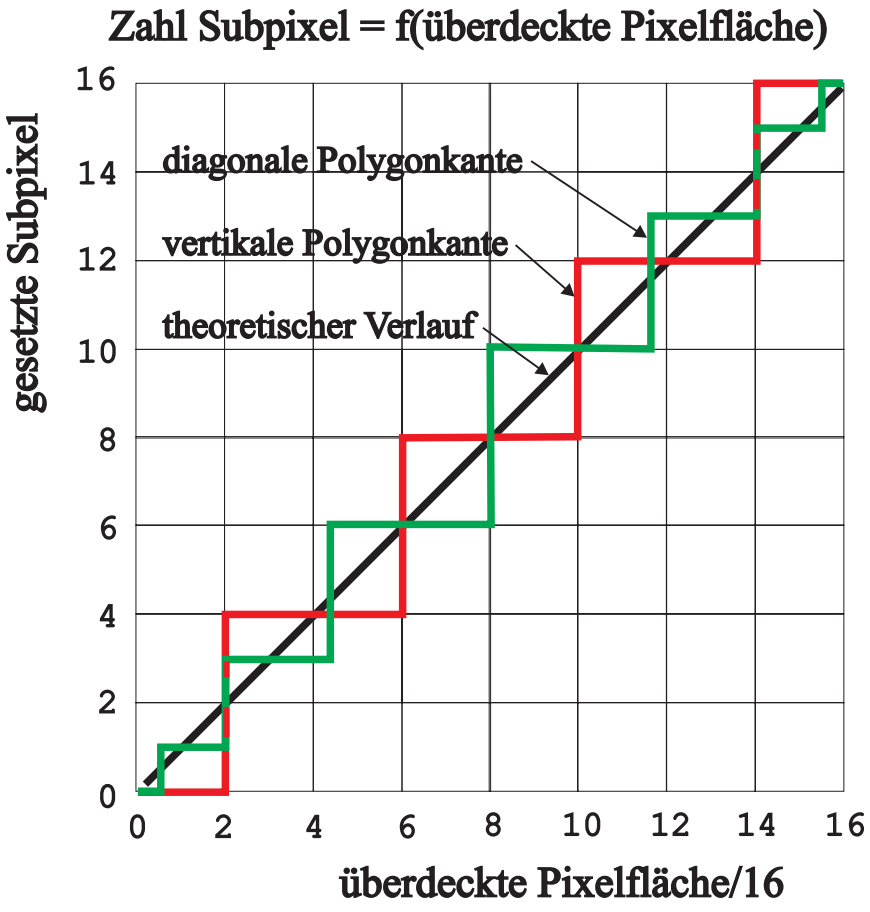
Adaptives Supersampling:
Es arbeitet nur in Bereichen, in denen Aliasing-Effekte auftreten können.

Stochastisches Supersampling:
Die Position der Abtastpunkte wird jeweils um einen zufälligen Betrag geändert. Die eingeführten Störungen mindern die Aliasing-Effekte.

Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

Reihenfolge der Überdeckung der Subpixel

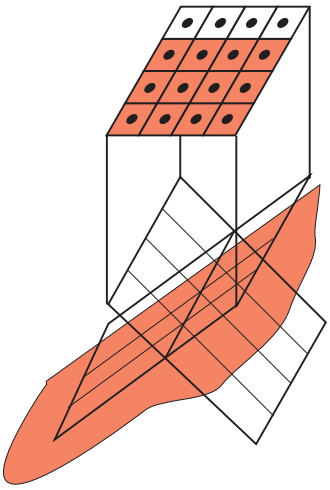




Technische Universität Dresden
Fakultät Informatik
Prof. Dr.-Ing.habil. Orlamünder

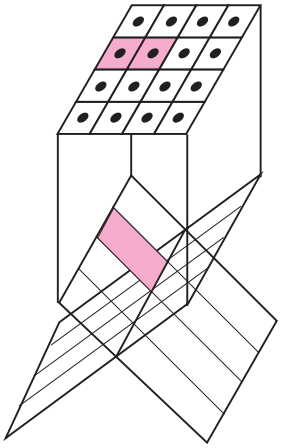
EXACT-Algorithmus
Wirkung der Prioritätsmaske P auf die Subpixelmasken A und B

A

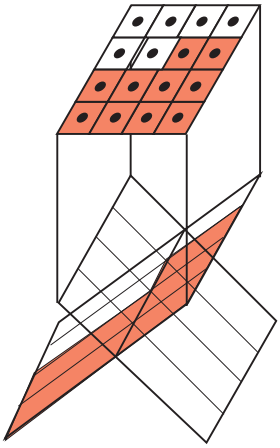


Objekt #1

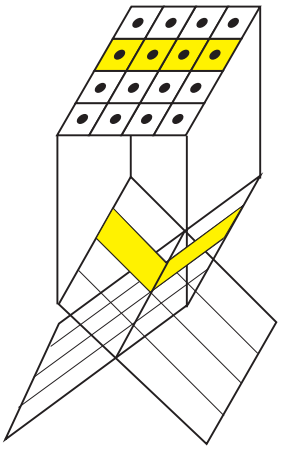
B'=A∧B∧$\overline{P}$



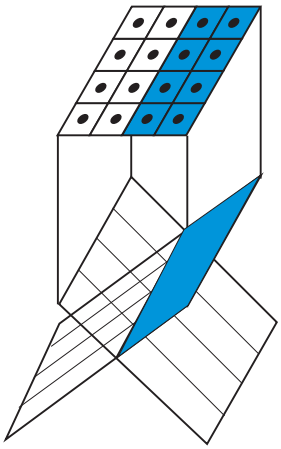
Aakt=A∧$\overline{B}$'



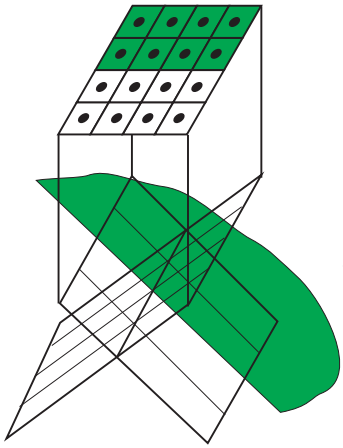
A∧B



P

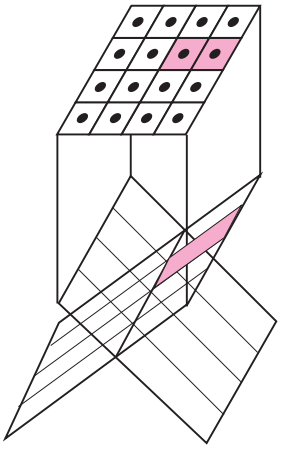


B



Objekt #2

A'=A∧B∧P



Bakt=B∧$\overline{A}$'