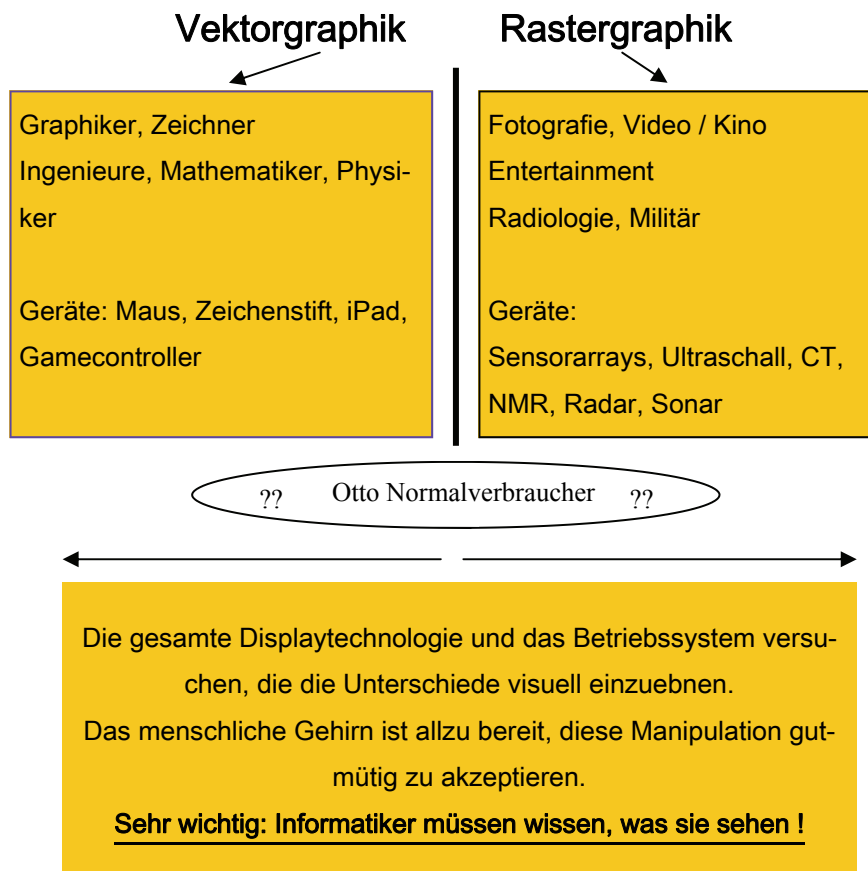


## Webmeeting Mi. 11.08. 2010: Bildverarbeitung 2

### Thema: Wanderung durch 2 Welten: Vektor und Raster



Links aus Bildverarbeitung 1:

Großer Vergleich: Vektorgraphik und Rastergraphik:

[http://miszalok.de/Lectures/L03\\_Raster/Raster\\_Graphics\\_d.htm#a4](http://miszalok.de/Lectures/L03_Raster/Raster_Graphics_d.htm#a4)

[http://miszalok.de/Lectures/L02\\_2DVector/2DVector\\_d.htm](http://miszalok.de/Lectures/L02_2DVector/2DVector_d.htm) und ff

[http://miszalok.de/Lectures/L06\\_3DVector/3D\\_Basics/3DBasics\\_d.htm](http://miszalok.de/Lectures/L06_3DVector/3D_Basics/3DBasics_d.htm) und ff

[http://miszalok.de/Lectures/L03\\_Raster/Raster\\_Graphics\\_d.htm](http://miszalok.de/Lectures/L03_Raster/Raster_Graphics_d.htm) und ff

Links aus Bildverarbeitung 2:

[http://www.miszalok.de/Lectures/L08\\_ComputerVision/DigitalTopology/DigitalTopology\\_d.htm](http://www.miszalok.de/Lectures/L08_ComputerVision/DigitalTopology/DigitalTopology_d.htm) und ff

## Vektor -> Raster Umwandlung

Geschieht hinter den Kulissen automatisch in den Graphikkarten.

Grund: Die Computerdisplays können nur Raster

Spektakulärer Nebeneffekt: Rastermatrizen sind enorm groß. Sie enthalten in der Regel Millionen von Pixeln. Die Datenmenge explodiert; Es entsteht enorme Redundanz.

### Probleme der Vektor -> Raster Umwandlung

#### Vorstellung des Laien:

Es ist egal, ob man einen Strich malt oder viele kleine Pünktchen hintereinander.

Man braucht nur genügend kleine und genügend viele Pünktchen,  
dann merkt keiner den Unterschied.

#### Einwände der Fachleute:

Die Pixelei zerstört alle Axiome der euklidischen Geometrie.

Beispiele für solche Zerstörungen:

#### Länge einer Geraden:

Vektor: Satz des Pythagoras; Raster: City-Block Abstand

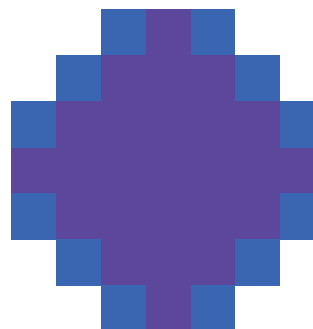
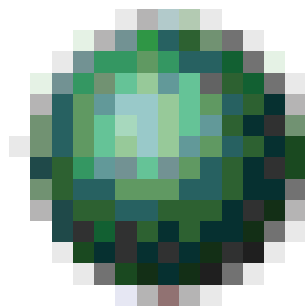
#### Wege zwischen zwei Koordinaten im Raum:

Vektor: es gibt genau einen kürzesten Weg; Raster: es gibt viele gleichlange Wege

#### Umfang eines Kreises:

Vektor:  $U = 2 * \pi * r$ ; Raster:  $U = 8 * r$

$$\begin{aligned} \text{Umfang} &= \\ 2 * 3,14 * 3,5 &= \\ 21,98 & \end{aligned}$$



$$\begin{aligned} \text{Umfang} &= \\ 8 * 3,5 &= 28 \\ &= \\ \text{Umfang des} & \\ \text{umschreibenden} & \\ \text{Rechtecks} & \end{aligned}$$

Links:

[http://www.miszalok.de/Samples/CV/StraightLine/straight\\_line.exe](http://www.miszalok.de/Samples/CV/StraightLine/straight_line.exe)

[http://www.miszalok.de/Samples/CV/StraightLine/straight\\_line.htm](http://www.miszalok.de/Samples/CV/StraightLine/straight_line.htm)

[http://www.miszalok.de/Samples/CV/StraightLine3D/straight\\_line3D.exe](http://www.miszalok.de/Samples/CV/StraightLine3D/straight_line3D.exe)

[http://www.miszalok.de/Samples/CV/StraightLine3D/straight\\_line3D.htm](http://www.miszalok.de/Samples/CV/StraightLine3D/straight_line3D.htm)

### Raster -> Vektor Umwandlung (Vektorisierung)

Geschieht vor den Kulissen und ist hohe Informatik.

Es gibt keine allgemein gültigen Lösungen und deshalb keine Hardwareunterstützung.  
Spektakulärer Nebeneffekt: Polygone sind viel kompakter als Rastermatrizen. Sie enthalten in der Regel nur wenige Vertices. Die Datenmenge implodiert; Es wird enorme Redundanz vernichtet.

Links:

[http://www.miszalok.de/C\\_CVCis/C3\\_Polygon/CVCisPolygon.exe](http://www.miszalok.de/C_CVCis/C3_Polygon/CVCisPolygon.exe)

[http://www.miszalok.de/C\\_CVCis/C3\\_Polygon/CVCisPolygon\\_d.htm](http://www.miszalok.de/C_CVCis/C3_Polygon/CVCisPolygon_d.htm)

Experimente mit CVCisPolygon.exe zeigen:

1. Exakte, pixelgenaue Vektorisierungen sind möglich, liefern aber eckige Polygone mit vielen Vertices.
2. Bessere und redundanzärmere Polygone bekommt man, wenn man einen Fehler Epsilon zulässt.
3. Epsilon von mindestens  $(\sqrt{2})/2$  = halbe Diagonale eines Pixels sind fast immer sinnvoll.
4. Epsilon von 1 und größer verkleinern das Gebiet,  
weil sie nach aussen stehende Pixel und Pixelreihen abschneiden.
5. Große Epsilon liefern grotesk deformierte Gebiete.
6. Ist Epsilon gleich oder größer als der Gebietsdurchmesser,  
schrumpft das Gebiet auf eine Strecke oder einen Punkt.
7. Es gibt kein überall sinnvoller Epsilon. Man muss Epsilon individuell für jeden Einzelfall neu bestimmen.

### Einteilung einer Rastermatrix in zusammenhängende Gebiete = Gebietsmarkierung = Labeling

Aufgabenstellung:

Gegeben eine Rastermatrix mit 256 Grauwerten.

Gesucht eine Rastermatrix mit  $n < 256$  Kennnummern, wo zusammenhängende Gebiete,  
die genügend homogen sind (gleiche oder ähnliche Grauwerte haben )  
mit einer Kennnummer markiert sind.

Links:

[http://www.miszalok.de/Samples/CV/ComponentLabel/simple\\_label.exe](http://www.miszalok.de/Samples/CV/ComponentLabel/simple_label.exe)

[http://www.miszalok.de/Samples/CV/ComponentLabel/simple\\_label.htm](http://www.miszalok.de/Samples/CV/ComponentLabel/simple_label.htm)

Experimente mit simple\_label.exe zeigen:

Gebietsmarkierung ist ein dreistufiger Algorithmus.

1. Durchgang: zeilenweise von links:  
Kommt ein neuer Grauwert, beginnt eine neue Kennnummer,  
es sei denn, man kann eine Kennnummer von oben herunterziehen.
2. Durchgang: zeilenweise von links:  
Jedes Pixel versucht, seine 4 oder 8 Nachbarn in seine Kennnummer einzusaugen.  
Ergebnis: Einzelne Kennnummern verschwinden vollständig.
3. Durchgang: Lückenlose Neuvergabe der Kennnummern nach Gebietsgröße.  
Das größte Gebiet bekommt die Nr. 1, das zweitgrößte die Nr. 2 usw.  
Die Anzahl der Gebiete wird identisch mit der höchsten Nr.

Spektakuläres Beispiele für Labeling in der Anatomie:

[http://www.voxel-man.de/gallery/visible\\_human/inner\\_organs/](http://www.voxel-man.de/gallery/visible_human/inner_organs/)