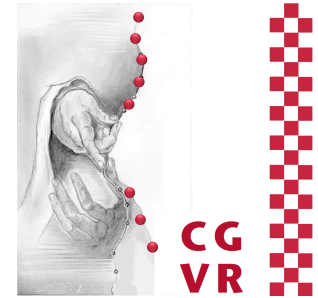


Bremen



Computergraphik I

Einführung & Displays



G. Zachmann

University of Bremen, Germany

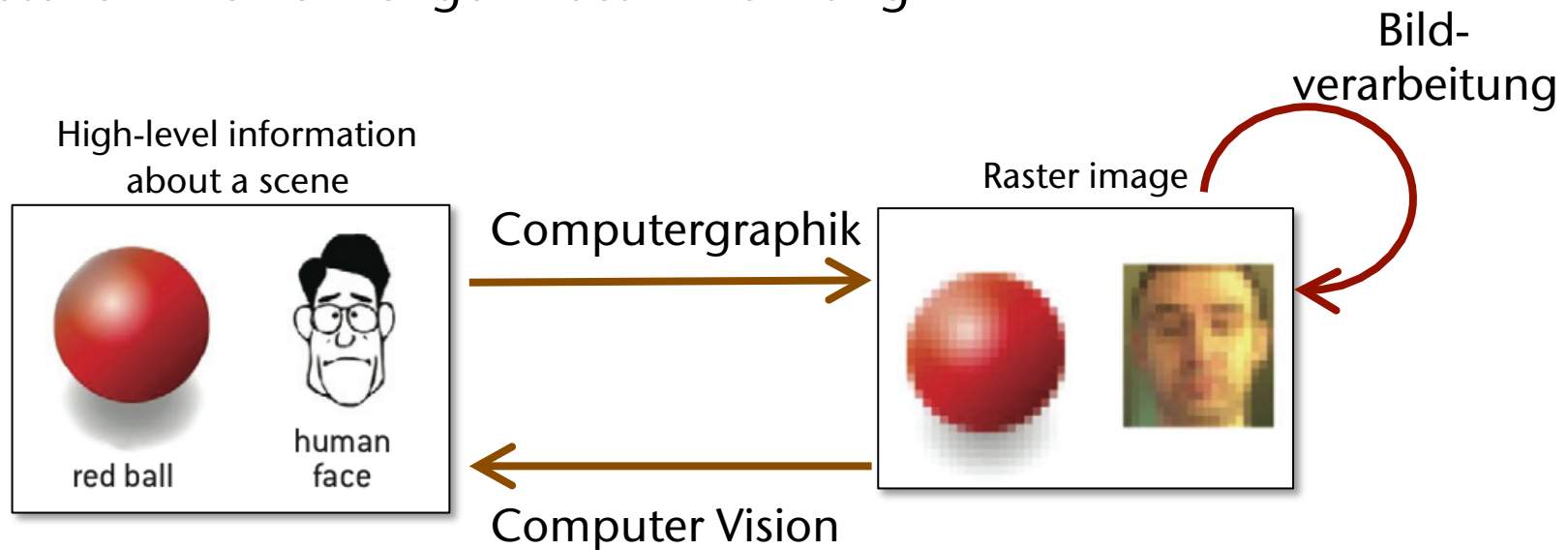
cgvr.informatik.uni-bremen.de

Ziel der Vorlesung

- Praxis: Sei in der Lage, ziemlich komplexe interaktive 3D Graphikprogramme zu schreiben (in OpenGL)
- Theorie: Verstehe den mathematischen Hintergrund und die grundlegenden Algorithmen der modernen 3D Graphiksysteme
- Diese Vorlesung behandelt **nicht** Graphikprogramme wie Maya, Alias, AutoCAD, Blender, 3DStudio Max, Photoshop, ...

Das Gebiet "Visual Computing"

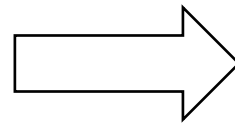
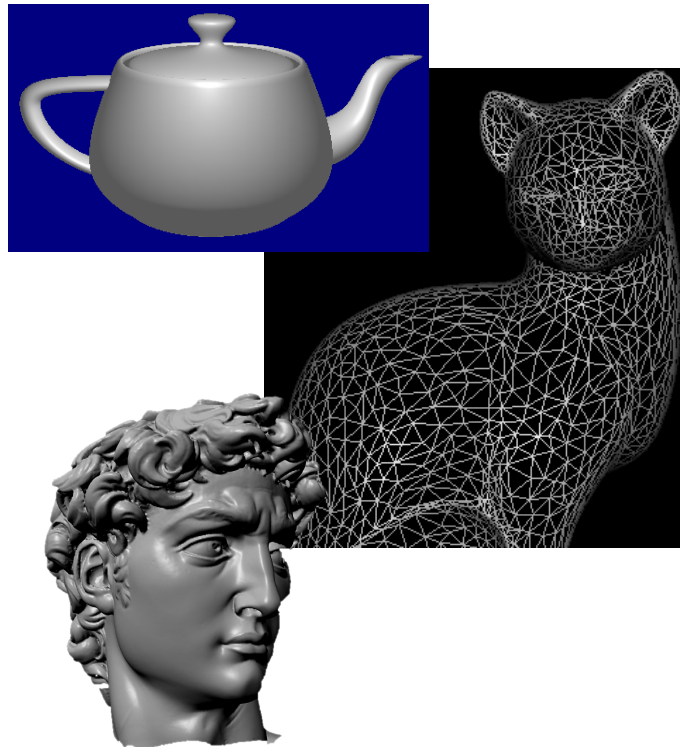
- *Visual Computing* = Informatik-Disziplinen "mit Bildern"
- Computergraphik, Maschinelles Sehen und Bildverarbeitung stehen in einem engen Zusammenhang



- Trend: Computergraphik und Computer Vision wachsen immer stärker zusammen ("ProCams")

Was ist Computergraphik?

Modelling
(Beschreibung der 3D
Geometrie und Szene)



Rendering
(Erzeugung der Bilder,
inkl. Shading, Lighting, Materials)



- Die wichtigsten Gebiete:
 - Modellierung
 - Festlegen der Form und Wirkung des äußeren Erscheinungsbildes
 - Rendering
 - Erzeugung des 2D Bildes aus einem 3D Modell
 - Animation / Simulation
 - Bewegung der Bilder
- Weitere Gebiete:
 - Interaktion mit dem Anwender (*Human-Computer Interaction - HCI*)
 - Virtual Reality (VR)
 - Visualisierung (*scientific / information visualization*)

- Wie beschreibt man ein Objekt einer Szene?
- Wie stellt man diese **schnell** dar?
- Was ist mit Lichtquellen?
- Wie erzeugt man Schatten? ... Verdeckungen? ... Tiefeneindruck?
- Was macht man bei "rauhem" Oberflächen?
- Was ist mit kleinen Partikeln wie Nebel, Rauch, Dunst, ... ?
- Physik?
- Animation?

- Videospiele
- Filme
 - Zeichentrickfilme
 - Computeranimationsfilme
 - Spezialeffekte
- CAD / CAM
- Simulationen
- Medizinische Visualisierung
- Visualisierung von Informationen
- Training (Flug-, Fahr-, Operationssimulator)



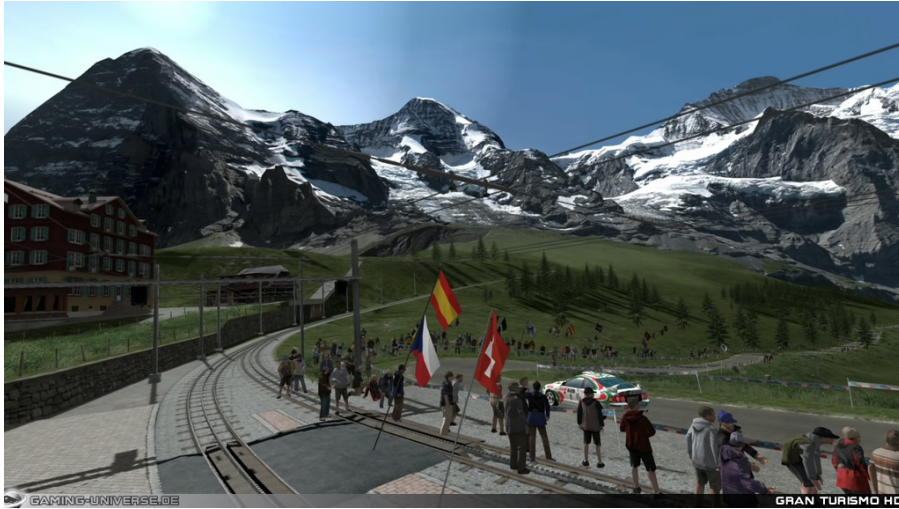
Pixar: Monster's Inc.

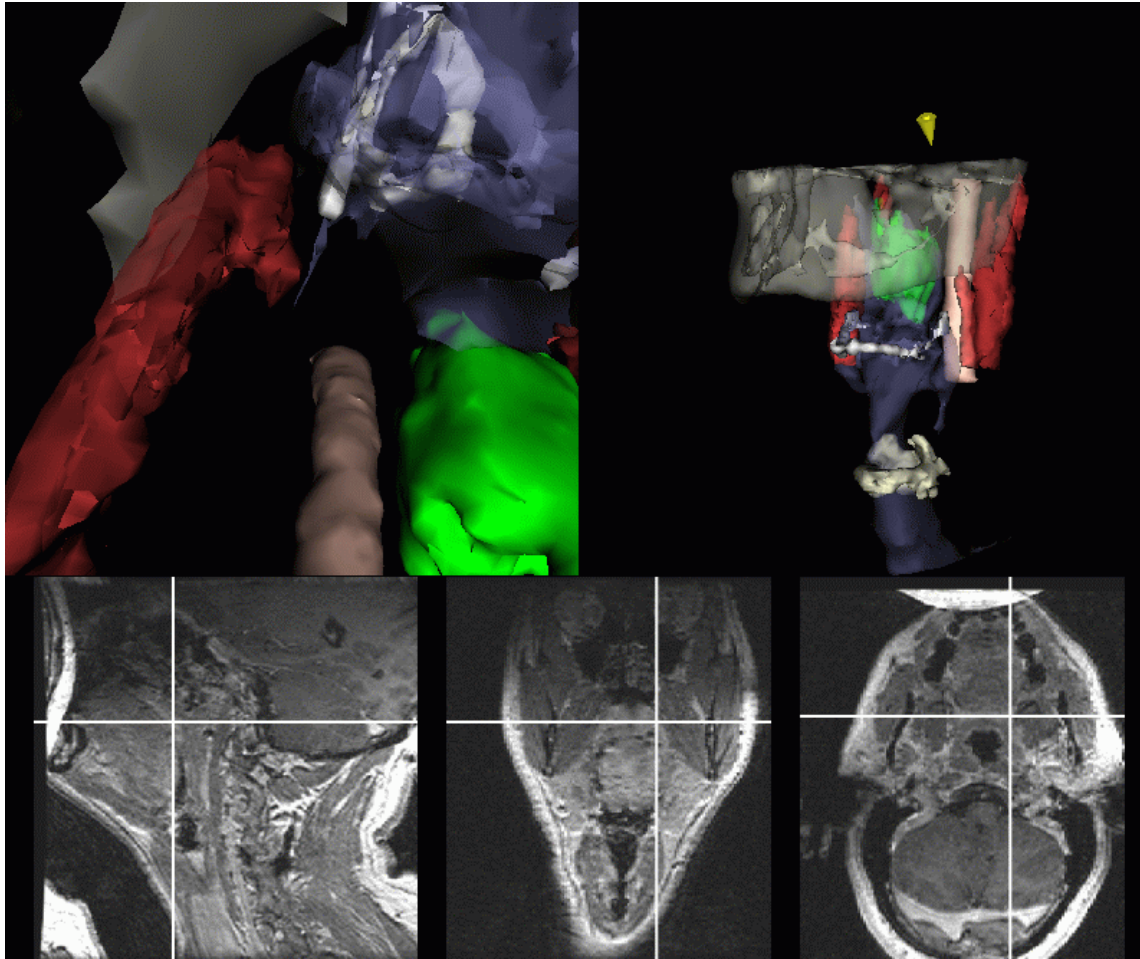


Square: Final Fantasy

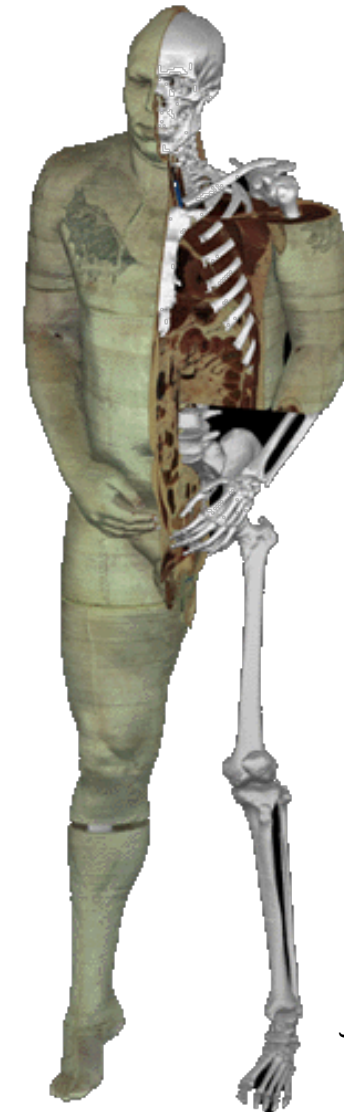


Spiele



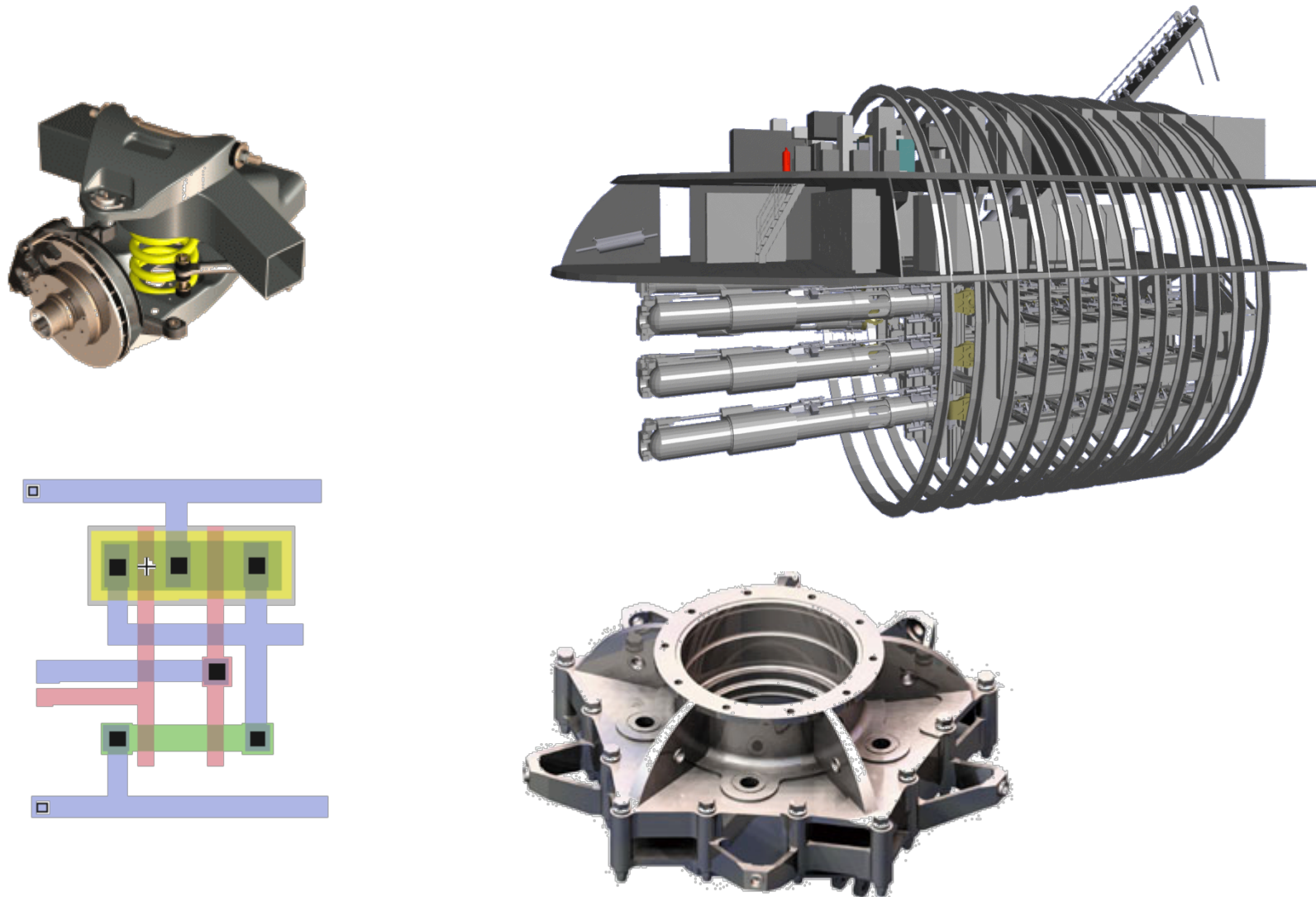


MIT: Image-Guided Surgery Project



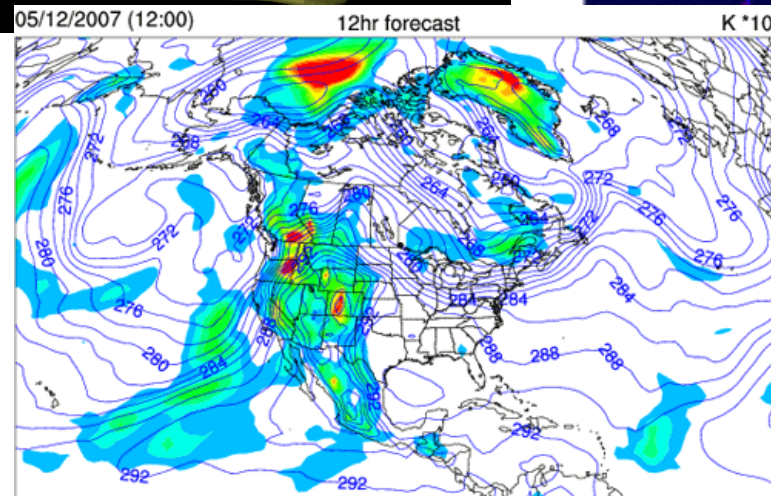
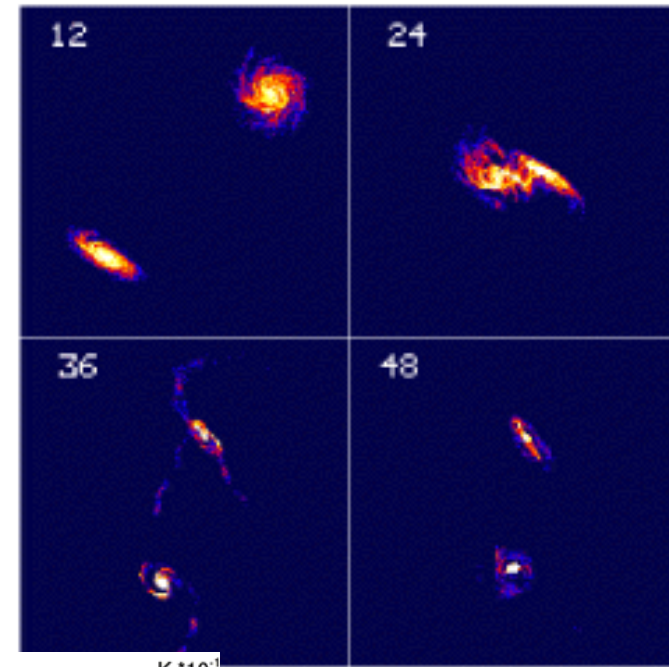
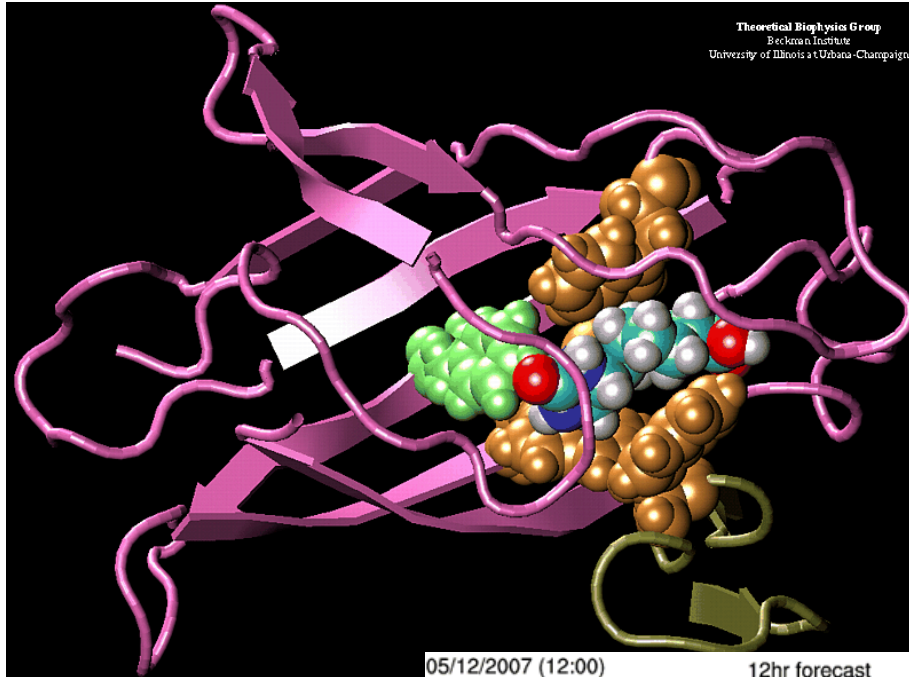
The Visible Human Project

Computer Aided Design (CAD)





Visualisierung wissenschaftlicher Daten

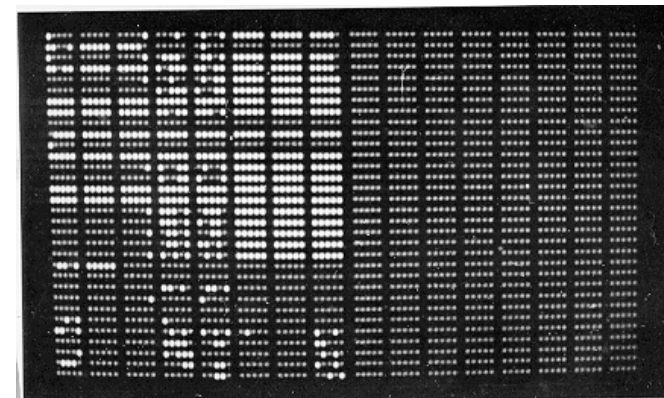
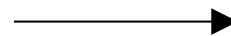


Kurze Historie der Computergraphik



Manchester Mark I

- Am Anfang: noch nicht einmal Text-Displays





Sketchpad (1963) & "The Mother of all Demos" (1968)



- Ivan Sutherland's Sketchpad
 - MIT, 1963
 - Der Beginn der modernen **interaktiven** Graphik
 - Sehr teuer!
 - Viele Konzepte findet man in heutigen Zeichensystemen wieder
 - Pop up Menü
 - Hierarchisches Modellieren
- Doug Engelbart
 - Maus
 - Hyperlinks / Hypertext
 - Email, CSCW
 - Telekonferenz, ...



Engelbart, 1968

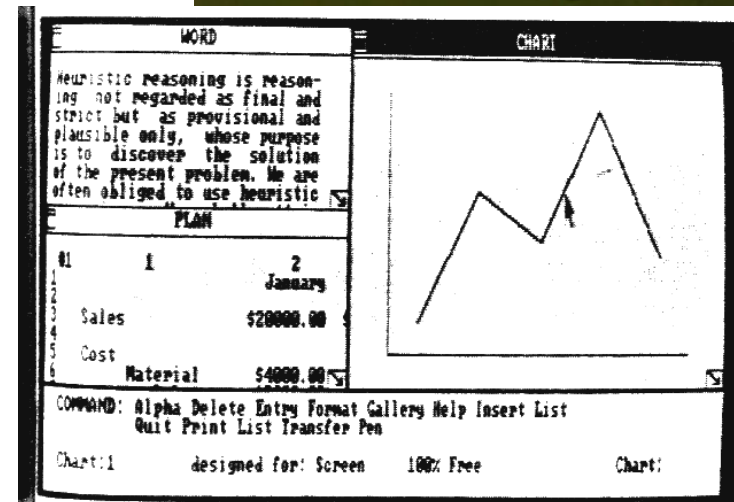
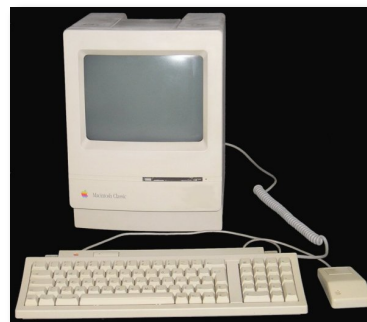
"The Mother of all Demos"



Von Text zu GUIs

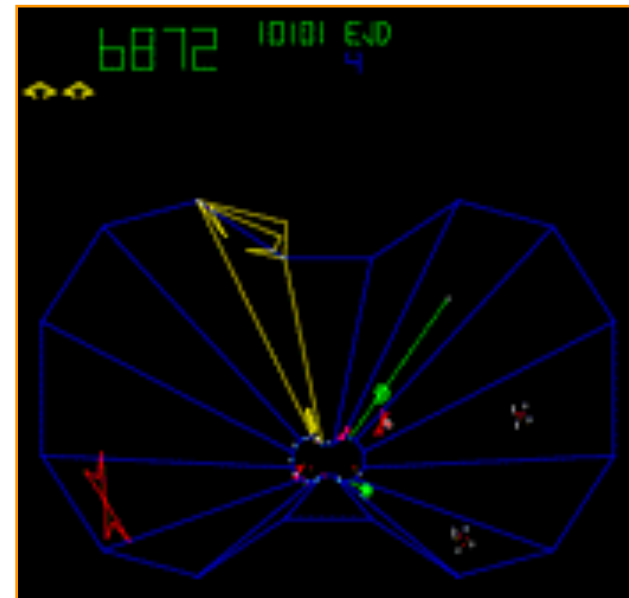
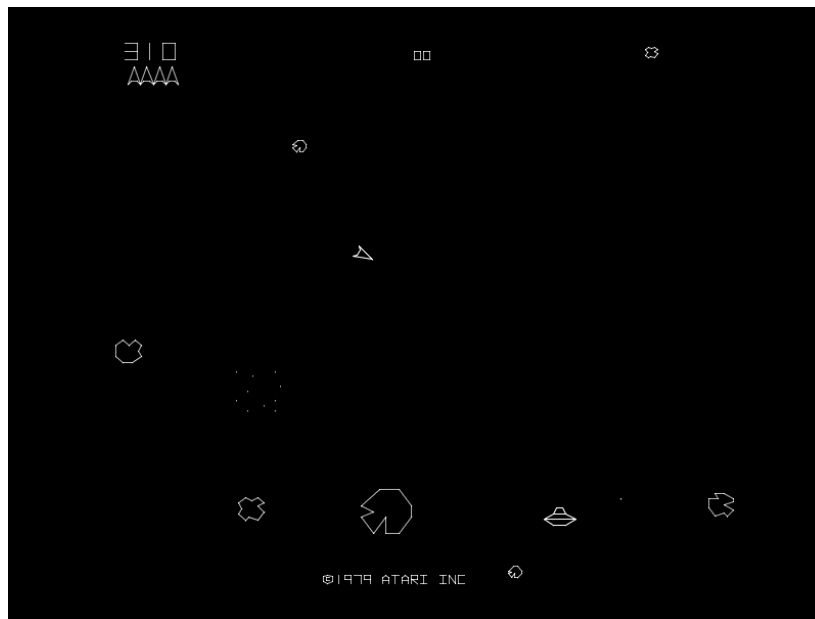


- Ausgedacht vom Xerox PARC etwa 1975
- 1981: "Echtzeit" Vektor-Displays, erste bezahlbare Rastergraphik (Apple II)
- "GUI / Desktop" zuerst kommerziell eingesetzt auf Apple Macintosh
- Mitte/Ende 80er: C64, IBM PC
 - PCs mit eingebautem Raster-Display
 - Bezahlbare Rastergraphik



Windows 1.0

- Zunächst noch reine Vektorgraphik:
 - Pong
 - Asteroids
 - Star Wars

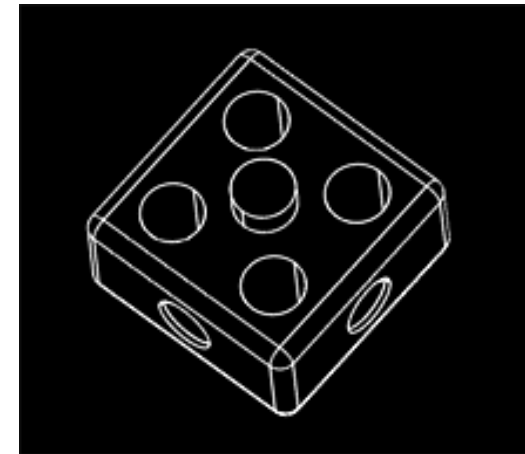


- Dann Rastergraphik:
 - Space Invaders
 - Pac Man



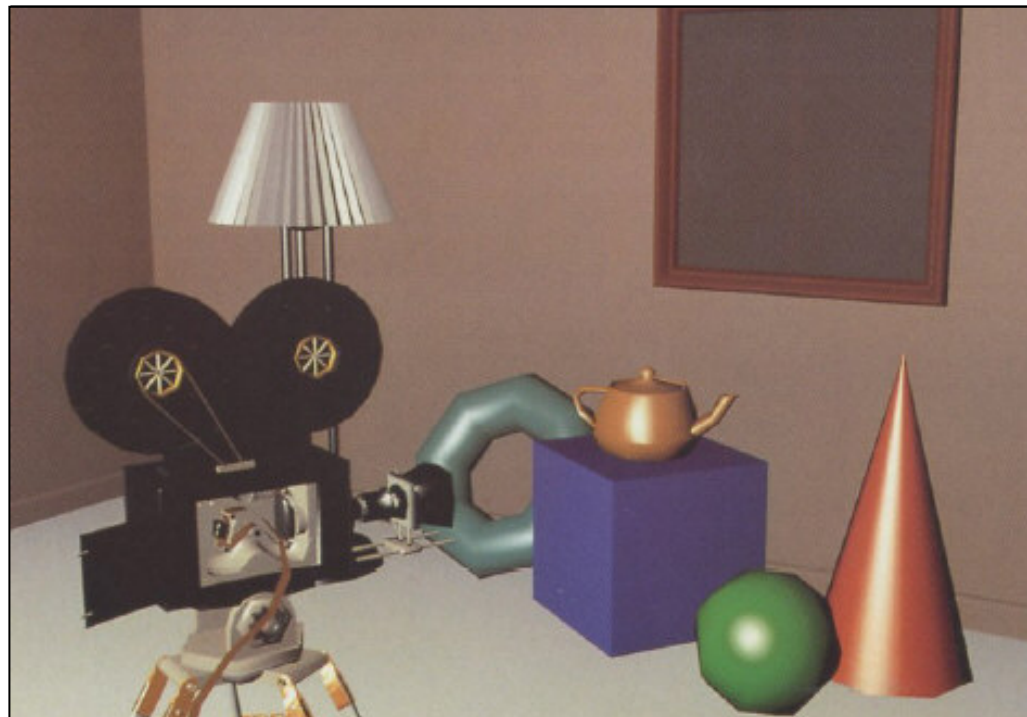
Rendering: 1960 (Sichtbarkeit)

- Roberts (1963), Appel (1967): verdeckte Linien
- Warnock (1969), Watkins (1970): verdeckte Flächen
- Sutherland (1974): Sichtbarkeit = Sortierung



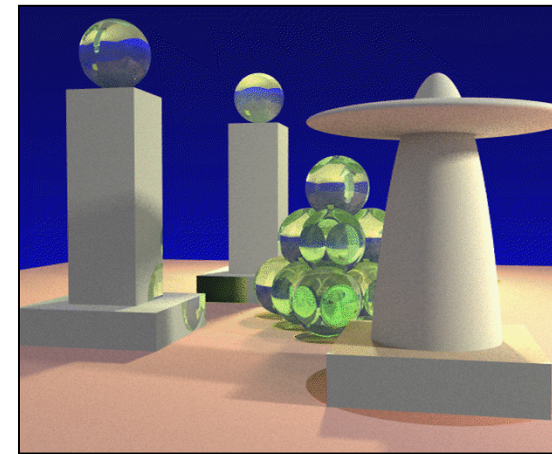
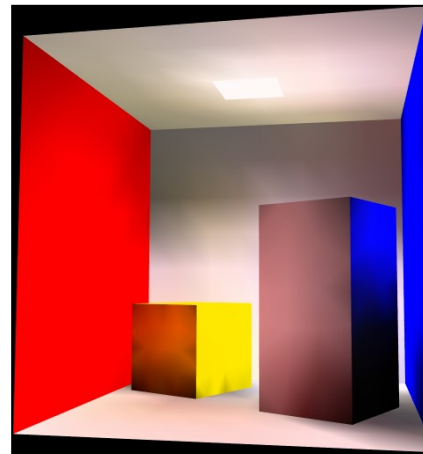
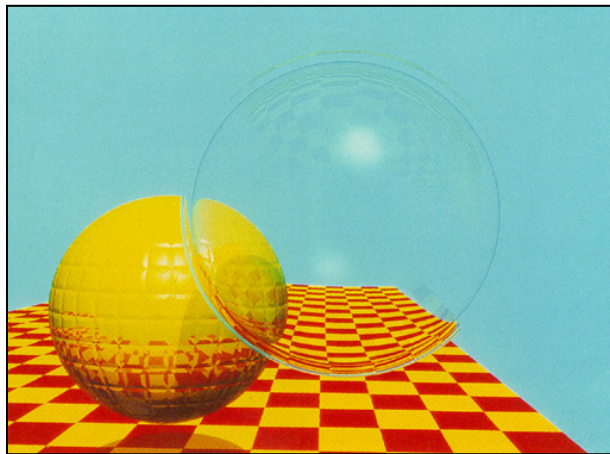
Rendering: 1970 (Beleuchtung)

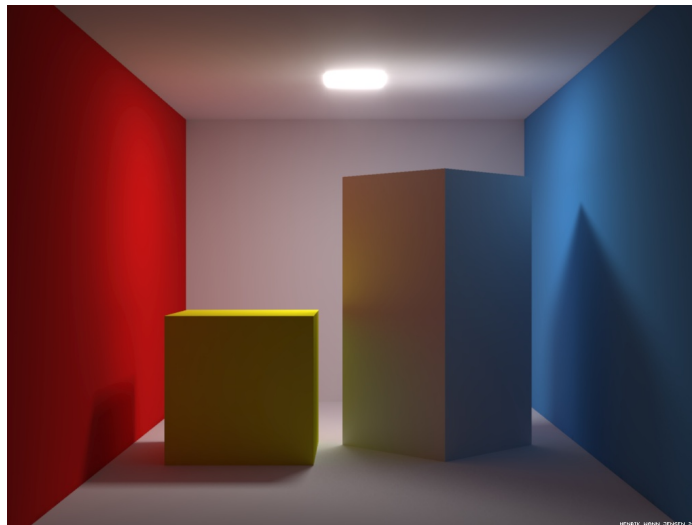
- Raster Graphiken:
 - Gouraud (1971) – diffuse Beleuchtung
 - Phong (1974) - spiegelnde Beleuchtung
 - Blinn (1974) – gewölbte Oberflächen, Texturen
 - Catmull / Straßer (1974) – verdeckte Flächen mittels Z-Buffer

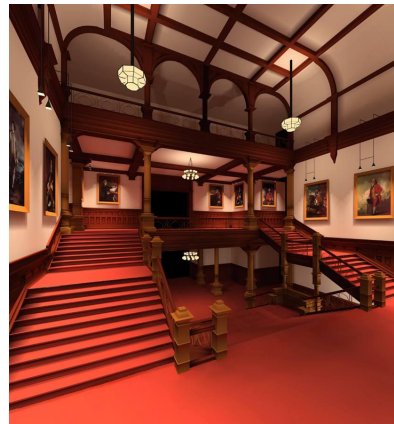
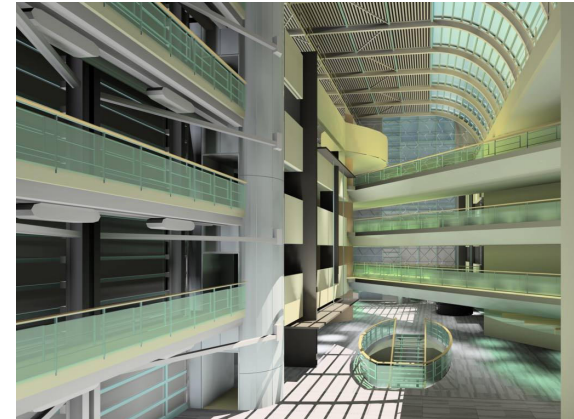


Rendering: 1980, 1990 (Globale Beleuchtung)

- Whitted (1980) : Ray-Tracing
- Goral, Torrance et al. (1984) : Radiosity
- Kajiya (1986) : Die Rendering-Gleichung







Beleuchtungseffekte bei polygonalem Rendering



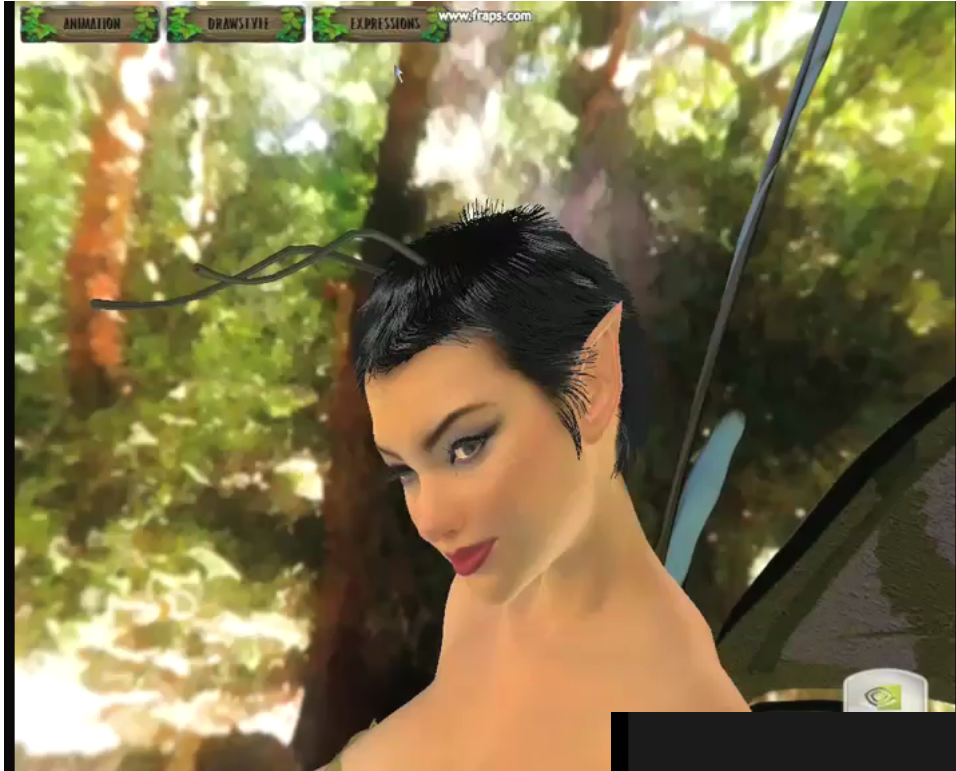
+





Polygonales Rendering heute ...

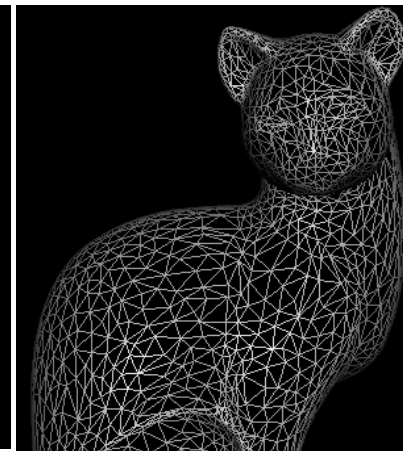




- Spline-Kurven, -Oberflächen: 70er – 80er
- Utah Teapot: Bekanntes 3D Modell
 - Von Hand modelliert von Newell
 - Zur "Folklore":
<http://www.sjbaker.org/teapot/>



- Erst kürzlich: Erstellung von Dreiecksnetzen von realen Objekten





Displays



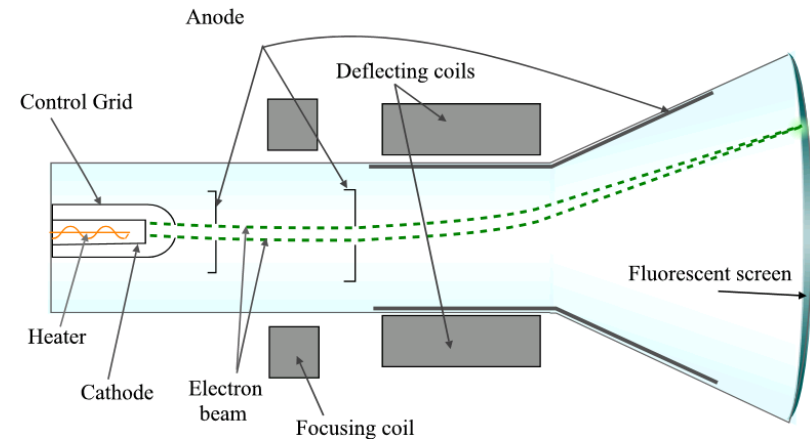
- CRTs,
- LCDs,
- DMDs,



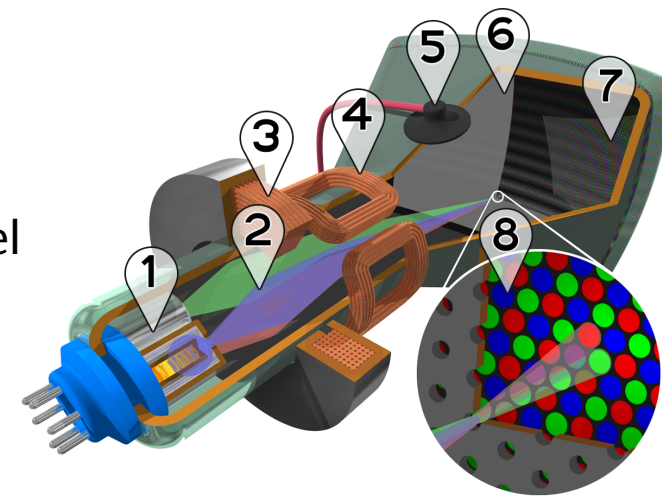
Kathodenstrahlröhre (CRT, Braunsche Röhre, 1897)



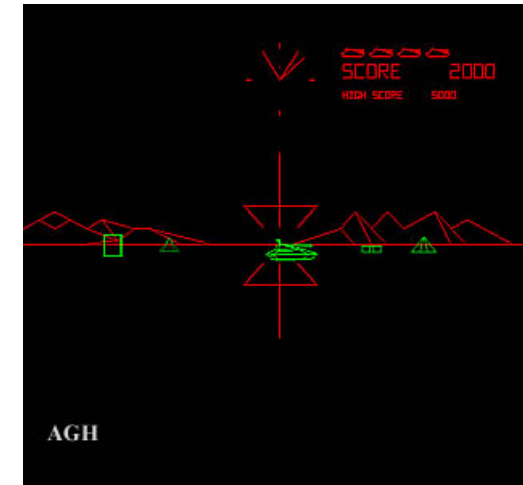
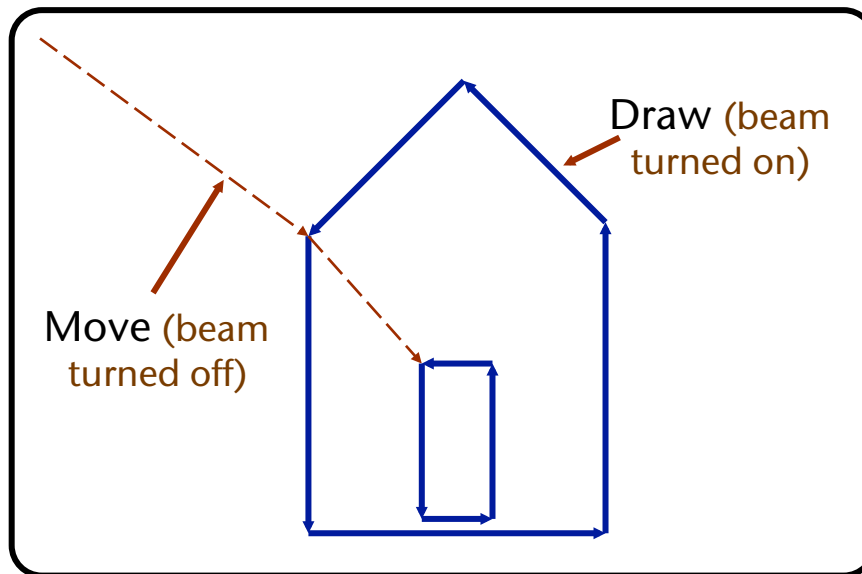
- Elektronen werden
 - erzeugt durch Erhitzung der Glühkathode
 - beschleunigt in Richtung Anode
 - fokussiert
 - Abgelenkt
 - gefiltert durch Lochmaske
 - treffen Phosphorpunkte



- Phosphor
 - Atome werden angeregt
 - bei Rückkehr zu normalem Energiepegel werden Photonen erzeugt
 - 3 Arten (rot, grün, blau) — später



- Bis Anfang / Mitte der 80er
 - Im Grunde Oszilloskope
 - Steuere X, Y durch die Spannung an den vertikalen/horizontalen Ablenkspulen
 - Oft wird Intensität durch Z geregelt



Battlezone



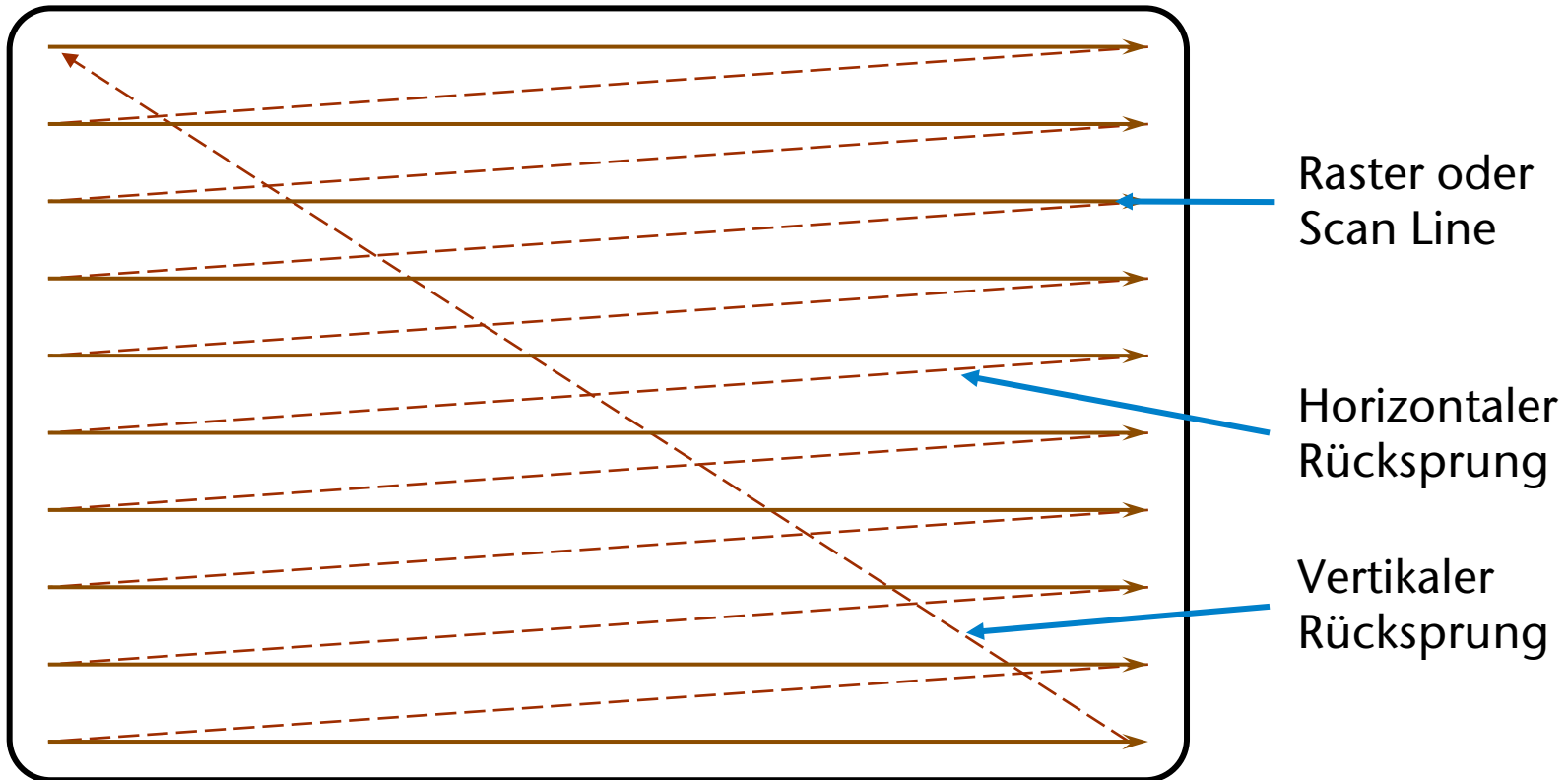
Tempest

Aktualisieren (Refresh)

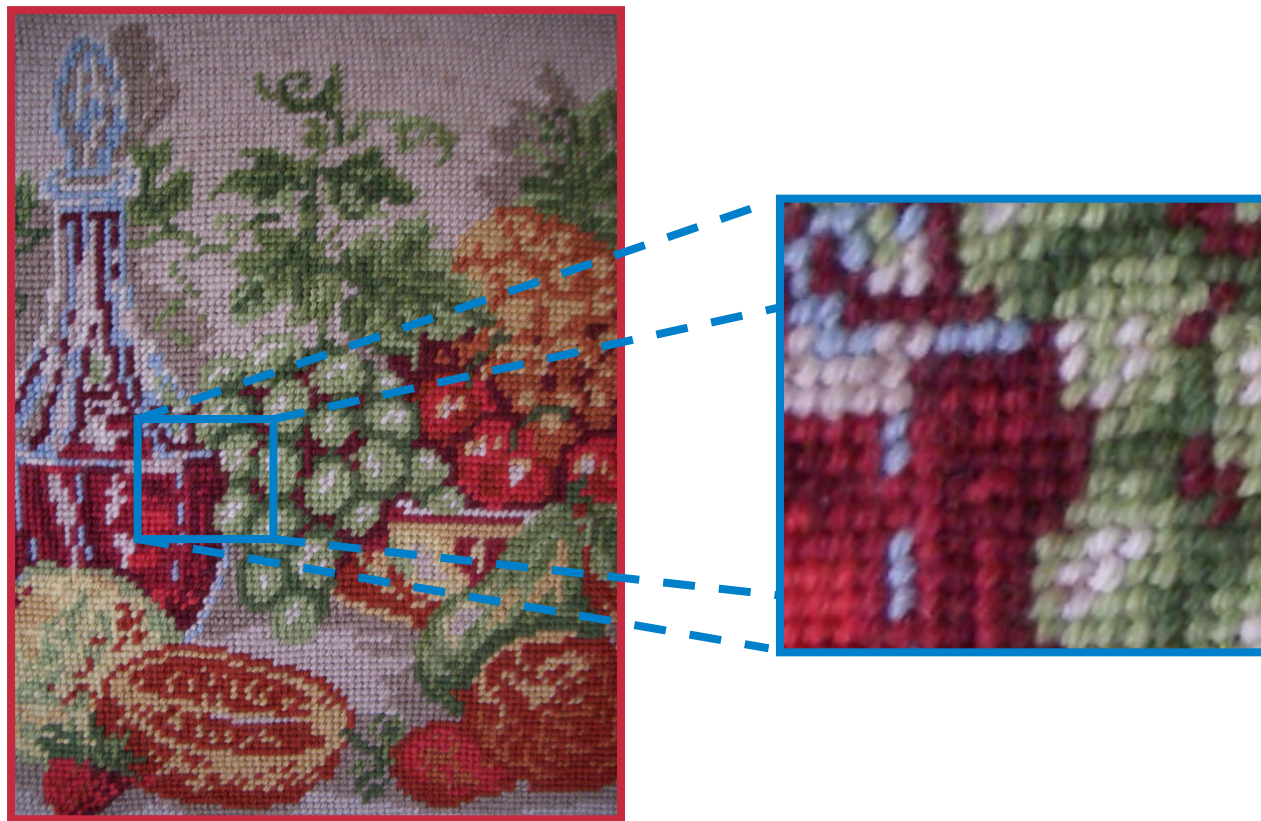
- Ein Bild muss aufgefrischt werden, um ein neues Bild darzustellen
 - Aktiviert der Elektronenstrahl eine Region von Phosphoratomen, so verblasst diese nach einer Weile (einige Millisekunden)
- Folge: der Elektronenstrahl muss regelmäßig alle Stellen des Bildes treffen, um Flimmern zu vermeiden
- Kritische Frequenz: 25 Hz (Vollbilder!)
- Max. mögliche Refresh-Rate hängt bei Vektordisplays von Anzahl und Länge der Linien ab → beschränkte Komplexität der Szene

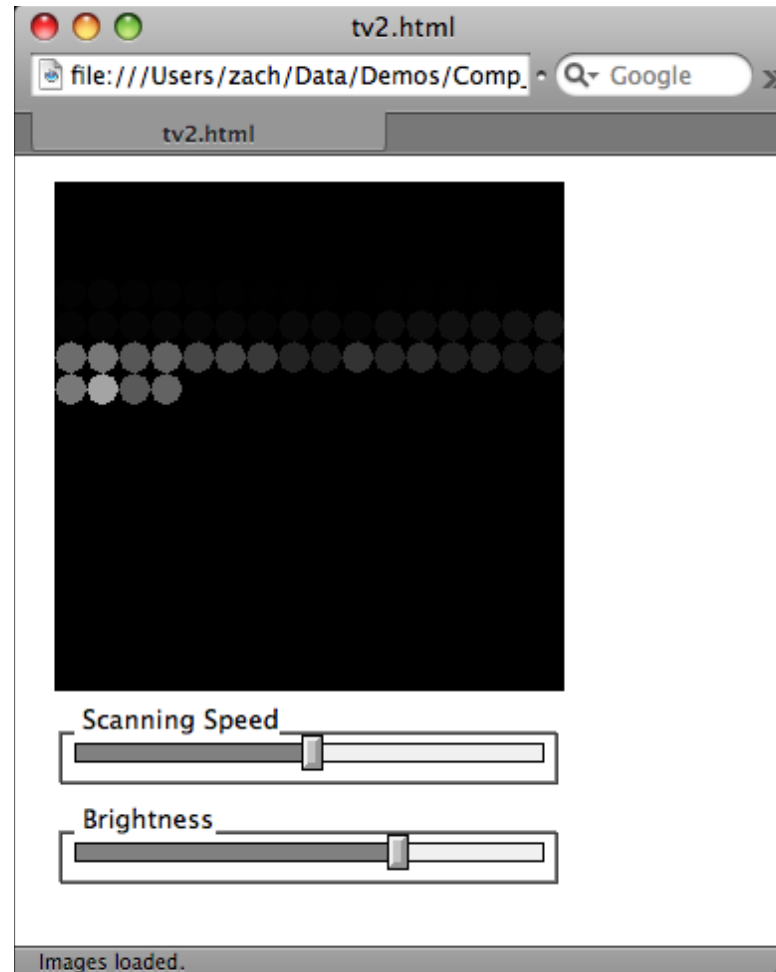
Rastergraphik (Raster / Scanline Displays)

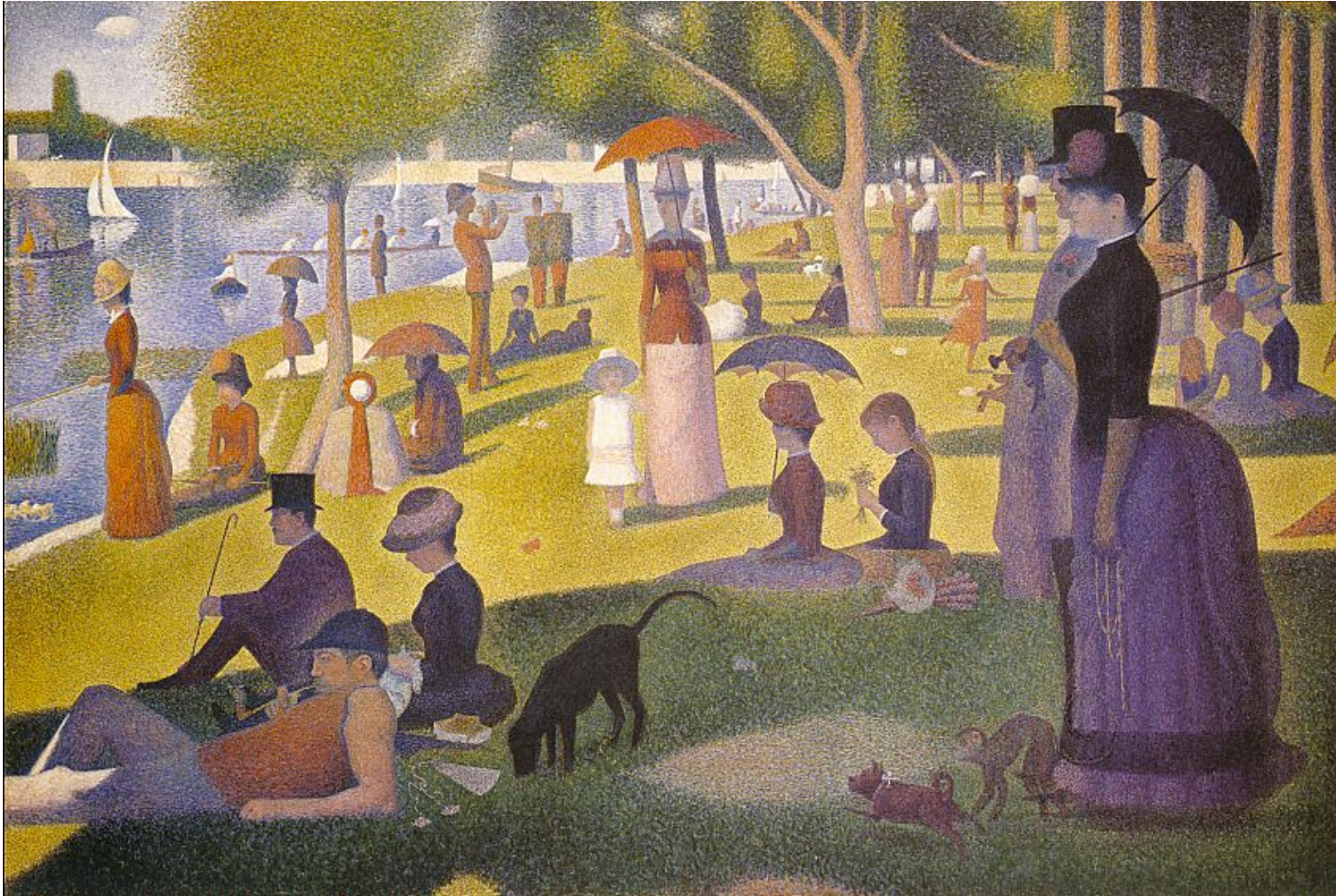
- Heutzutage sind fast alle Displays Raster-basiert



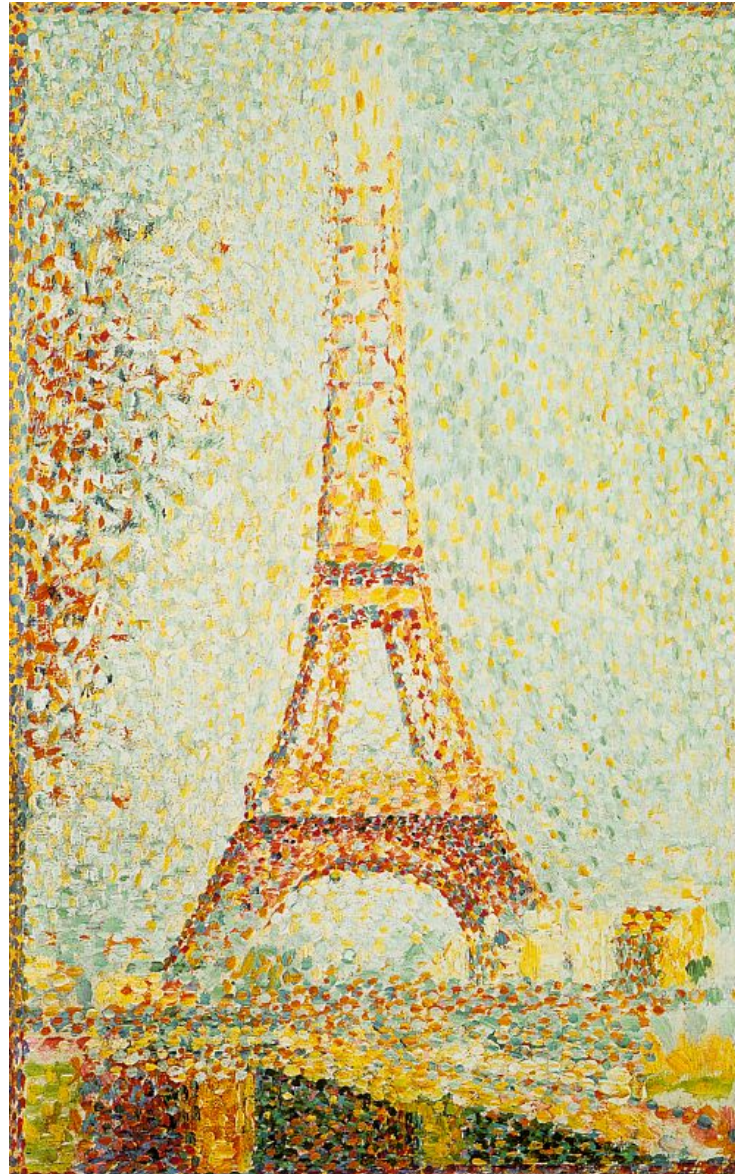
- Speicherung von Bildern als Bildpunktmatrix
 - Feste Informationsmenge pro Bildpunkt
 - Kompatibel zu Fernsehbildern





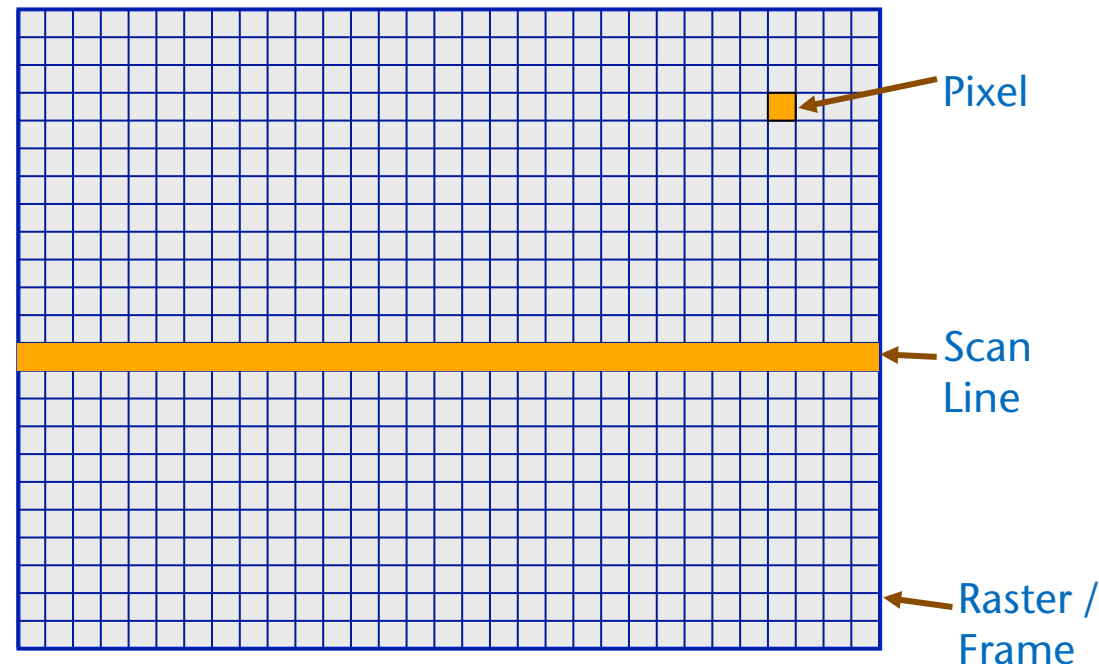


Seurat, *A Sunday Afternoon on the Island of La Grande Jatte*, 1884-86

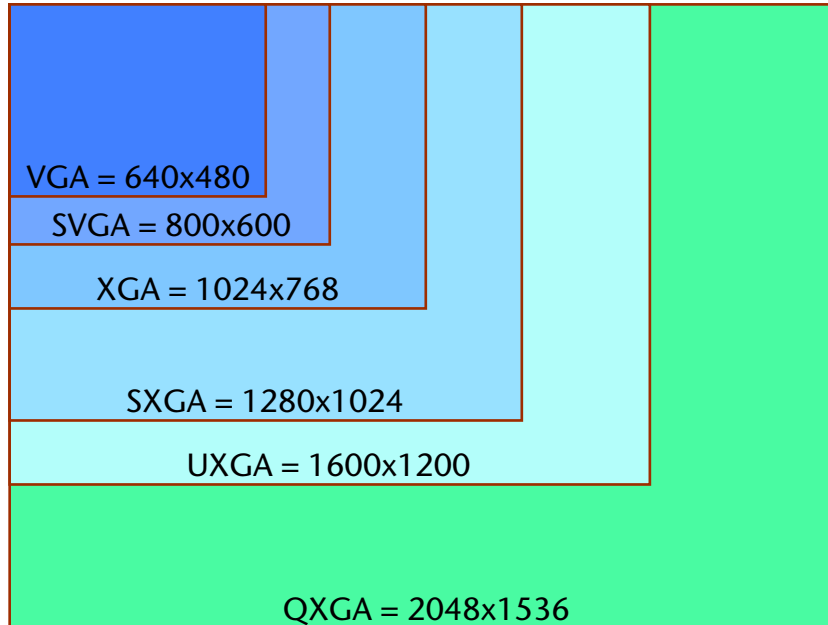


Seurat,
The Eiffel Tower,
1889

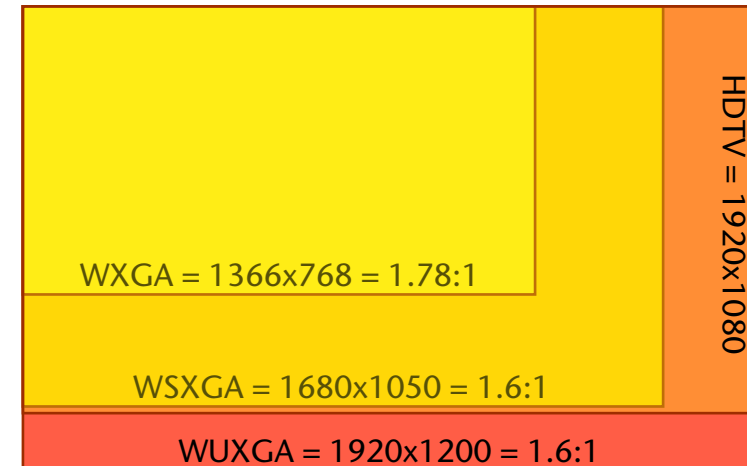
- Raster: ein rechteckiges Feld von Punkten oder Rasterpunkt
- **Frame**: Einzelbild, das auf dem Monitor dargestellt wird



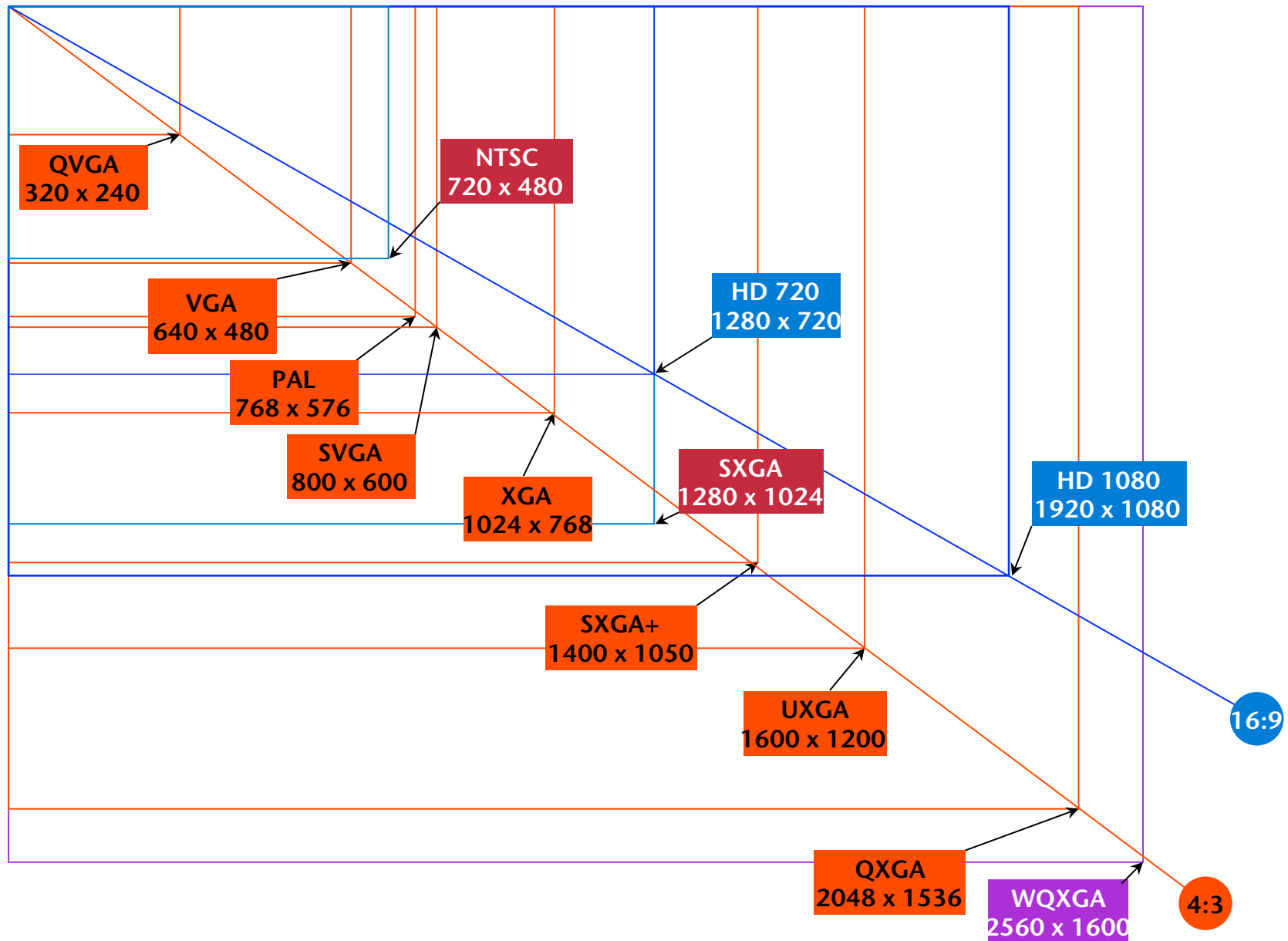
- **Pixel**: ein einzelnes Bildelement oder Rasterpunkt
- **Scanline**: eine Reihe von Pixel
- **Auflösung**: eigentlich Pixel pro Zoll; hier Größenbeschreibung von Bildern (640x480)
- **Aspect ratio** = Breite : Höhe (früher 4:3, jetzt immer mehr 16:9)



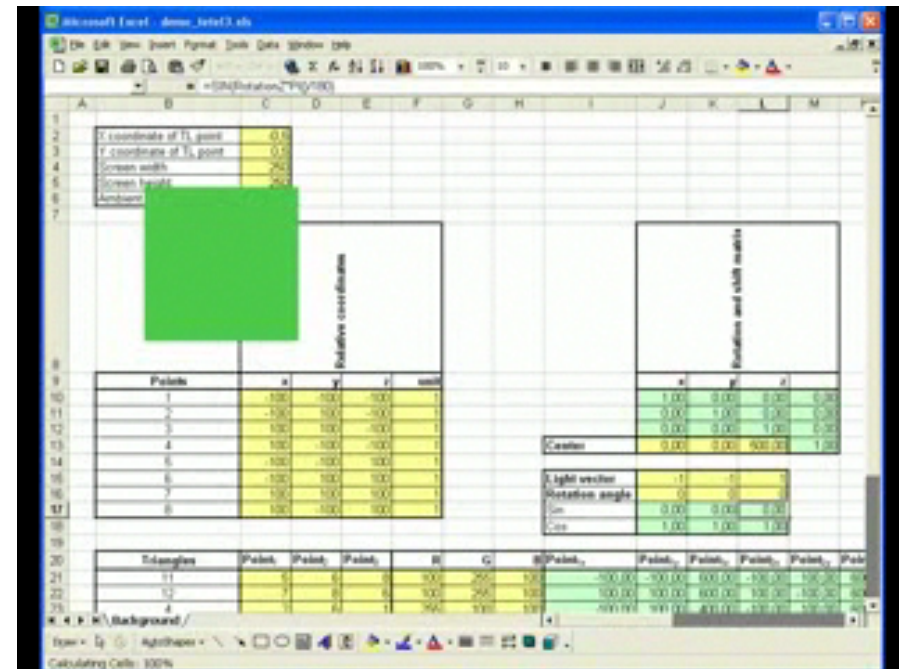
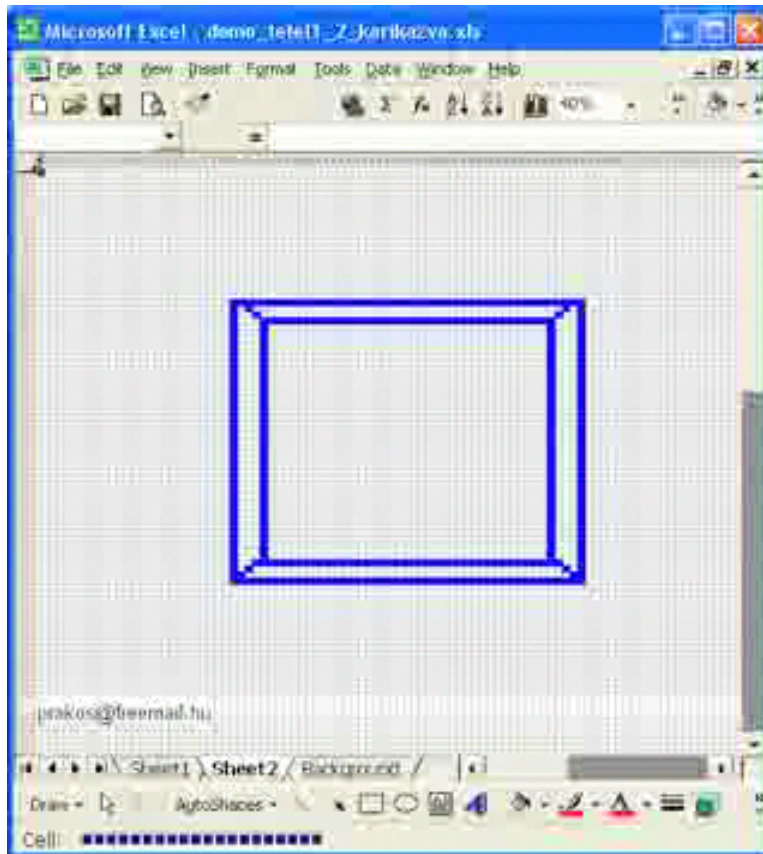
Standardauflösungen:
 Diese haben ein Seitenverhältnis (*aspect ratio*) von 4:3 = 1.33:1, außer SXGA mit 1.25:1



Wide-screen-Auflösungen:
Aspect ratio $\approx 16:9 \approx 1.78:1$.
 (Viele Kinofilme sind in 1.85:1 oder 2.35:1 $\approx 7:3$ gedreht.)

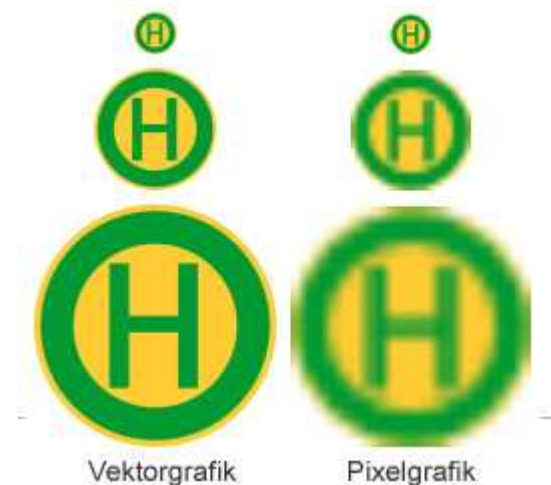
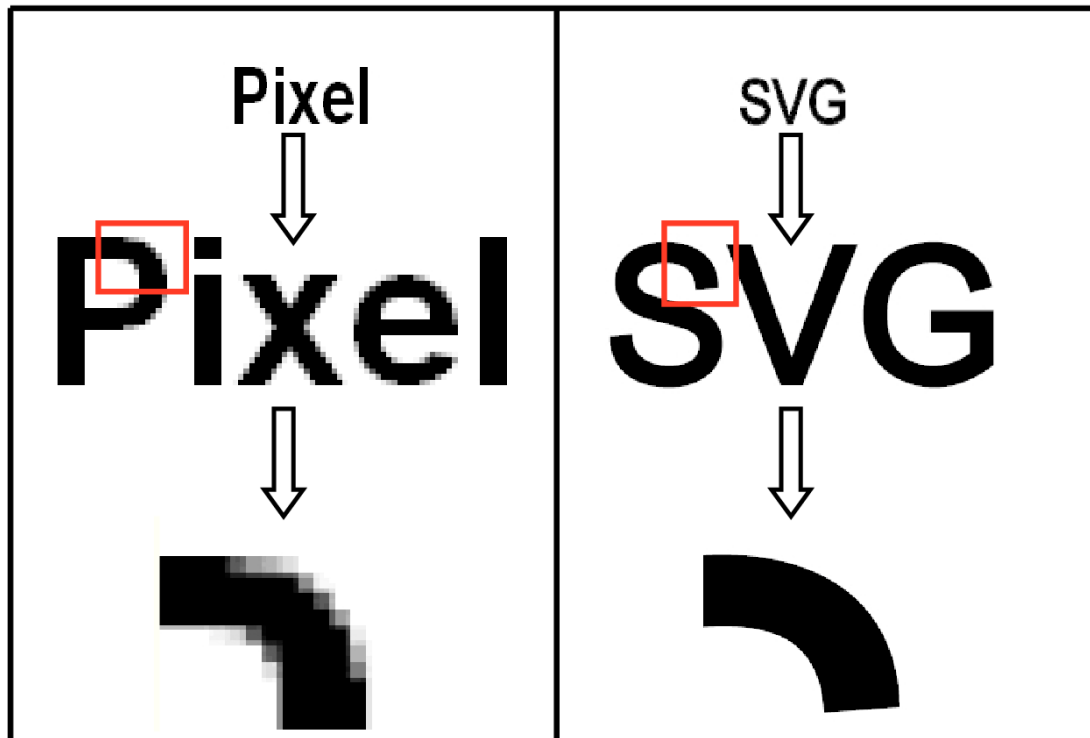


Microsoft Excel: Revolutionary 3D Game Engine? 😊



http://cgvr.cs.uni-bremen.de/teaching/cg_literatur/excel_3d_engine/

Pixelgraphik vs. Vektorgraphik





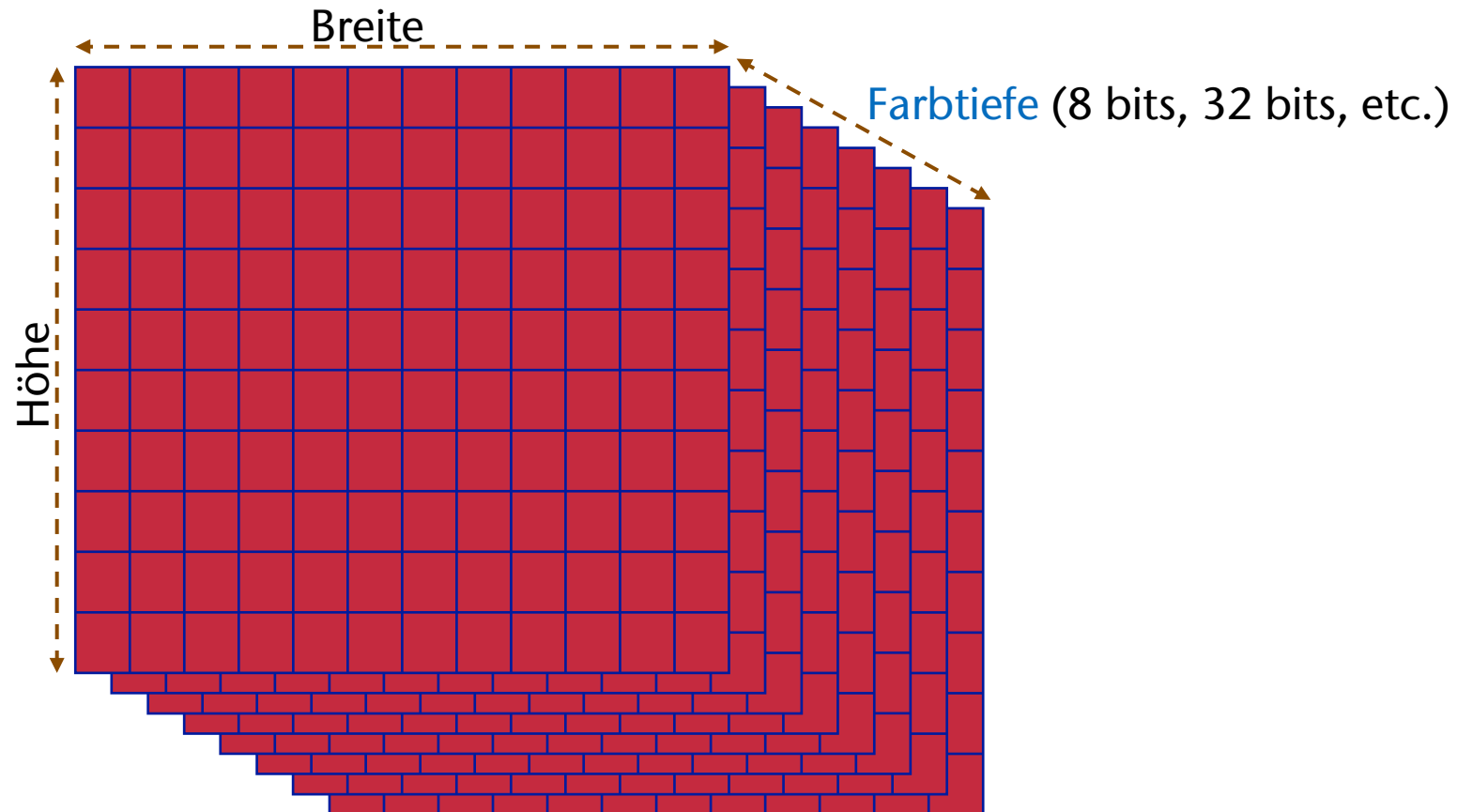
Vektorgraphik

- Linien- und Kurveninformationen, die geometrische Formen beschreiben
- Vorteile:
 - Skalierung ohne Qualitätsverlust
 - Kompression / geringe Dateigröße
 - Textbearbeitung
- PDF, SVG

Pixelgraphik

- Quadratische Bildpunkte, denen jeweils eine Farbe zugeordnet ist
- Vorteile:
 - Programmunabhängig
 - Bearbeitung jedes einzelnen Bildpunktes
 - Detailreicher
- JPG, PNG, BMP, GIF, ...

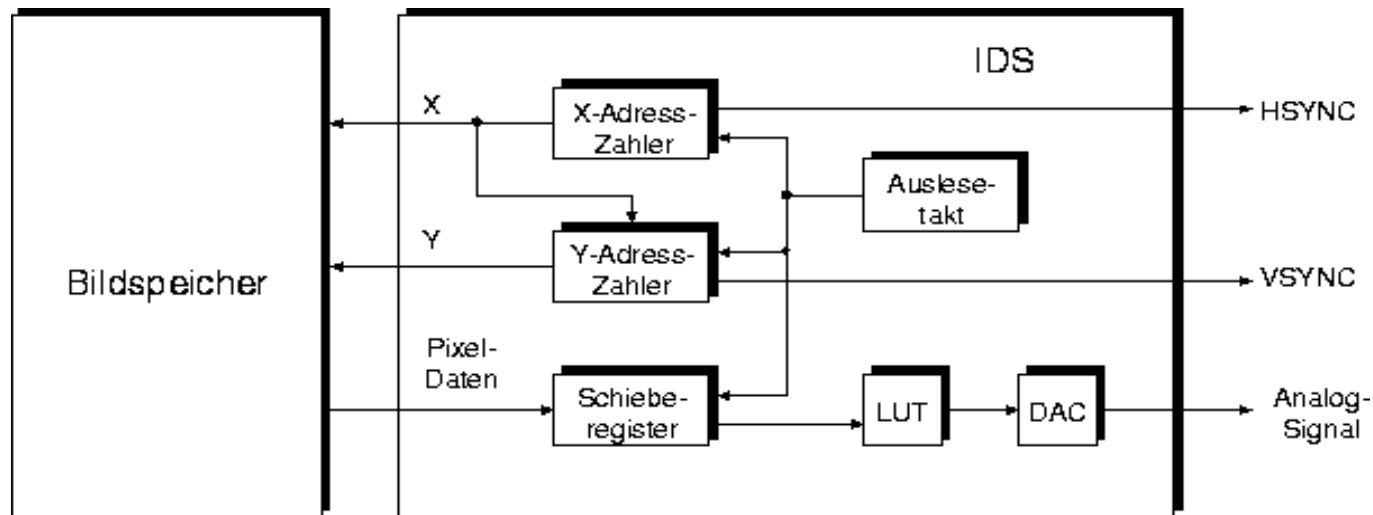
- Muss jetzt viele Bits pro Pixel spendieren (später noch mehr)



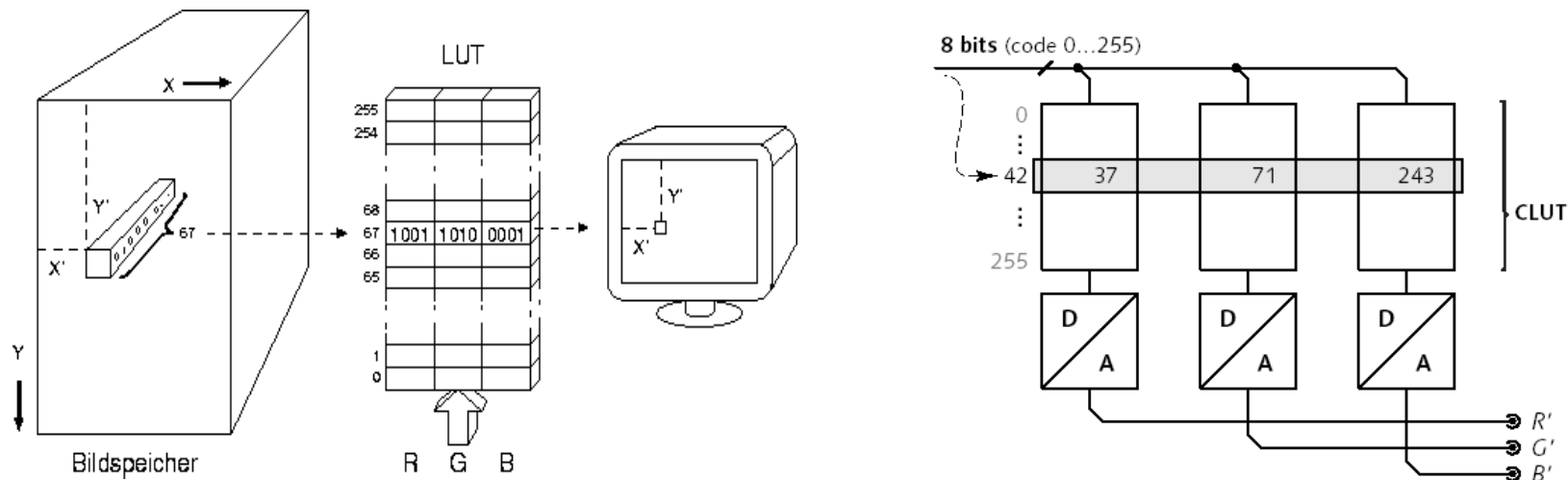
$$\text{Memory (Bits)} = \text{Breite} * \text{Höhe} * \text{Farbtiefe}$$

- Schwarz und Weiß: 1 Bit/Pixel
- Grauskala: 8 Bit/Pixel
- 8-bit Farbe: spart Speicherplatz, 3-2-2 oder Color Lookup Table
- 24-bit (RGB) Farbe: 8 Bit pro Farbkanal – rot, grün, blau
- Wie groß muss der Frame Buffer für ein 1600x1200 Pixel großes Bild in true color (RGB) sein?
 - 8 Bit für jeden RGB Farbkanal
 - Das sind 24 Bit/Pixel
 - Das ergibt $1600 \cdot 1200 \cdot 24 \text{ Bit} = 5.76 \text{ MBytes}$
 - Die meisten Graphikkarten reservieren 32 Bit/Pixel bei true color
= 7.68 MBytes
- Datenrate bei 30 frames per second (FPS): 230 Mbytes / sec

- Aufgaben des Video-Controllers
 - Erzeugen der horizontalen (HSYNC) und vertikalen (VSYNC) Synchronimpulse für das entsprechende Bildformat
 - Adressierung und Auslesen des Bildspeichers
 - Ansteuern des Monitors mit entsprechenden Intensitäts-/ Farbwerten, mit Dunkelsignal für H/V-Austastlücke und Digital-Analog-Wandlung (DAC).



- Idee, um Datenrate und Speicheraufwand zu senken:
 - Erstelle eine **Color Map** (Color Lookup Table, CLUT), welche alle im Bild benötigten Mischfarben enthält
 - Speichere pro Pixels nur einen Index (kleine Anzahl Bits) in die Color Map
 - Gibt keine direkte Abbildung vom Pixelwert auf den Farbwert der Color Map, aber Aufgrund der geringeren Anzahl an Bits pro Pixel spart man Speicherplatz & Datenrate

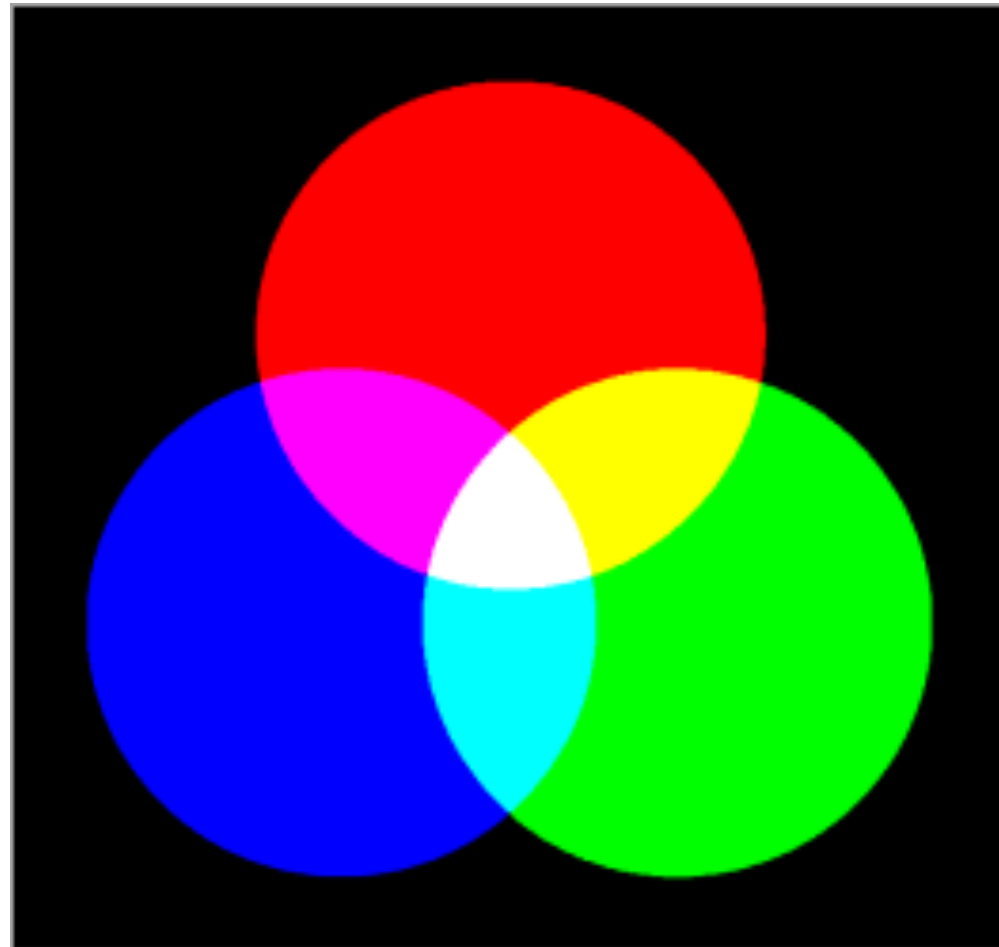


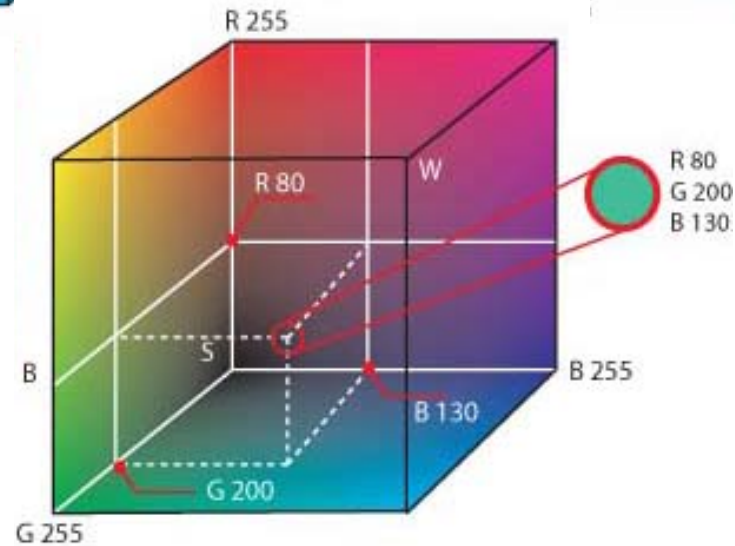
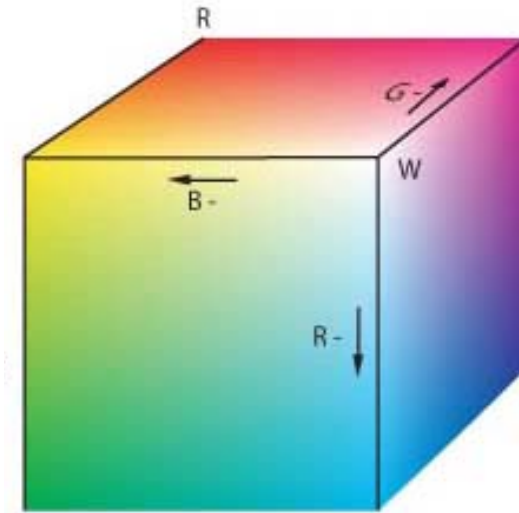
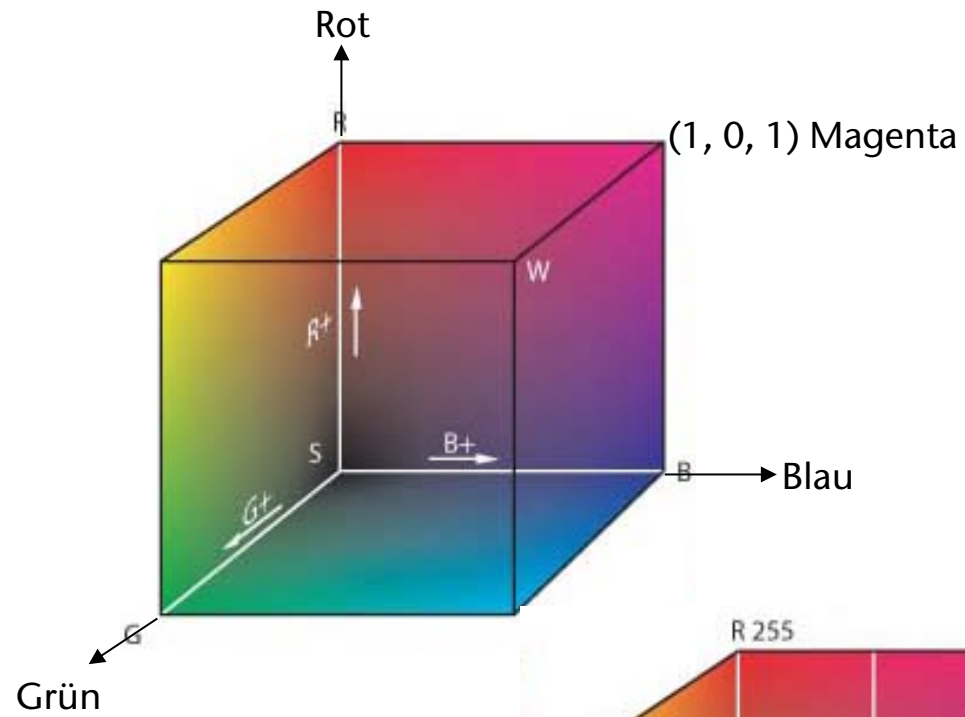
- Beispiel:
 - 8 Bit Farbe pro Pixel
 - 12 Bit breite Color Map
 - Das ergibt $2^{12} = 4096$ unterschiedliche Farben
 - Jeder Pixel durch 8 Bit dargestellt, kann nur $2^8 = 256$ Farben verwenden
 - Nehme 256 verschiedene Farben aus den möglichen 4096 und speichere sie in der Color Map
 - 8 Bit Farbwert eines Pixels indiziert einen Eintrag der Color Map
 - Die gespeicherte 12 Bit Farbe wird letztendlich angezeigt
- Wird heute nur noch selten gemacht, aber an anderer Stelle (in Algorithmen) taucht dieses Verfahren wieder auf ...

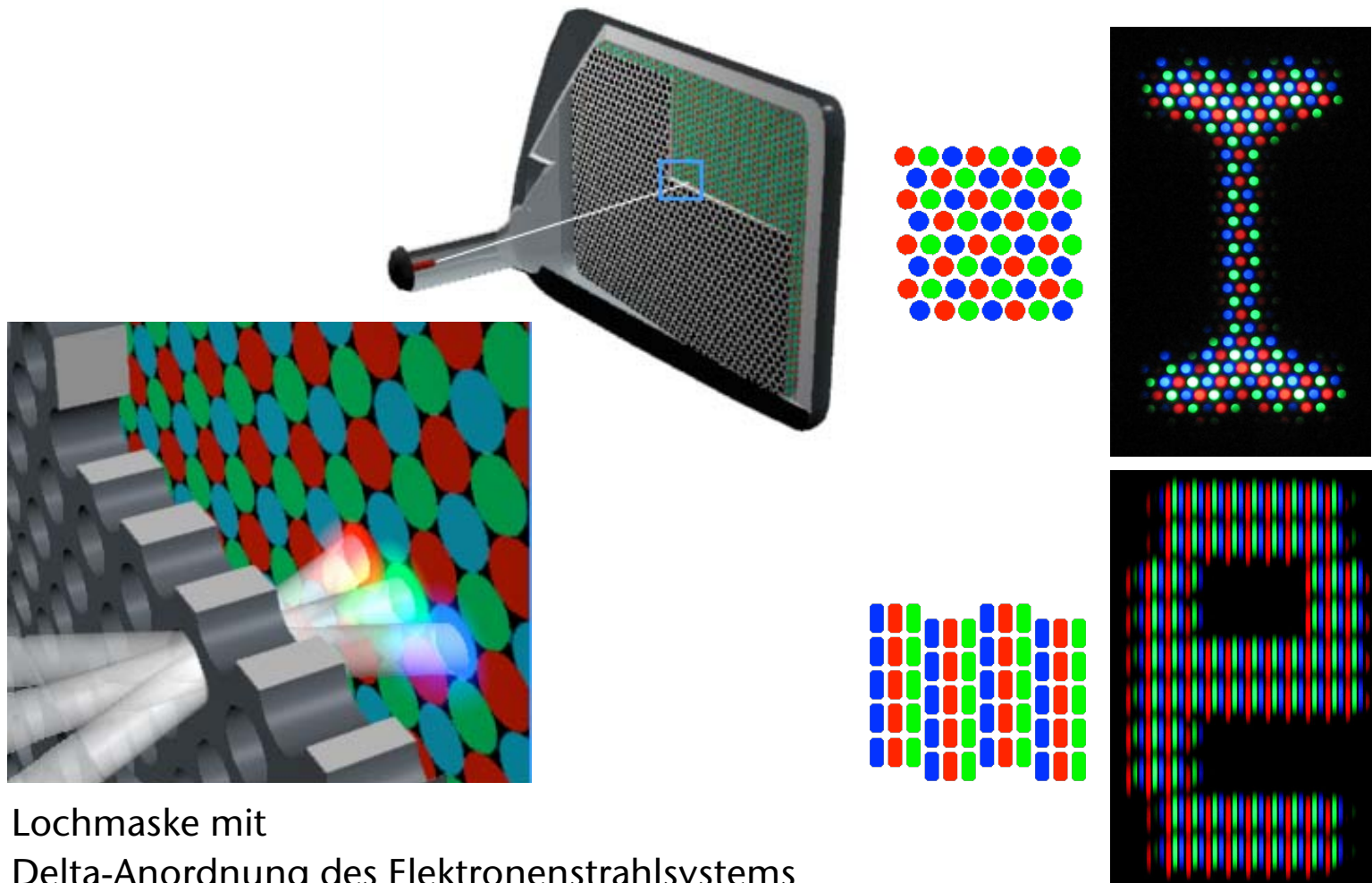
"640 Kilobyte ought to be enough for anybody."

Bill Gates, 1981

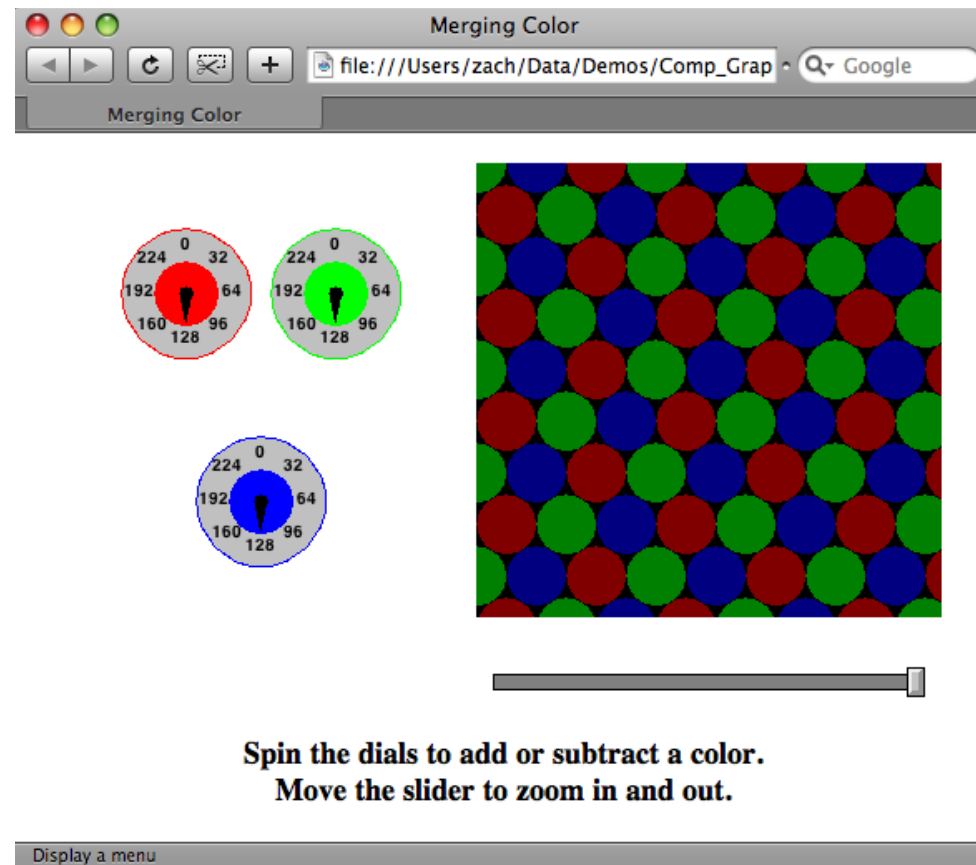
- RGB Farbmodell:
 - $(0, 0, 0)$ schwarz
 - $(1, 0, 0)$ rot
 - $(0, 1, 0)$ grün
 - $(0, 0, 1)$ blau
 - $(1, 1, 0)$ gelb
 - $(1, 0, 1)$ Magenta
 - $(0, 1, 1)$ cyan
 - $(1, 1, 1)$ weiß







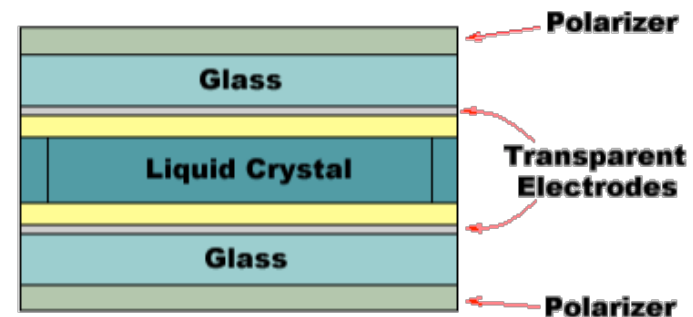
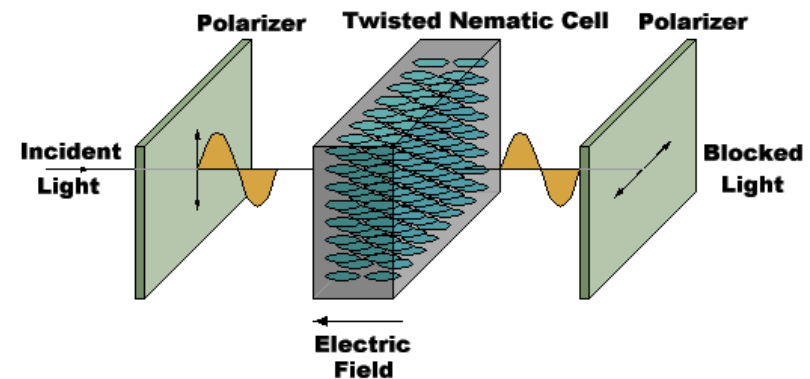
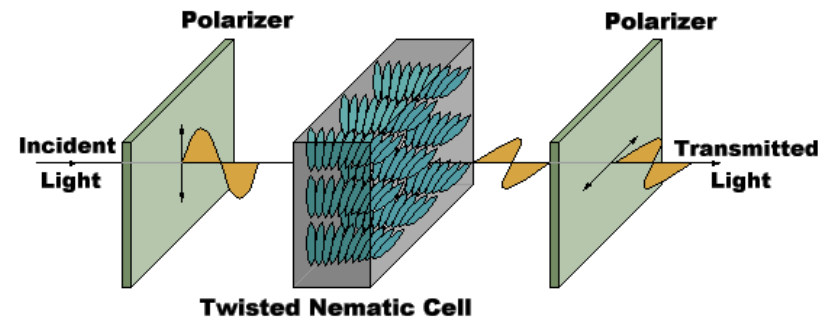
Lochmaske mit
Delta-Anordnung des Elektronenstrahlsystems
und punktförmigem Phosphor



http://www.colorado.edu/physics/2000/tv/merging_color.html

Liquid Crystal Displays (LCDs)

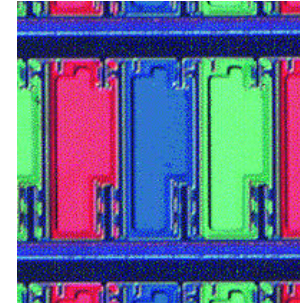
- LCDs lassen das Licht hindurch bzw. nicht und sind somit auf eine externe Lichtquelle angewiesen
- Laptop Bildschirme: von hinten beleuchtet, durchlässige Displays
- PDAs/ Handy: reflektierende Displays (+ Lichtquelle)



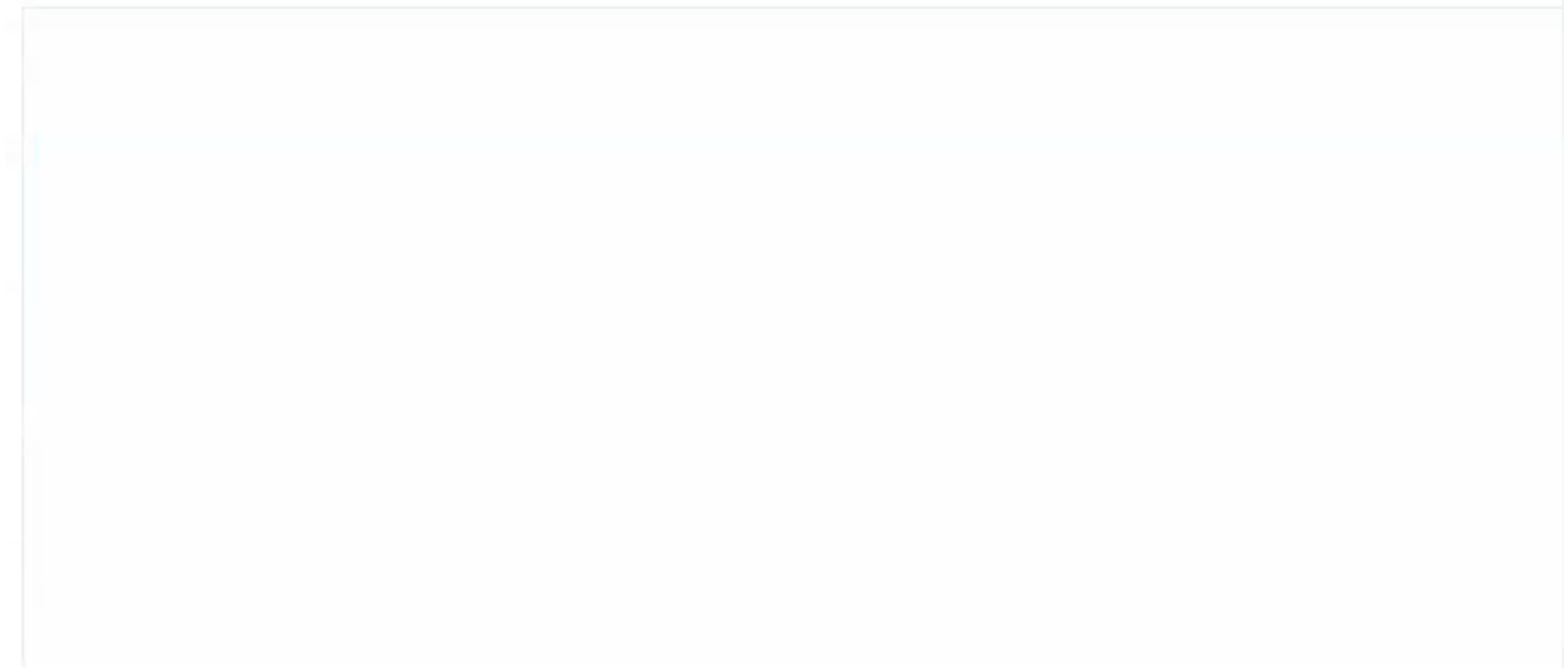
- Pixel besteht aus drei **Sub-Pixeln** mit R/G/B-Farbfiler
 - Jedes **Subpixel** ist ein Transistor!

- Leuchtmittel bei **transmissiven LCDs**
 - schmale Leuchtstofflampe oder LEDs an der Seite
 - Licht wird verteilt durch flachen Lichtleiter + Diffuser-Scheibe
 - Liefert etwas sichtbares Licht, vor allem UV-Spektrum
 - Beschichtung an der Innenseite des Glaspanels erzeugt daraus sichtbares Licht
 - Desktop-Displays an jeder Seite eine Röhre, Notebook-Displays nur zwei, um Strom zu sparen

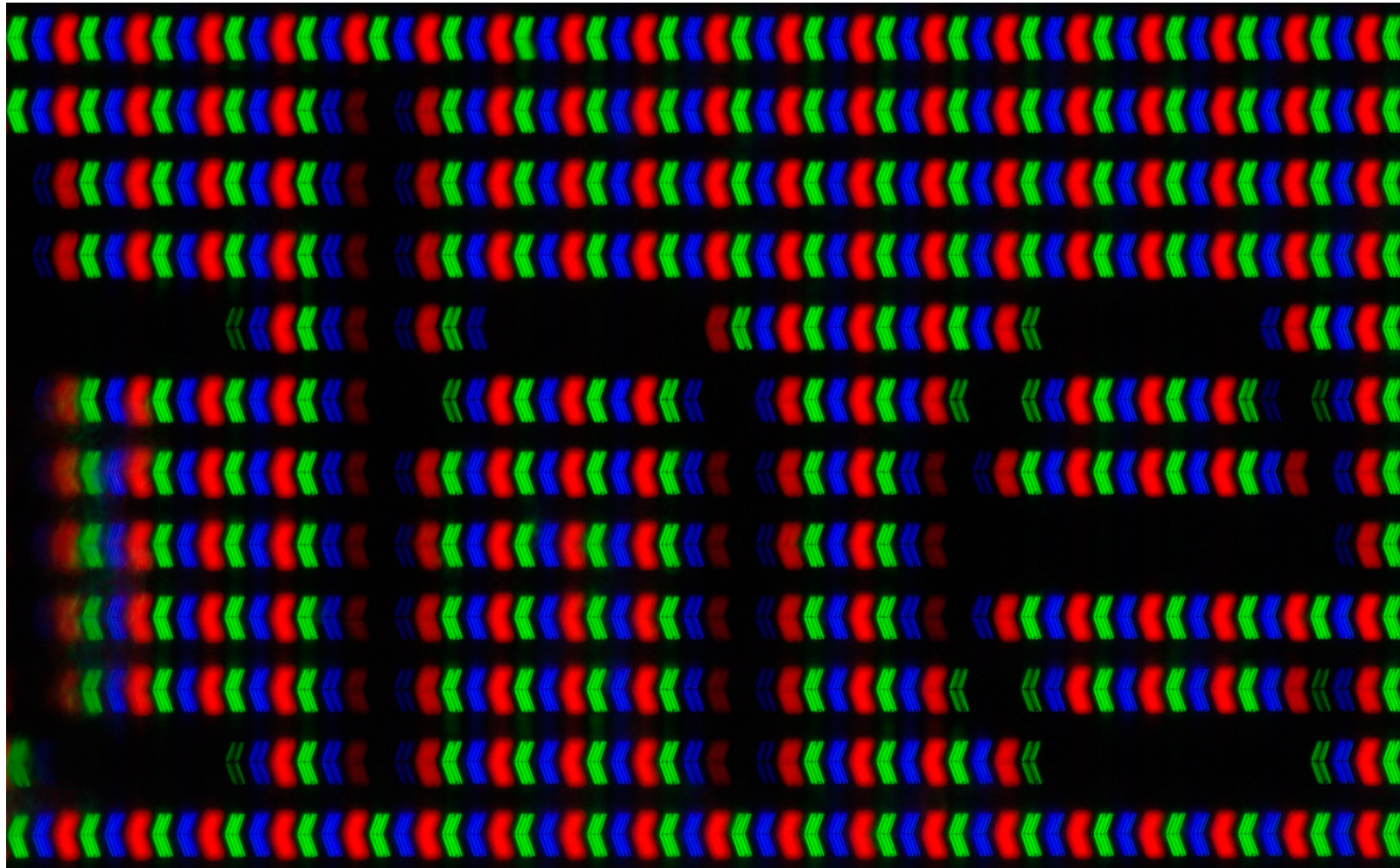
- **Reflektive LCDs** schalten Hintergrund nur bei Bedarf an



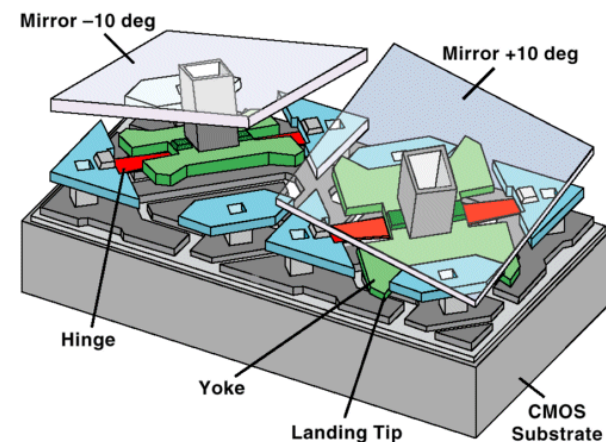
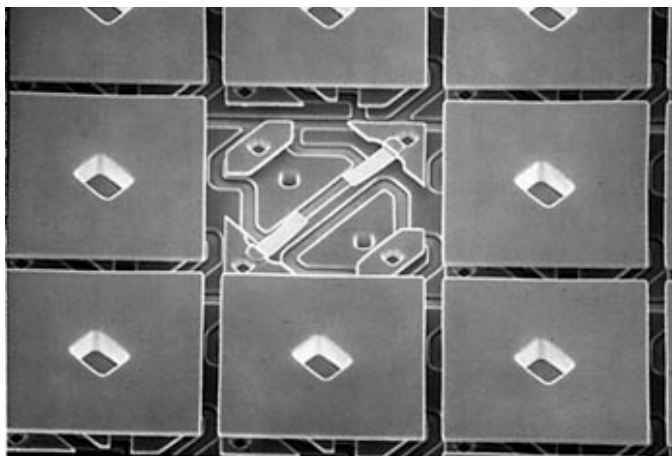
- Das Licht durchdringt den hinteren (vertikalen) Polarisator, wird dabei polarisiert (schwingt nur noch in einer Richtung)
- Nichtaktivierte Flüssigkeitskristalle drehen die Polarisierung um 90° → Licht gelangt durch vorderen (horizontalen) Polarisator
- Angeschaltete Transistoren erzeugen ein elektrisches Feld (in diesem Subpixel)
 - Das führt zu einer Drehung der Flüssigkristalle
 - Aktivierte Flüssigkeitskristalle ändern die Polarität des Lichtes **nicht**
 - Licht wird vom vorderen (horizontalen) Polarisator geblockt
- Die Transistoren werden Zeile für Zeile nach dem Scan-Line-Verfahren aktualisiert
- Die Kristalle müssen eine gewisse Zeit ausgerichtet bleiben, um Flimmern zwischen der Aktualisierung zu verhindern



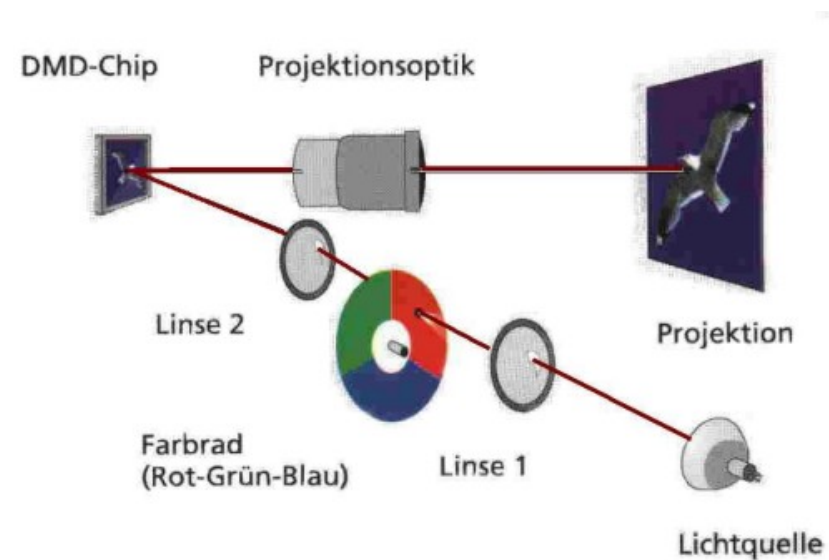
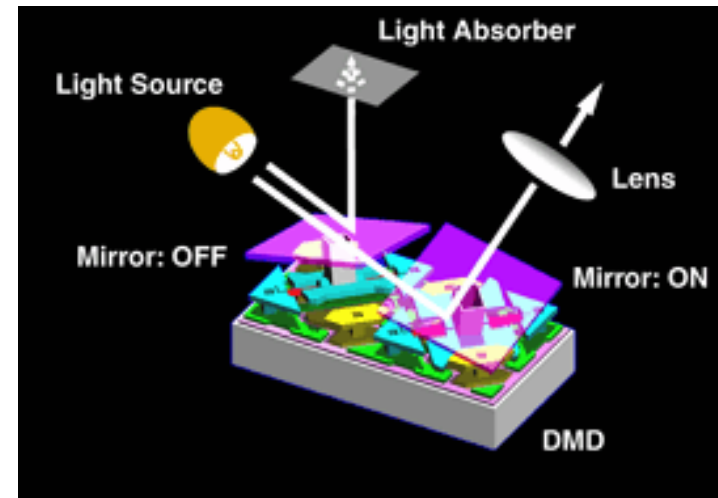
Ausschnitt aus http://solutions.3m.com/wps/portal/3M/en_US/Vikuiti1/BrandProducts/secondary/optics101/



- DMD-Chip (Digital Micromirror Device)
 - Kernstück eines DLP-Projektors
 - **Microelectromechanical** (MEM) Geräte werden mit VLSI Technik gefertigt
 - Auf 2 cm² über 508.000 reflektierende Mikro-Spiegel, jeder für sich um bis zu 10° kippbar
 - Jeder Spiegel kann einzeln elektrostatisch bewegt werden und schaltet genau ein Pixel hell oder dunkel



- DMDs haben „digitale“ Pixel →
 - Verschiedene Grauwerte durch Anpassen der Impulslänge
- Drei Grundfarben per rotierende RGB-Farbfiler-Scheibe oder mehrere Chips
- Vorteile:
 - Hochauflösend
 - Sehr flach
 - Sehr lichtstark
- Problem mit Flimmern





Quantum Dots



Weitere Displays

- Plasma
- OLEDs
- Laser ...

- Was ist hier der Unterschied?

Ohne Gammakorrktur



Mit Gammakorrktur

- Begriffe:

- Die **wahrgenommene(!) Helligkeit** = eine **physiologische** Größe
- Die **Intensität, I** = eine **physikalische** Größe
- **Dynamikbereich (dynamic range)** = Verhältnis max. / min. Intensität

1. Die Nichtlinearität im Auge:

- Beobachtung: eine Folge von Intensitäten I_0, I_1, \dots, I_k wird als **linear wahrgenommen** gdw.

$$\forall j : \frac{I_{j+1}}{I_j} \equiv \text{const.}$$

- Aufgabe: $k+1$ Intensitätsstufen I_j so im Intervall I_{min} bis I_{max} verteilen, dass die wahrgenommenen Helligkeitsstufen **linear** verlaufen
 - In der Praxis oft: $2^8 = 256$ oder $2^{12} = 4096$ Stufen

- Lösung: geometrische Folge

- $I_0 = I_{\min}$, $I_1 = r \cdot I_0$, \dots , $I_j = r^j \cdot I_0$, \dots , $I_k = r^k \cdot I_0$
- I_{\max} / I_{\min} kann man messen \rightarrow

$$r = \left(\frac{I_{\max}}{I_{\min}} \right)^{1/k}$$

- Korrektur der wahrnehmungspsychologischen Nicht-Linearität:

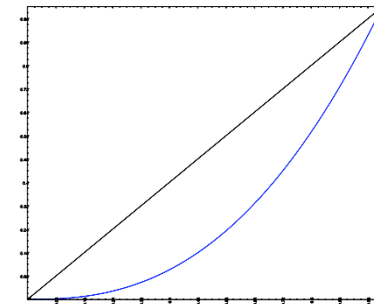
- Gegeben: $j = \text{Pixel-Wert im Framebuffer}$
- Bestimme: $I_j = r^j \cdot I_{\min}$
- Evtl. in LUT ablegen (als Preprocessing / in HW)

2. Die Nichtlinearität im Monitor:

- Beobachtung: bei Eingangsspannung V liefert ein Monitor eine Ausgangsintensität I (an einem Pixel) von

$$I = I_{\max} \left(\frac{V}{V_{\max}} \right)^\gamma$$

- Typischer Wert ist $\gamma = 2.5$



- Die Gamma-Korrektur:

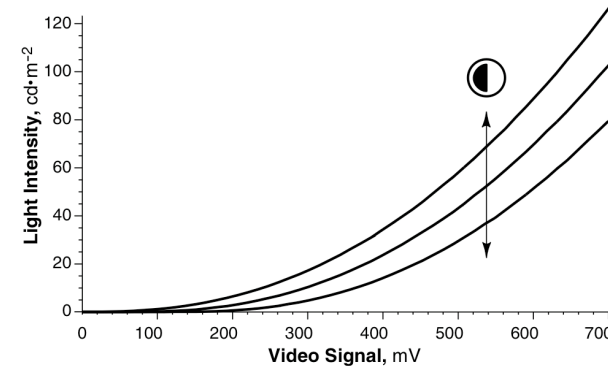
- Gegeben I_j

- Bestimme:

$$V_j = \left(\frac{I_j}{I_{\max}} \right)^{1/\gamma} \cdot V_{\max}$$

- Passiert nach der DA-Konvertierung (oder gleichzeitig) auf Rechnerseite

- Bemerkung: "Contrast"-Knopf am Monitor ändert einfach das Gamma des Monitors



- Wahrnehmungspsychologische Korrektur "für Arme":

- Approximiere

$$I_j = f(j) = r^j \cdot I_{\min}$$

durch

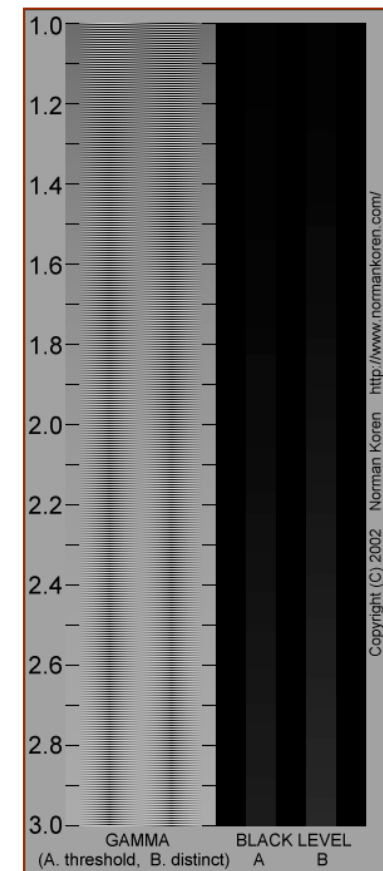
$$I_j \approx j^\gamma \cdot I_{\min}$$

- Damit wird

$$V_j = \left(\frac{I_j}{I_{\max}} \right)^{1/\gamma} \cdot V_{\max} \approx \left(\frac{j^\gamma \cdot I_{\min}}{I_{\max}} \right)^{1/\gamma} \cdot V_{\max} = j \cdot c$$

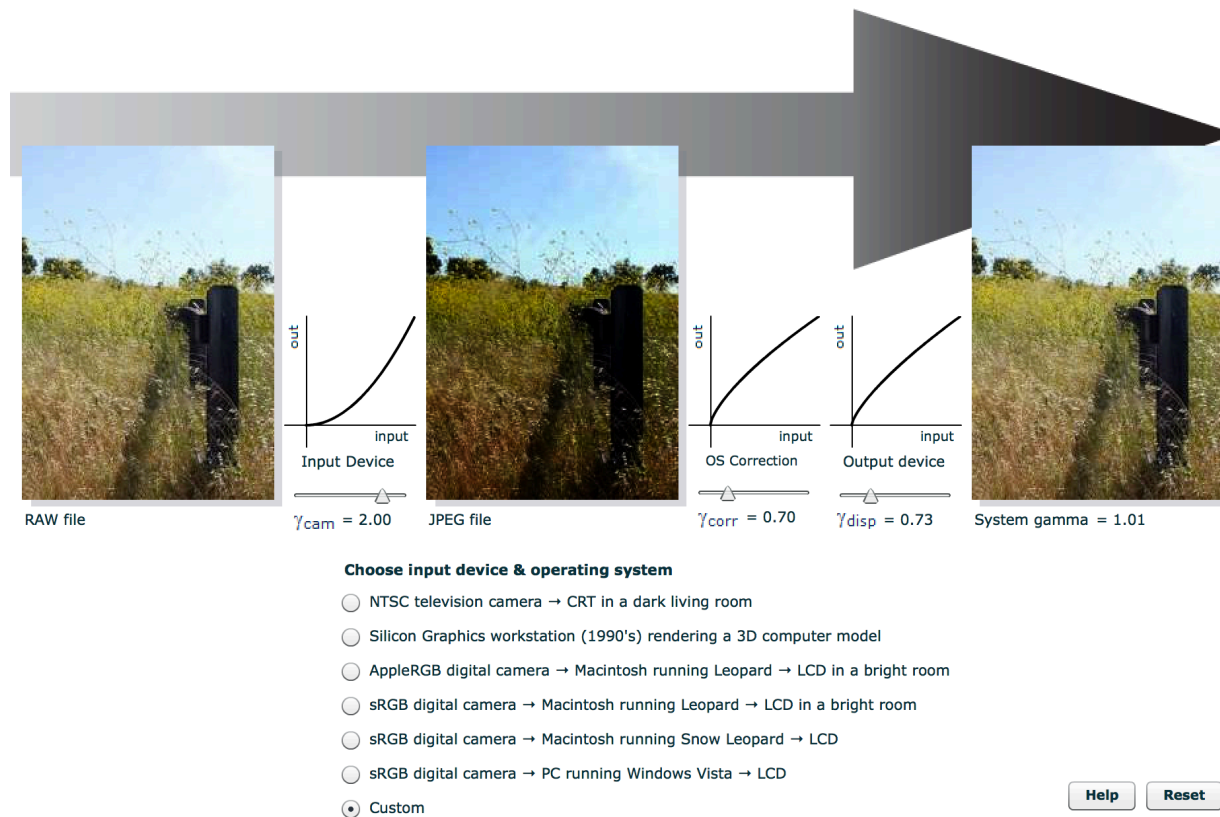
Wie bestimmt man das Monitor-Gamma?

- Achtung: (older / cheap) LCD screens are poorly suited for critical image editing because gamma is extremely sensitive to viewing angle!
 - Testbild zur visuellen Bestimmung des aktuellen Gammas des Gesamtsystems
 - Aktuelles Gamma ist dort, wo ein einheitlicher Grau-Level auf einer horizontalen Linie zu sehen ist
 - Die Methode:
 - Schwarze & weiße Pixel werden — **unabhängig von γ !** — als keine bzw. volle Helligkeit wahrgenommen →
 - Die beiden Streifenmuster werden aus der Entfernung als halbe Helligkeit wahrgenommen
 - Finde den Pixel-Grauwert a , so daß $\frac{1}{2} = a^\gamma$
- Führe das für verschiedene γ durch und eiche damit die Skala



- Das Chaos (jahrzehntelang):
 - Verschiedene Monitor- und System-Gammas
 - Unklarheit darüber, an welcher Stelle in der Pipeline die Gamma-Korrektur gemacht werden soll:
 - Im Bild? (manche Tools haben das Bild schon gamma-korrigiert abgespeichert!)
 - In der Software? (= vor dem sog. Frame-Buffer, d.h., im Browser, Video-Spiel, Photoshop, ...)
 - In der Graphikkarte? (= beim Auslesen und Konvertieren des Frame-Buffers)
 - Im Monitor? (= vor der Ansteuerung der CRT-Kanone / der LCD-Transistoren)
- Die Lösung: *Color Profiles*
 - Speichern Info, ob das Bild schon gamma-korrigiert wurde, und – falls ja – mit welchem Gamma (speichern noch viel mehr, u.a. den Farbraum)
 - Kann man in moderne Bildformate einbetten (z.B. JPG, TIF, PNG)

- Auf der Homepage der VL finden Sie ein interaktives Applet (cgvr.cs.uni-bremen.de → Teaching → Computergraphik 1):



RAW file $\gamma_{cam} = 2.00$ JPEG file $\gamma_{corr} = 0.70$ $\gamma_{disp} = 0.73$ System gamma = 1.01

Choose input device & operating system

- NTSC television camera → CRT in a dark living room
- Silicon Graphics workstation (1990's) rendering a 3D computer model
- AppleRGB digital camera → Macintosh running Leopard → LCD in a bright room
- sRGB digital camera → Macintosh running Leopard → LCD in a bright room
- sRGB digital camera → Macintosh running Snow Leopard → LCD
- sRGB digital camera → PC running Windows Vista → LCD
- Custom

