

Einführung in die Computergrafik

Prof. Dr. Matthias Hullin

Universität Bonn, Institut für Informatik,
Abteilung 2: Visual Computing

Organisatorisches

- Dozent (Vorlesung)
- Matthias Hullin (hullin@cs.uni-bonn.de)

Tel.: 0228-73 54169

Adresse: Friedrich-Hirzebruch-Allee 8
53115 Bonn (Room 3.029)

Sprechzeiten: nach Vereinbarung

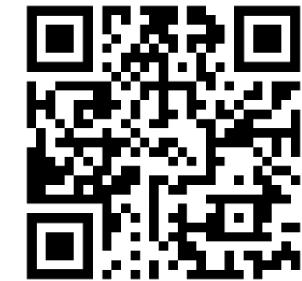


- Vorlesung (4h)
 - Montag: 14:15 – 15:45 (CP1-HS3)
 - Freitag: 12:15 – 13:45 (CP1-HS3)
- Ggf. werden manche Vorlesungen vorab aufgezeichnet und als Video bereitgestellt werden. Dies kündige ich rechtzeitig vorher an.

Kommunikation

Fachschafts-**Discord** - <https://discord.gg/mrDeFcqYHa>

Modulchannel freischalten:



- **eCampus:**



A screenshot of a computer screen showing a list of modules in eCampus. The list includes:

- Wahlpflichtbereich Informatik
- BA-INF 105 - Einführung in die Computergraphik und Visualisierung

The second item, "BA-INF 105 - Einführung in die Computergraphik und Visualisierung", is circled in green. A green arrow points from the word "Informatik" in the first item to the start of the circled module name.

- Rufen Sie Ihre „s6....@uni-bonn.de“ Mails regelmäßig ab!
Ggf. kann es kurzfristige/dringende Mitteilungen geben.
- **Bitte erinnern Sie mich, falls benötigte Materialien noch nicht online sind**

Organisation der Übung

- Orga/Tutor: **Julian Stamm** (s6justam@uni-bonn.de)
- **Übung (2h)**
 - Einteilung von Übungsgruppen über eCampus
 - Übungsblätter & Materialien über eCampus
 - „Lektion 1“ jetzt verfügbar
 - Diskussion in Übungsgruppe

Tutorien

- Organisation der Übungsblätter
 - 8 Blätter
 - Praktischer Teil: Programmieraufgaben (50 %)
 - Theoretischer Teil: Aufgabenblätter (50 %)
 - Abgaben in Gruppen von 3 Personen
- Inhalt des theoretischen Teils:
 - Nachvollziehen von Konzepten, Algorithmen, mathematischen Grundlagen. Klausurrelevant!!!
 - Korrektur und Präsentation in den Gruppen
 - Abgaben über eCampus als PDF (am liebsten LaTeX, notfalls handgeschrieben und klar und deutlich digitalisiert)

Tutorien

- **Inhalt des praktischen Teils**
- Einführung in Grafikprogrammierung mit C (C++) und OpenGL
- Benötigte Software: Editor, Compiler (gcc, clang, MSVC)
- Übungen bestehen i.d.R. aus unfertigem Code, der zu ergänzen und zu kommentieren ist
- Lösungen werden als .cpp-Dateien abgegeben

Tutorien

Übungsgruppen (ab übernächster Woche):

- Montag 12-14 (Raum angefragt)
- Donnerstag 10-12, 3.035b

Wichtig:

- **Anmeldung in eCampus** ist Voraussetzung für Übungsteilnahme.

CGIntro Kick-Off Helpdesk

Probleme mit dem Übungsframework?

- “Helpdesk” mit Julian:
im Anschluss an die Vorlesung am kommenden
Montag

Klausur (schriftlich)

Inhalt: Vorlesungsinhalt plus Inhalte aus den Übungen

Termine (voraussichtlich):

- ~~04.08.2024, 9:00-11:00, HSZ-HS1~~ tbd
- ~~18.09.2024, 13:00-15:00, Meckenheimer Allee 176, HS II~~

Zulassungsvoraussetzungen

- „Vorrechnen“ im Tutorium
- 50% der Übungspunkte für die erste Semesterhälfte
 - 50% korrekte praktische UND 50% korrekte theoretische Aufgaben für Klausur vorausgesetzt!
- Teilnahme an Computer Animation Competition

Moment --- Computer Animation Competition?

- Ja! In der 2. Semesterhälfte werden Sie in Teams eine Computeranimation erstellen.
- Freie Projektarbeit
- Konzipieren Sie eine kurze Animation (10-60 Sekunden) und erweitern Sie Ihre Grafikengine, um diese umzusetzen. Abgabe als Quellcode mit Doku
- Wählen Sie von uns vorgeschlagene technische Features aus, oder schlagen Sie eigene Ideen vor.
- Tutorien = "Helpdesk" für Animationsprojekte
- evtl. 1-2 Extra-Vorlesungen zu den Themen
 - Animation
 - Storytelling

Übungsblatt 9

- Produzieren Sie einen computergrafischen Kurzfilm.
- Abgabe: Heute in 6 Wochen
- *Scheitern ist erlaubt (wenn es gut dokumentiert ist)*

Computer Animation Festival

- In der letzten Vorlesungsstunde: Vorführung Ihrer Animationen im feierlichen Rahmen

Einige Beiträge aus vergangenen Jahren finden Sie hier:

https://youtube.com/playlist?list=PLBQN9Lxu5Q5_MPKG6SQYKbXw6MOEEEJBe



Geschichte der Computergrafik

- 1950 **Kathodenstrahlröhre** zur Bildausgabe unter Computersteuerung
- 1955 **Lichtgriffel** zur Anwahl von Punkten auf dem Schirm
- 1963 **Sketchpad**-System (Sutherland)
Datenstrukturen / Bildhierarchien, Bildkomposition, Interaktionstechniken (Tastatur, Lichtgriffel)
- 1959-1961 **Bézierkurven** (de Casteljau, Bézier) werden bei Citroën und Renault entwickelt und eingesetzt Erste CAD-Systeme für Freiformmodellierung
- Mitte 60-er Jahre: Erstes **Rasterdisplay**
- 1965 **Maus**, TVEdit: Screeneditor (D. Engelbart, Stanford)
- 1968 Computer Supported Cooperative Work (Engelbart)
– “**The Mother of All Demos**” ← check it out!!
- 1969 **VR/AR Headset** (Sutherland)

Classic Vector and Raster Fonts

! " # \$ % & ' () *
+, - . / 0 1 2 3 4 5
6 7 8 9 : ; < = > ? @
A B C D E F G H I J K
L M N O P Q R S T U V
W X Y Z [\] ^ _ ' a
b c d e f g h i j k l
m n o p q r s t u v w
x y z { | } ~

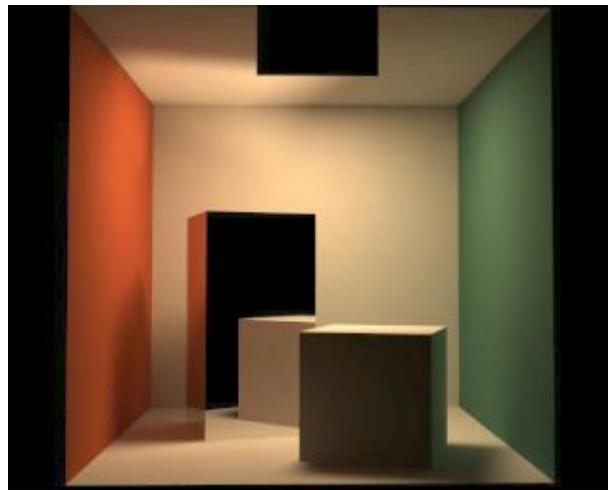


Dr. Allan Hershey,
ca. 1967

80er-Jahre MacOS

Geschichtlicher Überblick

- Anfang der achtziger Jahre **Hardwareunterstützung von 2D-Operationen**
(rasante technologische Entwicklung von Grafikhardware)
- 1983 **Erstes GUI-System** (Apple)
- ~1985 **Fotorealistische Grafik** (Raytracing, Whitted 1980; Radiosity, Goral 1984)



Cornell Box [Goral et al. 1984]. Von links: Foto, Rendering, Differenzbild.

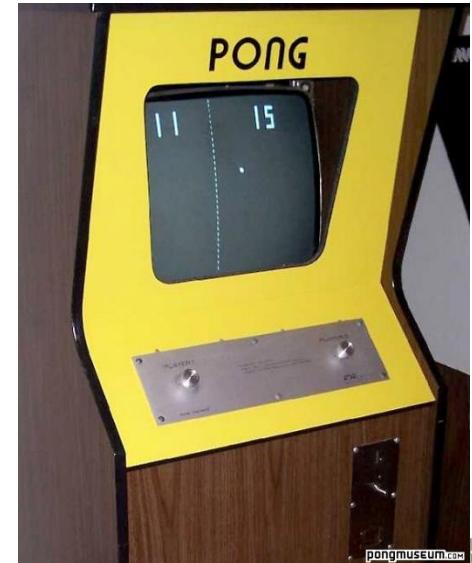
Geschichtlicher Überblick

Ära der Vektorgrafikspiele

- 1962 **Spacewar!**
auf PDP-1, Steve Russell
- 1972, **Pong** (Atari)
- 1979, **Asteroids** (Atari)



- 80er-Jahre: Heimkonsolen
(Atari 7800, NES, etc.);,
Übergang zu Rastergrafik



Grafikstandards und GPUs

- CORE 1977, GKS 1985, PHIGS 1988, PHIGS+ 1992, OpenGL 1992, Direct3D 1996, QuicktimeVR 1997
- 1976 “**Pixie**” video signal generator (RCA)
- Seit 1980er rasante Entwicklung bei 2D-Beschleunigung, später 3D
- 1980er “**Pixel Planes**” (University of North Carolina, H. Fuchs)
- 1993 **T&L** (Transform, Clipping, Lighting) Prozessoren für SEGA-Arcadesysteme
- 1997-98 Erste **3D-Beschleuniger** für PC-Spiele (PowerVR, 3dfx Voodoo, ATI Rage Pro, Nvidia Riva, ...)

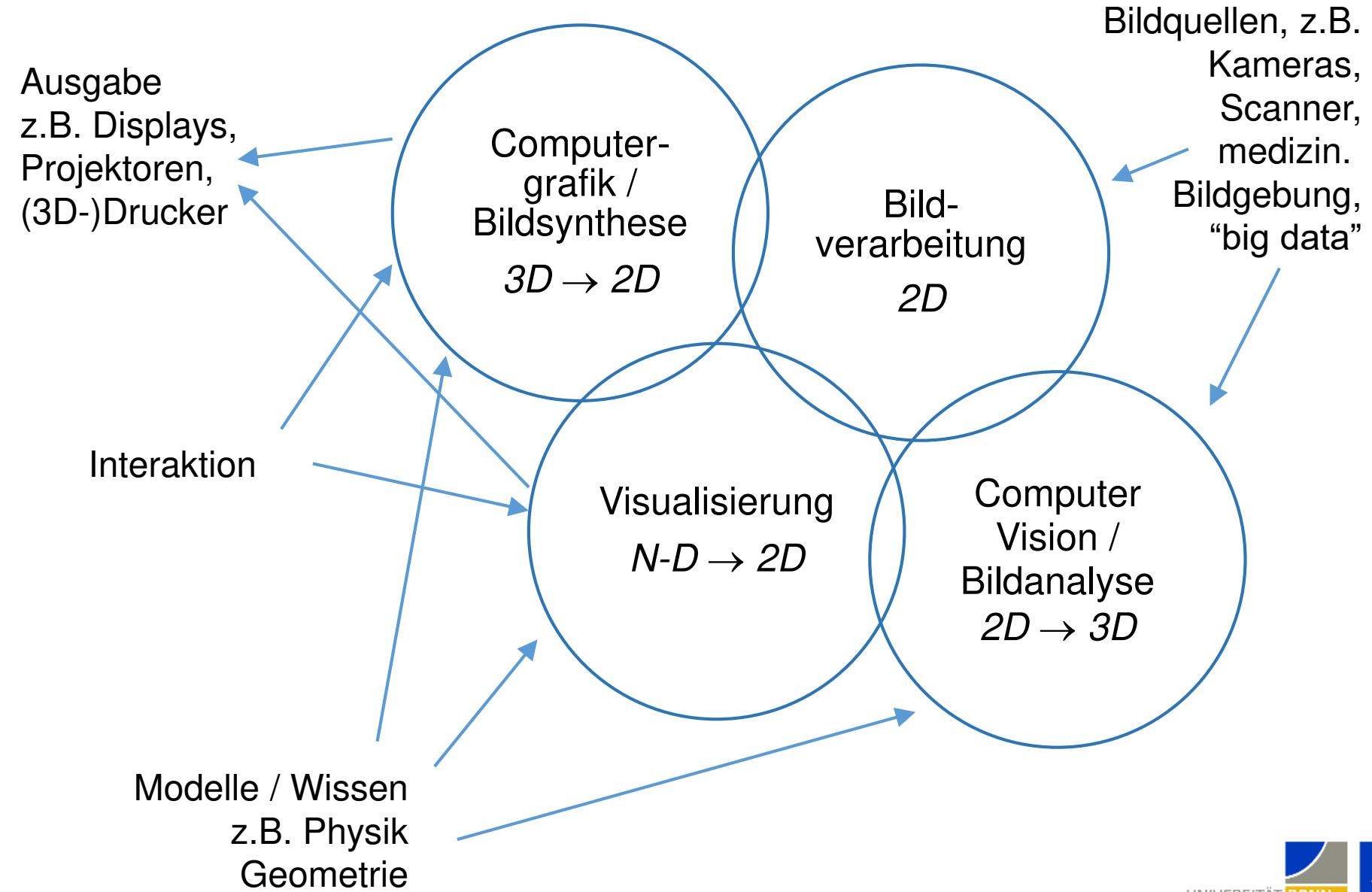
Das GPU-Zeitalter

- Heutige GPUs haben tausende Kerne – sehr effizient für paralleles Rechnen
- Allgemeine und spezialisierte Kerne: Rasterisierung, Vektorrechnung, Texturierung

GPGPU (General-Purpose Computing on GPUs)

- Erste Verwendung von GPUs für allgemeine mathematische Berechnungen (Matrix-Vektor-Multiplikation) um 2001; LU-Faktorisierung [Galoppo 2005]
- heute: OpenCL, CUDA, “compute shaders” ermöglichen deep learning, wiss. Simulation, Verschlüsselung, Videokompression, Kryptowährungen, ...
- Seit 2018: Raytracing in Hardware gegossen

Visual Computing / Grafische Datenverarbeitung



Technik und Methodik

- Hochleistungsrechnen (CPU, GPU)
- Mathematik
- Datenstrukturen und Algorithmen
- Physik (Optik, Mechanik, ...)
- menschliche Wahrnehmung und Interaktion
- Ein- und Ausgabegeräte (Sensoren, Displays)
- ...

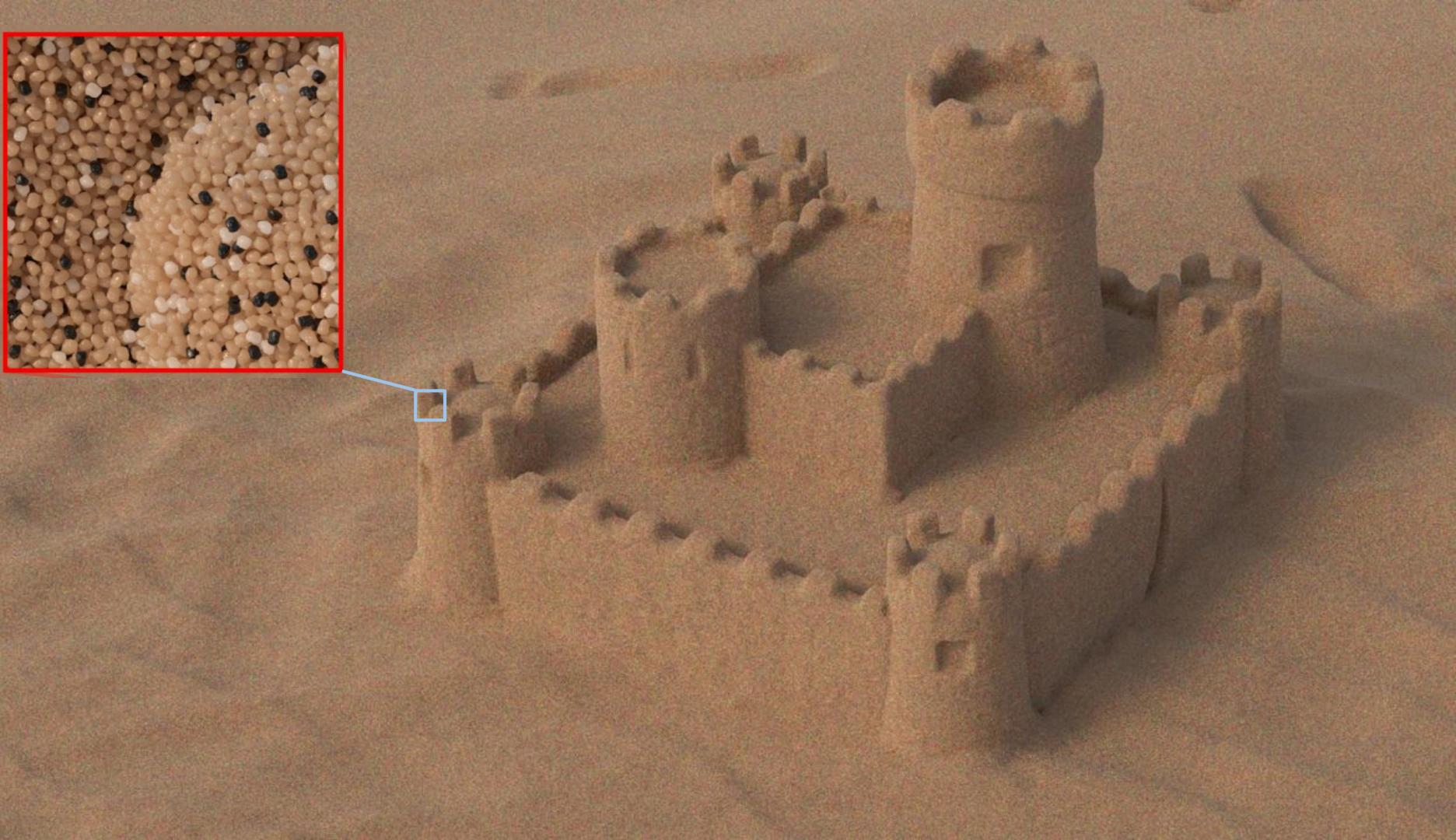
Technical Papers 	Art Papers 	Talks 	Production Sessions 	Panels 	Courses 
PROGRAM	PROGRAM	PROGRAM	PROGRAM	PROGRAM	PROGRAM
Education Forum 	Frontiers 	Awards 	Electronic Theater  Computer Animation Festival	Real-Time Live! 	VR Theater 
PROGRAM	PROGRAM	PROGRAM	PROGRAM	PROGRAM	PROGRAM
Immersive Pavilion 	Art Gallery 	Partners 	ACM SIGGRAPH Village 	Labs Experience Hall 	Emerging Technologies  Experience Hall
PROGRAM	PROGRAM	PROGRAM	PROGRAM	PROGRAM	PROGRAM
Games 	Diversity, Equity, and Inclusion 	Birds of a Feather 	Appy Hour 	Student Volunteers 	Student Research Competition 
PROGRAM	PROGRAM	PROGRAM	EVENT	PROGRAM	EVENT
Lead 	Keynote Presentations 				
PROGRAM	PROGRAM				

Anwendung

- Spiele
- Filme
- Produktdesign
- Architektur
- Kunst
- Simulation
- Wissenschaftliche und Informations-Visualisierung
- Virtual/Augmented Reality
- Therapie (Traumabewältigung)
- ...

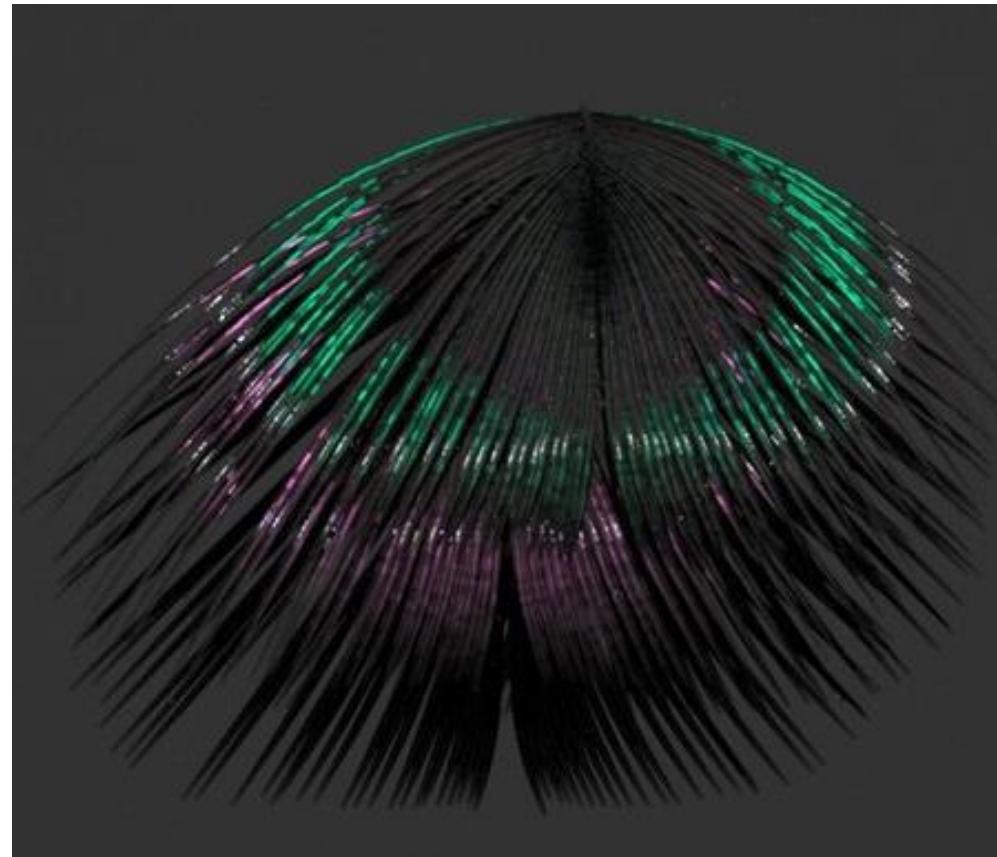
Beispiele aus der Forschung

Sand



[Meng et al., SIGGRAPH 2015]

Haare und Federn



[Huang et al., EGSR 2022 und SIGGRAPH 2022]

Transluzenz von Zähnen



[Velinov et al., SIGGRAPH Asia 2018]

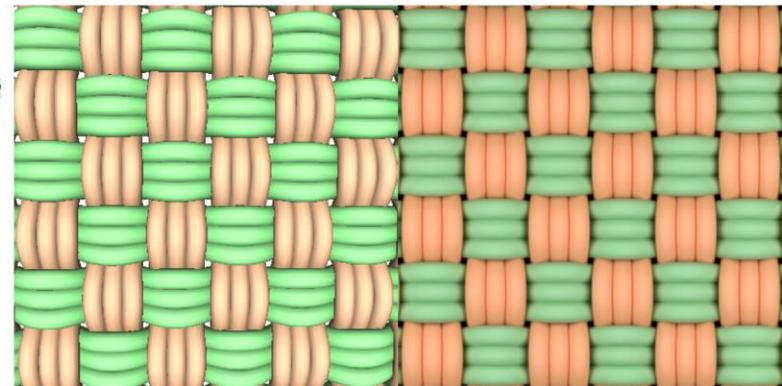
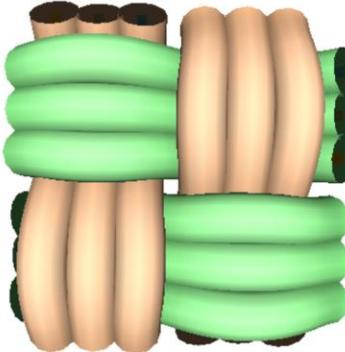
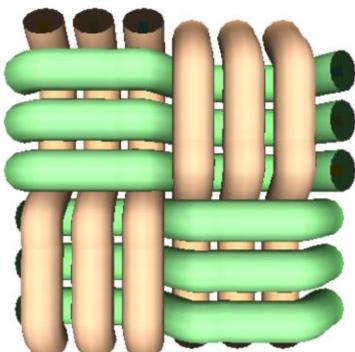
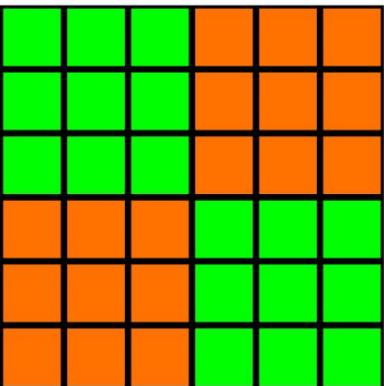
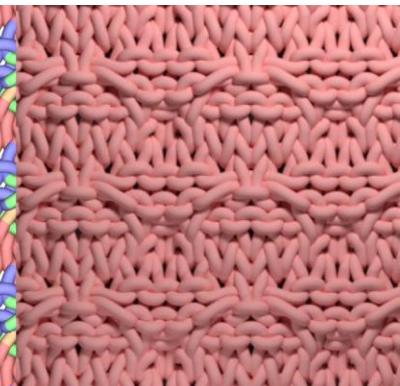
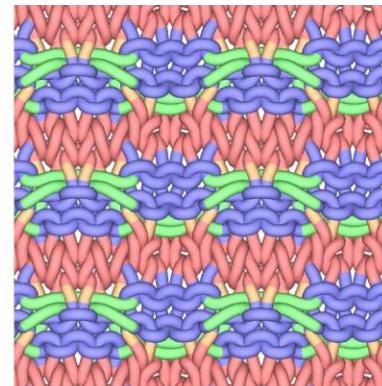
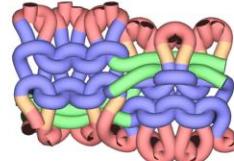
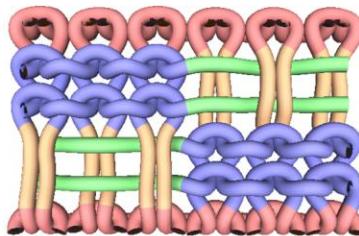
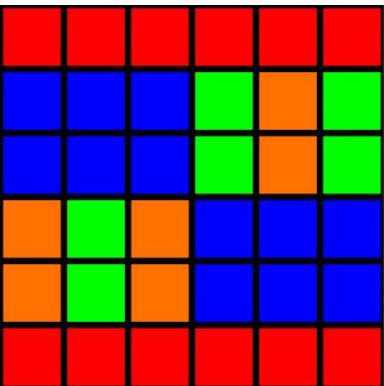
Geschichtete Materialien



[Zeltner und Jakob, SIGGRAPH 2018]

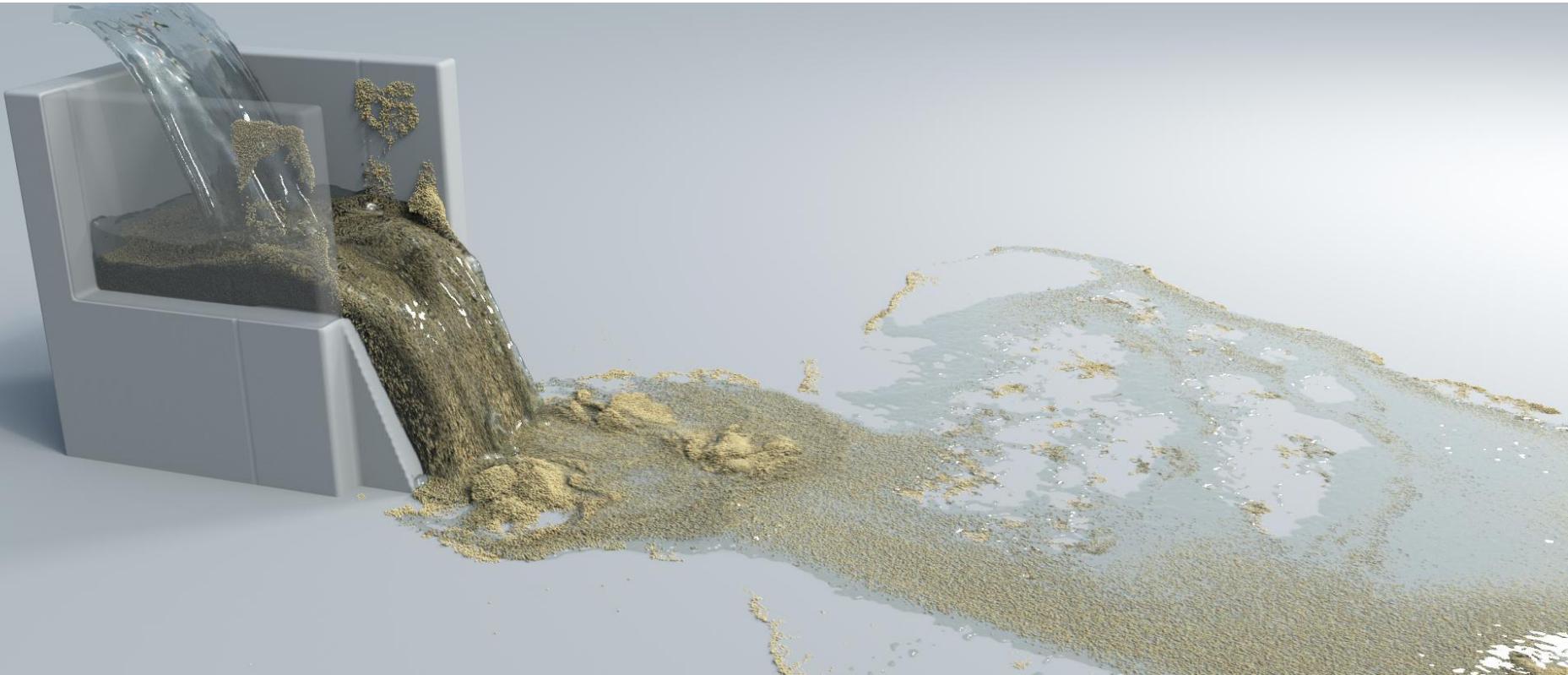
Web- und Strickmuster entwerfen

[Leaf et al., SIGGRAPH 2015]



Matsch simulieren

[Tampubolon et al. SIGGRAPH 2017]



3D mit Reißverschluss



[Schüller et al., SIGGRAPH 2018]

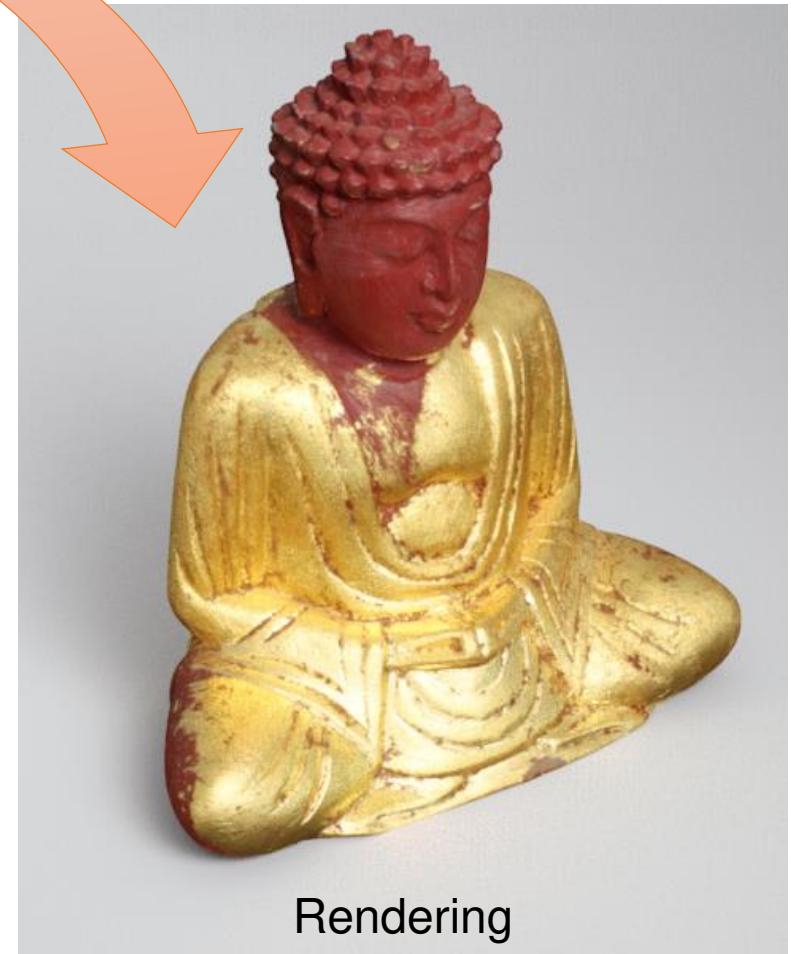
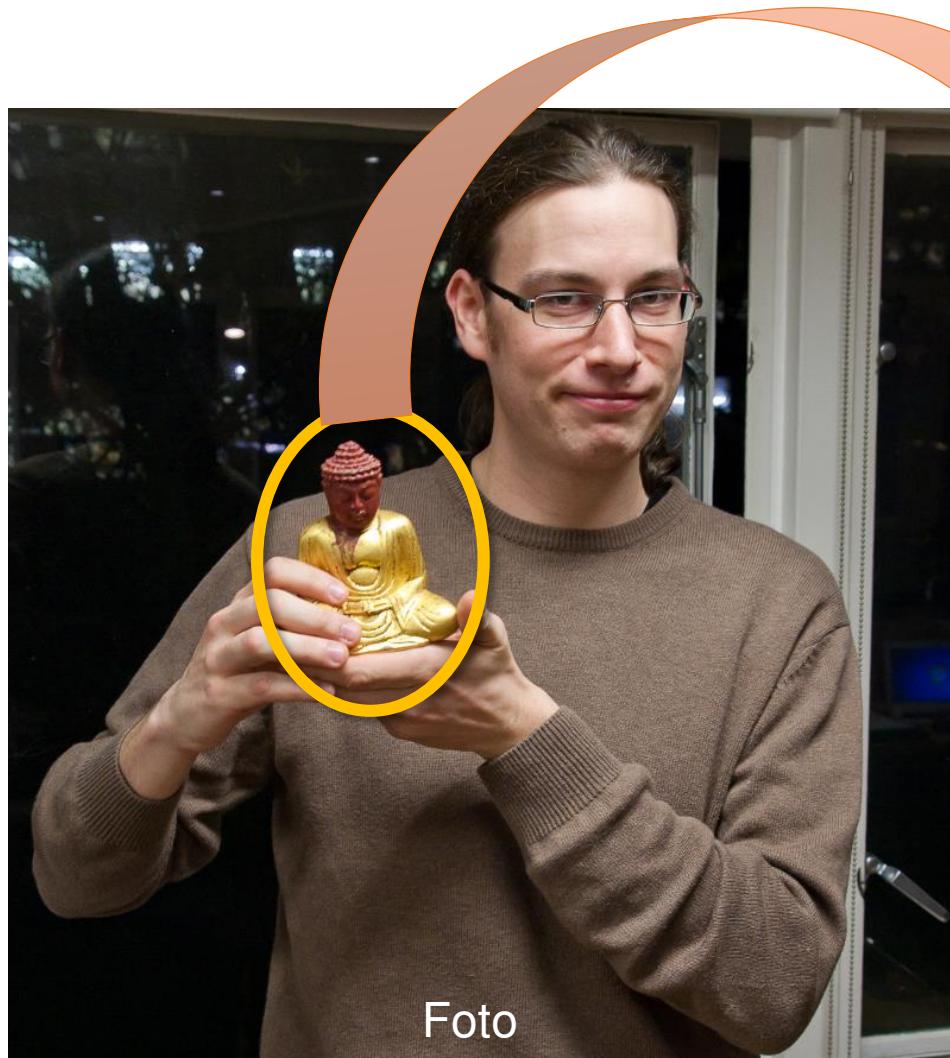
3D-Modelle komplexer realer Szenen

- [Kerbl et al., SIGGRAPH 2023]



Computergrafik in Bonn

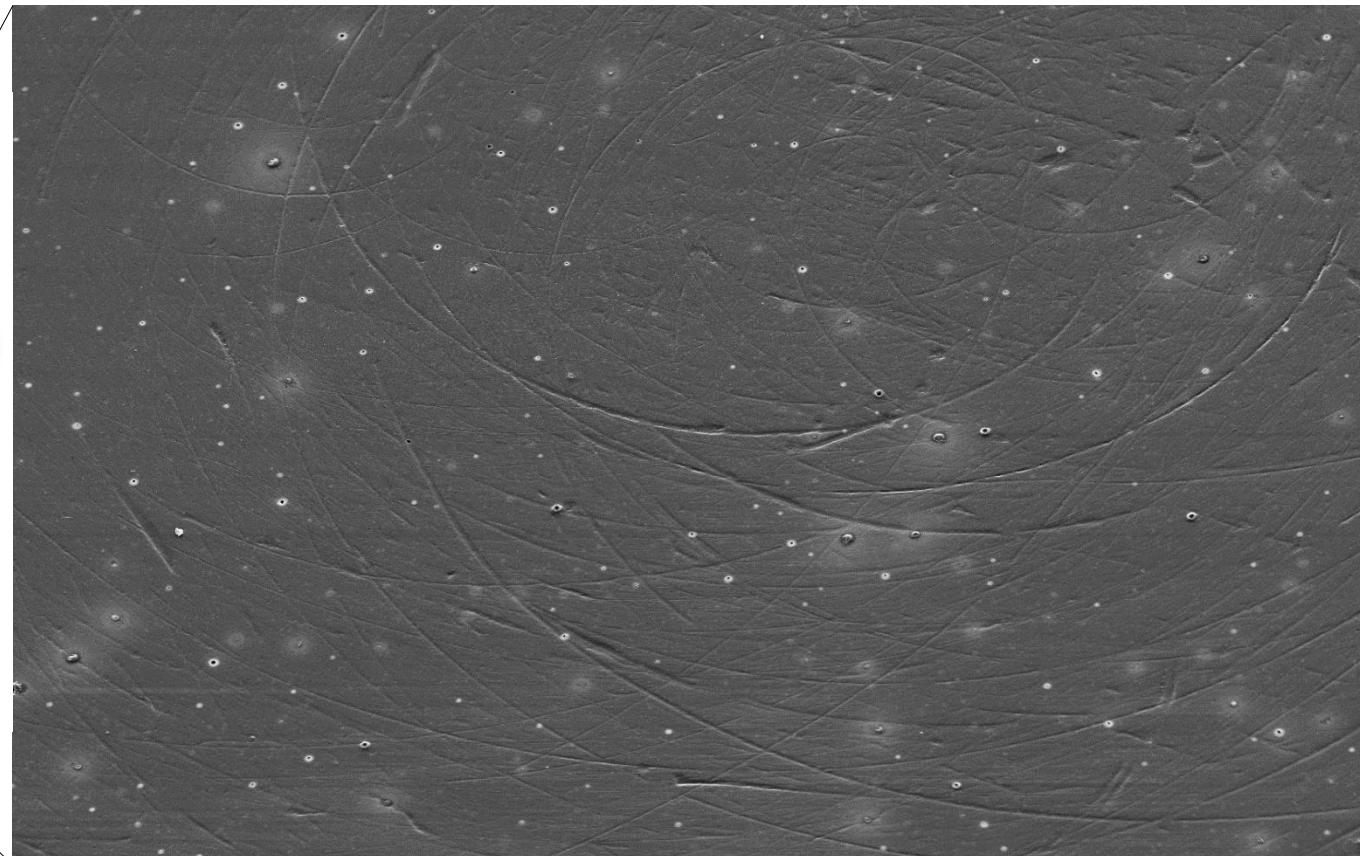
Objekte und Materialien digitalisieren



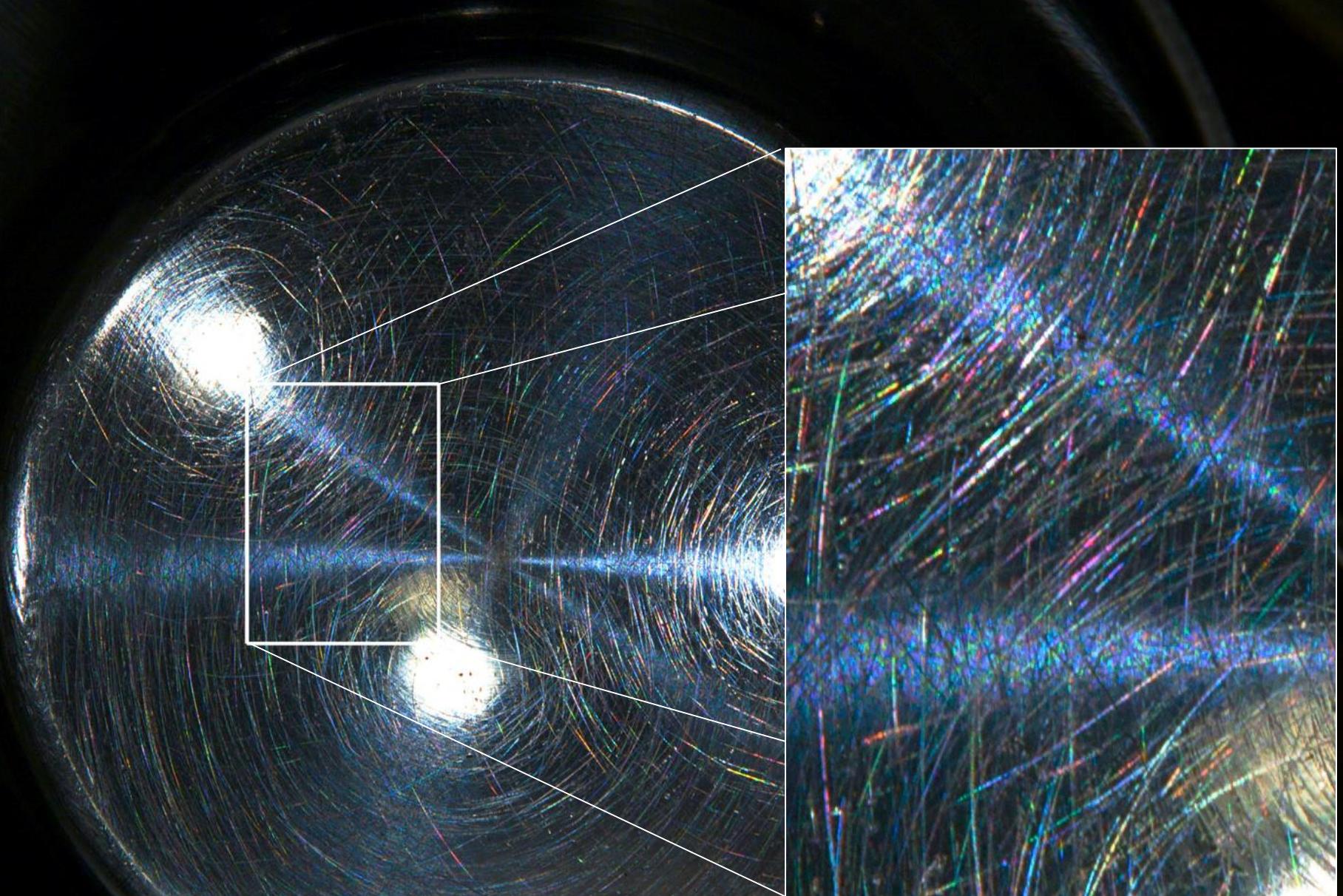
Schillernde Kratzer [Werner et al., SIGGRAPH Asia 2017]



IKEA 365+ cooking pot,
REM image of inner bottom



— 400 µm —





File View Window

Scratch Properties

▼ Dimensions

◀ Width us: 3 ▶

◀ Width (Std Dev) us: 3 ▶

Width Distribution:

Gaussian ▼

◀ Depth us: 3 ▶

◀ Depth (Std Dev) us: 3 ▶

Depth Distribution:

Gaussian ▼

◀ Length us: 5000 ▶

▼ Distribution

◀ Radius us: 2000 ▶

◀ Count 10 ▶

▼ Brush Properties

Brush ▼

Add

Brush

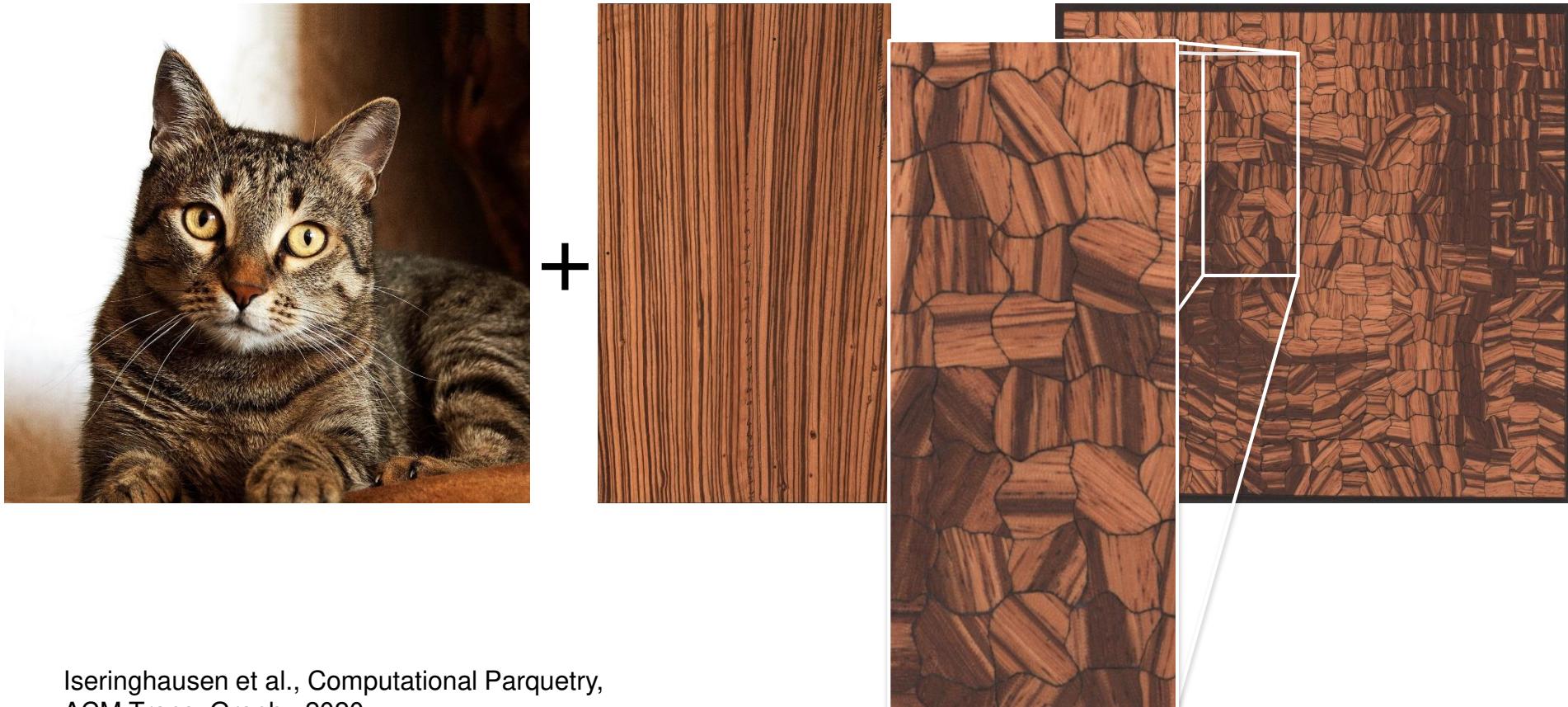
Distribute



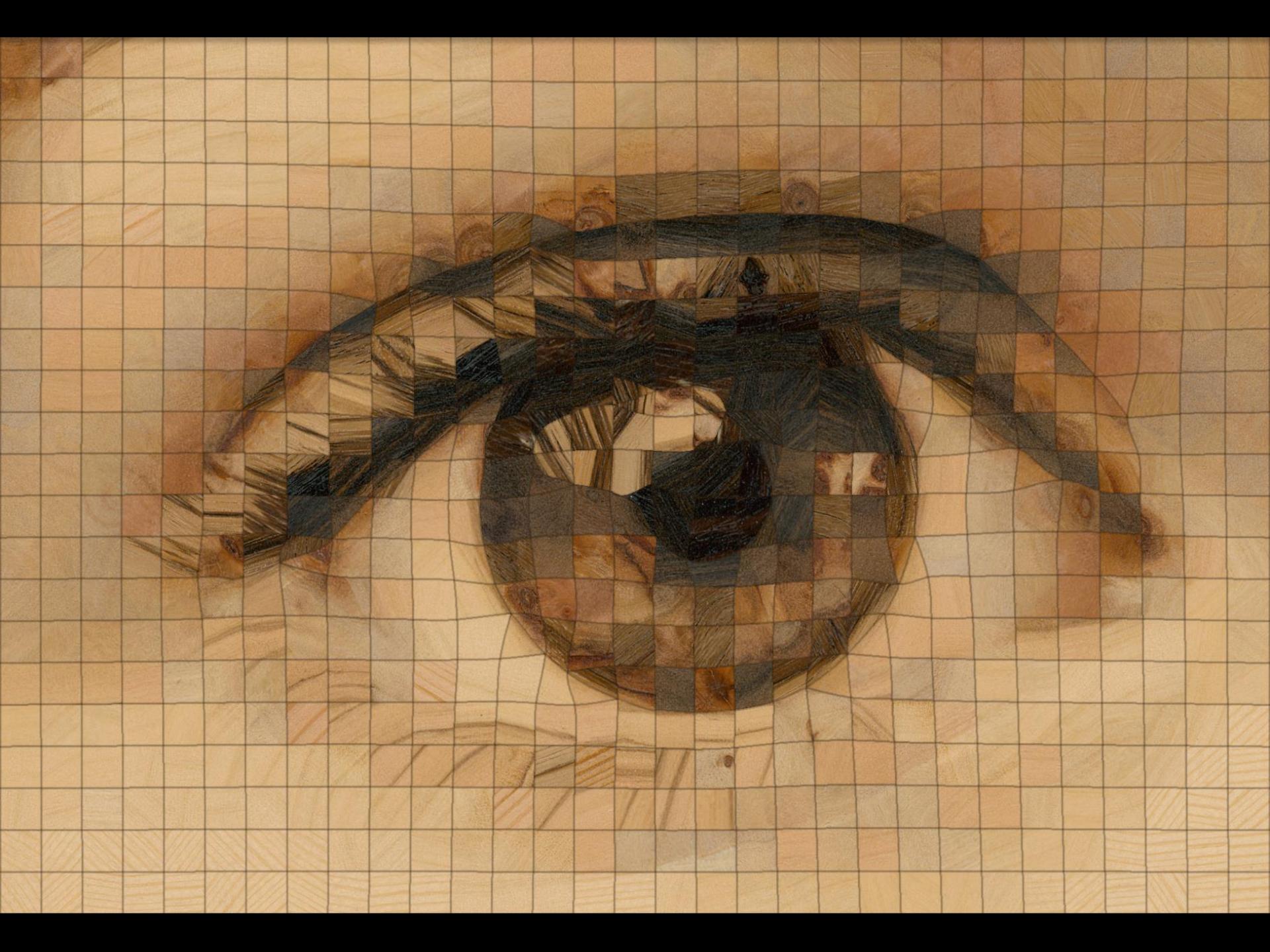
[Velinov et al., Eurographics 2018]

Computergrafik für Fabrikation

- Pixel aus Holz



Iseringhausen et al., Computational Parquetry,
ACM Trans. Graph., 2020



In dieser Veranstaltung lernen Sie

- Konzepte und Grundlagen
 - Licht, Farbe und Wahrnehmung
 - Mathematik (Transformationen, Signalverarbeitung)
 - Modellierung (3D-Objekte, Animation, Physik)
 - Renderingtechniken (Raytracing, Rasterisierung)
 - Texturierung und Schattenwurf
- Fortgeschrittene Themen
 - Beschleunigungsstrukturen
 - Lokale und globale Beleuchtungsmodelle
- Software
 - Echtzeit-Grafikumgebung OpenGL mit Shadern
 - Blender
- Animation und Storytelling

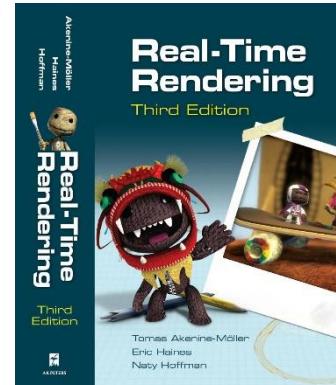
Buchempfehlung

- “Die Bibel”: Akenine-Möller et al., **Real-Time Rendering, 4th Ed.**, 2018. AK Peters/CRC Press, ISBN13: 9781138627000

<http://www.realtimerendering.com/>

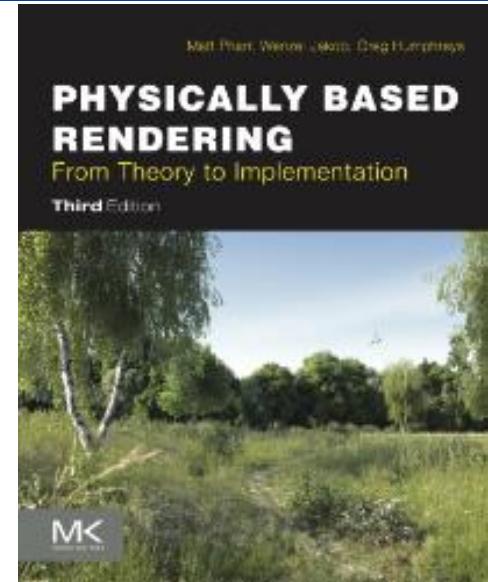


- PDFs der 3. Ausgabe (2008) spuken im Netz herum – ohne Garantie für Legalität.
Für diese Vorlesung aktuell genug.



Weitere Literatur

- Pharr, Jakob, Humphreys.
Physically Based Rendering (3rd Ed.)
Kompletter Text online verfügbar:
<https://www.pbrt.org/>



- Thomas Theis. Einstieg in C++.
Rheinwerk Verlag (2017)
- Torsten T. Will. C++: Das umfassende
Handbuch. Rheinwerk Verlag (2017).

Computergrafik

Grundlagen:
Licht und die menschliche Wahrnehmung

Die menschliche Wahrnehmung

- **Sehen**

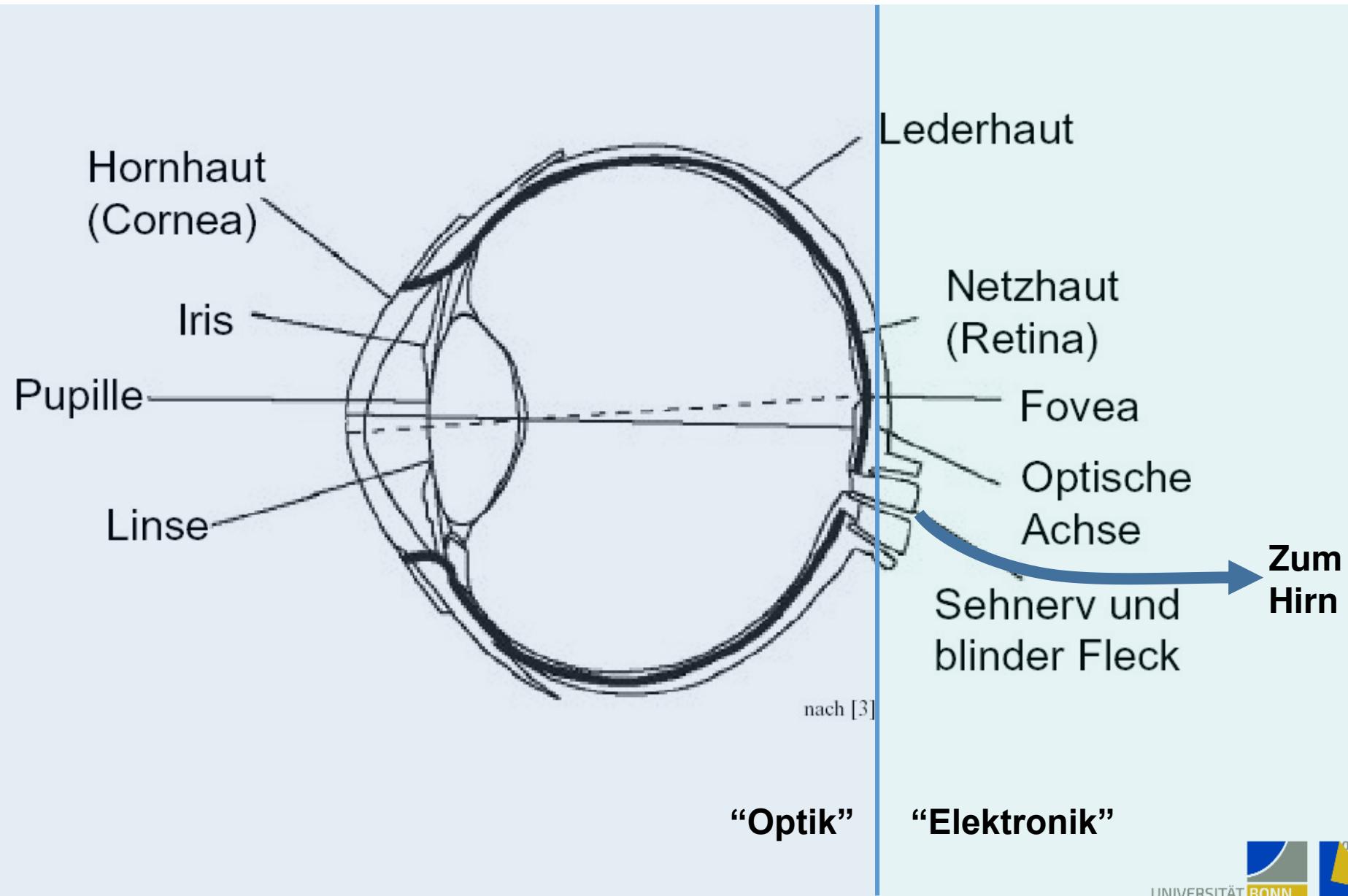
Hier konzentrieren wir uns auf den dominanten Sinn, das Sehen.

- Gehör

- Geruchssinn und Schmecken

- Tastsinn / Haptik

Das menschliche Auge



Optik = Lichtkunde

Strahlenmodell

(Geometrische Optik)

Strahl; Lichtenergie, die sich auf wohldefiniertem Pfad ausbreitet (im Vakuum: entlang einer Geraden)



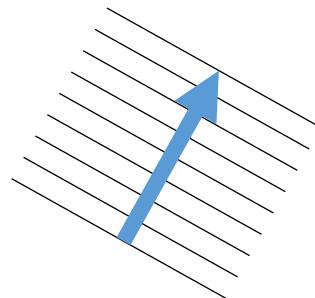
Modell gilt für inkohärente, makroskopische Lichtausbreitung (>99% der Computergrafik)

Ray tracing = Strahlen verfolgen

Wellenmodell

(Physikalische Optik)

Welle; räumlich und zeitlich veränderliches elektromagnetisches Feld

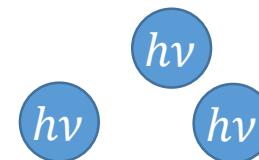


Gilt insbes. für Streuung/Beugung an kleinen Strukturen, kohärentes Licht (Interferenz)

Teilchenmodell

(Quantenoptik)

Photon; Teilchen mit Energie $E = h\nu = \frac{hc}{\lambda}$; fliegt mit Lichtgeschwindigkeit; keine Ruhemasse



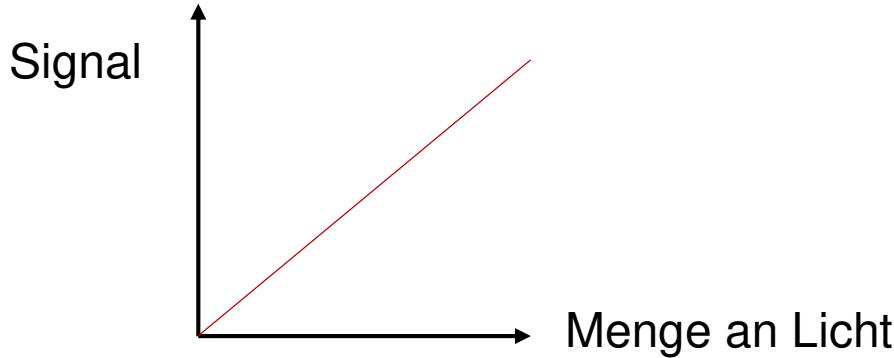
Gilt auch für Wechselwirkung mit Elektronen sowie für sehr geringe Intensitäten

Licht

- ...ist eine Form der Energie (Zeichen Q , Einheit 1 Joule)
- Strahlungsfluss $\Phi = \frac{dQ}{dt}$
(Energie pro Zeit; Einheit 1 Watt = 1 Joule pro Sekunde)
- Bildsensoren (wie auch das menschliche Auge) sollen Bestrahlungsstärke (“Irradianz”) E messen:
 $E = \frac{d\Phi}{dA} = \frac{d^2Q}{dt dA}$ (Energie pro Zeit pro Fläche).
- Sonneneinstrahlung (“Solarkonstante”), $E_{\text{Sonne}} \approx 1.4 \frac{kW}{m^2}$
- Zu alledem später mehr...

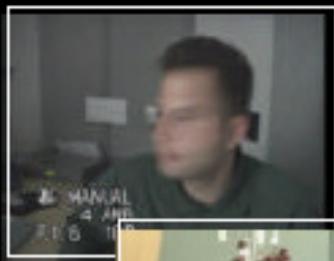
Strahlung messen und repräsentieren

- “Radiometrie”: Lehre von der Strahlungsmessung
- Ein idealer Detektor würde ein Signal (z.B. eine analoge Spannung oder einen digitalen Messwert) liefern, das proportional zur Strahlungsdichte ist.



- Herausforderung: die reale Welt hat einen immensen Dynamikumfang (Verhältnis von hellstem zu dunkelstem Messwert).

Dynamic Range in the Real World



1



1500



25,000



400,000



2,000,000,000

The real world is
high dynamic range.

Save the Date

Freitag 26.4.2024, 12:15 (CGIntro-Vorlesungsslot), HS2

Prof. Paul Debevec (Netflix/Eyeline/USC):

From Virtual Cinematography to Virtual Production

(Uni Bonn Visual Computing Incubator /
InVirtuo 4.0 Distinguished Lecture)



8-Bit Intensitätswerte

- Um diesem hohen Dynamikumfang Rechnung zu tragen, müssten Helligkeitswerte eigentlich als Fließkommazahlen gespeichert und verarbeitet werden.
 - Technische Einschränkungen erforderten lange Zeit die Repräsentation von Helligkeiten mit 8 Bit Wortbreite: Werte 0, 1, 2, ..., 254, 255
 - Auch heute noch arbeiten die meisten Anzeigegeräte und –treiber mit 8 Bit pro Farbkanal.
-
- Hinzu kommen historische Altlasten: z.B. Nichtlinearität von Bildröhren
 - Ungefähr quadratische Response: doppelte Spannung erzeugt etwa viermal so viel Licht.

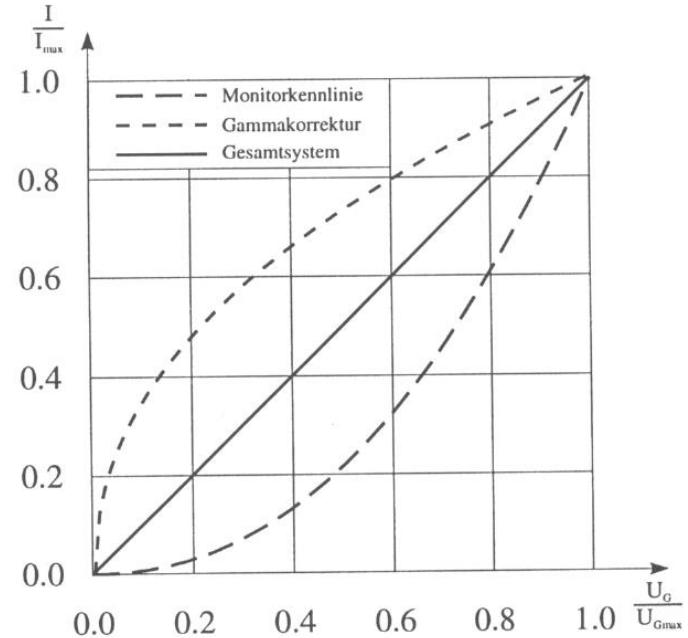
“Gamma-Korrektur”

- Annahme: Wertebereich [0, 1.0]
- Potenzmodell für Ansprache des Anzeigegeräts:

$$I_{\text{out}} = I_{\text{in}}^{\gamma}$$

z.B. typischerweise mit $\gamma = 2.2$ (Windows)

- Anzeigewerte müssen vorher korrigiert werden, um dies zu kompensieren:

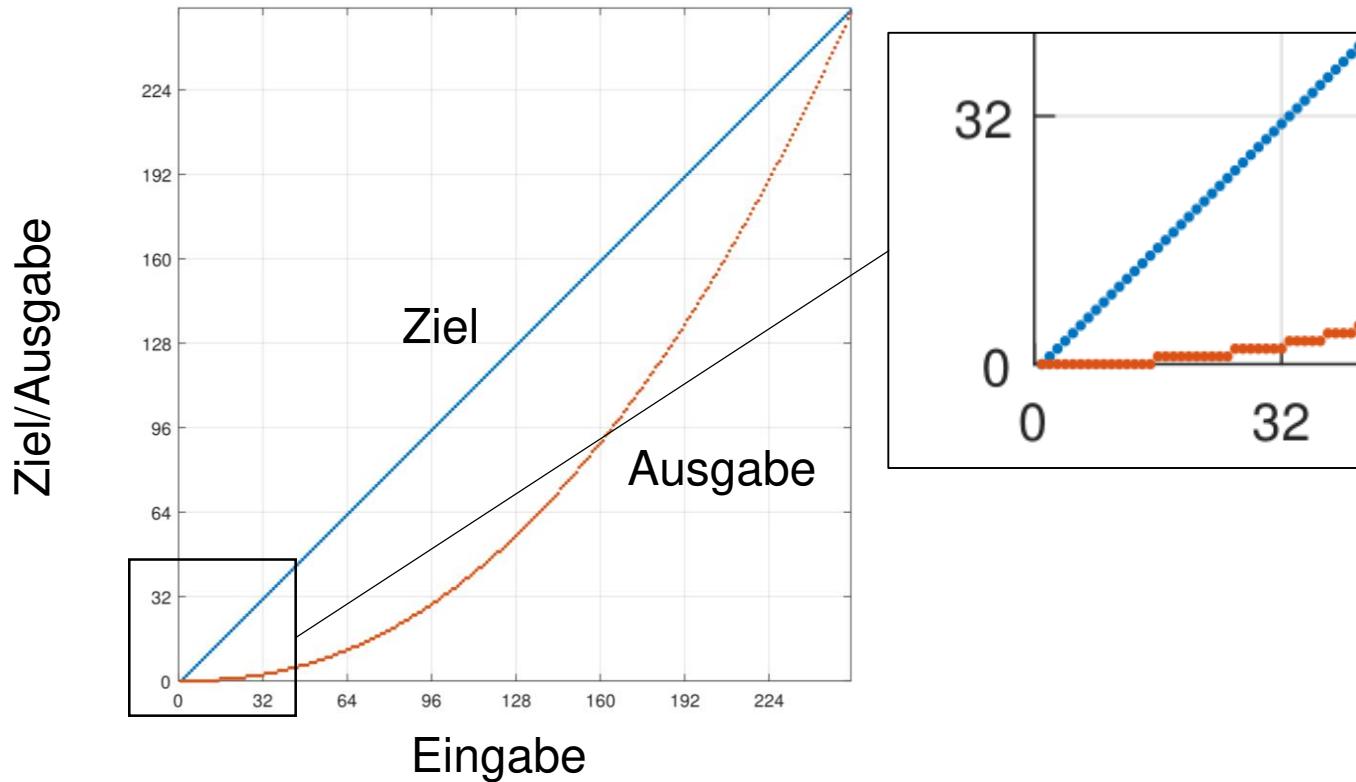


Vorsicht beim Arbeiten mit 8-Bit-Bilddaten!

Problem 1: Auflösungsverlust

- Eingabe (Zielwert): $I_{\text{Ziel}} \in \{0, \dots, 255\}$

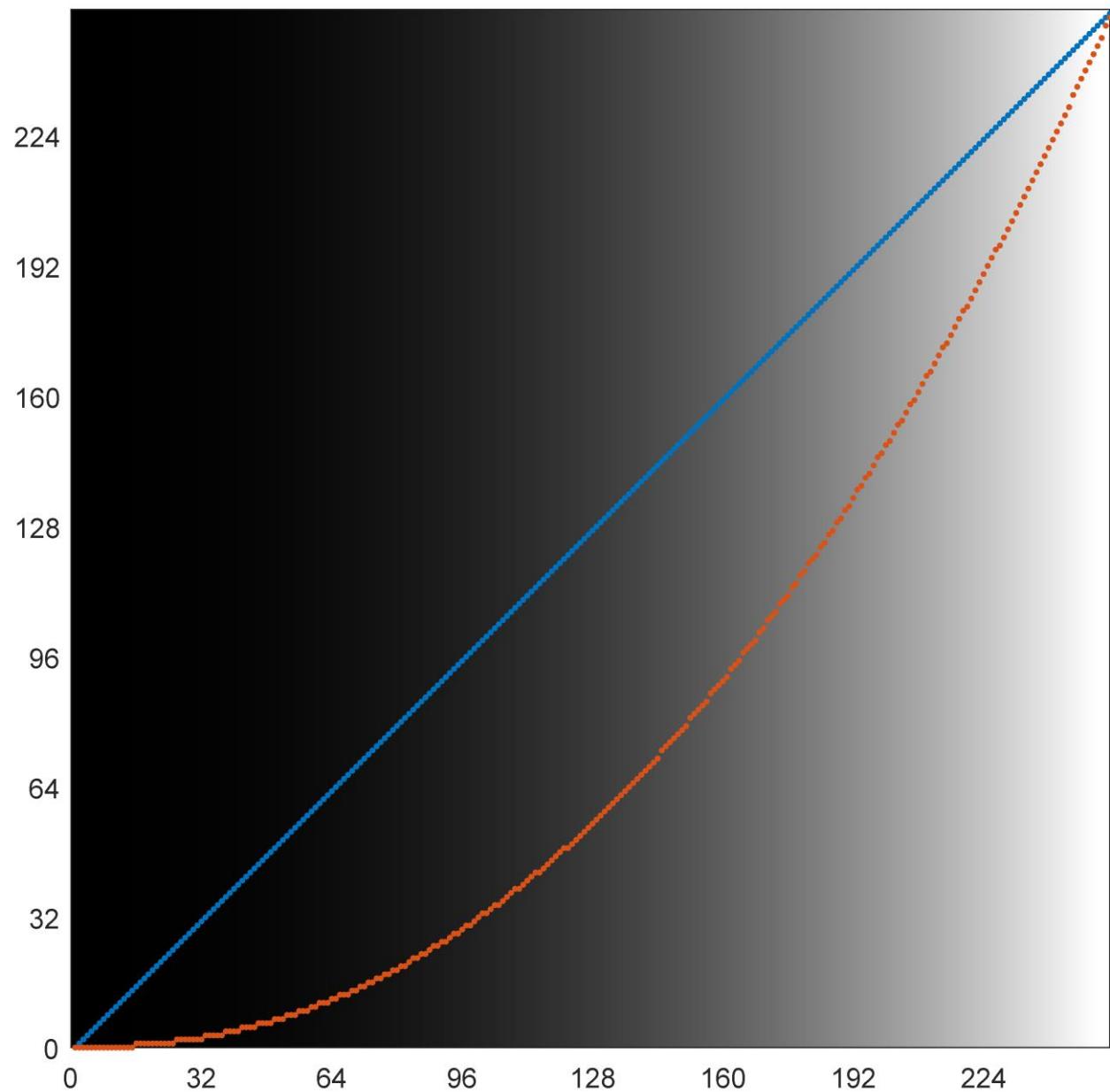
- Ausgabewert: $I_{\text{Ausgabe}} = \text{round} \left(255 \cdot \left(\left(\frac{I_{\text{Ziel}}}{255} \right)^{2.2} \right) \right)$



Vorsicht beim Arbeiten mit 8-Bit-Bilddaten!

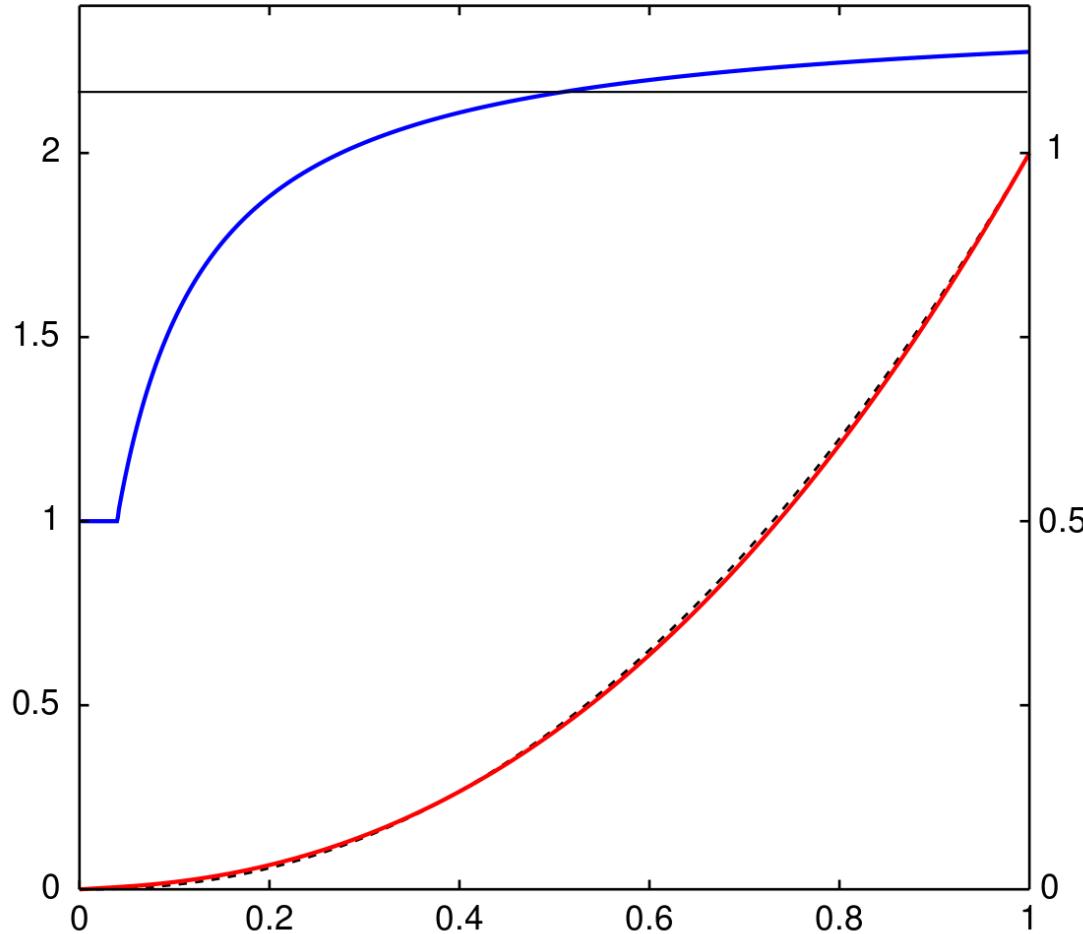
Problem 2:
Banding

(evtl. in der
Projektion nicht
sichtbar)



sRGB Gamma-Kurve

[Wikipedia]



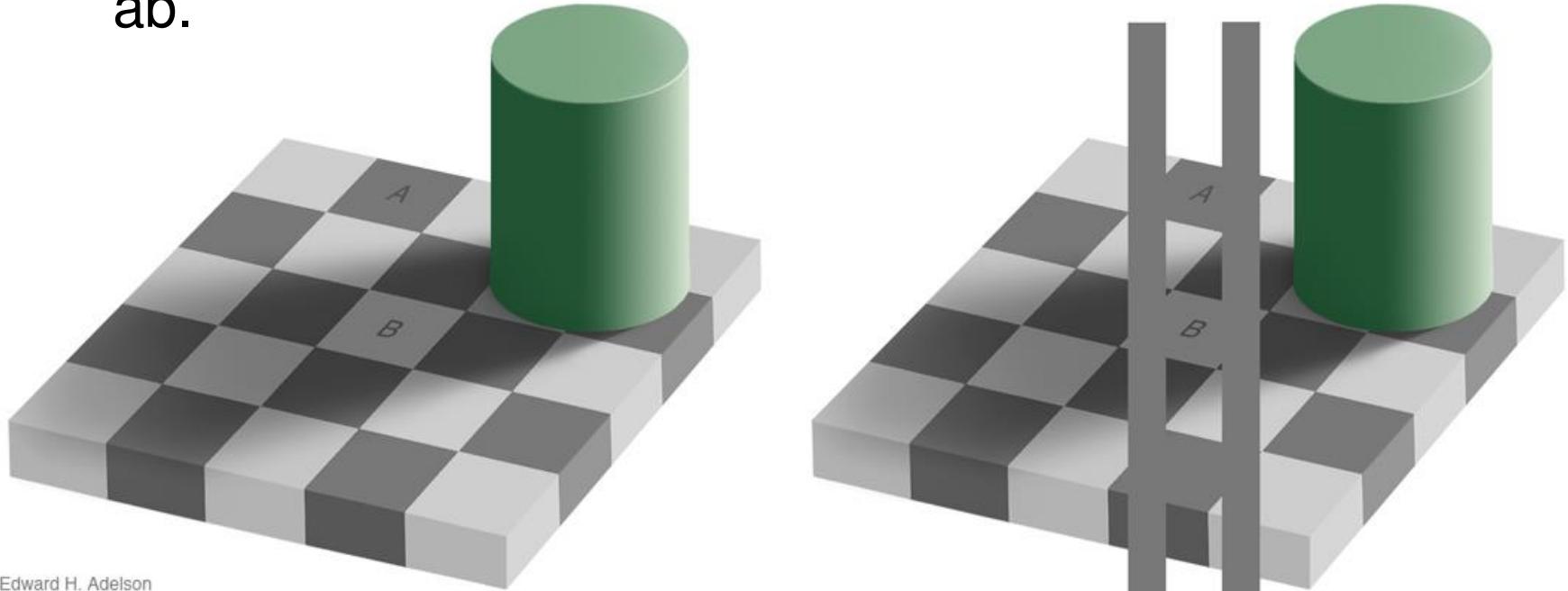
The overall sRGB gamma is approximately 2.2, consisting of a linear (gamma 1.0) section near black, and a non-linear section elsewhere involving a 2.4 exponent and a gamma (slope of log output versus log input) changing from 1.0 through about 2.3.

Photometrie

- Das menschliche Sehen ist hochgradig nichtlinear, wie auch die meisten Sensoren nicht perfekt linear sind. Zudem ist jedes Auge verschieden.
- Die “Photometrie” (Lehre von der Messung des Lichts) versucht die Ansprache unseres Auges durch einen “Standardbeobachter” einheitlich zu verstehen und näherungsweise zu beschreiben. Einheiten sind “Lichtstrom” (gemessen in $1 \text{ lm} = 1 \text{ cd} \cdot \text{sr}$), Leuchtdichten (in $\frac{\text{cd}}{\text{m}^2}$), usw.
- Auch hierzu (vielleicht) später mehr...

Semantisches Sehen / Konstanz der “Helligkeit”

- Unser Erfahrungswissen und unsere Erwartungen tragen dazu bei, was wir sehen.
- Beim Betrachten dieses Bildes erwartet unsere Wahrnehmung, dass Objekte im Schatten dunkler erscheinen, und gleicht diese Helligkeit entsprechend ab.



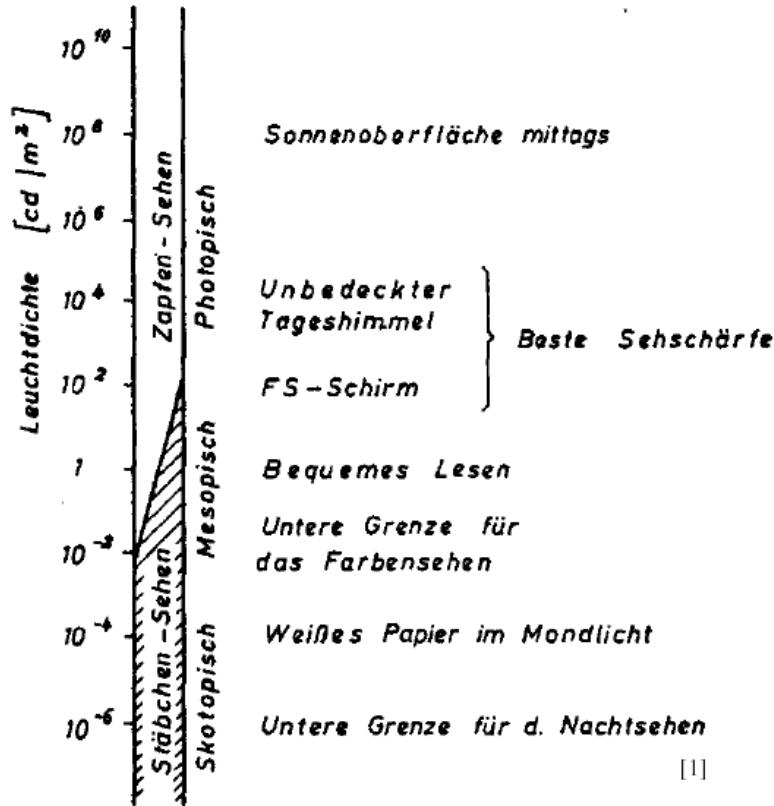
Edward H. Adelson

Reflektanz A != Reflektanz B;

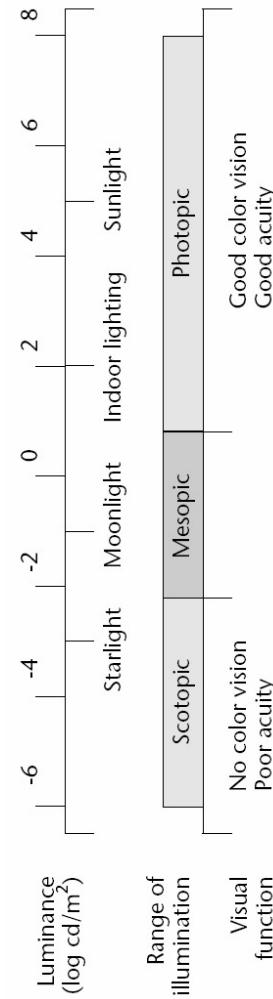
Luminanz A = Luminanz B

Dynamik

- Visuelle Reize in der realen Welt haben einen immensen Dynamikumfang von kompetter Dunkelheit bis zu direktem Sonnenlicht.



[1]

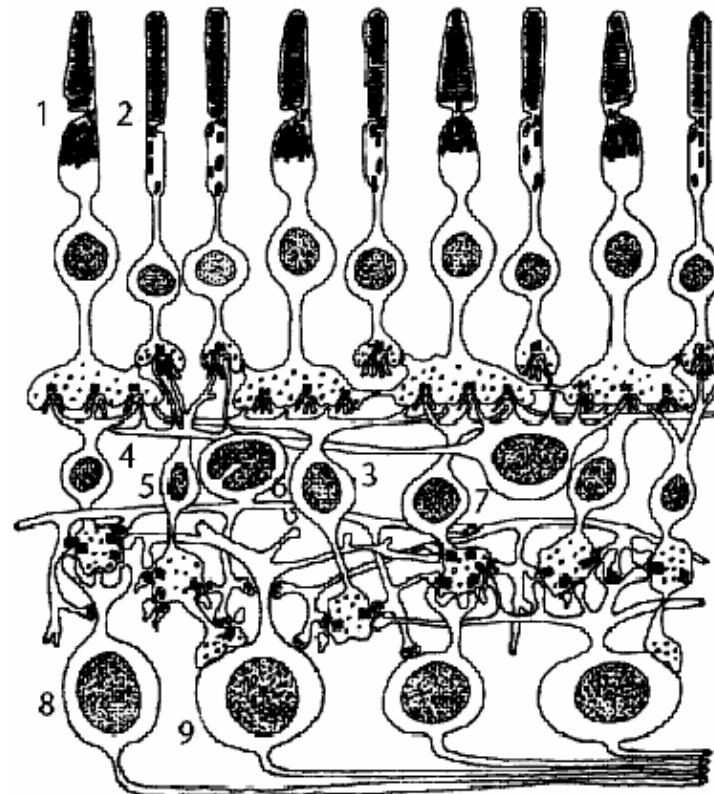


Sehzellen

- Unsere Augen haben verschiedene Detektorzellen, die auf schwaches bzw. starkes Licht spezialisiert sind.

Aufbau der Netzhaut:

1. Zapfen
2. Stäbchen
- 3.-5. Bipolarzellen
- 6.-7. Amakrinzellen
- 8.-9. Ganglienzellen



Rezeptive Felder

- Ca. 130 Mio. Photorezeptoren werden mit ca. 1 Mio. Ganglien verbunden
- Sie bilden zirkuläre rezeptive Felder unterschiedlicher Größe
- 2 Typen
 - ON: Erregung im Zentrum
Hemmung in Peripherie
 - OFF: Erregung in Peripherie
Hemmung im Zentrum
- Reagieren auf Intensitäts- und Farbunterschiede
- Erkennen von Orientierung und Textur

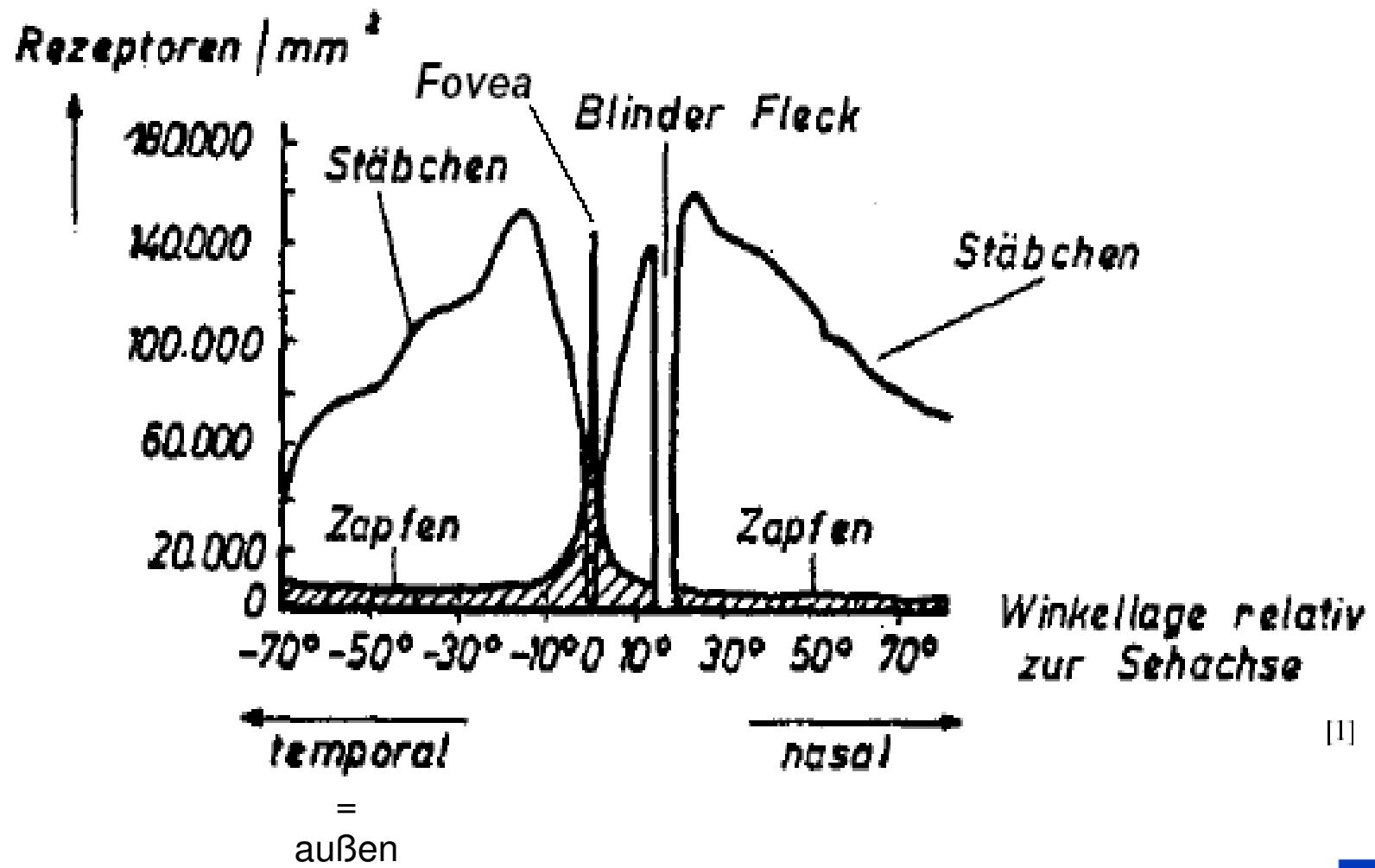


Zum visuellen Kortex

Rezeptoren in der Retina

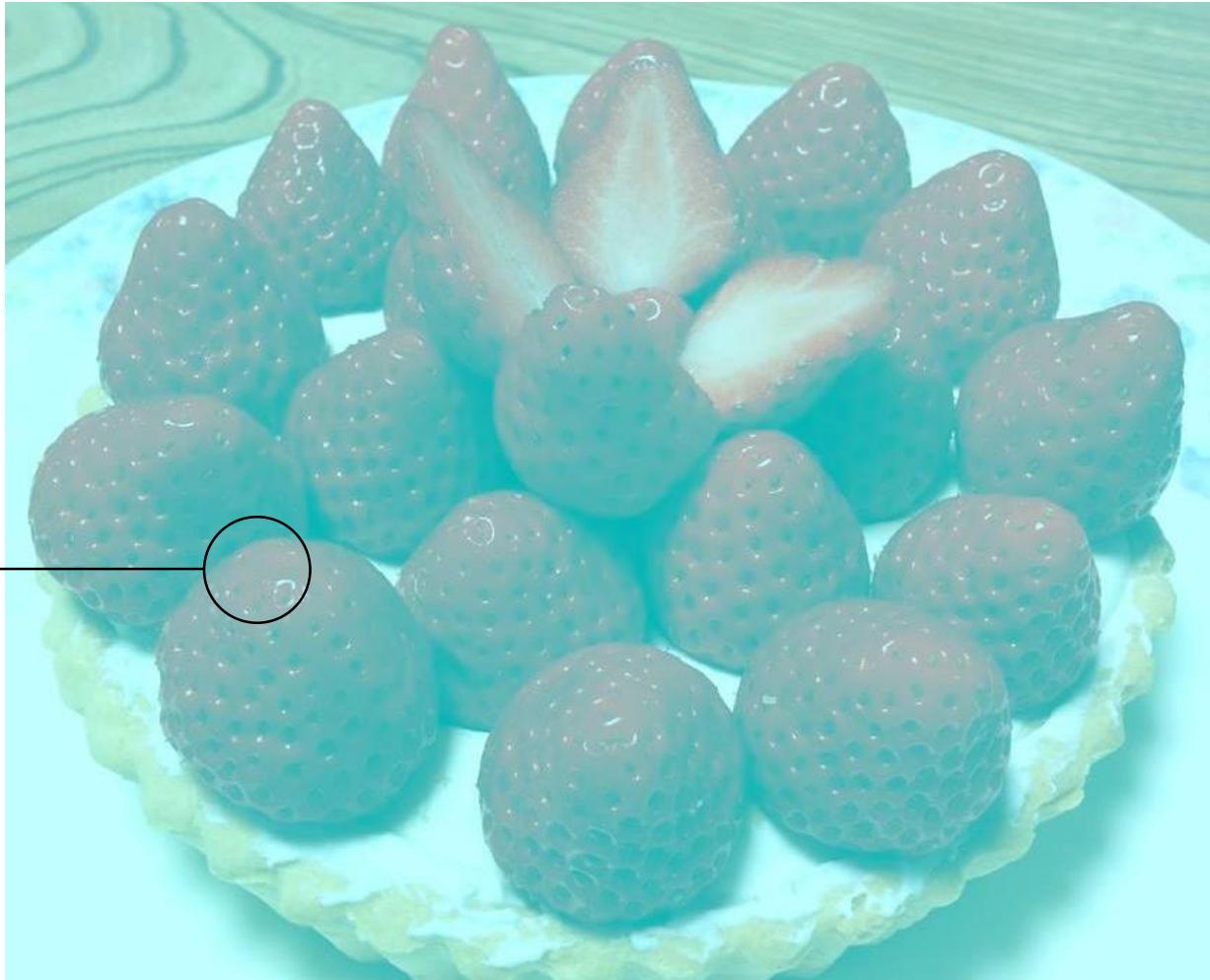
- Stäbchen (rods):
 - 120 Mio. hauptsächlich am Rand (max. bei 20°)
 - Grauwerte, geringe Helligkeit
 - am empfindlichsten bei 500 nm (grün)
 - Zapfen (cones):
 - 7 Mio. konzentriert in der Mitte
 - Farben, große Helligkeit
 - Scharfsehen
 - 1 mm² Netzhaut ≈ 1 Grad Blickfeld
≈ 2,5 cm² auf Oberfläche in 1 m Entfernung
- peripheres skotopisches Sehen
- foveales photopisches Sehen

Dichte der Zapfen und Stäbchen entlang des horizontalen Meridians der Retina



Farbe

- Farbe ist nicht physikalisch definiert, sondern das Ergebnis unserer Wahrnehmung.



Chromatische Adaptation

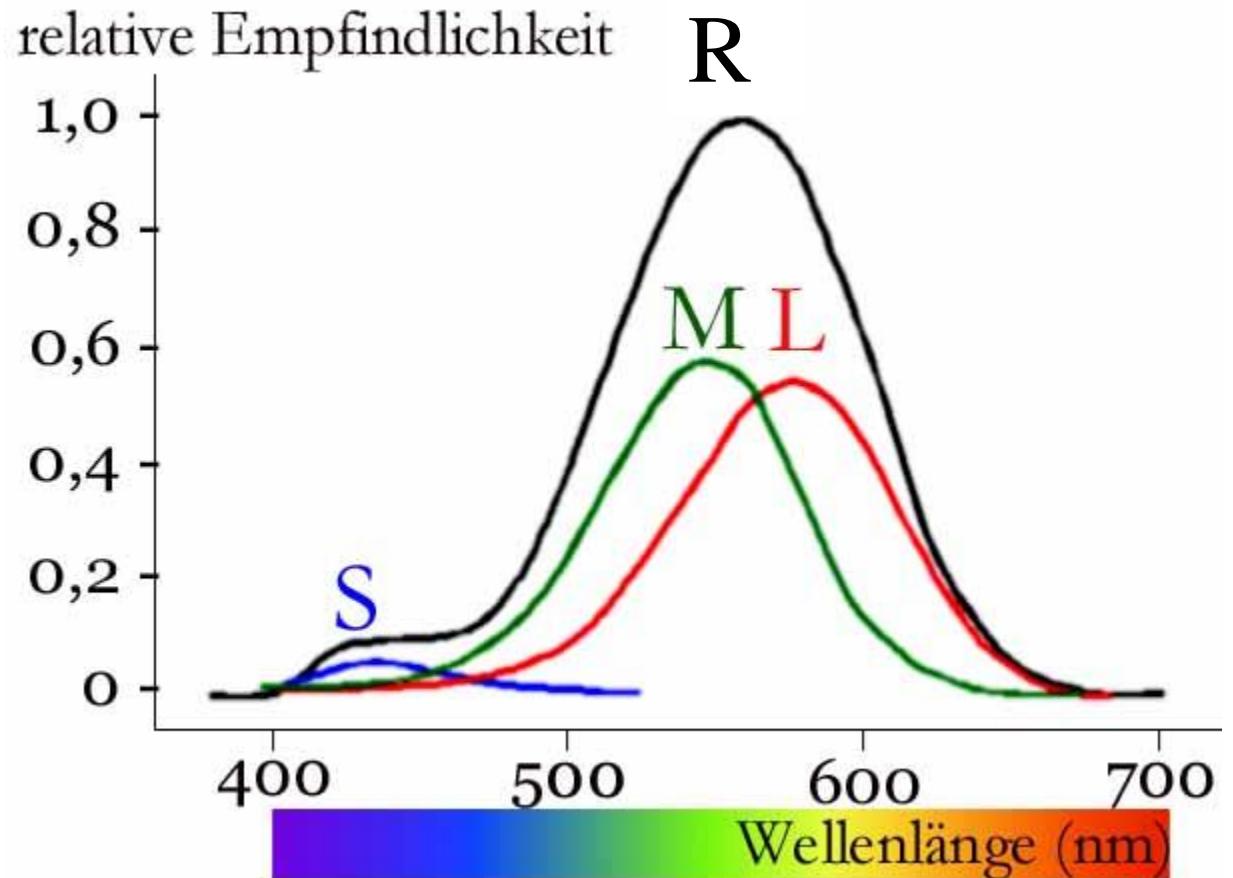


Chromatische Adaptation



Empfindlichkeit von Zapfen und Stäbchen

- 3 Sorten Zapfen: S, M und L
("short", "medium", "long" wavelength)



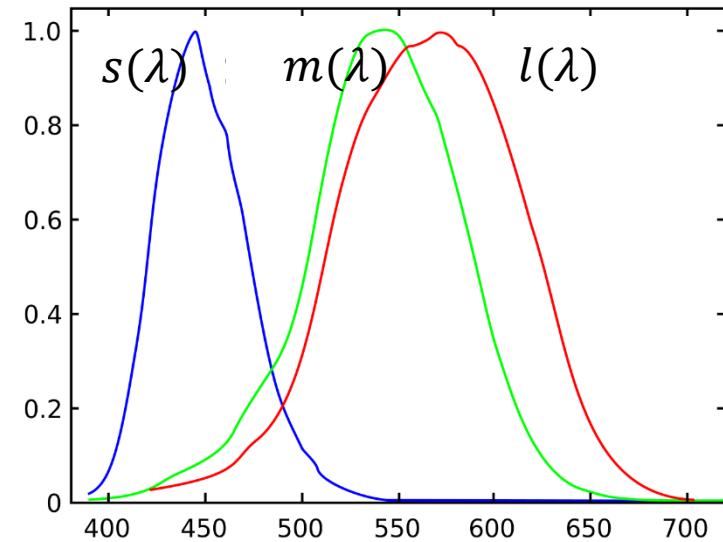
Tristimulus-Farträume

- Tristimulus = Farbe als Kombination dreier Primärfarben.
- Koeffizienten sind “Gewichte” der jeweiligen Grundfarbe (als Basisfunktion)
- Gegeben: Spektrale Intensitätsverteilung $I(\lambda)$, Primärfarbenspektren $\{l, m, s\}(\lambda)$
- LMS-Farbkoordinaten:

$$L = \int_{400}^{\infty} l(\lambda)I(\lambda)d\lambda$$

$$M = \int_{400}^{\infty} m(\lambda)I(\lambda)d\lambda$$

$$S = \int_{400}^{\infty} s(\lambda)I(\lambda)d\lambda$$



Jeder Mensch sieht anders

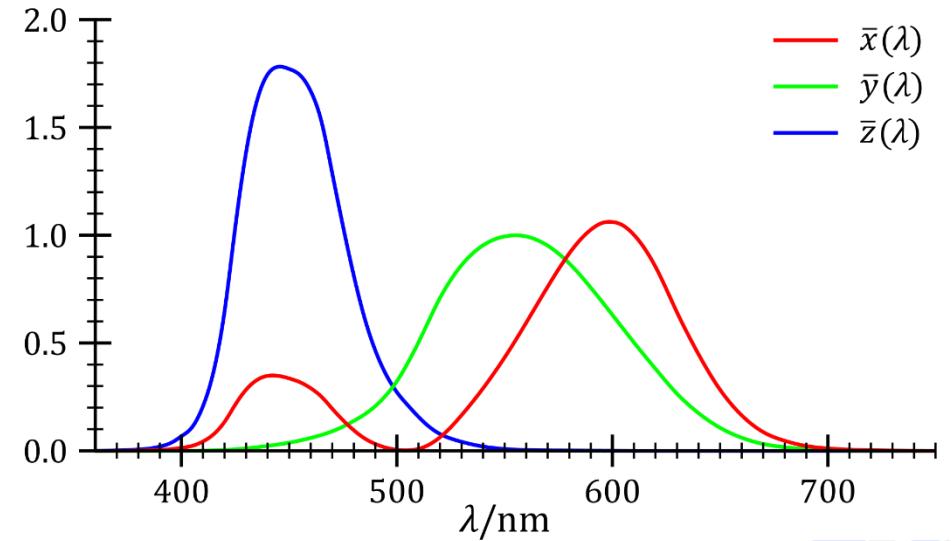
- ...darum hat die Internationale Beleuchtungskommission (Commission Internationale de l'Éclairage, CIE) im Jahr 1931 Standardbeobachter definiert.
- Der gängigste Farbraum ist CIE XYZ mit Tristimuluskurven (“color matching functions”) $\{\bar{x}, \bar{y}, \bar{z}\}(\lambda)$

- XYZ-Farbkoordinaten:

$$X = \int \bar{x}(\lambda) I(\lambda) d\lambda$$

$$Y = \int \bar{y}(\lambda) I(\lambda) d\lambda$$

$$Z = \int \bar{z}(\lambda) I(\lambda) d\lambda$$



Wechsel zwischen Farträumen

- So definierte *lineare* Farbräume lassen sich in der Regel einfach ineinander überführen.
- Zum Beispiel:

CIE RGB in CIE XYZ (von der CIE exakt definierte Werte):

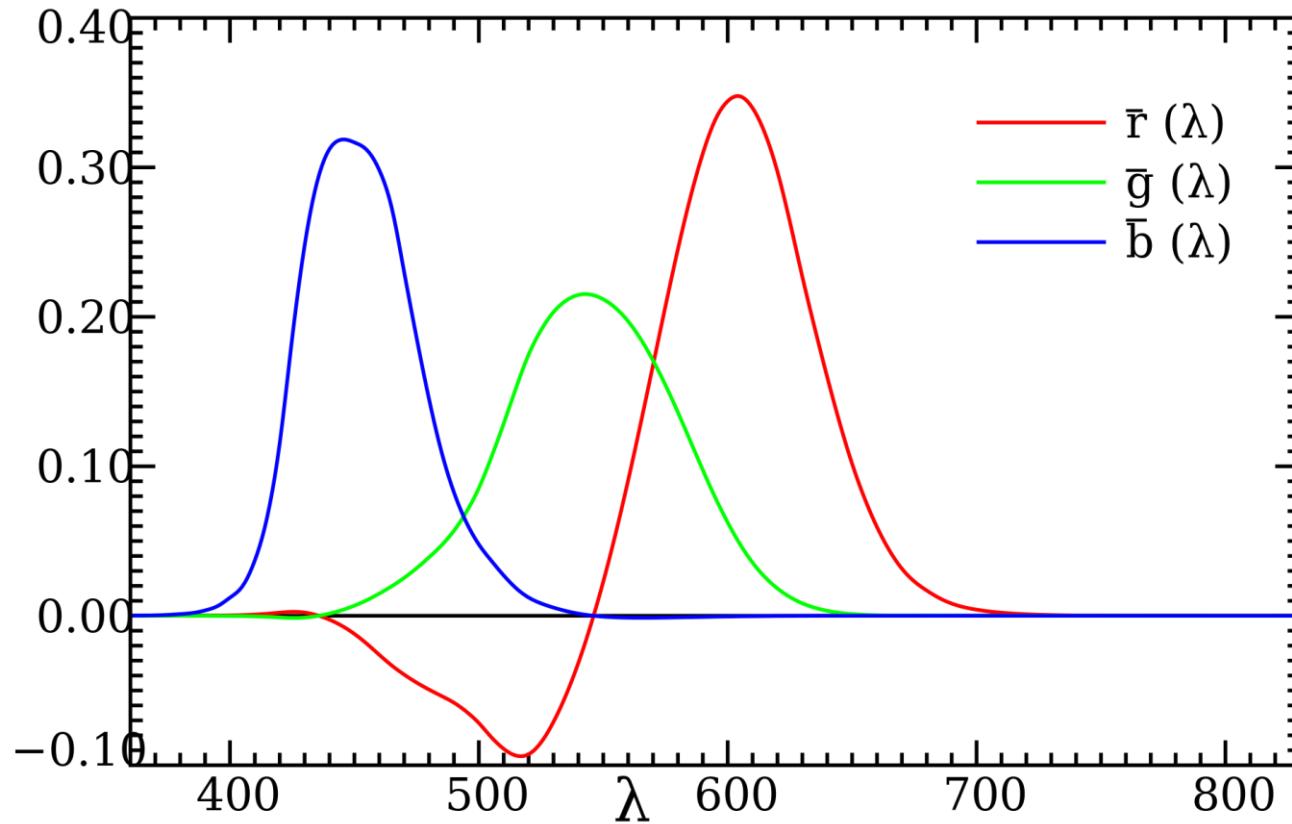
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{b_{21}} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \frac{1}{0.176\ 97} \begin{bmatrix} 0.490\ 00 & 0.310\ 00 & 0.200\ 00 \\ 0.176\ 97 & 0.812\ 40 & 0.010\ 63 \\ 0.000\ 00 & 0.010\ 00 & 0.990\ 00 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

und umgekehrt (ungefähre Werte):

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.418\ 47 & -0.158\ 66 & -0.082\ 835 \\ -0.091\ 169 & 0.252\ 43 & 0.015\ 708 \\ 0.000\ 920\ 90 & -0.002\ 549\ 8 & 0.178\ 60 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Farbkoordinaten können negativ sein!

- In der Regel kein Grund zur Sorge



Unser erstes Bild (Rastergrafik)

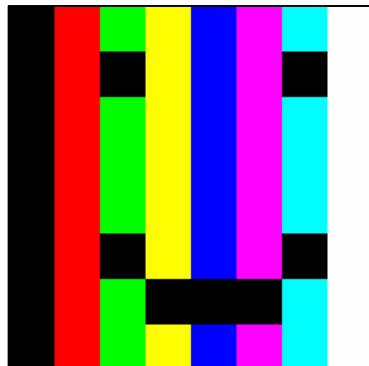
beispiel.ppm

P3 # "Magic Number" for Plain PPM

8 8 # Width Height

15 # Maxvalue

0	0	0	15	0	0	0	15	0	15	15	0	0	15	15	0	15	15	15		
0	0	0	15	0	0	0	0	0	15	15	0	0	15	15	0	0	0	15	15	15
0	0	0	15	0	0	0	15	0	15	15	0	0	15	15	0	15	15	15	15	
0	0	0	15	0	0	0	15	0	15	15	0	0	15	15	0	15	15	15	15	
0	0	0	15	0	0	0	15	0	15	15	0	0	15	15	0	15	15	15	15	
0	0	0	15	0	0	0	15	0	15	15	0	0	15	15	0	15	15	15	15	
0	0	0	15	0	0	0	0	0	15	15	0	0	15	15	0	0	0	15	15	15
0	0	0	15	0	0	0	15	0	0	0	0	0	0	0	0	0	15	15	15	15
0	0	0	15	0	0	0	15	0	15	15	0	0	15	15	0	15	15	15	15	



Unser zweites Bild (Vektorgrafik)

beispiel2.svg

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 100 100">
  <path d="M30,76q6-14,13-26q6-12,14-23q8-12,13-17q3-4,6-6q1-1,5-2q8-1,12-
1q1,0,1,1q0,1-1,2q-13,11-27,33q-14,21-24,44q-4,9-5,11q-1,2-9,2q-5,0-6-1q-1-1-
5-6q-5-8-12-15q-3-4-3-6q0-2,4-5q3-2,6-2q3,0,8,3q5,4,10,14z" fill="green" />
</svg>
```

q: Quadratic spline to
(relative Koordinaten)

M: Move to
(absolute Koordinaten)

z: close path



Ausblick auf nächstes Mal

- Rechnen mit Punkten, Vektoren und Strahlen
- Konvexe/Affine Kombinationen
- Transformationen
- Homogene Koordinaten