

IVC Interactive Visual Computing: Computer Graphics

Leonie Dreschler-Fischer

Fachbereich Informatik der Fakultät für Mathematik, Informatik und
Naturwissenschaften

WS 2014/2015



Ansage

Prüfungsunterlagen: Das Skript

Online: Computer Vision, Computer Graphics (als pdf, mit Index und Anhang zur Prüfungsvorbereitung),

Übungsaufgabe: /home/tis_6

Erste Übung:

Klausur: 2 Termine

- ① Termin IVC: Mittwoch, 04.02.2015,
13:30-15:30 Uhr, ESA A
- ② Termin IVC: Freitag, 06.03.2015,
10:30-12:30 Uhr, Phil B

<http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Lehre/Lehre.html>

Themen der Vorlesung

- 1 Geometrische Grundlagen: Koordinaten und Transformationen ▶ Koordinaten
- 2 Geometrische Grundlagen: Abbildung des Raumes ▶ Projektionen
- 3 Photometrische Grundlagen ▶ Licht und Farbe
- 4 Bilder und Bildfunktionen ▶ Ikonische Bildverarbeitung
- 5 Binarisieren ▶ Segmentierung
- 6 Geometrische Szenenrekonstruktion ▶ Szenenrekonstruktion
- 7 Mehrfache Objektansichten ▶ Mehrfache Objektansichten
- 8 Übungen: Werkzeuge, Handbücher und Beispiele ▶ Übungen
- 9 Heute:
 - ▶ Koordinatentransformation ▶ Transformationen
 - ▶ Übungen ▶ Übungen

Einführung und Grundbegriffe
Anwendungen und Beispiele
Organisatorische

Teil I

Einführung

Entwicklung



1 Einführung und Grundbegriffe

- Entwicklung
- Computergrafik vs. Bildverarbeitung
- Methoden und Modelle

2 Anwendungen und Beispiele

3 Organisatorische

Einführung

BEGINN 1963: Ivan Sutherland legt die software-theoretischen Grundlagen, erste Algorithmen für grafische Aufgaben. Computergrafik war zunächst nur auf teurer Spezialhardware möglich, die *Vektorgrafik* unterstützte.

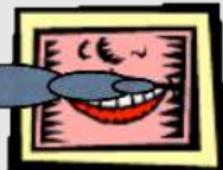
SEITHER: Rasante Entwicklung, Markt mit mehreren Milliarden Dollar Umsatz für Hardware und Software.

Es gibt leistungsfähige, preiswerte Grafikkarten für PCs, heute überwiegend *Rastergrafik*.

GRUND: Aussagekraft von Bildern: (Diagrammen, Karten, Konstruktionszeichnungen; grafische Interaktionsmöglichkeiten, Benutzungsoberflächen...)

Zur Aussagekraft von Bildern

**Ein Bild sagt mehr
als tausend Worte ...**



Die scherzhafte Rechnung von Fellner:

- ▶ Visueller Kanal des Menschen:
Datenrate: 30–40 Mio. Bit pro Sekunde.
- ▶ Das entspricht einem Text von ca. 1 Million Worten pro Sekunde, mit folgenden Schätzwerten:
7 Bit pro Zeichen, 5 Zeichen pro Wort.
- ▶ Leserate für gedruckten Text: 600–1200 Worte pro Minute: um mehr als einen Faktor 1000 kleiner.

Definition (Computergrafik)

Die Computergrafik ist die Wissenschaft von den Methoden und Techniken zur Konvertierung von Daten in und aus grafischer Darstellung mit Hilfe von elektronischen Rechenanlagen. (ISO)

Sie hat die drei Teilgebiete:

- ▶ Grafische Datenverarbeitung,
- ▶ Bildverarbeitung,
- ▶ Bildverstehen / Mustererkennung.

Grafische Datenverarbeitung

Definition (**Grafische Datenverarbeitung**)

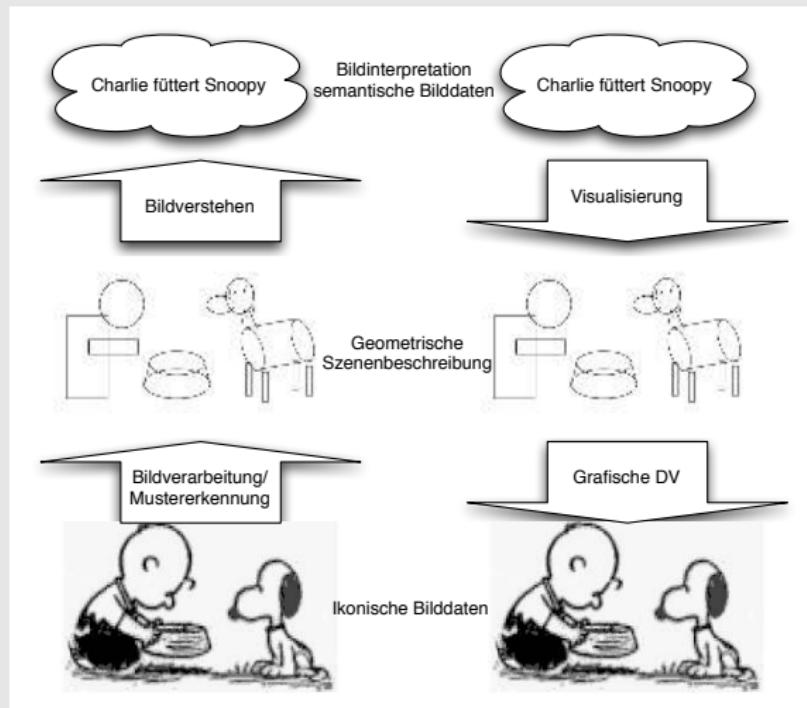
Die grafische Datenverarbeitung umfaßt Verfahren zur Eingabe und Manipulation von Bildbeschreibungen sowie deren Umwandlung in grafische Darstellungen.

Probleme:

Grafische Datenverarbeitung: Komplexität des Problems

Bildverstehen: Mehrdeutigkeit von Bildern

Prozesse der grafischen DV



Ebenen der Bildbeschreibung

Ikonische Ebene: Das ist die Ebene der Bilder als physikalisches Signal, die Beschreibung von Helligkeiten und Farben als Funktion des Ortes im Bild.

Die geometrische Ebene: Die geometrische Szenenbeschreibung beschreibt die Anordnung von den im Bild abgebildeten Szenenkomponenten und die Form der Szenenkomponenten.

Die Bildinterpretation: Auf der semantischen Ebene wird die Bedeutung einer im Bild dargestellten Szene beschrieben: Welche Objekte gibt es? Was tun sie? Welche Ziele verfolgen sie?

Verarbeitungsprozesse

Bildverarbeitung: (ikonische) Bildverarbeitung im engeren Sinne bezeichnet alle Verarbeitungsverfahren, bei denen Bilder transformiert werden (Bildrestauration, Registrierung, Falschfarbendarstellung usw.) oder elementare Merkmale der Bilder abgeleitet werden.

Niedere Bilddeutung: Die niedere Bilddeutung leitet aus Bildern eine geometrische Bildbeschreibung ab. Hierzu müssen die Konturen von Objekten anhand der Bildstrukturen ermittelt und mit Objektmodellen verglichen werden.

Verarbeitungsprozesse 2

Grafische Datenverarbeitung: Die Gegenrichtung zu den bisher genannten Verfahren — die Aufgabe der grafischen DV ist die Synthese von Bildern, ausgehend von einer geometrischen Szenenbeschreibung. Die beschriebenen Komponenten der Szene werden mittels Projektion und Renderingverfahren abgebildet.

Visualisierung und Bildverstehen: Die Verarbeitungsprozesse, die zwischen der geometrischen Beschreibung und der Interpretation vermitteln, gehören nicht mehr zum Bereich der Computergrafik, wie er seinerzeit von der ISO festgelegt wurde. Es sind aber wichtige Forschungsgebiete der Künstlichen Intelligenz und der Bildverarbeitung im weiteren Sinne.

Verarbeitungsrichtung

- ▶ Die Prozesse der graphischen Datenverarbeitung sind also im Wesentlichen „**top-down**“ Prozesse — von der Beschreibung zum Bild,
- ▶ während die Prozesse des Bildverständens „**bottom up**“ Prozesse sind — vom Bild zur Bildbeschreibung.
- ▶ Beide Teilgebiete der Computergrafik nutzen als gemeinsames methodisches Repertoire dieselben Repräsentationen auf den verschiedenen Abstraktionsebenen.

Probleme

Bildverstehen und Computergrafik haben mit sehr unterschiedlichen Schwierigkeiten zu kämpfen:

- ▶ Das größte Problem beim **Bildverstehen** ist die mangelnde Eindeutigkeit: Jedes Bild kann das Abbild von unendlich vielen verschiedenen Szenen sein.
- ▶ Bei der **grafischen DV** dagegen ist in der Regel die Lösung eindeutig, aber die Aufgabe besteht darin, Algorithmen zu entwickeln, die diese eindeutigen Lösungen in vertretbarer Zeit finden.

Einführung und Grundbegriffe
Anwendungen und Beispiele
Organisatorische

Entwicklung
Computergrafik vs. Bildverarbeitung
Methoden und Modelle

Teil I

Einführung

Methoden und Modelle



1 Einführung und Grundbegriffe

- Entwicklung
- Computergrafik vs. Bildverarbeitung
- Methoden und Modelle

2 Anwendungen und Beispiele

3 Organisatorische

Bildentstehung

Die grafische Datenverarbeitung formalisiert den Vorgang der **Bildentstehung (image formation)** über **Modelle**.

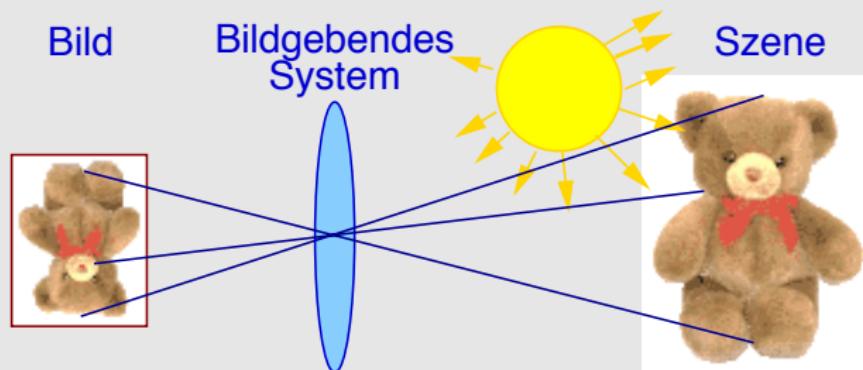


Bild $\xrightarrow{\text{Bildverstehen}}$ Szenenbeschreibung

Computergrafik

Bild $\xleftarrow{\text{Bildsynthese}}$ Szenenbeschreibung

Modelle und Abstraktionen:

Bildfunktion Die grundlegende Abstraktion eines Bildes.

Geometrische Modelle dienen zur Modellierung der Form und Anordnung von Objekten, sowie der Abbildung von dreidimensionalen Szenen auf zweidimensionale Bildräume.

Radiometrische Modelle beschreiben, wie die Anordnung von Licht, Objektoberflächen und Bildwandlern die Helligkeit und Farbe im Bild beeinflußt.

Farbmodelle beschreiben die Beziehung zwischen den Farben im Bild und spektralen Messungen.

Digitalisierungsmodelle beschreiben den Prozeß der Diskretisierung von analogen Bildfunktionen.

Modelle und Abstraktionen:

- ▶ **Bildverstehen:** Die geometrischen und radiometrischen Modelle versuchen den physikalischen Prozeß der Bildentstehung möglichst genau zu erklären.
- ▶ **Computergrafik:** Die geometrischen und radiometrischen Modelle sind Approximationen an die physikalischen Prozesse, die adäquate Algorithmen ermöglichen.
- ▶ **Virtuelle Welten:** Die Modelle in der Computergrafik können unmögliche Welten beschreiben, in denen andere physikalische Gesetze gelten als in unserem Universum.
- ▶ **Visualisierung** von konzeptuellen Welten (abstrakten oder unsichtbaren Phänomenen).

Anwendungen und Beispiele



1 Einführung und Grundbegriffe

2 Anwendungen und Beispiele

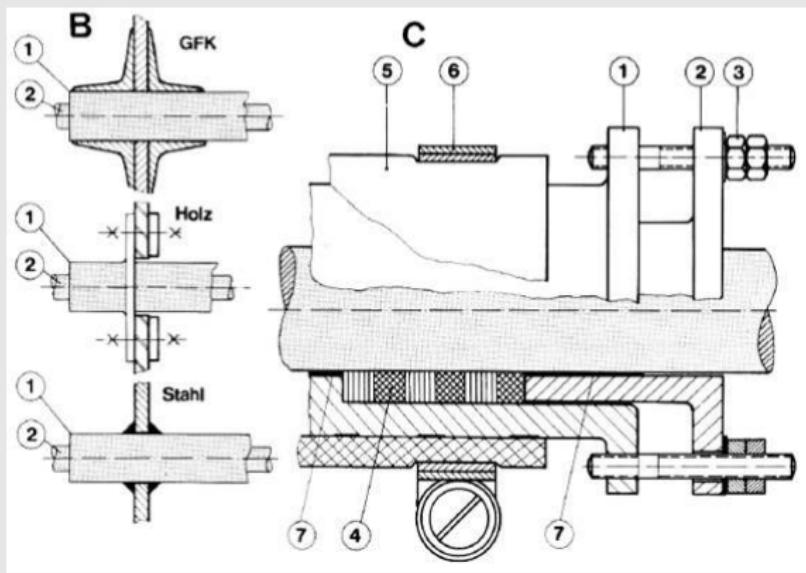
- Konstruktion und Fertigung
- Anwendungen in der Bildverarbeitung
- Echtzeitbilddarstellungen
- Visualisierung
- Fotorealismus

3 Organisatorische

Anwendungen

- ▶ Computergestützte Konstruktion und Fertigung (CAD/CAM usw.)
- ▶ Präsentationsgraphiken
- ▶ Bildverarbeitung, Spielfilm
- ▶ Photogrammetrie
- ▶ Kartographie, geographische Informationssysteme
- ▶ Interaktive Programmoberflächen
- ▶ Visualisierung für spezielle Anwendungen, Simulation
- ▶ Virtuelle Welten
- ▶ Kunst

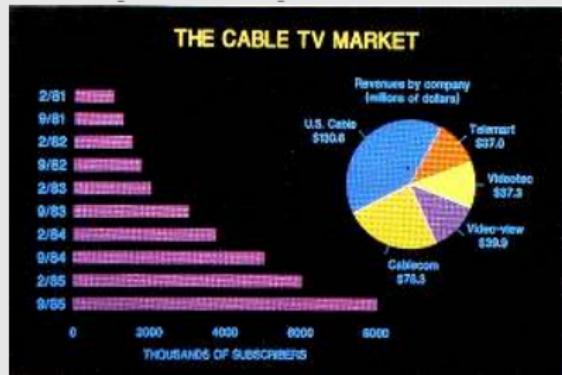
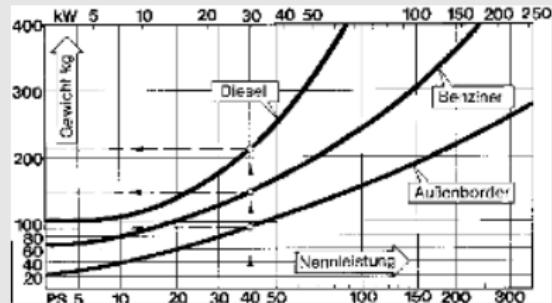
Anwendung: Technische Zeichnungen:



Vorteile des CAD/CAM:

- ▶ Arbeit beschränkt sich auf den kreativen Anteil, Routinearbeit übernimmt das System.
- ▶ Fehlerarme Entwürfe.
- ▶ Objektmodelle eignen sich für weitere Anwendungen.
- ▶ Simulation möglich, bevor die Teile in die Fertigung gehen.
- ▶ Rationalisierung.

Anwendung: Präsentationsgrafiken:



Präsentationsgrafiken:

Visualisierung von Statistiken, Tabellen und anderen Entscheidungsinformationen.

- ▶ Leichte Erkennbarkeit von Trends und Zusammenhängen.
- ▶ Reduktion von Information: *80% der Entscheidungen können mit 20% der Information getroffen werden.* [Fellner, 2004]
- ▶ Ausgabe auf Bildschirmen, Folien, Dias.

Anwendung: Bildverarbeitung:

- ▶ Bildrestaurierung
- ▶ Robotik,
- ▶ Fernerkundung,
- ▶ Qualitätskontrolle, Überwachung

Bildrestauration

Beseitigung von Bewegungsunschärfe



Echtzeitbilddarstellungen:

- ▶ Seit ca. 60 Jahren Flugsimulatoren zur Pilotenausbildung,
- ▶ Schiffssimulationsanlage SUSAN (FH Hamburg),
- ▶ Planetarien,
- ▶ Videospiele,
- ▶ Ärzteausbildung, Operationsplanung,
- ▶ Telekonferenzen, virtuelle Studios.

Beispiel: ShipAnim

Fernerkundung



Simulierter Flug über den Mars (Olympus Mons)

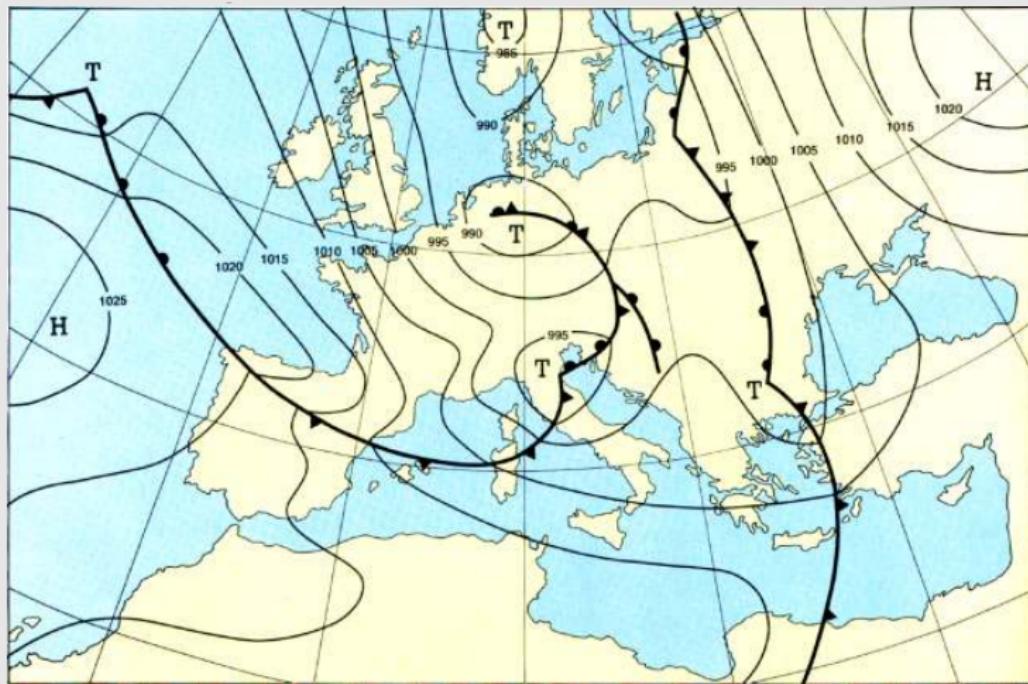
Simulierter Flug über den Mars

Simulierter Flug über den Mars (Olympus Mons)

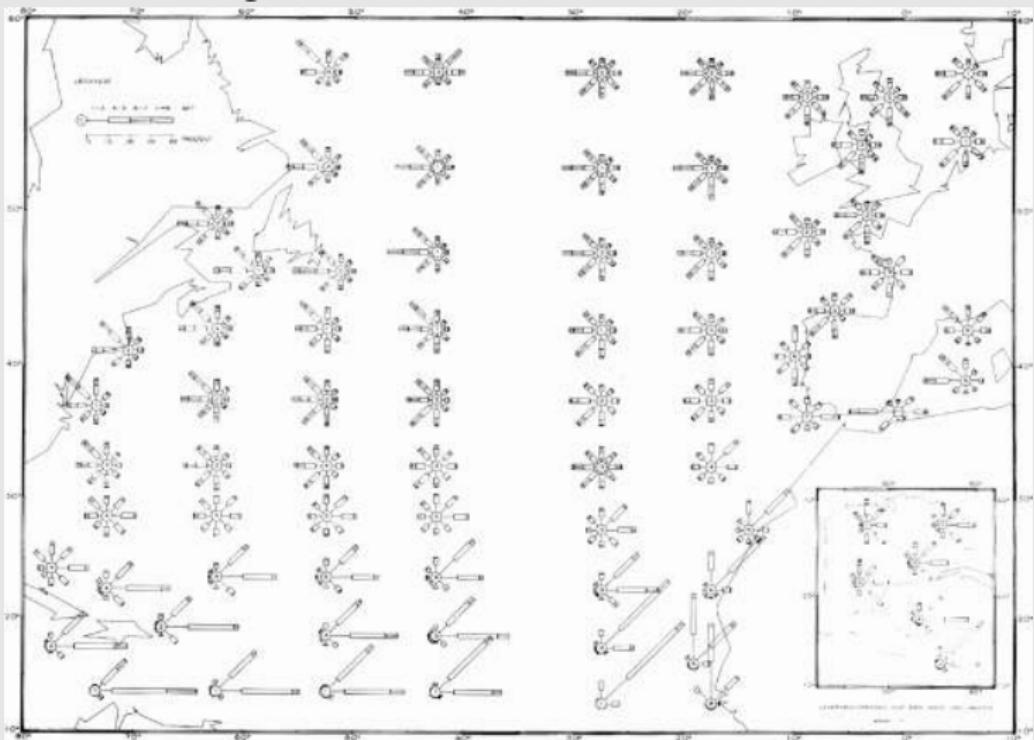
Visualisierungen

- ▶ Veranschaulichung von Prozessen und Vorgängen
- ▶ Darstellungen von Funktionen und Objekten in mehreren Dimensionen
- ▶ Interaktives Optimieren von Parametern
- ▶ Sichtbar machen von unsichtbaren Phänomenen (Strömungen, Röntgenbilder, Spannungen usw.)
- ▶ Visualisierung von Entwürfen (Architektur, Stadtplanung usw.)

Visualisierung des Luftdrucks durch Isobaren



Visualisierung der Windverhältnisse



Visualisierung von Strömungen



Visualisierung von Strömungen

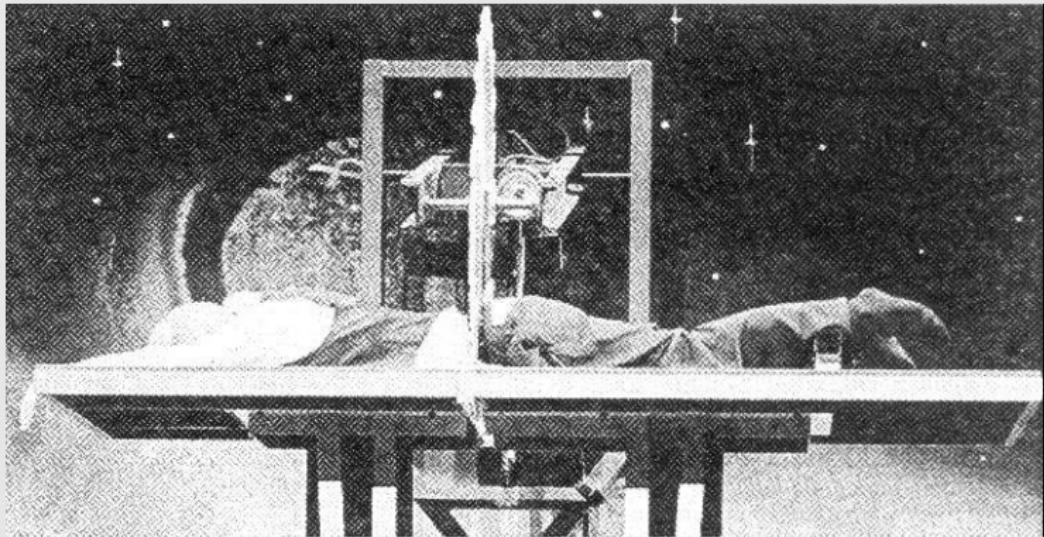
Simulierte Strömungen im Windkanal

Fotorealismus

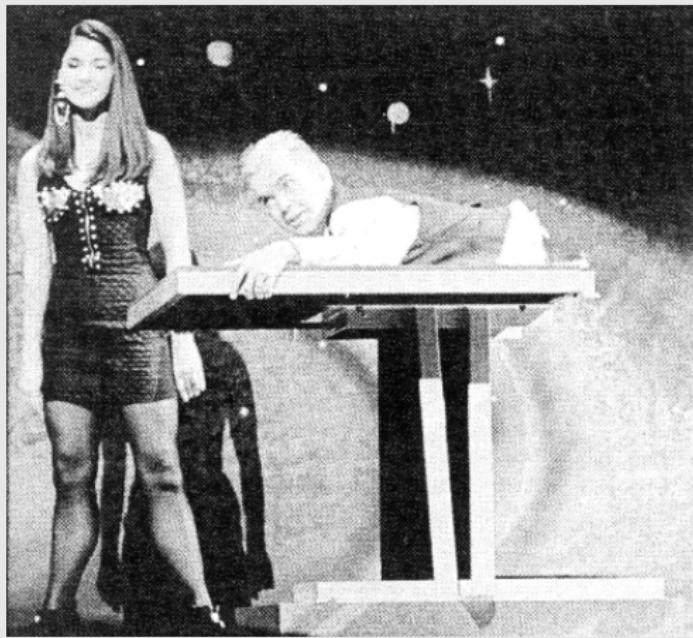
Erzeugung von Bildern oder Filmen, die von natürlichen nicht zu unterscheiden sind.

- ▶ Spielfilm,
- ▶ Werbung,
- ▶ Kunst,
- ▶ Simulation.

Eine Zaubershow im vergangenen Jahrhundert;
Tricks mit Spiegeln: *vorher*



Nachher:



Animationen für Computerspiele



Ein Bewegungsspiel

Rette den Ball vor dem Moskito

Anwendungen beim Spielfilm

Virtuelle Welten: Animation künstlicher Wesen (Jurassic Parc, Toy Story).

Tricks: Unmögliches möglich machen:

Forrest Gump schüttelt J.F. Kennedy die Hand und zeigt Elvis den Hüftschwung.

Vereinfachung der Produktion: ▶ Virtuelle Statisten,

virtuelle Kulissen, Vereinfachen von „stunts“,

▶ Verlagerung der Aufnahme ins Studio oder an preiswertere Drehorte.

Nachbearbeitung, künstlerischer Ausdruck:

Lichtverhältnisse, Wetter, Retouche.

Retouche von Zivilisationsspuren



Entfernen von Zivilisationsspuren

Entfernen von Oberleitungen und Straßen

Kaschieren von Ruinen



Kaschieren von Ruinen

Schloß Camelot als Pappkulisse

Entschärfen gefährlicher Szenen



Entschärfen gefährlicher Szenen

Eine Kutschfahrt im Studio

Entschärfen gefährlicher Szenen



Entschärfen gefährlicher Szenen



Ein Nahkampf ohne Schwerter

Synthetische Schwertklingen

Massenszenen

Vervielfältigung von Statisten



Massenszenen

Vervielfältigung von Statisten

Nur weniger Reiter sind nötig

Vervielfältigung von Kulissen



Vervielfältigung von Kulissen

Aus Tag wird Nacht

Einbringen von Modellen



Einbringen von Modellen

Raumschiffe als Modell

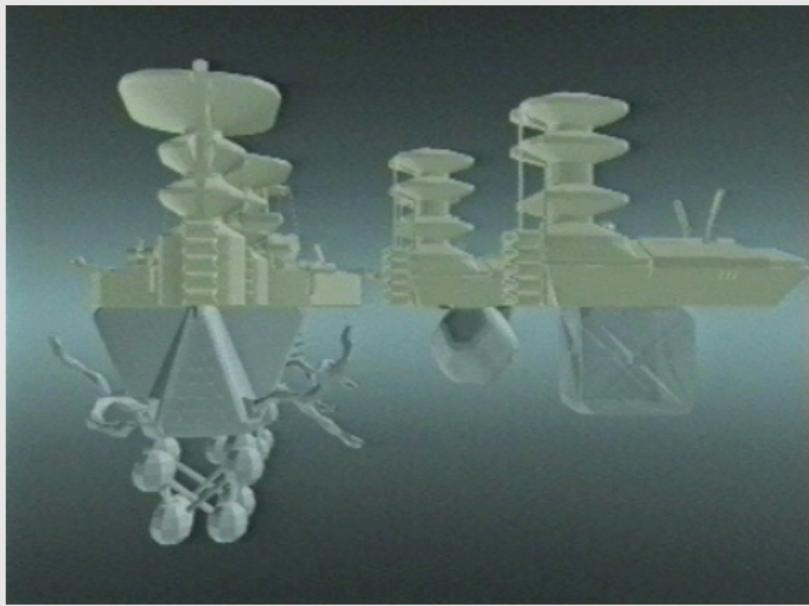
Virtuelle Kulissen



Virtuelle Kulissen

Kulissenumbau im Rechner

Eine virtuelle Welt, Computerkunst

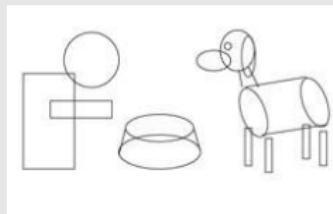


aus dem Film 'Neulich in Flockville" von Christine Heubi

Eine virtuelle Welt, Computerkunst

neulich in Flockville

Organisatorische



- 1 Einführung und Grundbegriffe
- 2 Anwendungen und Beispiele
- 3 Organisatorische
 - Übungsaufgabe
 - Literatur

Die Übungsaufgabe

Die Übungen:

- ▶ Für den Übungsschein ist eine kurze POV-Ray **Animationssequenz** zu erstellen und zu dokumentieren.
- ▶ Das Thema ist Ihnen freigestellt.
- ▶ Erstellen Sie ein Filmplakat zu Ihrem Film.
- ▶ Abgabe des Films und des Plakats: bis Montag, 26.1.2015, 9:30 Uhr
- ▶ Abgabe der Dokumentation: Bis Mittwoch, 28.1.2015 in der Vorlesung.

Anforderungen

Die folgenden Elemente sollen enthalten sein:

- ① Wenigstens ein Objekt in der Szene sollte **bewegte Gliedmaßen** haben.
- ② Wenigstens eine Szene sollte die **Kameraeinstellung** variieren, z.B. in die Szene hineinfahren, schwenken oder zoomen.
- ③ Setzen Sie ein verarbeitetes **Bild** ein, z.B. als Höhenprofil, Kulisse oder Textur.
- ④ Erstellen sie einen animierten **Titel**.
- ⑤ Setzen Sie **Überblendungen** ein, um den Schnitt zwischen zwei Kamerapositionen oder anderen Bildwechseln zu betonen oder zu kaschieren.
- ⑥ Setzen Sie **Sound-Effekte** zum Vertonen ein.

POV-Ray im Netz

- ▶ <http://povray.org/documentation/>
- ▶ <http://povray.org/download/>
- ▶ MacMegaPOV <http://www.macupdate.com/info.php?id/15692/macmegapov>

Auf den iMacs läuft zur Zeit nur MacMegaPOV, aber unter Linux und Windows die offizielle POV-Ray-Version.

Aufgabeformat

Wichtig!

- ▶ Schneiden Sie Ihren Film auf den iMacs des Informatikrechenzentrums (D-011 – D-013). Wir nehmen **nur** Quicktime-Filme an, die wir auf einem **Mac** abspielen können.
- ▶ Im KOGS-Labor (R-131) steht Ihnen auch ein Midi-Keyboard zur Verfügung, mit dem Sie Ihr Meisterwerk vertonen können.

So bekommt man den IVC-Schein

Und so entsteht wie von selbst ein Film....

Materialien:

- ▶ Die Wep-page zur Vorlesung (im Lehre-Blog):
<http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Lehre/Lehre.html>

Materialien:

- ▶ Die Wep-page zur Vorlesung (im Lehre-Blog):
<http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Lehre/Lehre.html>
- ▶ Folienskript als pdf:
<http://kogs-www.informatik.uni-hamburg.de/~dreschle/informatik/Skripte/IVChandouts.pdf>

Materialien:

- ▶ Die Wep-page zur Vorlesung (im Lehre-Blog):
<http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Lehre/Lehre.html>
- ▶ Folienskript als pdf:
<http://kogs-www.informatik.uni-hamburg.de/~dreschle/informatik/Skripte/IVChandouts.pdf>
- ▶ Beispieldateien aus den Übungen:
http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Lehre/Entries/2009/12/23_Studentische_Filme_zu_CGB_und_IVC.html

Literatur zur Bildentstehung

-  Glassner, A. S. (1995).
Principles of Digital Image Synthesis, volume 2 of *The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling*.
Morgan Kaufmann Publ. Inc., Los Altos/CA.
-  Fellner, W. (2004).
Computergrafik, volume 58 of *Reihe Informatik*.
BI Wissenschaftsverlag, Mannheim – Wien – Zürich,
3. edition.
-  A. Nischwitz and P. Haberäcker (2004).
Masterkurs Computergrafik und Bildverarbeitung
Vieweg.

Literatur zur Szenenanalyse

Steinmueller-2008: Ein gut verständliches, einführendes Buch zur Szenenanalyse. Als eBook mit Campuslizenz im Web verfügbar:

- ▶ <http://dx.doi.org/10.1007/978-3-540-79743-2>
(Volltextzugang Campus)
- ▶ [http://emedien.sub.uni-hamburg.de/han/
SpringerEbooks/dx.doi.org/10.1007/
978-3-540-79743-2](http://emedien.sub.uni-hamburg.de/han/SpringerEbooks/dx.doi.org/10.1007/978-3-540-79743-2) (Nur für Angehörige der
Universität Hamburg: Volltextzugang von ausserhalb
des Campus mit Bibliotheksausweis)



Steinmüller, Johannes (2008)

Bildanalyse: Von der Bildverarbeitung zur räumlichen Interpretation von Bildern, Reihe eXamen.press
Springer Verlag, Berlin – Heidelberg.

Literatur zur Szenenanalyse 2

- ❖ Gonzales, R. C. and Wood, R. E. (2008).
Digital Image Processing.
Pearson Prentice Hall, New Jersey, 3rd edition.
- ❖ Sonka, M., Hlavac, V., and Boyle, R. (1999).
Image Processing, Analysis, and Machine Vision.
PWSPublishing.
- ❖ Davies, E. R. (1997).
Machine Vision.
Academic Press, 2nd edition.
- ❖ Horn, B. K. P. (1986).
Robot Vision.
The MIT Press, Cambridge, Mass. McGraw-Hill, New York.

Geometrische Grundlagen

Koordinaten und Transformationen

4

2D-Koordinatensysteme und Transformationen

- 2D-Koordinatensysteme
- 2D-Transformationen
 - 2D-Translation
 - 2D-Skalierung
 - 2D-Rotation
- Zusammengesetzte Transformationen

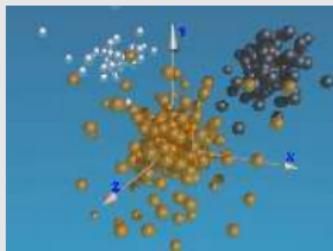
5

3D-Koordinatensysteme

6

Transformationen in POV-Ray

Koordinatensysteme und Transformationen



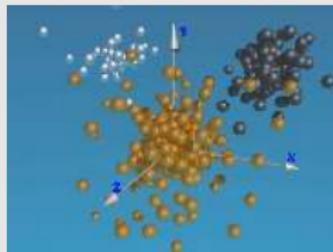
4 2D-Koordinatensysteme und Transformationen

- 2D-Koordinatensysteme
- 2D-Transformationen
- Zusammengesetzte Transformationen

5 3D-Koordinatensysteme

6 Transformationen in POV-Ray

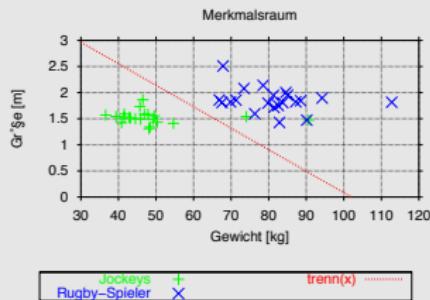
Anwendungen von Koordinatensystemen:



Ein
3D-Koordinaten-
system

- ▶ Angabe der Position und Orientierung von modellierten Objekten, Lichtquellen, Kameras,
- ▶ Relative Orientierung von Koordinatensystemen,
- ▶ Transformation von Objekten.

2D-Koordinatensysteme



2D-Koordinatensysteme dienen in der Computergrafik und Bildverarbeitung zur Beschreibung der relativen Anordnung von Bildelementen.

Ein
2D-Koordinatensystem

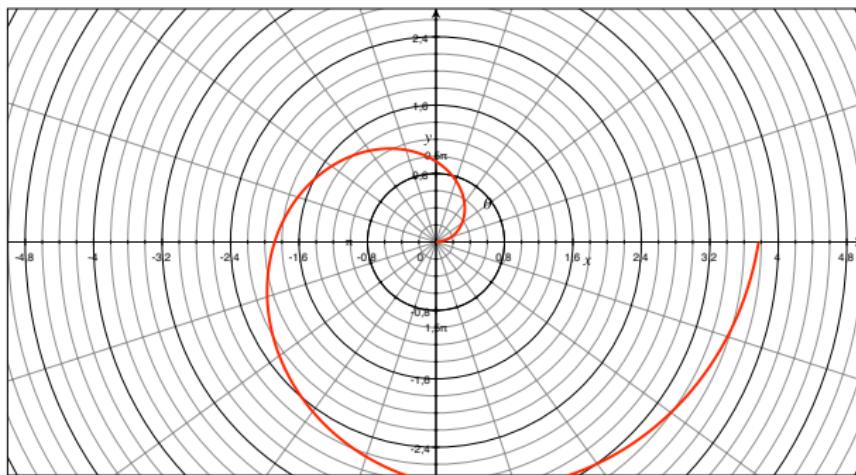
Gebräuchlich sind:

Kartesische Koordinaten: Ursprung links unten, x-Achse nach rechts, y-Achse nach oben.

Polarcoordinaten: (für mathematische Anwendungen).

Beispiel:

Eine Spirale in Polarkoordinaten

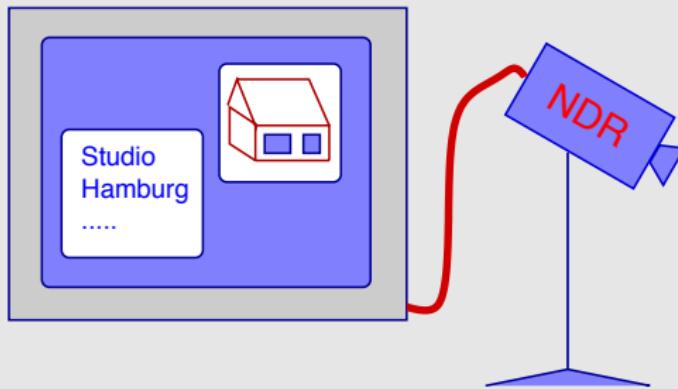


Spezifische Koordinatensysteme

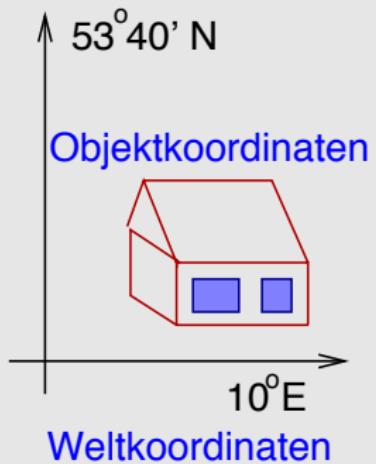
- ▶ Die Koordinaten, die von einem **spezifischen Ausgabegerät** verwendet werden, nennt man **Gerätekordinaten**.
- ▶ Die BenutzerInnen definieren ihre Koordinaten in anwendungsspezifischen **Weltkoordinatensystemen**. Diese sind unabhängig von den speziellen Gerätekordinaten.
- ▶ Häufig werden in einer Anwendung mehrere Weltkoordinatensysteme verwendet.
- ▶ Die verschiedenen Weltkoordinatensysteme werden meist vor der Abbildung auf die Gerätekordinaten auf ein *normalisiertes Koordinatensystem* (*normalized device coordinates*) abgebildet.

Hierarchie der Koordinatensysteme

Normalisierte Koordinaten



Gerätekoordinaten



Weltkoordinaten

Hierarchie der Koordinatensysteme

Weltkoordinatensystem: Anordnung der Szenenobjekte,
Betrachtungsparameter.

Objektkoordinatensystem: Anordnung der *Komponenten*
von Objekten, relativ zum Objekt definiert.

Bildkoordinatensystem: Anordnung der
Bildkomponenten.

Koordinatensysteme der Grafiksysteme:

- ▶ Normalisierte Gerätekoordinaten,
- ▶ Viewport-Koordinaten,
- ▶ Bildschirmkoordinaten.

2D-Transformationen

2D-Transformationen dienen zur Manipulation von ebenen grafischen Darstellungen, beispielsweise:

- ▶ einfache Animationen,
- ▶ Grafikeditoren,
- ▶ Koordinatensystemwechsel.

Es gibt elementare (Translation, Rotation, Skalierung, Scherung) und zusammengesetzte Transformationen (Koordinatentransformation).

Translation

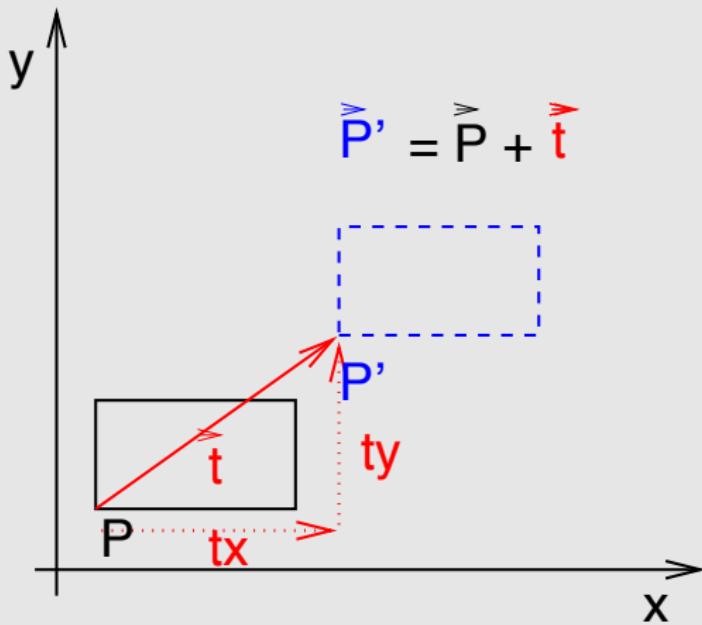
Definition (**Translation**)

Unter Translation versteht man eine geradlinige Verschiebung eines Objektes oder eines Definitionspunktes. [Fellner, 2004]

Definitionspunkte sind Punkte, die die Lage, Größe oder Form eines Objektes festlegen:

- ▶ Eckpunkte eines Rechtecks,
- ▶ Mittelpunkt eines Kreises usw.

Translationsvektor



Verschobener Definitionspunkt:

Neue Koordinaten

Die Koordinaten eines *verschobenen Punktes* $P' = (x', y')$ berechnen sich aus der ursprünglichen Position $P = (x, y)$ und dem **Translationsvektor** $\vec{t} = (t_x, t_y)$

$$P' = P + \vec{t} = (x, y) + (t_x, t_y) = (x + t_x, y + t_y).$$

Die Translation hat normalerweise keine *Fixpunkte*, d.h. keine Koordinate bleibt erhalten, es sei denn $\vec{t} = \vec{0}$.

Skalierung

Definition (**Skalierung**)

Mit Skalierung bezeichnet man die Vergrößerung bzw. Verkleinerung von Objekten.[Fellner, 2004]

Bei der **uniformen Skalierung** wird das Objekt in alle Richtungen gleich skaliert, andernfalls kann eine Verzerrung erfolgen.

Der Fixpunkt der Skalierung ist der Ursprung.

Skalierungsfaktoren

Die Koordinaten eines **skalierten Punktes** $P' = (x', y')$ berechnen sich aus der ursprünglichen Position $P = (x, y)$ und den **Skalierungsfaktoren** $t = (s_x, s_y)$

$$P' = (x', y') = (s_x \cdot x, s_y \cdot y).$$

Uniforme Skalierung: $s_x = s_y$.

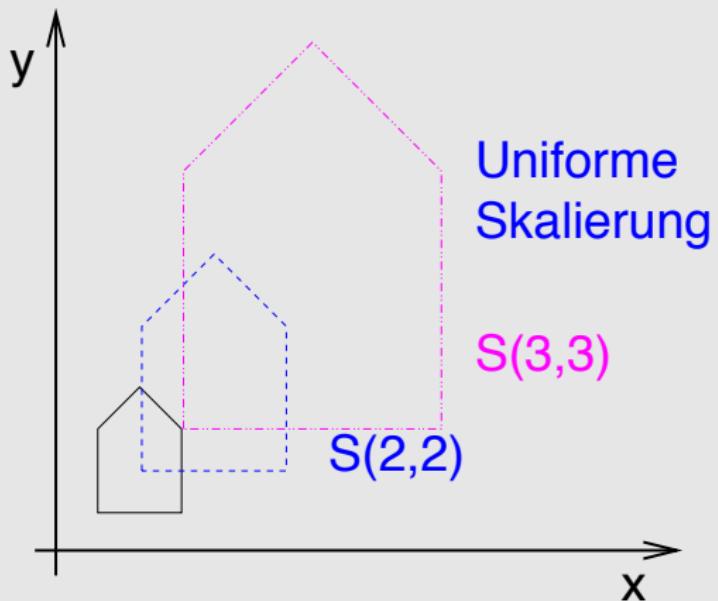
Beliebiger Fixpunkt: c_x, c_y

Translation zum Fixpunkt: $t(-c_x, -c_y)$,

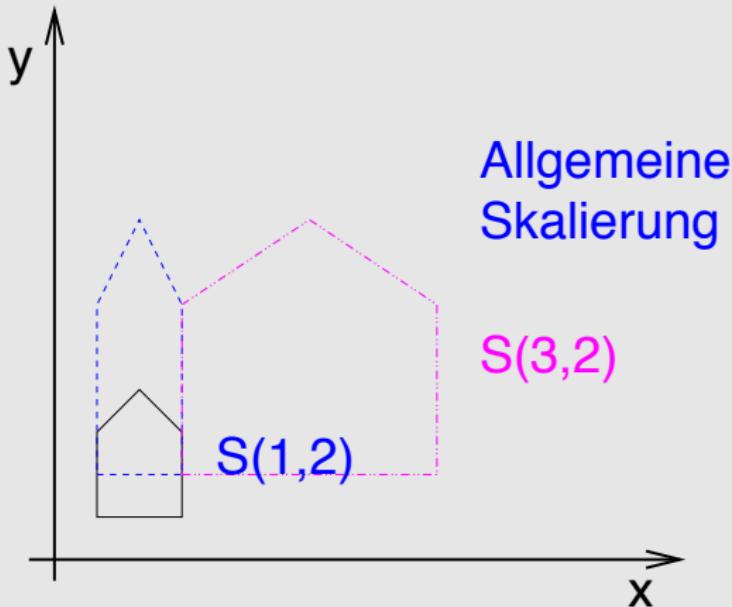
Skalierung um den Fixpunkt: $s(s_x, s_y)$,

Translation in die alte Lage: $t(c_x, c_y)$.

Beispiel: Uniforme Skalierung



Allgemeine Skalierung



Rotation

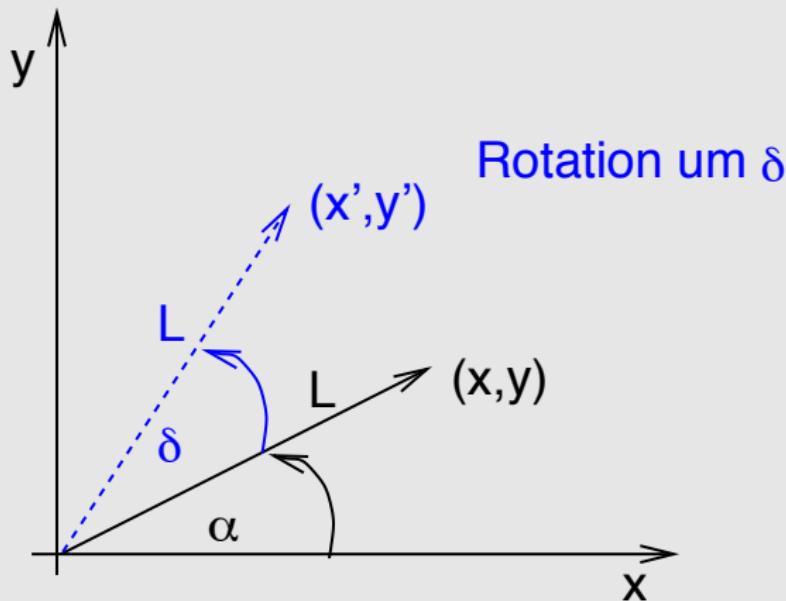
Definition (Rotation)

Die Rotation eines Objekts wird durch den Rotationswinkel δ und das Zentrum R_x, R_y festgelegt.

- Positive Werte für δ bewirken eine Drehung gegen den Uhrzeigersinn,
- negative Werte eine Drehung im Uhrzeigersinn.
- ▶ Das Zentrum ist ein Fixpunkt der Rotation.
- ▶ Bei der Rotation bleiben alle Abstände zwischen Punkten erhalten.

Beispiel:

Rotation von $P = (x, y)$ um den Winkel δ



Herleitung der Rotationsmatrix

Rotation von (x, y) nach (x', y') um den Winkel δ ;
Herleitung über den Winkelsummensatz:

$$\sin(\alpha + \delta) = \frac{y'}{L} = \cos(\delta) \sin(\alpha) + \sin(\delta) \cos(\alpha)$$

$$y' = L \cdot \cos(\delta) \cdot \frac{y}{L} + L \sin(\delta) \frac{x}{L}$$

$$y' = y \cdot \cos(\delta) + x \cdot \sin(\delta)$$

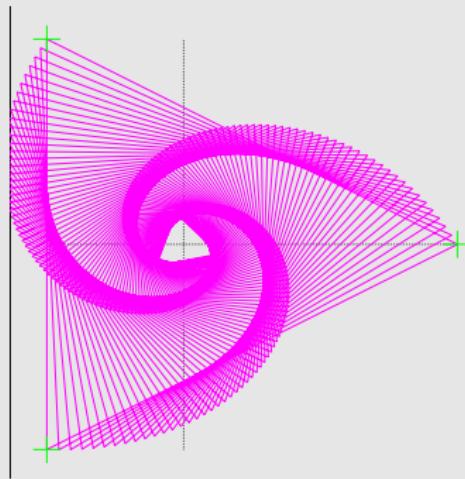
$$\cos(\alpha + \delta) = \frac{x'}{L} = \cos(\delta) \cos(\alpha) - \sin(\delta) \sin(\alpha)$$

$$x' = L \cdot \cos(\delta) \cdot \frac{x}{L} - L \sin(\delta) \frac{y}{L}$$

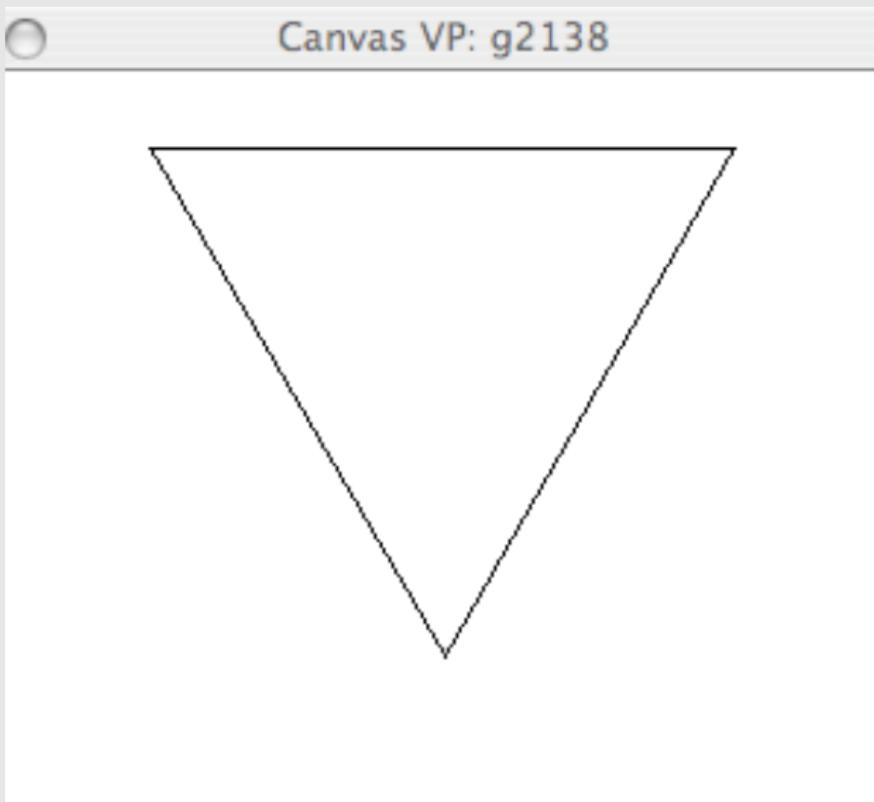
$$x' = x \cdot \cos(\delta) - y \cdot \sin(\delta)$$

Beispiel:

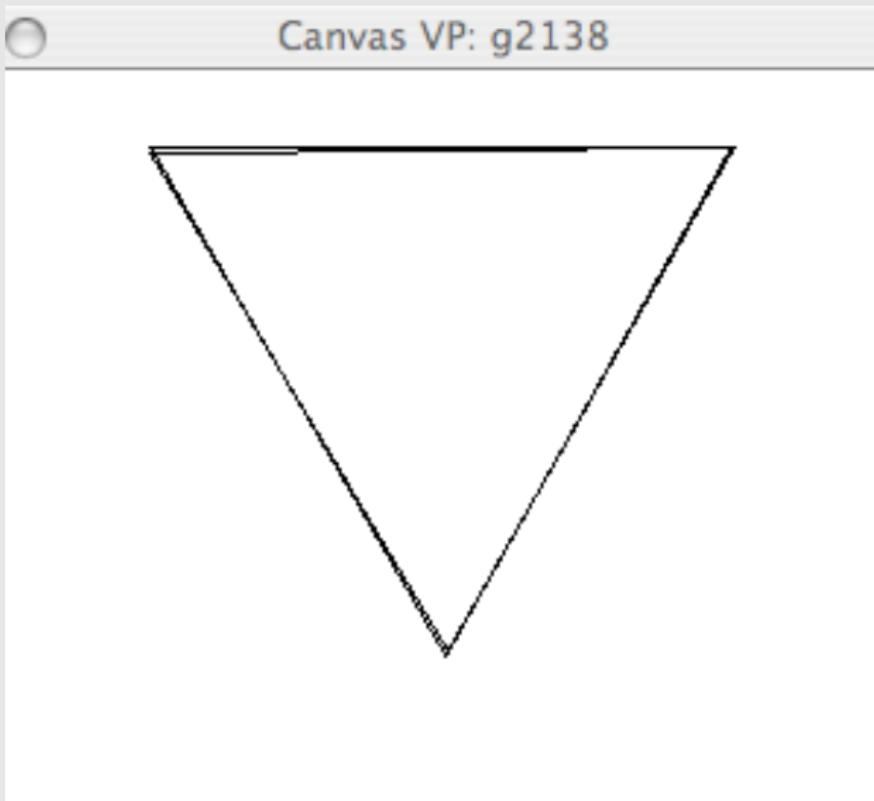
Rotation und Skalierung von Dreiecken



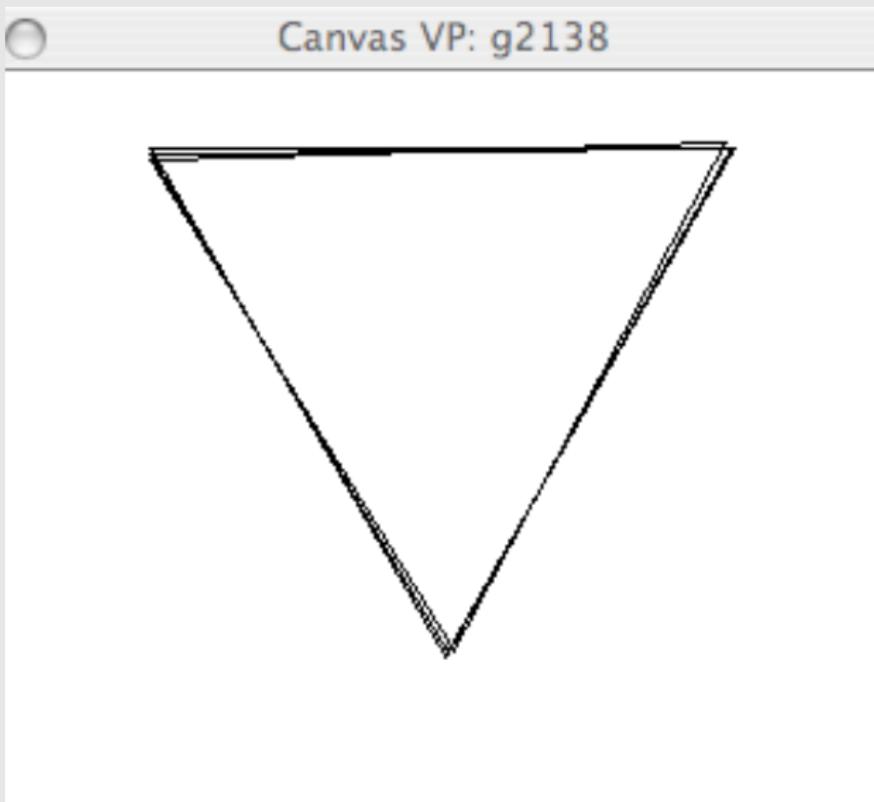
Ein linear-rekursiver Prozeß



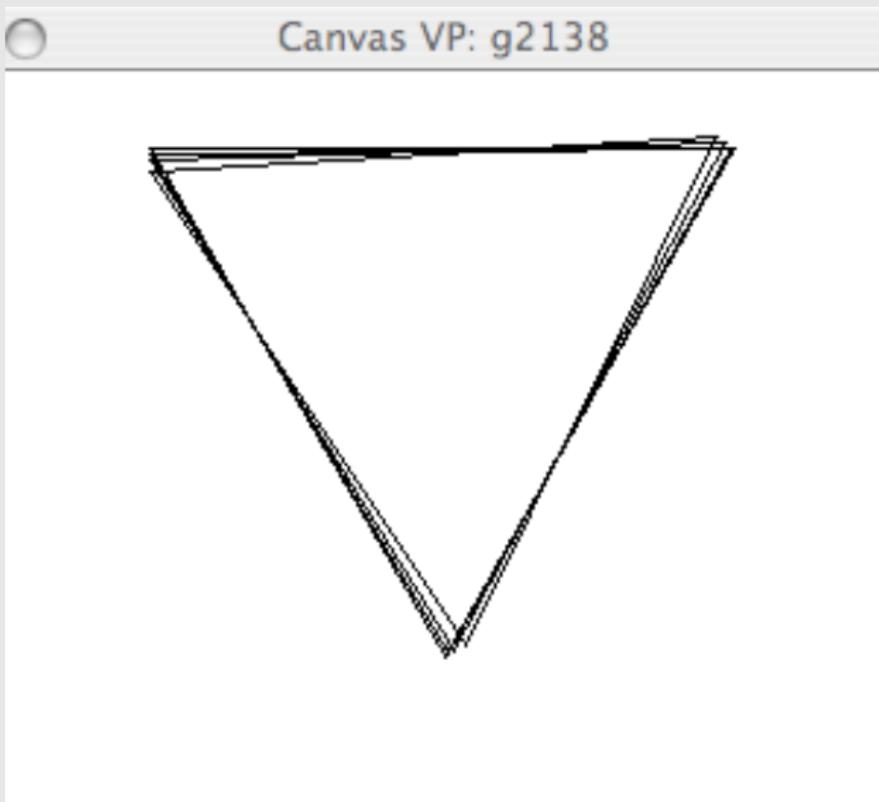
Ein linear-rekursiver Prozeß



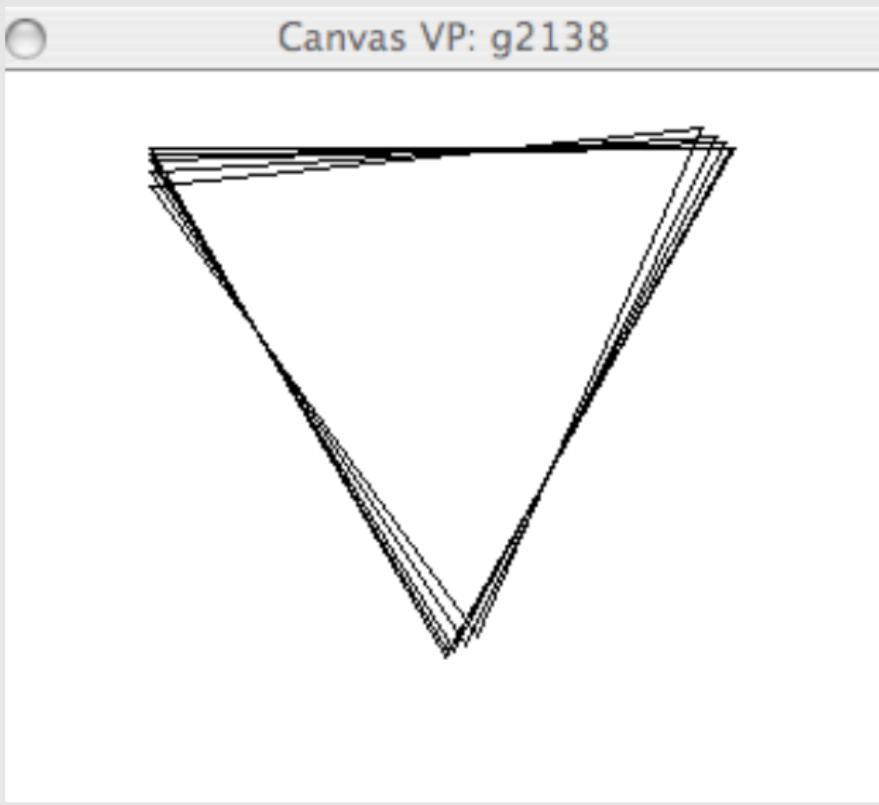
Ein linear-rekursiver Prozeß



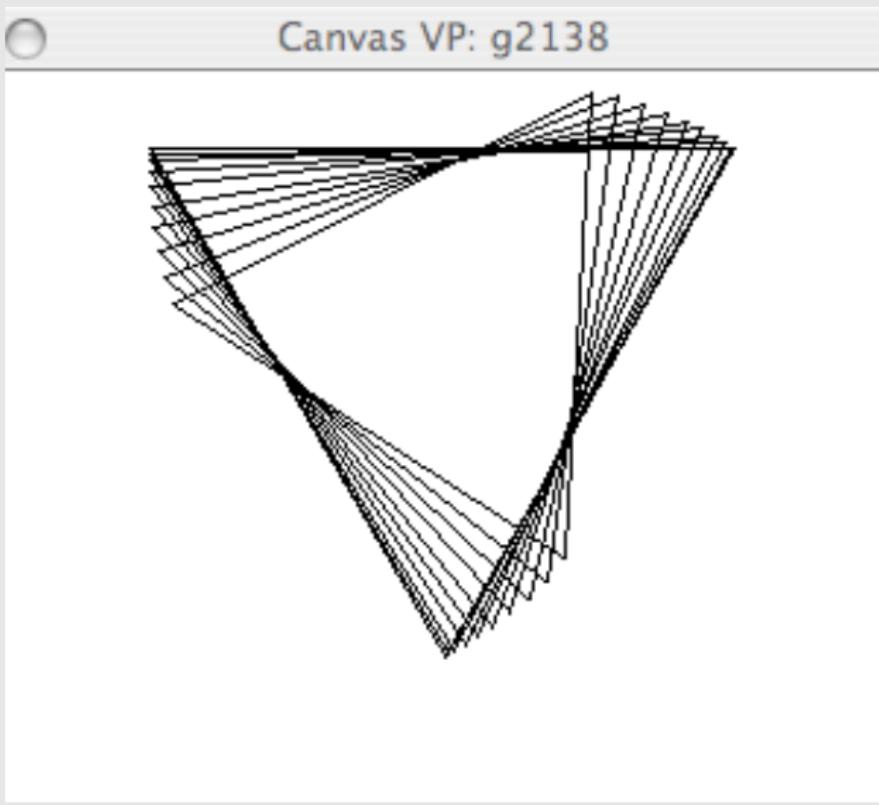
Ein linear-rekursiver Prozeß



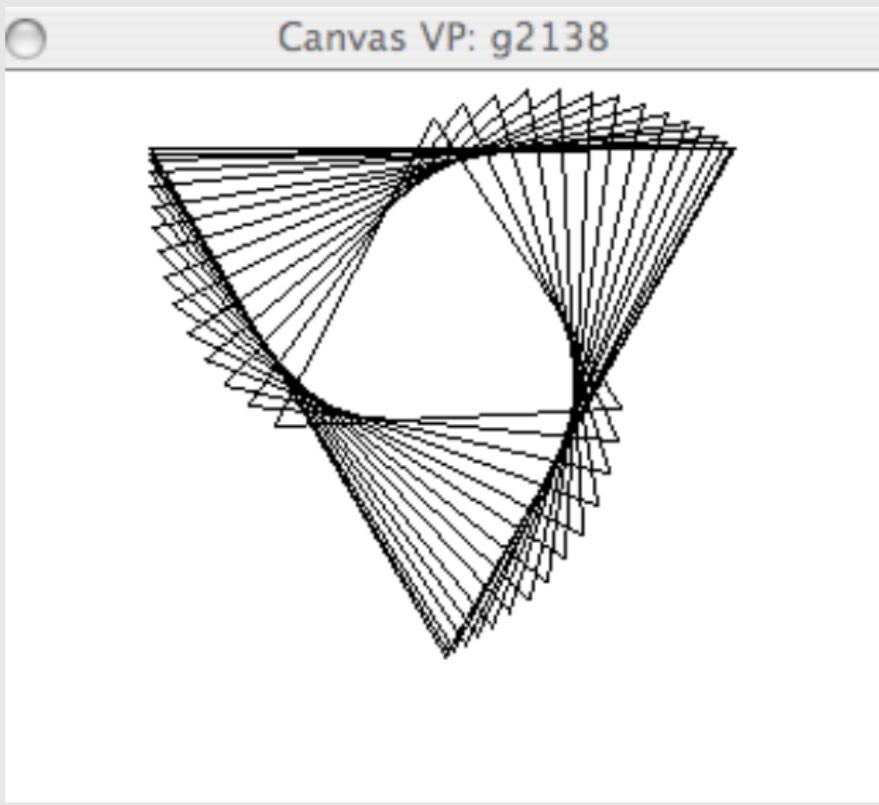
Ein linear-rekursiver Prozeß



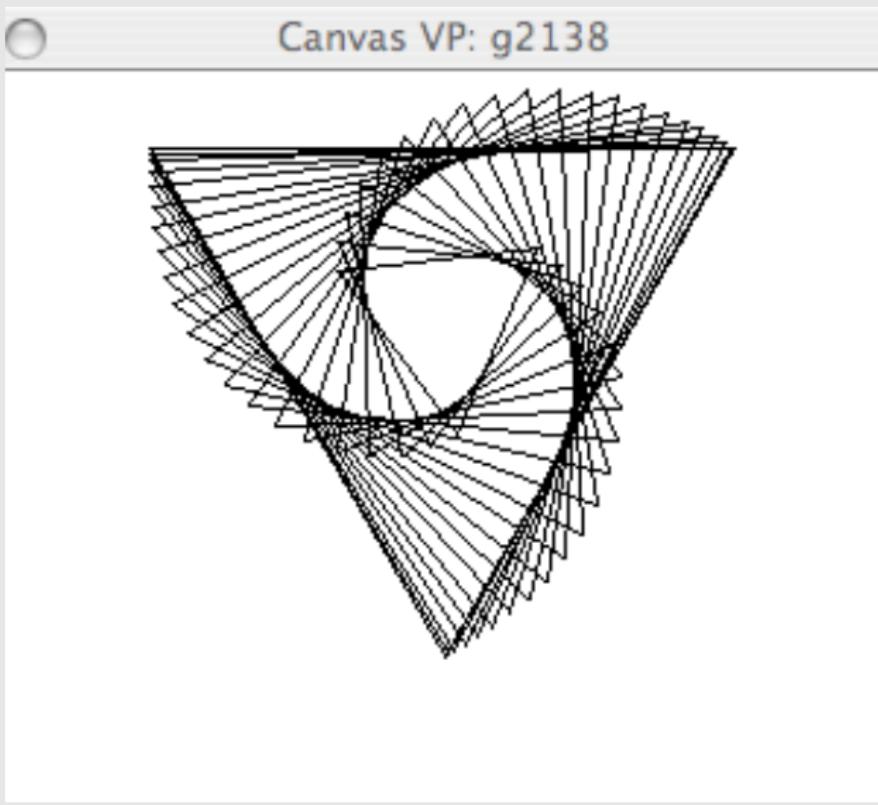
Ein linear-rekursiver Prozeß



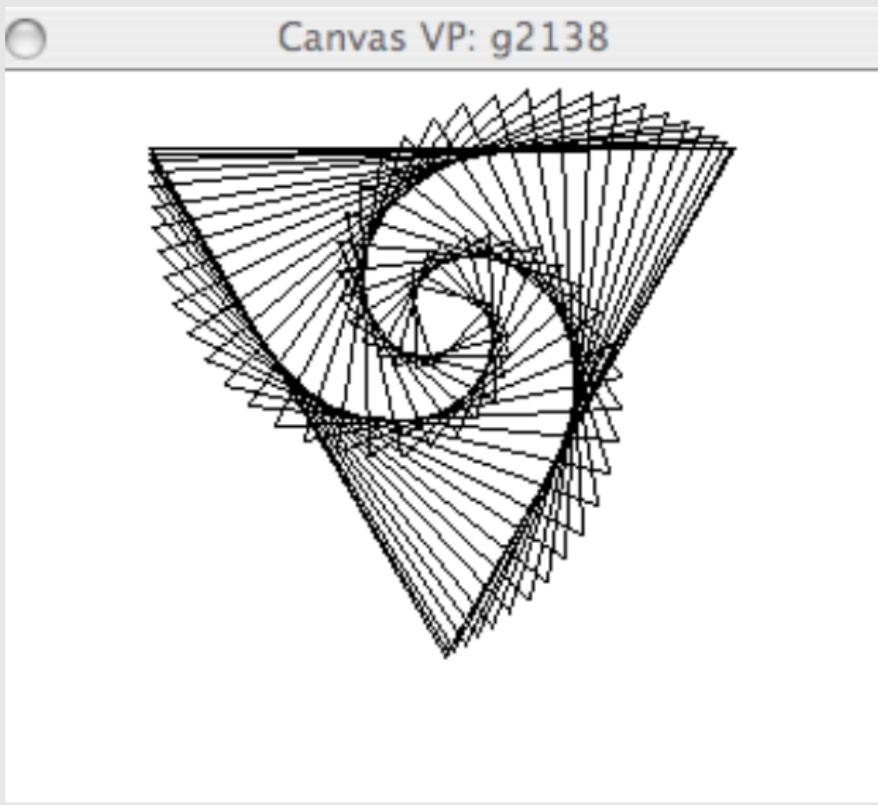
Ein linear-rekursiver Prozeß



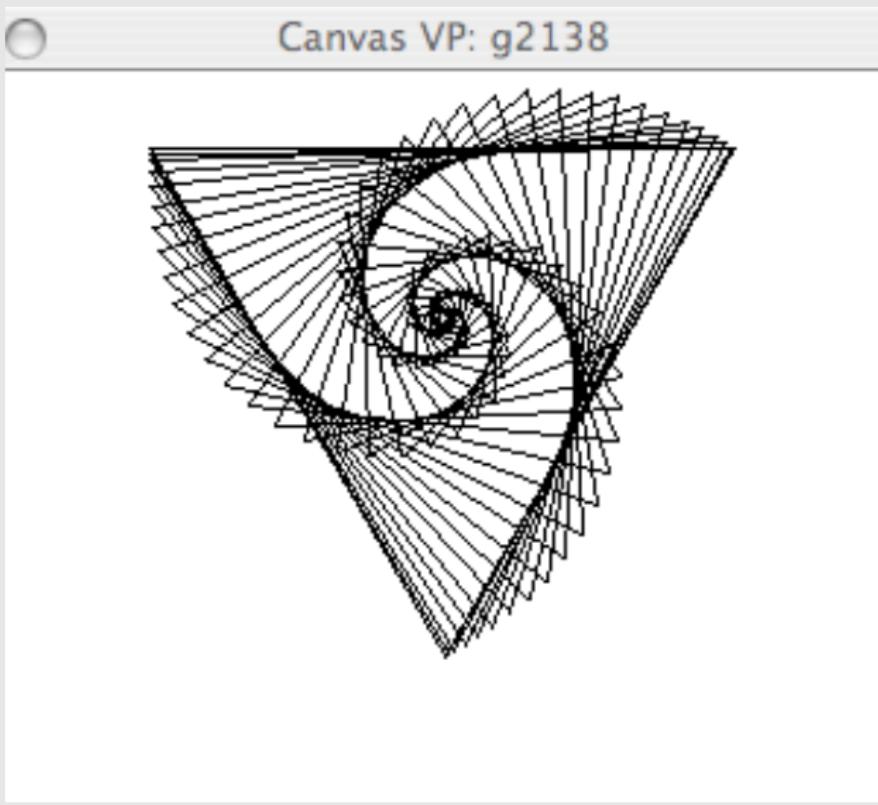
Ein linear-rekursiver Prozeß



Ein linear-rekursiver Prozeß

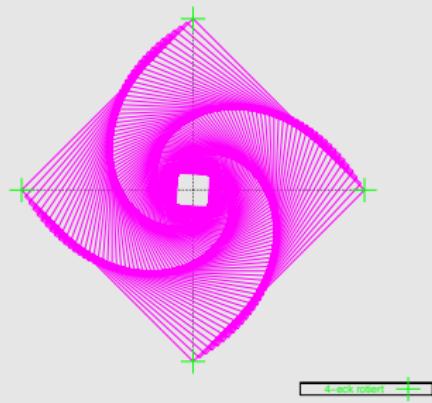


Ein linear-rekursiver Prozeß



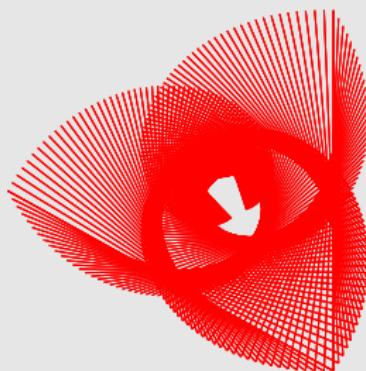
Beispiel:

Rotation und Skalierung von Quadraten



Beispiel:

Sechs rotierte Kopien der rotierten Dreiecke



Zusammengesetzte Transformationen

Bei der Abbildung von Weltkoordinaten auf Gerätekordinaten ist eine Folge von Transformationen auszuführen, die für viele Punkt gleich sind.

- ▶ Um Rechenzeit zu sparen und um möglichst wenig Genauigkeit zu verlieren, werden elementare Transformationen zu **zusammengesetzten Transformationen** komponiert.
- ▶ Alle bisher angesprochenen Transformationen lassen sich in **homogenen Koordinaten** als Matrixoperation darstellen.
- ▶ Homogene Koordinaten sind notwendig, da sonst nicht alle Transformationen linear wären.

Homogene Koordinaten

Ein Punkt $P(x, y)$ wird in **homogenen Koordinaten** durch das Tripel $\tilde{P} = [x \cdot w, y \cdot w, w]$ repräsentiert, mit $w \neq 0$.

Normalisierte Darstellung:

$$w = 1 : \quad \tilde{P} = [x, y, 1]$$

- ☞ Jeder Punkt hat unendlich viele äquivalente Repräsentationen in homogenen Koordinaten.
- ☞ Zum Vergleich muß daher die normalisierte Darstellung genommen werden.

Translation in homogenen Koordinaten:

Translation mit dem Translationsvektor (t_x, t_y) in homogenen Koordinaten:

$$[x' y' 1] = [x + t_x \ y + t_y \ 1] = [xy1] \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

$$P' = P \cdot T(t_x, t_y), \quad \text{mit} \quad T(t_x, t_y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

Skalierung

$$\text{Skalierung mit } s_x, s_y : S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotation

Zur Erinnerung:

Rotation von (x, y) nach (x', y') um den Winkel δ ;

$$y' = y \cdot \cos(\delta) + x \cdot \sin(\delta)$$

$$x' = x \cdot \cos(\delta) - y \cdot \sin(\delta)$$

Rotation um δ :

$$[x'y'1] = [xy1] \cdot R(\delta)$$

$$R(\delta) = \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Aufbau der Rotationsmatrix:

Rotation um δ : $R(\delta) = \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$

- ▶ Die Zeilen der Rotationsmatrix sind *orthogonale Einheitsvektoren* (wg. $\sin^2 \delta + \cos^2 \delta = 1$).
- ▶ Sie zeigen in Richtung der um δ gedrehten Koordinatenachsen.

Rechenregeln:

Für Folgen von Transformationen:

Satz (Assoziativgesetz:)

Es gilt das Assoziativgesetz:

$$(A \cdot B) \cdot C = A \cdot (B \cdot C), \quad ((P \cdot A) \cdot B) \cdot C = P \cdot (A \cdot B \cdot C).$$

Wenn eine Folge von Punkten P in gleicher Weise transformiert werden muß, kann eine gemeinsame, akkumulierte Matrix (A · B · C) für alle Punkte genutzt werden.

Satz (Kommutativgesetz)

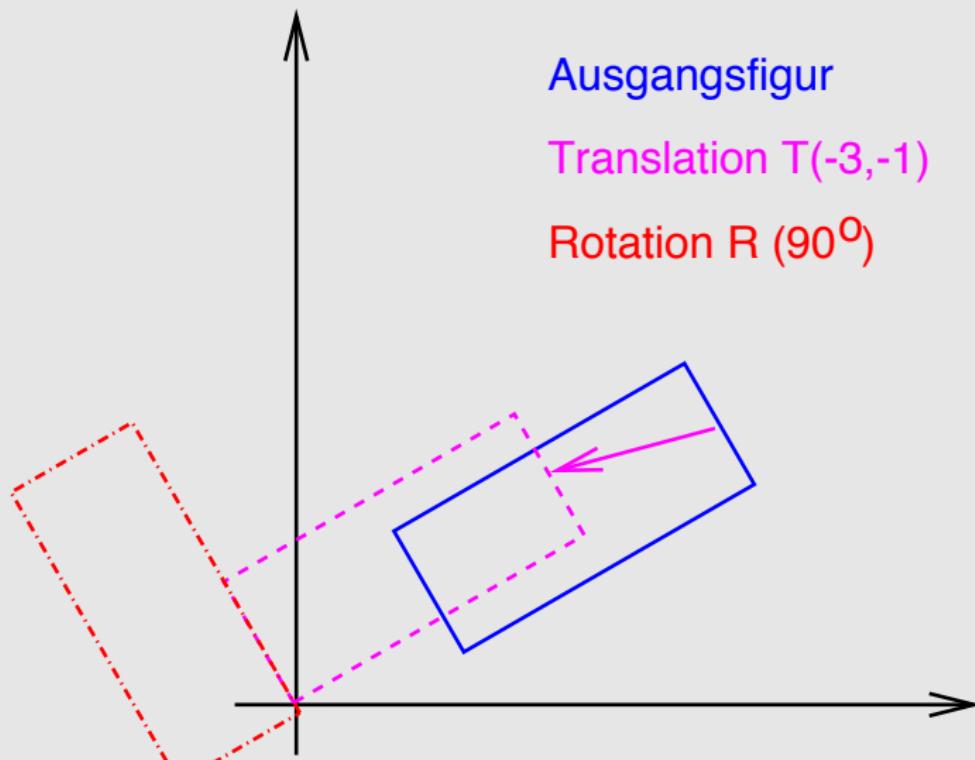
Das Kommutativgesetz gilt meistens nicht:

$$A \cdot B \neq B \cdot A,$$

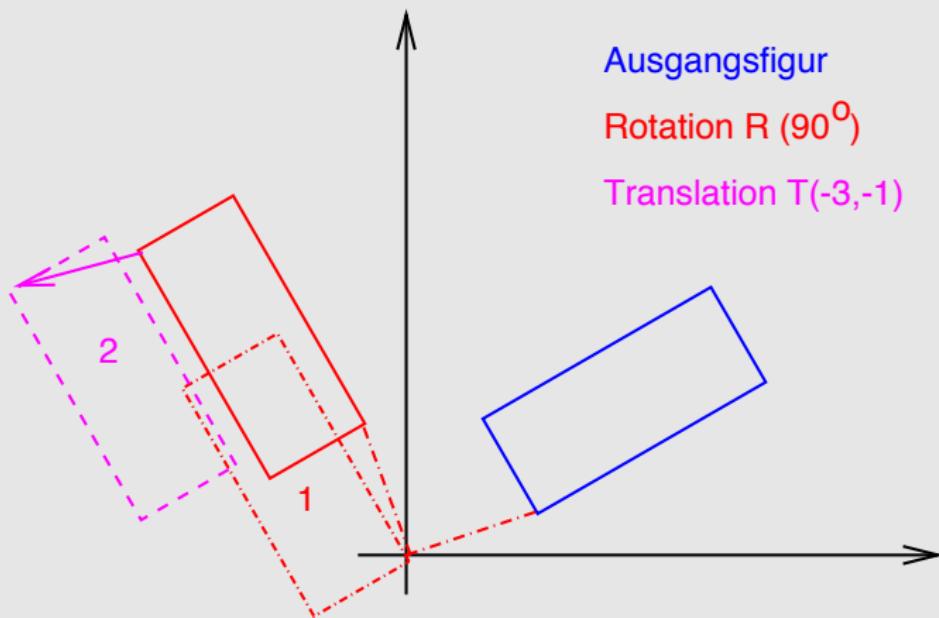
*nur innerhalb derselben Gruppe von Transformationen
(z.B. eine Folge von Rotationen oder Skalierungen) gilt:*

$$R(\alpha) \cdot R(\beta) = R(\beta) \cdot R(\alpha)$$

Erst Translation, dann Rotation



Erst Rotation, dann Rotation



Weitere Rechenregeln:

Beliebige Fixpunkte:

$$T(t_{x1}, t_{y1}) \cdot T(t_{x2}, t_{y2}) = T(t_{x1} + t_{x2}, t_{y1} + t_{y2})$$

$$S(s_{x1}, s_{y1}) \cdot S(s_{x2}, s_{y2}) = S(s_{x1} \cdot s_{x2}, s_{y1} \cdot s_{y2})$$

$$R(\delta_1) \cdot R(\delta_2) = R(\delta_1 + \delta_2)$$

Zooming: Fixpunkt (c_x, c_y)

$$Z((c_x, c_y), s_{x1}, s_{y1}) = T(-c_x, -c_y) \cdot S(s_x, s_y) \cdot T(c_x, c_y)$$

Rotation: Fixpunkt (r_x, r_y)

$$R((r_x, r_y), \delta) = T(-r_x, -r_y) \cdot R(\delta) \cdot T(r_x, r_y)$$

Affine Transformation

Definition (Affine Transformation)

Eine Transformation heißt **affin**, wenn die Transformationsmatrix A die folgende Form hat:

$$A = \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 1 \end{bmatrix}$$

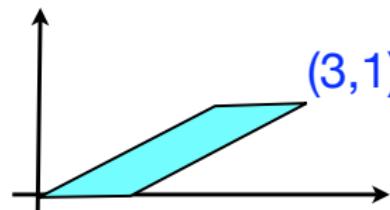
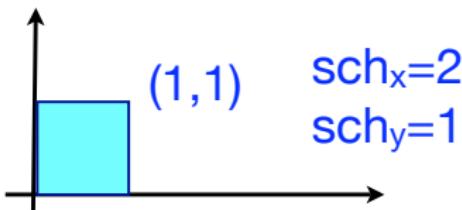
Alle bisher eingeführten Transformationen sind affine Transformationen. Weitere affine Transformationen:

- ▶ Scherung, Spiegelung, Zentralprojektion (3D).

Scherung

Scherung in x und y: sch_x, sch_y :

$$SCH(sch_x, sch_y) = \begin{bmatrix} 1 & sch_y & 0 \\ sch_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Vorteile

- ▶ Grafikpakete führen für jedes Objekt eine *Transformationsmatrix*, in der alle anfallenden Transformationen akkumuliert werden werden.
- ▶ Diese Matrix wird mit der *Einheitsmatrix* initialisiert.
- ▶ Vor der Anzeige werden die Definitionspunkte des Objekts mit der Transformationsmatrix multipliziert.
- ▶ Die Multiplikation wird meist durch geeignete Hardware unterstützt.

$$x' = ax + by + c \quad y' = dx + ey + f$$

- ▶ Der Rechenaufwand für die Anzeige ist konstant und unabhängig von der Zahl der Transformationen.

Vorteile 2

- ▶ Rundungsfehler durch die Abbildung auf ganzzahlige Raster werden klein gehalten, da erst am Ende gerundet wird.
- ▶ Einige Grafiksysteme bieten Transformationsmatrizen als eigene Datenstruktur an (GKS, POV-Ray).
- ▶ Beachten Sie bitte, daß die Transformationen hier für *Zeilenvektoren* definiert sind. Für Spaltenvektoren (wie in der Bildverarbeitungsliteratur meist üblich) müssen die Gleichungen umgestellt und die Transformationsmatrizen an der Hauptdiagonalen gespiegelt werden.

Geometrische Grundlagen:

3D-Koordinatensysteme:

4

2D-Koordinatensysteme und Transformationen

5

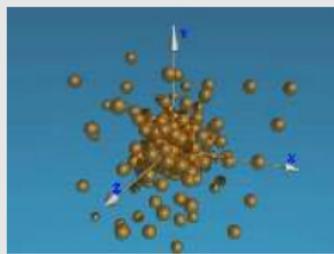
3D-Koordinatensysteme

- Händigkeit eines Koordinatensystems
- 3D-Transformationen
 - Transformationen in POV-Ray
 - 3D-Translation
 - 3D-Skalierung
 - 3D-Rotation

6

Transformationen in POV-Ray

3D-Koordinatensysteme:



4 2D-Koordinatensysteme und Transformationen

5 3D-Koordinatensysteme

- Händigkeit eines Koordinatensystems
- 3D-Transformationen

6 Transformationen in POV-Ray

Händigkeit eines Koordinatensystems

Es gibt zwei Möglichkeiten, die kartesische XY-Ebene durch eine Z-Koordinate (senkrecht auf der XY-Ebene) zu einem dreidimensionalen Koordinatensystem zu erweitern.

Linkshändiges Koordinatensystem:

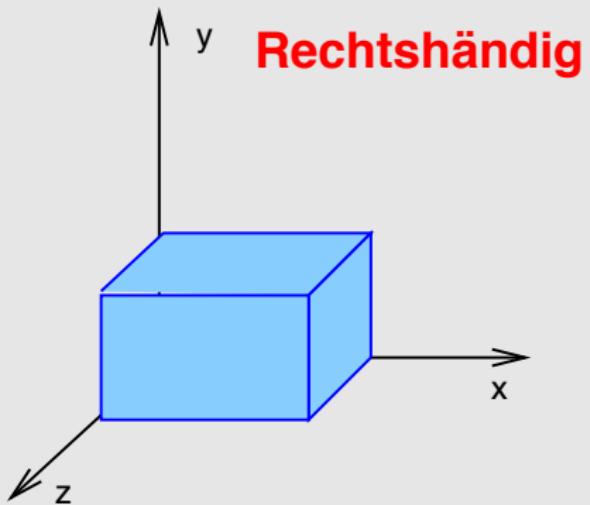
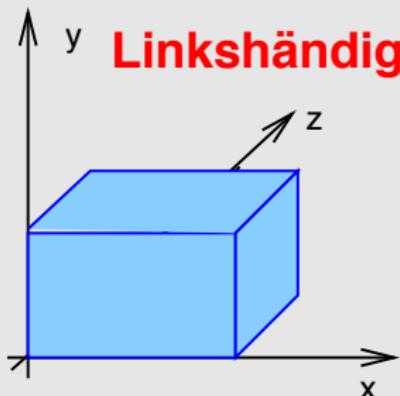
Die Z-Achse zeigt vom Betrachter weg in die Tiefe.

Rechtshändiges Koordinatensystem:

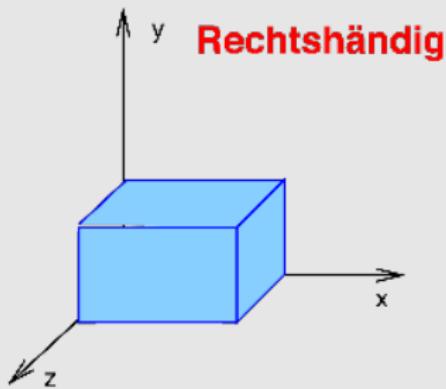
Die Z-Achse zeigt zum Betrachter.

Die beiden Koordinatensysteme sind spiegelbildlich und nicht durch Drehung ineinander zu überführen.

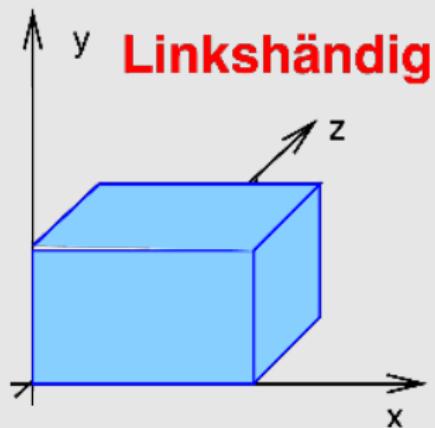
Händigkeit eines Koordinatensystems



Eselbrücke: Rechtshändig



Eselbrücke: Linkshändig

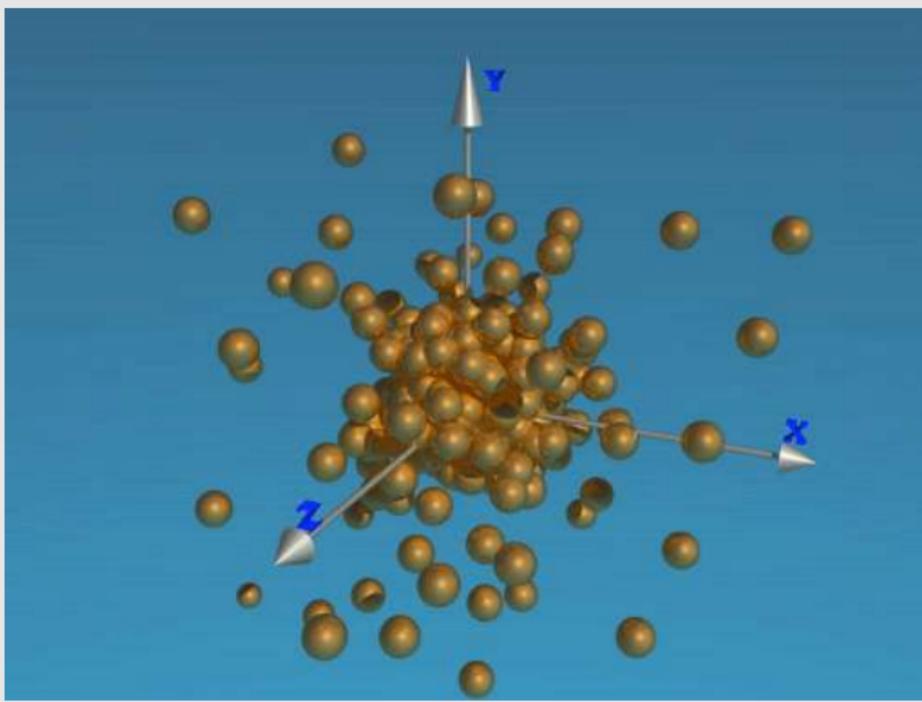


3D-Transformationen: Motivation

Eine wichtige Motivation ist die **Wiederverwendbarkeit** von Objekten in allgemeiner Lage:

- ▶ Objektmodelle werden in einem **objektbezogenen Koordinatensystem** definiert und durch Transformationen in die richtige Position, Orientierung und Größe gebracht.
- ▶ So kann ein Modell an vielen Stellen in einer Szene zum Einsatz kommen.

Beispiel: Ein Objektmodell mehrfach verwendet



Translationen in POV-Ray

```
translate < tx , ty , tz >
// Translation mit dem Vektor (tx ,ty ,tz )
translate y*3
// Translation entlang der Y-Achse (3 Einh.)
// 3*y = <0,3,0>
translate y*3+x
// 3*y+x = <1,3,0>
```

Definition von Objekten

Beispiel

Ein Beispiel für die Definition eines Objektmodells (stardot) und die mehrfache Nutzung in der Szene.

```
#declare stardot = sphere { 0, 0.7
    texture { T_Gold_1B }
}
object { stardot
    translate < 0.839746, -14.392706, 12.386859>
}
object { stardot
    translate <-2.100542, 6.854274, 3.663232>
}
object { stardot
    translate <-5.767194, -0.599780, 18.099405>
}
object { stardot
    translate < 2.928079, 3.419351, 1.299496>
}
...
```

3D-Transformationen

- ▶ Auch im 3D-Raum gibt es die Transformationen Translation, Rotation, Skalierung, Scherung.
- ▶ Die Rotation ist allgemeiner, da jetzt nicht nur ein Zentrum und ein Winkel, sondern auch ein beliebiger Vektor als Drehachse gewählt werden kann.
- ▶ Die homogene Darstellung der Koordinaten hat jetzt entsprechend vier Koordinaten.

3D-Translation in homogenen Koordinaten

Translation mit dem Translationsvektor (t_x, t_y, t_z) in homogenen Koordinaten:

$$[x + t_x \quad y + t_y \quad z + t_z \quad 1] = [xyz1] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

$$P' = P \cdot T(t_x, t_y, t_z), \text{ mit } T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{bmatrix}$$

3D-Skalierung in homogenen Koordinaten

Skalierung mit s_x, s_y, s_z :

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Skalierungen werden in POV-Ray mit dem Schlüsselwort
scale angegeben:

Uniforme Skalierung: scale *Faktor*

Ungleichförmige Skalierung:

scale <*Faktor x, Faktor y, Faktor z*>

Skalierung mit Fixpunkt

Zooming mit Fixpunkt $Z = (c_x, c_y, c_z)$, Skalierung s_x, s_y, s_z :

$$S(Z, s_x, \dots) = T(-c_x, -c_y, -c_z) \cdot S(s_x, s_y, s_z) \cdot T(c_x, c_y, c_z)$$

$$= \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ (1 - s_x) \cdot c_x & (1 - s_y) \cdot c_y & (1 - s_z) \cdot c_z & 1 \end{bmatrix}$$

Rotation in drei Dimensionen

3D-Rotation:

Eine **3D-Rotation** wird durch

- ▶ einen Punkt (das **Rotationszentrum**),
- ▶ einen Vektor (die Richtung der **Drehachse**)
- ▶ und den **Rotationswinkel** angegeben.

Rotationszentrum: Ein Punkt auf der Drehachse.

Drehachse: Die Achse, um die die Drehung erfolgt.

Nomenklatur: Drehungen um die Hauptachsen

Drehungen im Ursprung um die Hauptachsen heißen:

Roll: Kippung um die Z-Achse ψ .

Pitch: Drehung um die Y-Achse χ .

Yaw: Neigung um die X-Achse ϕ .

Roll

Roll: Rotation im Ursprung um die z-Achse, Winkel ψ

$$x' = x \cdot \cos(\psi) - y \cdot \sin(\psi)$$

$$y' = x \cdot \sin(\psi) + y \cdot \cos(\psi)$$

$$z' = z$$

$$R_z(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Yaw und Pitch

Und die Drehungen um die anderen beiden Achsen:

$$\text{Yaw: } R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) & 0 \\ 0 & -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Pitch: } R_y(\chi) = \begin{bmatrix} \cos(\chi) & 0 & -\sin(\chi) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\chi) & 0 & \cos(\chi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotationsrichtung und Händigkeit

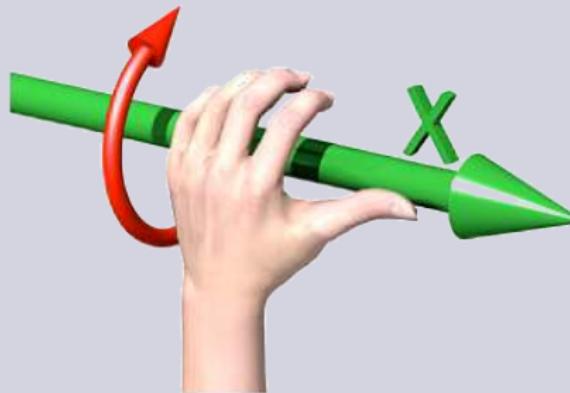
Achtung: Bei der Rotation ist die Händigkeit des Koordinatensystems zu beachten:

Linkshändiges System: Ein **positiver** Drehwinkel bewirkt eine Drehung **im** Uhrzeigersinn ⌂.

Rechtshändiges System: Ein **positiver** Drehwinkel bewirkt eine Drehung **gegen** den Uhrzeigersinn ⌂.

Wenn wir die Finger um die Drehachse legen und dabei den Daumen in die positive Richtung der Drehachse zeigen lassen, dann zeigen die Finger in Richtung der positiven Drehwinkel (linke Hand für ein linkshändiges System, rechte Hand für ein rechtshändiges).

Eselbrücke: Rotation und Händigkeit



Wenn wir die Finger um die Drehachse legen und dabei den Daumen in die positive Richtung der Drehachse zeigen lassen, dann zeigen die Finger in Richtung der **positiven** Drehwinkel (linke Hand für ein linkshändiges System, rechte Hand für ein rechtshändiges).

Linkshändig

Linkshaendig

Rechtshändig

Rechtshaendig

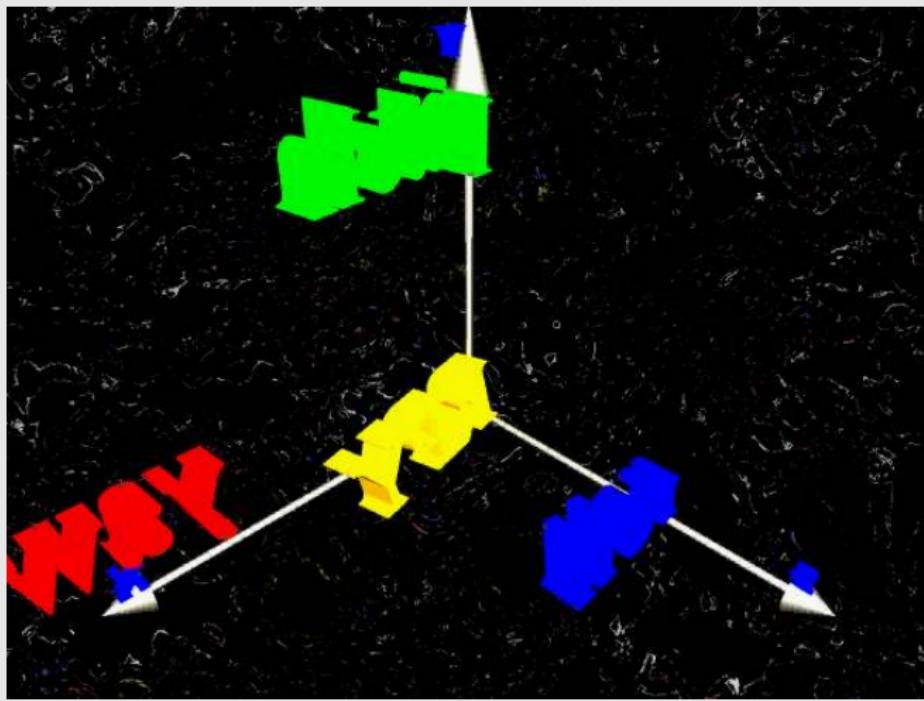
3D-Rotation in POV-Ray

Rotationen werden in POV-Ray durch einen (ϕ, χ, ψ) -Vektor angegeben, der eine Folge von Drehungen (in Grad) im Ursprung um die Hauptachsen beschreibt, zuerst *yaw*, dann *pitch*, dann *roll*.

```
rotate <20, 10, 3>
```

Drehe 20° um die x-Achse, dann 10° um y, dann 3° um z.

Roll, Pitch und Yaw



Roll, Pitch, and Yaw

Roll pitch yaw

Roll, Pitch, and Yaw: Artistik

Roll pitch yaw:Artistik

Rotationen in POV-Ray

- ▶ Die Rotationen können auch kaskadiert werden.
- ▶ Die Vektoren können als Ausdrücke unter Bezug auf vordefinierte Namen angegeben werden.

```
rotate 20*x  
rotate 10*y  
rotate 3*z
```

Drehe 20° um die x-Achse, dann 10° um y, dann 3° um z.

Drehungen um beliebige Achsen

Drehungen um *beliebige Achsen* werden aus elementaren Transformationen zusammengesetzt:

- ▶ Translation der Rotationsachse (und des zu drehenden Objekts), so daß die Achse durch den Ursprung verläuft.
- ▶ *Yaw* der Drehachse um die x-Achse in die xz-Ebene.
- ▶ *Pitch* der Drehachse um die y-Achse in die z-Achse.
- ▶ *Roll* des Objekts um die z-Achse (jetzt die Drehachse) mit dem Winkel δ .
- ▶ Rücktransformation der Drehachse in ihre ursprüngliche Lage durch die inversen Transformationen.

Teil II

Geometrische Grundlagen: Koordinaten und Transformationen

Zusammengesetzte Transformationen

(affin)

- ▶ Zusammengesetzte affine Transformationen können in POV-Ray durch eine 4×3 - Matrix angegeben werden.
- ▶ Die vierte Spalte wird automatisch zu [0001] ergänzt.

```
matrix < m00, m01, m02,  
          m10, m11, m12,  
          m20, m21, m22,  
          m30, m31, m32 >
```

Scherungsmatrix in POV-Ray

Eine Scherung in y-Richtung:

```
object { MyObject // irgend ein Objekttyp
matrix < 1, 1, 0,
          0, 1, 0,
          0, 0, 1,
          0, 0, 0 > }
```

Scherung einer Kugel:

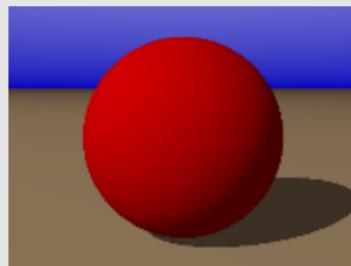
Animation

Beispiel

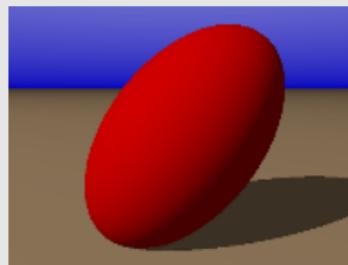
Eine Kugel wird in Abhängigkeit von der Uhrzeit entlang der Y-Achse geschert.

```
sphere {0, 1
pigment {red} }
matrix <1, clock, 0,
          0, 1, 0,
          0, 0, 1,
          0, 0, 0 >
```

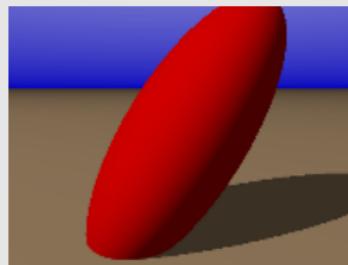
Scherung einer Kugel



Scherung einer Kugel



Scherung einer Kugel



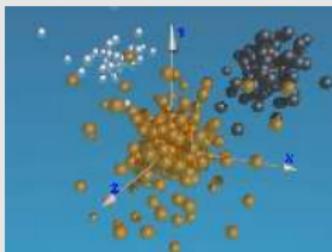
Scherung entlang aller drei Achsen

```
// Reorients and deforms object so:  
// x axis points along A, original y along B,  
// and original z along C.  
#macro Shear_Trans(A, B, C)  
    transform {  
        matrix < A.x, A.y, A.z,  
              B.x, B.y, B.z,  
              C.x, C.y, C.z,  
              0, 0, 0>  
    }  
#end  
// aus: transforms.inc
```

Scherung

Scherung der X-Achse

Koordinatentransformation



4 2D-Koordinatensysteme und Transformationen

5 3D-Koordinatensysteme

6 Transformationen in POV-Ray

- Affine Transformationen in POV-Ray
- Koordinatentransformation und Frames
- Ein virtueller Schneemann
- Iterative Konstruktion von Objekten

Koordinatentransformation

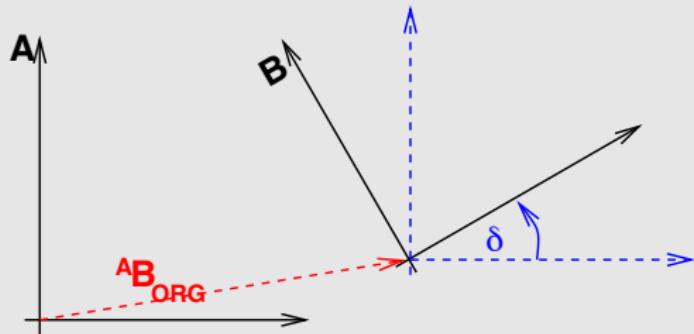
Die Rotation um eine beliebige Achse kann als eine Koordinatentransformation aufgefaßt werden:

- ▶ Wir verschieben den Ursprung des Koordinatensystems in den Drehpunkt.
- ▶ Wir drehen die z-Achse parallel zur Rotationsachse.
- ▶ Im transformierten Koordinatensystem ist die Rotation dann trivial.
- ▶ Es ist wichtig, die Übersicht zu behalten, in welchen Koordinatensystemen die jeweiligen Koordinaten und Transformationen gelten.

Zur Notation:

- ▶ Koordinatensysteme werden mit großen Buchstaben bezeichnet: A, B, \dots
- ▶ Falls nötig, wird der Name des Koordinatensystems links oben an Koordinaten angehängt: ${}^A P$ ist ein Punkt, dessen Koordinaten im Koordinatensystem A angegeben sind.
- ▶ Bei Transformationen zwischen Koordinatensystemen steht links unten der Index des Quellkoordinatensystems und links oben der Index des Zielkoordinatensystems.
 ${}^A_B R$ rotiert Punkte von System B nach System A ,
 ${}^A_B T$ ist die Translation von System B nach System A .

Transformation zwischen Koordinatensystemen



$$[{}^A P | 1] = [{}^B P | 1] \cdot \left[\begin{array}{c|c} {}^A R_B & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} \\ \hline {}^A B_{ORG} & 1 \end{array} \right]$$

Beispiel

$${}^A_B T = \left[\begin{array}{ccc|c} 0.866 & 0.500 & 0.000 & 0.000 \\ -0.0500 & 0.866 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ \hline 10.000 & 0.000 & 5.000 & 1.000 \end{array} \right]$$

$${}^B P = [3.0 \ 3.0 \ 7.0 | 1.0]$$

$${}^A P = {}^B P \cdot {}^A_B T = [11.098 \ 4.098 \ 12.0 | 1.0]$$

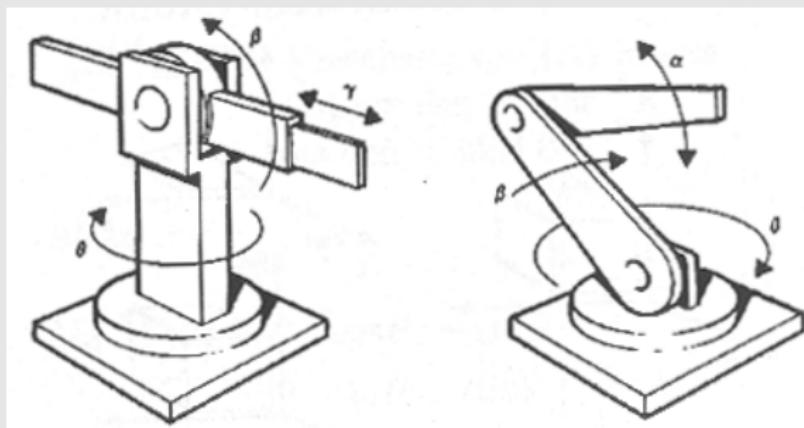
Orientierungen (Frames)

- ▶ Eine Transformation ${}^A_B T$ erfüllt einen doppelten Zweck:
- ▶ Sie beschreibt, wie Koordinaten von System B in Koordinaten des Systems A gewandelt werden.
- ▶ Sie beschreibt die Lage und Orientierung von System B relativ zu System A .
- ▶ In der Computergrafik werden solche Transformationen daher auch für beide Aufgaben eingesetzt,
 - ▶ zur Beschreibung der Lage und Orientierung von Objekten
 - ▶ und zur Umrechnung der Koordinaten.

Orientierungen in der Robotik

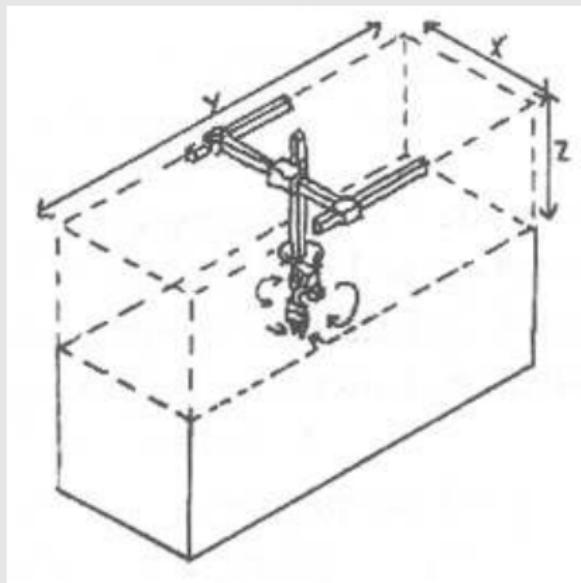
- ▶ Auch in der Robotik spielen Koordinatentransformationen eine große Rolle, da Handhabungsgeräte aus gegeneinander beweglichen Gliedern aufgebaut sind (Translationsgelenke, Rotationsgelenke).
- ▶ In der Robotik heißt ein Paar aus Translation und Rotation *frame* und wird zur Beschreibung der Lage der Glieder eines Roboters eingesetzt.

Bauformen von Robotern

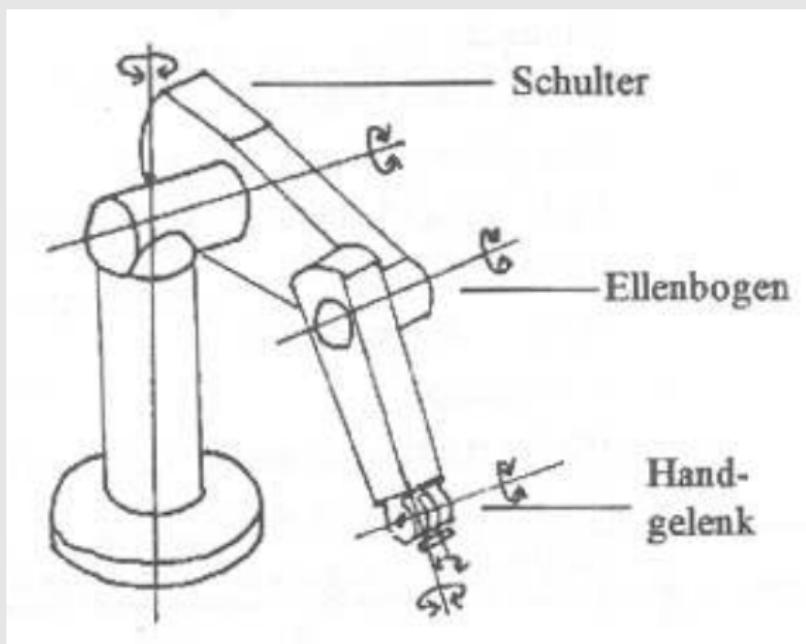


Translationsgelenke und Rotationsgelenke

Ein Roboter mit Translationsgelenken:



Ein Roboter mit Rotationsgelenken:

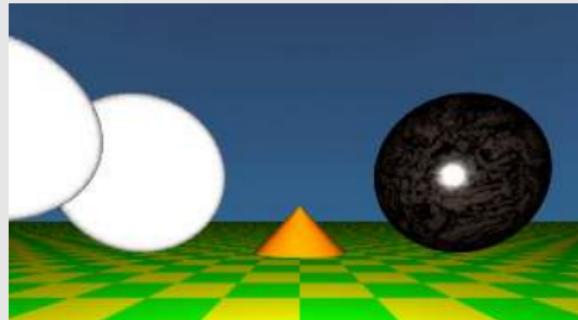


Ein virtueller Schneemann

Beispiel

Wir konstruieren einen Schneemann aus gegebenen Grundobjekten, die wir durch geeignete Transformationen zusammenfügen.

Das Baumaterial: Weiße und schwarze Kugeln,
ein oranger Kegel



Deklaration der Basisobjekte

```
#include "snowman.inc"
object { snowball // Baustelle
    scale 50 /*x, y, z um Faktor 50 skaliert */
    translate <-100, 50, 0>}
object { snowball
    scale 30
    translate <-100, 50, -100>}
object { coal
    scale 50
    translate <100, 50, 0>}
object { carrot scale 30}
```

Konstruktion des Schneemanns:

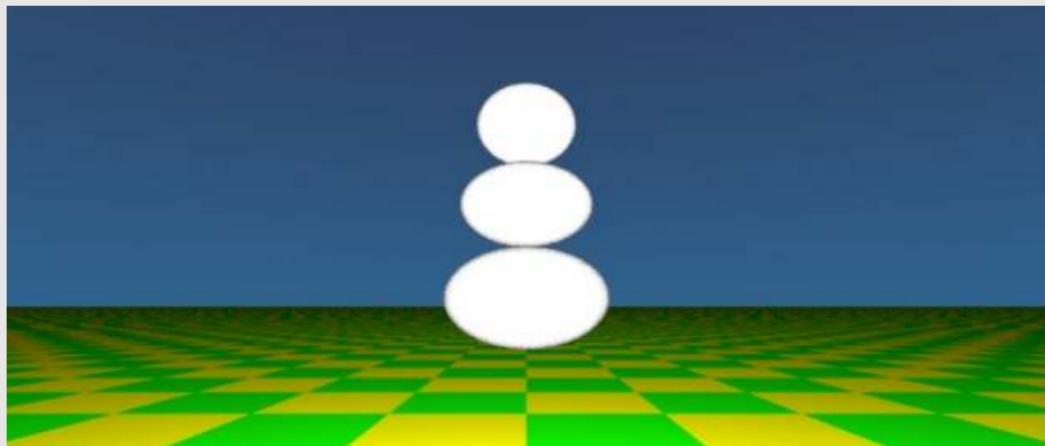
Der Körper: Der Körper wird aus passend skalierten weißen Kugeln zusammengesetzt, die durch Translation übereinander gestapelt werden.

Die Nase: Für die **Nase** wird der orange Kegel im Ursprung skaliert, um die x-Achse in die horizontale Lage rotiert und durch Translation zur Gesichtsmitte gebracht.

Die Augen: Die Augen sind skalierte schwarze Kugeln. Sie werden durch Translation zur Gesichtsmitte gebracht und dann um die y-Achse nach links oder rechts rotiert.

Die Knöpfe: Rotation im Brustmittelpunkt um die X-Achse.

Kopf, Brust und Bauch aus skalierten Kugeln



Konstanten für relative Abmessungen

```
#declare snowmanheight=120;  
  
#declare bellyscalex=0.25*snowmanheight;  
#declare bellyscaley=0.2*snowmanheight;  
#declare bellyscalez=bellyscalex;  
  
#declare breastscalex=0.2*snowmanheight;  
#declare breastscaley=0.16*snowmanheight;  
#declare breastscalez=breastscalex;  
  
#declare headscale=0.15*snowmanheight;
```

```
object { snowball /* belly */  
  scale <bellyscalex , bellyscaley ,  
          bellyscalez>  
  translate <0,bellyscaley , 0>}  
object { snowball /* breast */  
  scale <breastscalex , breastscaley ,  
          breastscalez>  
  translate <0,2*bellyscaley+breastscaley , 0>}  
object { snowball /* head */  
  scale headscale  
  translate  
  <0,2*bellyscaley +2*breastscaley+headscale ,0>}
```

Das Gesicht und die Knöpfe



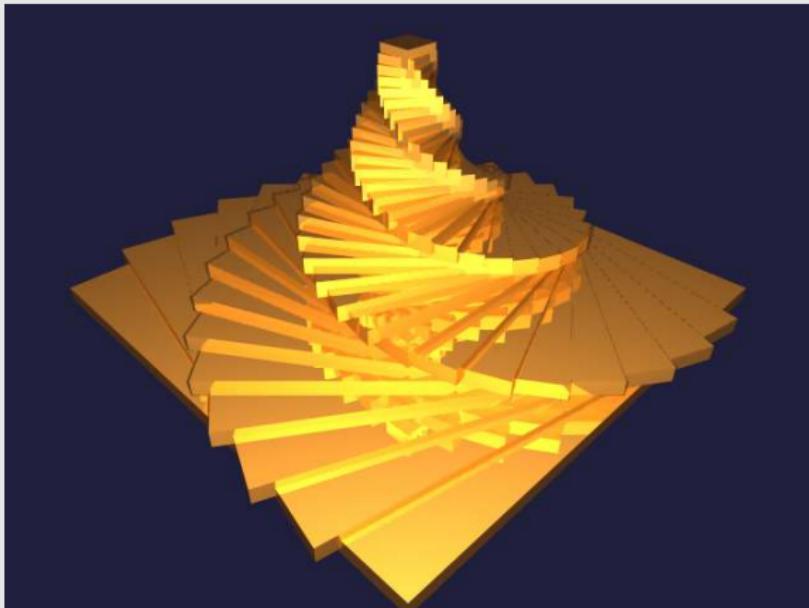
Mit Hut und Schal



Ein Schneemann

Linkshaendig

Linear rekursive Objekte



Eine Pyramide, konstruiert mittels 3-D-Transformationen

Beispiel

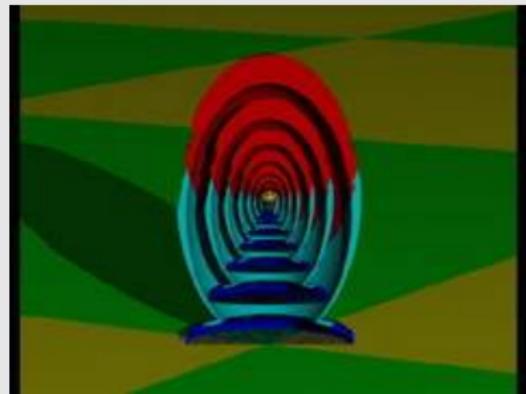
Die Pyramide wurde aus einer Folge von Quadern mit quadratischer Grundfläche modelliert, wobei jeder Quader gegenüber dem vorherigen

- ▶ um einen konstanten Faktor verkleinert,
- ▶ um einen konstanten Winkel rotiert
- ▶ und um eine konstante Distanz in der Höhe verschoben wurde.

Eine linear rekursive Pyramide

Linkshaendig

Eine Schachtelpuppe, aufgeschnitten



Eine Schachtelpuppe

Matruschka

Eine Schachtelpuppe, rekursiv geöffnet



Eine Schachtelpuppe

Matruschka

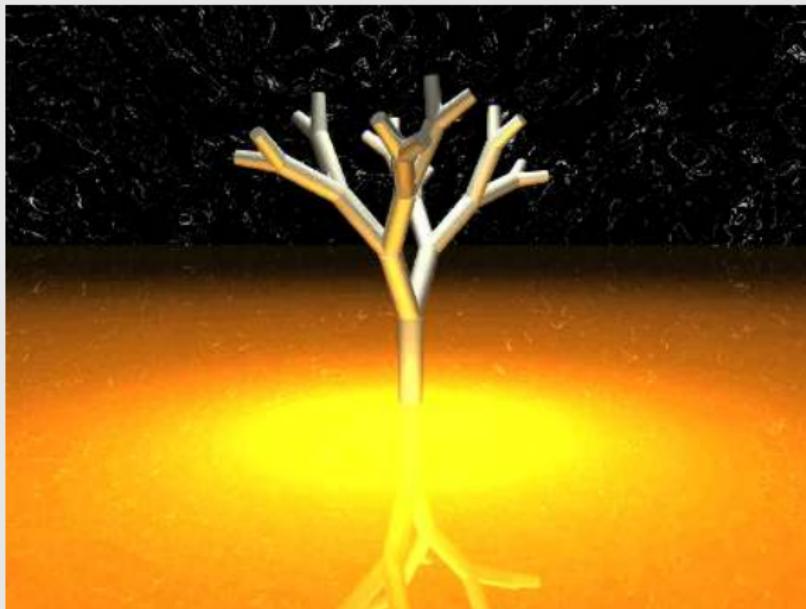
Ein Ammonit



Ein Ammonit

Lineare Rekursion: Ammonit

Eine baumrekursive Struktur



Baumrekursive Objekte

Baumrekursive Skalierung

Geometrisches Modellieren

Einführung

- 7 Einführung: Zweck und Anwendungen des geometrischen Modellierens
- 8 Repräsentation geometrischer Modelle

Zweck und Anwendungen des geometrischen Modellierens



- 7 Einführung: Zweck und Anwendungen des geometrischen Modellierens
- 8 Repräsentation geometrischer Modelle

Zweck des geometrischen Modellierens

Das geometrische Modellieren ist ein Grundlagengebiet für diverse Natur- und Ingenieurwissenschaften; im Bereich des **CAD** (Computer Aided Design) dient es

- ▶ dem Entwurf und
- ▶ der (möglichst exakten) Beschreibung der Form von Gegenständen zum Zwecke der
 - ▶ **Analyse** für
 - ▶ Bauingenieure,
 - ▶ Maschinenbauingenieure,
 - ▶ Schiffbauingenieure,
 - ▶ Chemiker und Biologen,
 - ▶ Mediziner,
 - ▶ etc.

Repräsentation geometrischer Modelle

Es gibt zwei grundsätzlich unterschiedliche Methoden, geometrische Modelle zu repräsentieren:

- ▶ Volumenbasierte Repräsentation
- ▶ Begrenzungsbasierte Repräsentation

Sie haben sich aus den Ansprüchen und Erfahrungen unterschiedlicher Anwendungsgebiete heraus entwickelt und besitzen i.d.R. in Bezug auf das jeweilige Anwendungsgebiet ein hohes Maß an Intuition.

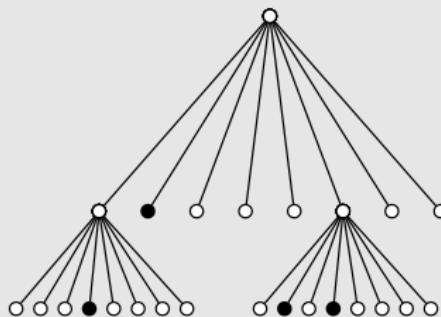
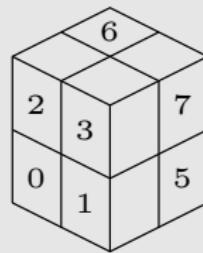
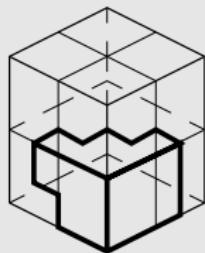
Volumenbasierte Repräsentationsformen

Octree

Prinzip: sukzessive Unterteilung der Objekthülle (Hüllquader) in “Achtelquader”, (bzw. Zusammenfassung von Voxelclustern)

Anwendungsgebiet: volumenorientierte Objektrekonstruktion (z.B. Tomographie)

Octree, Beispiel



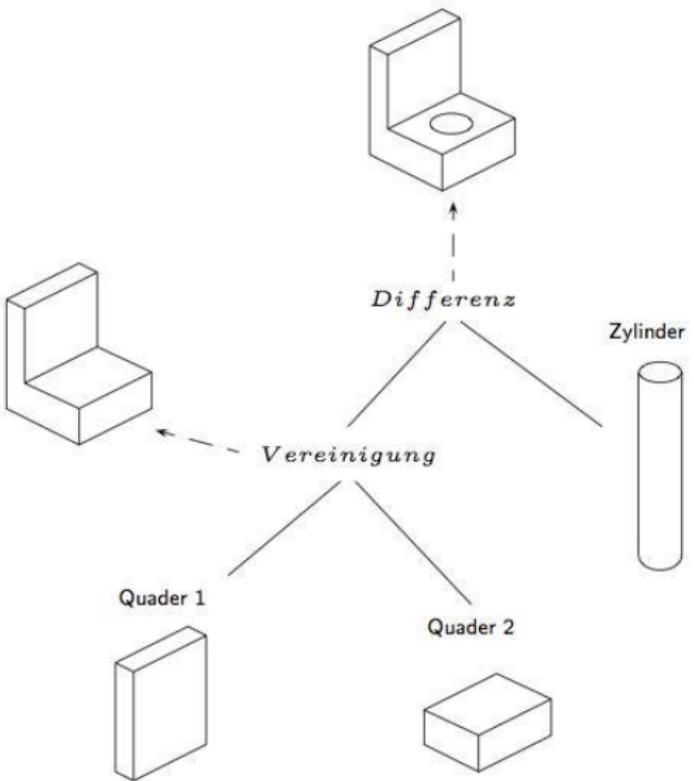
Constructive Solid Geometry (CSG)

Constructive Solid Geometry (CSG)

Prinzip: Verknüpfung “primitiver” Objekte (Quader, Zylinder, Kegel, Kugel, etc.) durch Mengenoperationen

Anwendungsgebiet: volumenorientierte Objektrekonstruktion (z.B. Tomographie)

CSG, Beispiel



- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

- ▶
- ▶
- ▶

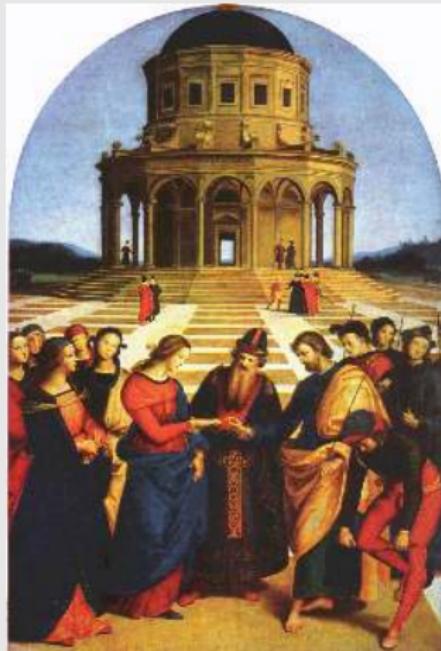
- ▶
- ▶
- ▶

to be added again

Teil IV

Geometrische Grundlagen: Abbildung des Raumes

Projektionen



9

Projektionen

- Grundbegriffe
- Lochkamera

10

Parallelprojektion

11

Zentralprojektion

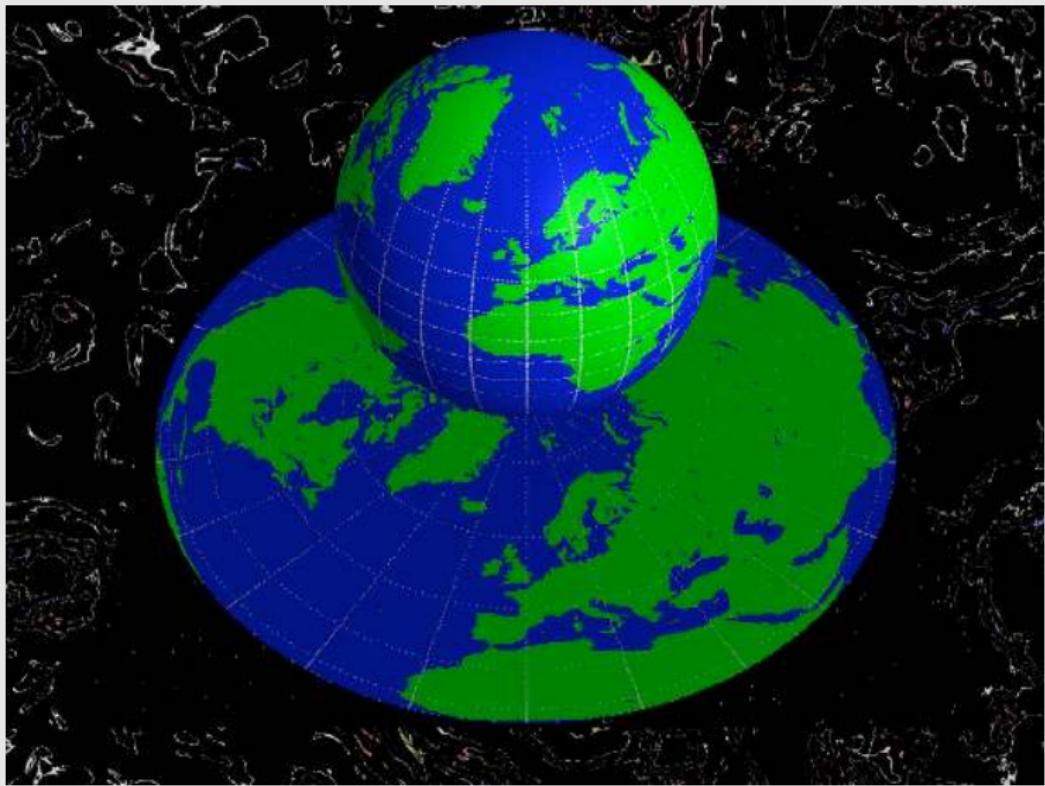
Rafael von Urbino

Definition (Projektion)

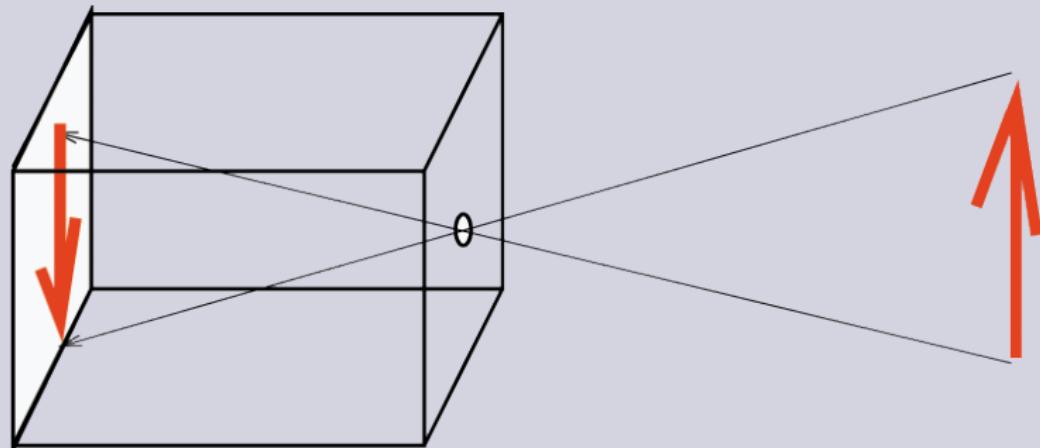
Eine Projektion ist die Abbildung eines Vektors \vec{x} aus einem n -dimensionalen Vektorraum X auf einen Vektor \vec{u} in einem m -dimensionalen Unterraum U für $m < n$.

- ▶ Wir werden hier nur Abbildungen des 3D-Raumes auf eine Ebene betrachten, d.h. nur **planare** Projektionen mit $n = 3, m = 2$.
- ▶ Nicht-planare Projektionen werden häufig in der Kartographie eingesetzt: Zylinderprojektion, Kegelprojektion usw.

Planare und nicht-planare Projektionen



Die Abbildung mittels einer Lochkamera

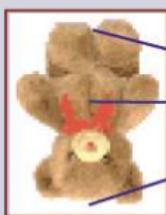


- ▶ Das Bild entsteht im Schnittpunkt der Projektionsstrahlen mit der Bildebene.
- ▶ Das Bild steht auf dem Kopf.

Abstraktion vom Gehäuse

Nur ein Loch und eine Bildebene

Bild

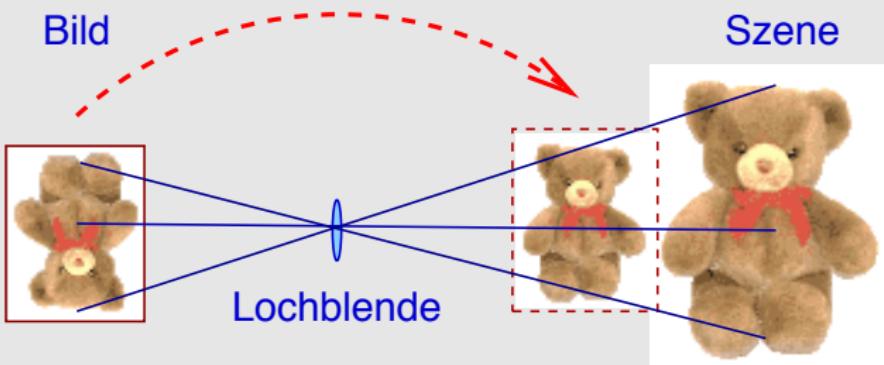


Szene



Lochblende

Gleichwertiges Modell: Zentralprojektion



- ▶ Die Bildebene wird um 180° um das Loch rotiert.
- ▶ Die Schnittpunkte der Projektionsstrahlen mit der Bildebene ergeben jetzt ein Bild, das aufrecht steht.

Definition (Zentralprojektion)

Eine **Zentralprojektion** wird festgelegt durch die Angabe

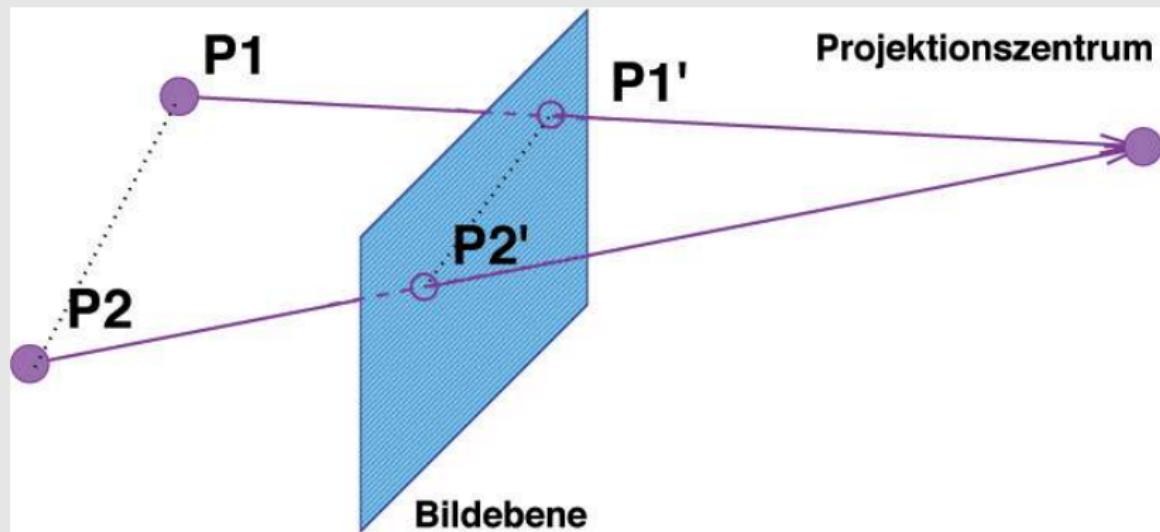
- ▶ eines **Projektionszentrums (Augpunkt)**
- ▶ und einer **Projektionsebene**.

Projektionsstrahlen sind Geraden durch den Augpunkt und die abzubildenden Punkte.

Das Abbild des Punktes ist der Schnittpunkt seines Projektionsstrahls mit der Bildebene (die Projektion).

Im Grenzfall, wenn das Projektionszentrum unendlich weit von der Bildebene entfernt ist, ergibt sich eine **Parallelprojektion**.

Zentralprojektion



Parallelprojektion



Orthogonale
Projektionen

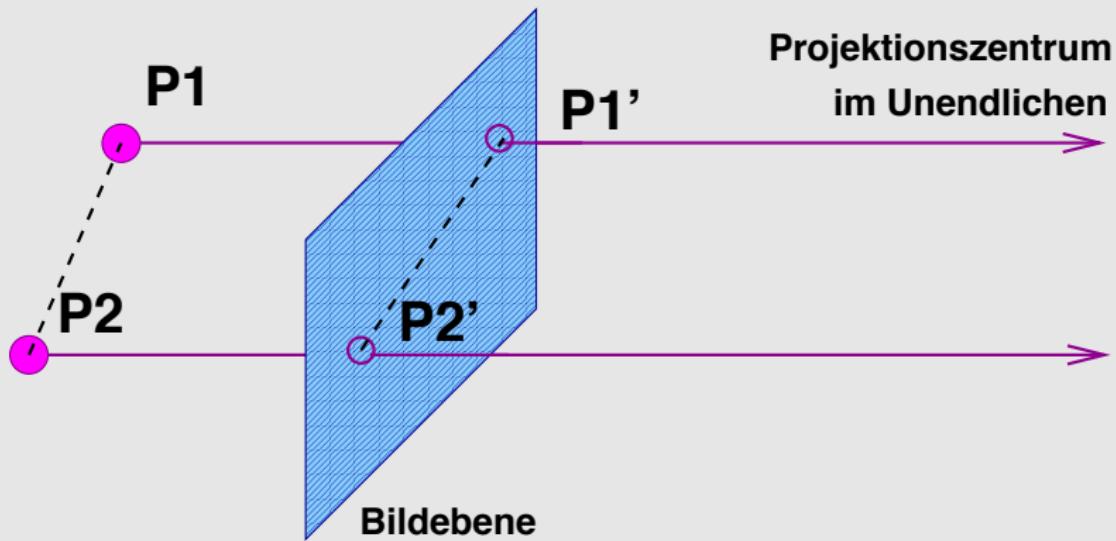
9 Projektionen

10 Parallelprojektion

- Orthogonale Projektion
- Schiefe Parallelprojektion

11 Zentralprojektion

Parallelprojektion



Parallelprojektion

Parallelprojektion

- ▶ Die Punkte des Objekts werden entlang paralleler Strahlen auf die Projektionsebene abgebildet.
- ▶ Parallele Linien bleiben parallel.
- ▶ Wird die Bildebene parallel zu den Projektionsstrahlen verschoben, bleibt das Bild gleich.

Typen von Parallelprojektionen

Die Parallelprojektionen werden in Abhangigkeit vom Winkel, unter dem die Projektionsstrahlen die Bildebene schneiden, in zwei Gruppen unterteilt:

- ▶ Orthogonale Projektion
- ▶ Schiefe Parallelprojektion

Definition: Orthogonale Projektion

Definition (Orthogonale Projektion)

Eine Parallelprojektion ist eine **orthogonale Projektion** (engl. orthographic projection), auch **Normalprojektion** genannt, wenn die Projektionsstrahlen senkrecht zur Bildebene stehen.

Definition

Schiefe Parallelprojektion, Schrägriß

Definition (Schiefe Parallelprojektion, Schrägriß)

Eine Parallelprojektion ist eine **schiefe Parallelprojektion** (engl. oblique projection), wenn sie keine Normalprojektion ist, also wenn die Projektionsstrahlen schräg auf die Bildebene treffen.

Orthogonale Projektion

- ▶ Orthogonale Projektionen sind im Ingenieurbereich beliebt, da in der Ebene parallel zur Bildebene alle Winkel und Längen erhalten bleiben.
- ▶ Parallelprojektionen auf die Grundebenen des Koordinatensystems heißen Risse.

Standardrisse

Wenn das Weltkoordinatensystem so festgelegt wird, daß die Achsen parallel zu wichtigen Flächen und Symmetriearchsen der abgebildeten Objekte sind, dann haben die Risse eine bestimmte Bedeutung und festgelegte Namen:

Grundriß: Projektion in die xz -Ebene.

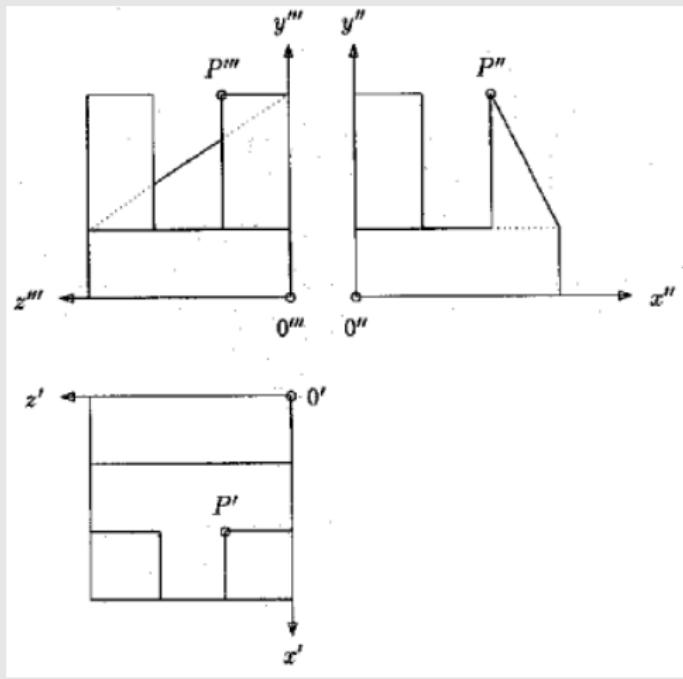
Aufriß: Projektion in die xy -Ebene.

Seitenriß, Kreuzriß: Projektion in die yz -Ebene.

Die Interpretation von Rissen

- ▶ Die Interpretation von Rissen erfordert viel Übung.
- ▶ Meist sind mehrere Ansichten nötig.
- ▶ Nicht jedes Objekt kann eindeutig aus den drei Rissen rekonstruiert werden.

Aufriß, Seitenriß und Grundriß eines Objektes.



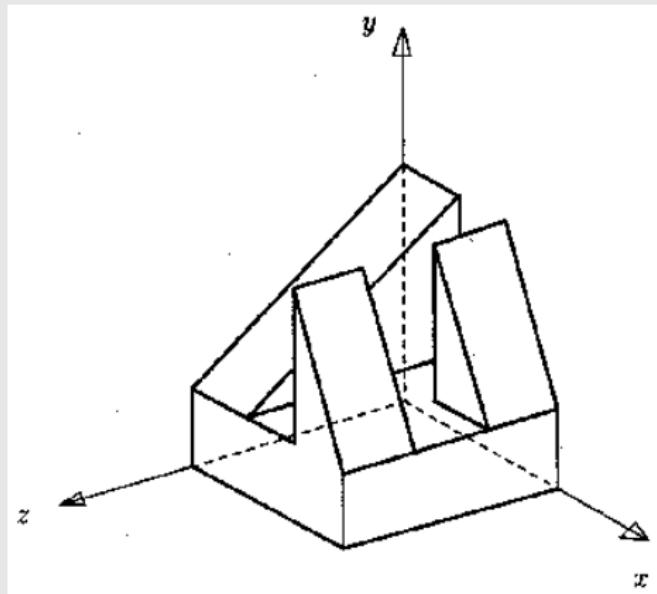
Definition (Normalaxonometrische Projektion)

Eine orthogonale Projektion heißt **Normalaxonometrische Projektion**, wenn keine Achse des Weltkoordinatensystems senkrecht zur Bildebene ist.

- ▶ Zueinander senkrechte Objektflächen können gleichzeitig dargestellt werden, auch wenn diese parallel zu den Grundebenen des Weltkoordinatensystems sind.
- ▶ Parallel Kanten werden gleich verkürzt.
- ▶ Die Abbildung ist der Zentralprojektion von weit entfernten Objekten ähnlich.

Normalaxonometrischer Projektion

Das Objekt von Folie 230 in normalaxonometrischer Projektion:



Die normalaxonometrischen Projektionen werden danach unterschieden, wie die Bildebene die Achsen des Koordinatensystems schneidet.

Isometrische Projektion: Die Schnittpunkte der Bildebene mit den Hauptachsen sind alle gleich weit vom Ursprung entfernt: alle Koordinaten werden gleich verkürzt.

Dimetrische Projektion: Zwei Koordinatenachsen werden in der gleichen Entfernung vom Ursprung geschnitten.

Trimetrische Projektion: Alle Koordinatenachsen werden in unterschiedlicher Entfernung vom Ursprung von der Bildebene geschnitten, d.h. alle Koordinaten werden paarweise ungleich verkürzt.

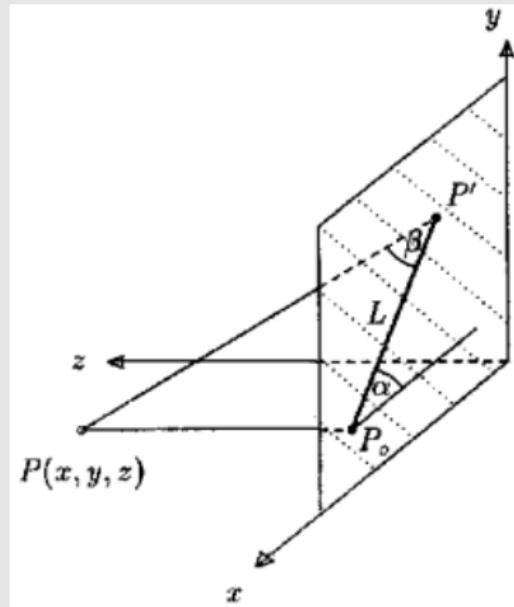
Transformationsgleichungen für die orthogonale Projektion

Die Transformation erfolgt in zwei Schritten:

- 1 Transformation des Weltkoordinatensystems:
 - Translation, so daß die Bildebene im Ursprung liegt.
 - Rotation, so daß die Normale der Bildebene parallel zur z-Achse ist.
- 2 Die Bildkoordinaten sind dann gleich den Koordinaten im transformierten System, abgesehen von der z-Koordinate, die gleich 0 ist.

Schiefe Parallelprojektion, Schrägriß

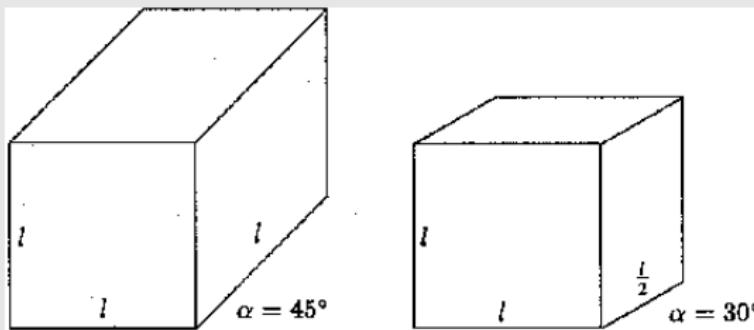
Die Projektionsstrahlen stehen **nicht senkrecht** auf der Bildebene, Angabe durch α und β .

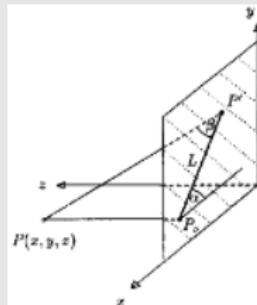


Einige schiefen Parallelprojektionen:

Kavalierprojektion: $\tan \beta = 1$

Kabinettprojektion: $\tan \beta = \frac{1}{2}$





Sei $P'(x', y', 0)$ die Projektion des Punktes $P(x, y, z)$:

$$\tan(\beta) = \frac{z}{L}, \quad L = \overline{P_0P'} = \frac{z}{\tan(\beta)}$$

$$x' = x - L \cdot \cos(\alpha) = x - z \cdot \frac{\cos(\alpha)}{\tan(\beta)}$$

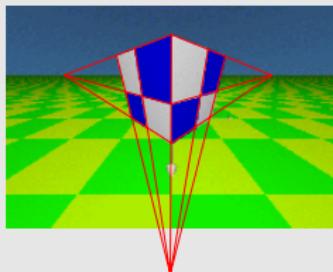
$$y' = y + L \cdot \sin(\alpha) = y + z \cdot \frac{\sin(\alpha)}{\tan(\beta)}$$

$$x' = x - L \cdot \cos(\alpha) = x - z \cdot \frac{\cos(\alpha)}{\tan(\beta)}$$

$$y' = y + L \cdot \sin(\alpha) = y + z \cdot \frac{\sin(\alpha)}{\tan(\beta)}$$

$$\begin{bmatrix} x' & y' & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\cos(\alpha)}{\tan(\beta)} & \frac{\sin(\alpha)}{\tan(\beta)} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{P_{\text{schief}_{xy}}}$$

Zentralprojektion



Ein Würfel in
Dreipunktprojek-
tion

9 Projektionen

10 Parallelprojektion

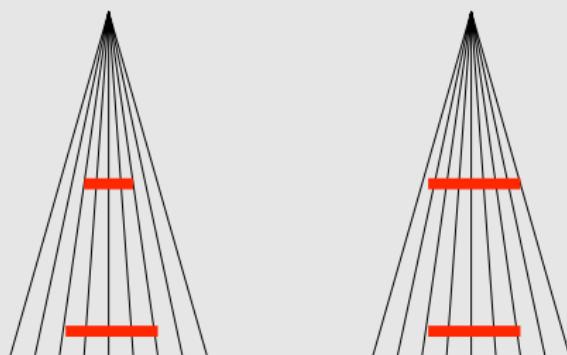
11 Zentralprojektion

- Eigenschaften der Zentralprojektion
- Projektionsgleichungen

Eigenschaften der Zentralprojektion

- ▶ Die Zentralprojektion entspricht der Abbildung durch eine Sammellinse (Kameras, unser Auge) oder eine Lochkamera.
- ▶ Sie vermittelt durch die perspektivische Verkürzung einen räumlichen Eindruck.
- ▶ Die Abbildungen paralleler Geraden enden in einem **Fluchtpunkt**.
- ▶ Winkel bleiben nur dann erhalten, wenn die Geraden parallel zur Bildebene sind.

Zur perspektivischen Verkürzung



- ▶ Ein entferntes Objekt muß größer sein als ein nahes Objekt, um denselben Raumwinkel einzunehmen.
- ▶ Bei gleicher Größe erscheint das nahe Objekt größer als ein weiter entferntes Objekt.



Fluchtpunkte: Sommerdeich Hallig Hooge



Fluchtpunkte

Avenida de 9. Julio (Buenos Aires)



Projektionsgleichungen

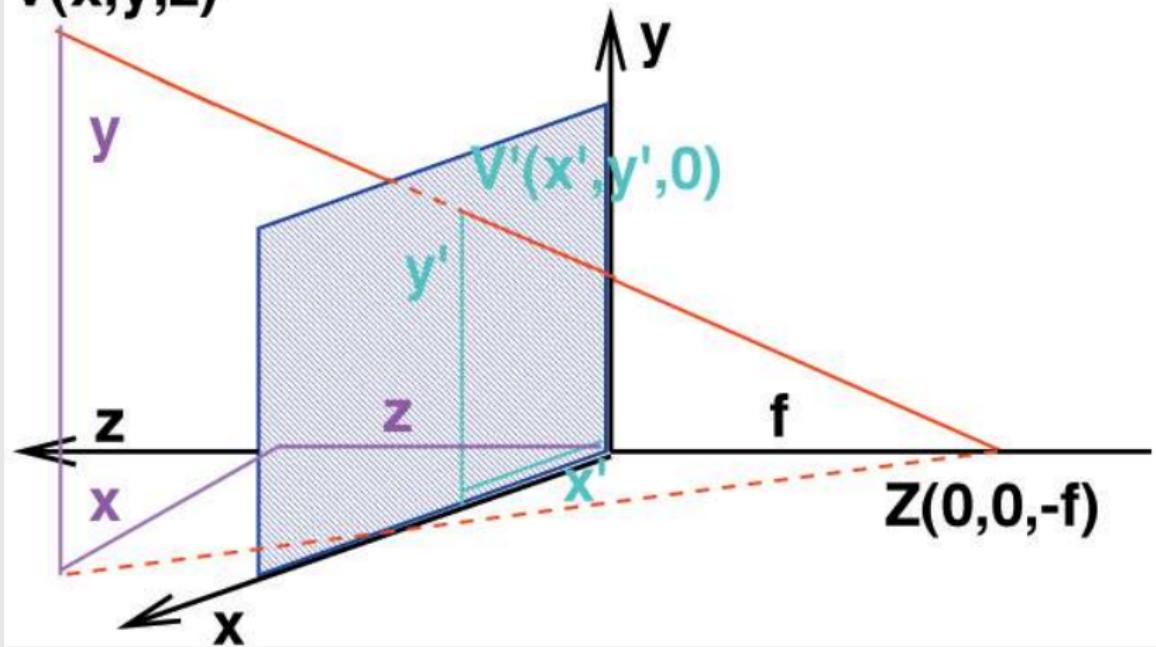
Wahl des Koordinatensystems

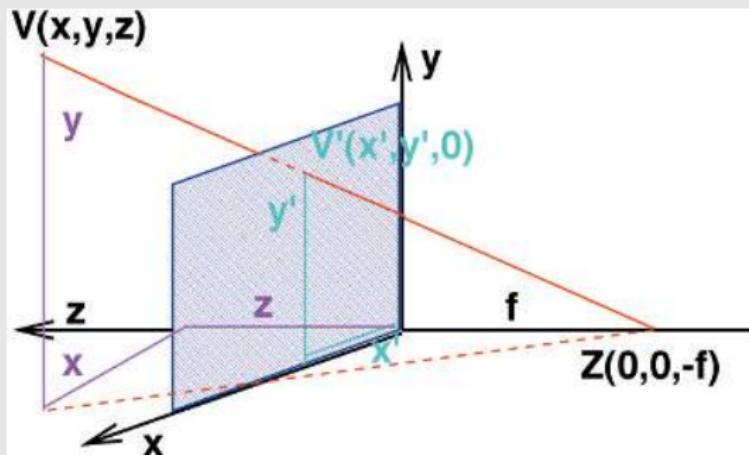
Sei f der Abstand des Projektionszentrums von der Bildebene. Wir legen fest:

- ▶ Der Ursprung des Koordinatensystems liegt in der Bildebene.
- ▶ Die z-Achse steht senkrecht auf der Bildebene, die x-Achse und y-Achse liegen in der Bildebene.
- ▶ Das Koordinatensystem ist rechtshändig.
- ▶ Das Projektionszentrum Z liegt im Punkt $(0, 0, -f)$.

Herleitung über ähnliche Dreiecke

$V(x,y,z)$





$$\frac{x'}{f} = \frac{x}{z+f} \quad \text{Strahlensatz}$$

$$x' = \frac{f \cdot x}{z+f} = \frac{x}{\frac{z}{f} + 1}$$

$$x' = \frac{f \cdot x}{z + f} = \frac{x}{\frac{z}{f} + 1}$$

$$\text{Analog: } y' = \frac{f \cdot y}{z + f} = \frac{y}{\frac{z}{f} + 1}$$

$z' = 0$ in der Bildebene

$$\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 0 & \frac{z}{f} + 1 \end{bmatrix}$$
$$= \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ▶ Alle x und y-Koordinaten werden bei der Projektion ins Bild **proportional** zur Entfernung z verkürzt.
- ▶ Die Verkürzung ist umso stärker, je kleiner f ist.

Die inverse Zentralprojektion

$$P_Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{nicht invertierbar, Rang 3}$$

$$P_{IZ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_{IZ}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- ▶ Alle Punkte entlang eines Projektionsstrahls haben denselben Bildpunkt.
- ▶ Das Ergebnis der Rücktransformation vom Bild in den 3D-Raum sollte daher kein einzelner Punkt sondern eine Geradengleichung sein, die den Projektionsstrahl beschreibt.
- ▶ Wir wählen die z' -Koordinate eines Bildpunktes als freie Variable. Die inverse Zentralprojektion liefert dann eine Geradengleichung für x, y, z in Abhängigkeit von z' ,
→ [Duda and Hart, 1973, S. 384].

$$V = [x', y', z', 1] \cdot P_{IZ}^{-1}$$

Perspektivische Effekte



Raffael von
Urbino:
Vermählung
Mariä

12

Gesetze der Perspektive: Fluchtpunkte

- Projektionsgleichung für Weltkoordinaten
- Hauptfluchtpunkte
- Perspektivische Elemente in der Malerei

13

Illusionen

14

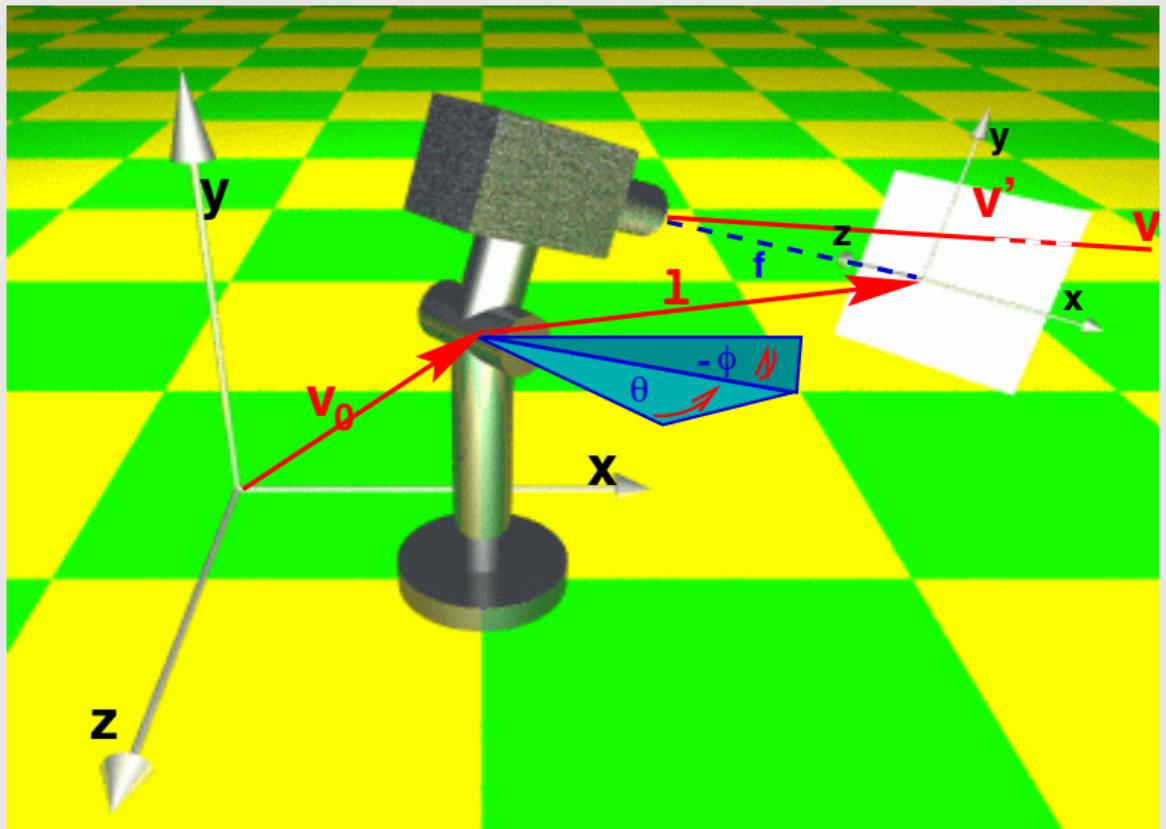
Virtuelle und reale Kameras

15

Höherdimensionale Räume

Weltkoordinatensystem und Kamerakoordinatensystem

- ▶ Es ist unpraktisch, eine Szene im Kamerakoordinatensystem zu modellieren. Meist wird daher ein Weltkoordinatensystem verwendet.
- ▶ **Zur Notation:** Wir werden Koordinaten im Kamerakoordinatensystem mit einem hochgestellten K bezeichnen, Koordinaten im Weltkoordinatensystem mit einem W .
- ▶ Man unterscheidet Klassen von Zentralprojektionen danach, wie das Kamerakoordinatensystem K relativ zum Weltkoordinatensystem W liegt, siehe Hauptfluchtpunkte.



Projektionsgleichung für Weltkoordinaten

Translation zum Drehpunkt des Dreh-Neige-Kopfs
(gimbal center) des Stativs ${}^G_W T = T(-{}^W \vec{v}_0)$.

Rotation im Dreh-Neige-Kopf ${}^{G2}_G R$: Drehung um die
y-Achse um den Winkel θ und Neigung um
die z-Achse um den Winkel φ
 ${}^{G2}_G R = R({}^G \hat{z}, \varphi) \cdot R({}^G \hat{y}, \theta)$.

Translation vom Drehpunkt zum Ursprung des
Kamerakoordinatensystems
 ${}^{K2}_{G2} T = T(-{}^G \vec{l} - f \cdot {}^G \hat{z})$.

Zentralprojektion in die Bildebene: ${}_K P_Z$.

Projektionsgleichung in homogenen Koordinaten

In homogenen Koordinaten können wir die Transformationen als Matrizen zusammenfassen:
Der projizierte Punkt ${}^K V'$ des Punktes ${}^W V$ ist:

$${}^K V' = {}^W V \cdot {}_W^G T \cdot {}_G^{G2} R \cdot {}_{G2}^K T \cdot {}_K^K P_Z.$$

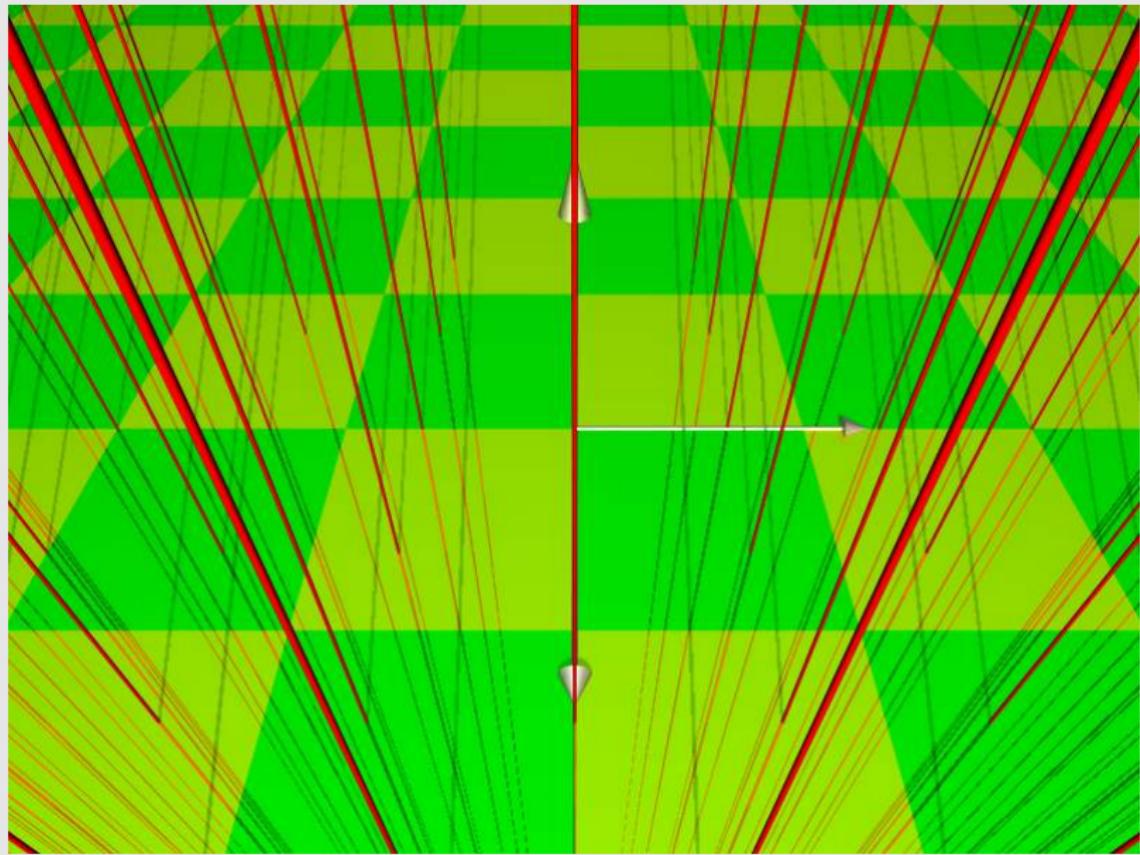
Und die inverse Zentralprojektion:

$${}^W V = {}^K V' \cdot {}_K^K P_{IZ}^{-1} \cdot {}_{G2}^K T^{-1} \cdot {}_G^{G2} R^{-1} \cdot {}_W^G T^{-1}.$$

Definition (**Hauptfluchtpunkt**)

Ein Hauptfluchtpunkt ist der Fluchtpunkt von Geraden, die parallel zu einer der Achsen des **Weltkoordinatensystems** sind.

- ▶ Es gibt **mindestens einen und höchstens drei** Hauptfluchtpunkte.
- ▶ Entsprechend der Anzahl der Hauptfluchtpunkte klassifiziert man die Zentralprojektionen:
 - Einpunktprojektion:** Ein Hauptfluchtpunkt.
 - Zweipunktprojektion:** Zwei Hauptfluchtpunkte.
 - Dreipunktprojektion:** Drei Hauptfluchtpunkte.



Der Fluchtpunkt von vertikalen Geraden

Vereinfachte Projektionsgleichung: Alle Parameter gleich Null, mit Ausnahme des Neigungswinkels φ und der Brennweite f .

$$\text{Gimbal center } \vec{v}_0 = \vec{0}$$

$$\text{Gimbal offset } \vec{l} = \vec{0}$$

$$\text{Neigungswinkel } \varphi = \text{beliebig}$$

$$\text{Brennweite } f = \text{beliebig}$$

$${}^K_w P = R(\hat{x}, \varphi) \cdot P_z$$

$$\begin{aligned}
 {}^K_w P &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) & 0 \\ 0 & -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) & \frac{\sin(\varphi)}{f} \\ 0 & -\sin(\varphi) & \cos(\varphi) & \frac{\cos(\varphi)}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^K V' &= {}^W \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot {}^K_w P
 \end{aligned}$$

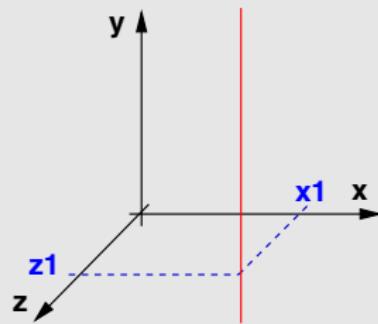
$$\kappa V' = \kappa \begin{bmatrix} x \\ y \cdot \cos(\varphi) - z \cdot \sin(\varphi) \\ 0 \\ \frac{y \cdot \sin(\varphi)}{f} + \frac{z \cdot \cos(\varphi)}{f} + 1 \end{bmatrix}^t$$

Division durch die homogene Koordinate w

$$x' = \frac{f \cdot x}{y \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f}$$

$$y' = \frac{f \cdot y \cdot \cos(\varphi) - f \cdot z \cdot \sin(\varphi)}{y \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f}$$

Eine vertikale Gerade durch den Punkt ${}^W(x_1, 0, z_1)$:



$${}^W [\begin{array}{cccc} x & y & z & 1 \end{array}] = {}^W [\begin{array}{cccc} x_1 & y & z_1 & 1 \end{array}]$$

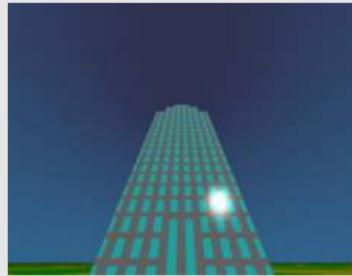
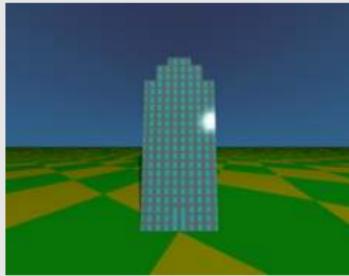
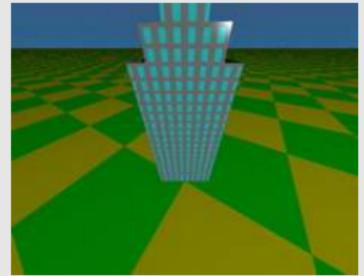
$$x' = \frac{f \cdot x_1}{y \cdot \sin(\varphi) + z_1 \cdot \cos(\varphi) + f}$$

$$y' = \frac{f \cdot y \cdot \cos(\varphi) - f \cdot z_1 \cdot \sin(\varphi)}{y \cdot \sin(\varphi) + z_1 \cdot \cos(\varphi) + f}$$

$$\lim_{y \rightarrow \infty} x' = 0$$

$$\lim_{y \rightarrow \infty} y' = f \cdot \frac{\cos(\varphi)}{\sin(\varphi)} = \frac{f}{\tan(\varphi)}$$

- ▶ Die Grenzwerte $\lim_{y \rightarrow \infty} x'$ und $\lim_{y \rightarrow \infty} y'$ sind unabhängig von x_1 und y_1 und damit für alle Vertikalen gleich!
- ▶ Der Punkt $(0, \frac{f}{\tan(\varphi)})$ ist der **Hauptfluchtpunkt** der Parallelen zur y-Achse!

 $\varphi = -50^\circ$  $\varphi = 0^\circ$  $\varphi = 50^\circ$

Fluchtpunkt vertikaler Geraden

Fluchtpunkt vertikaler Geraden

Der Fluchtpunkt von horizontalen Geraden

Vereinfachte Projektionsgleichung: Alle Parameter gleich Null, mit Ausnahme des Neigungswinkels φ , der Brennweite f und **der Höhe über der Grundebene**.

Gimbal center $\vec{v}_0 = (0, y_0, 0)$

Gimbal offset $\vec{l} = \vec{0}$

Neigungswinkel $\varphi = \text{beliebig}$

Brennweite $f = \text{beliebig}$

$$x' = \frac{f \cdot x}{(y - y_0) \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f}$$

$$y' = \frac{f \cdot (y - y_0) \cdot \cos(\varphi) - f \cdot z \cdot \sin(\varphi)}{(y - y_0) \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f}$$

Geraden in der xz-Ebene, Steigung m , z-Achsenabschnitt c

$${}^W \begin{bmatrix} x & y & z \end{bmatrix} = {}^W \begin{bmatrix} x & 0 & m \cdot x + c \end{bmatrix}$$

$$x' = \frac{f \cdot x}{-y_0 \cdot \sin(\varphi) + (m \cdot x + c) \cdot \cos(\varphi) + f}$$

$$y' = \frac{f \cdot y_0 \cdot \cos(\varphi) - f \cdot (m \cdot x + c) \cdot \sin(\varphi)}{y_0 \cdot \sin(\varphi) + (m \cdot x + c) \cdot \cos(\varphi) + f}$$

$$\lim_{x \rightarrow \infty} x' = \frac{f}{m \cdot \cos \varphi} \quad \text{abhängig von } m$$

$$\lim_{x \rightarrow \infty} y' = \frac{-f \cdot m \cdot \sin(\varphi)}{m \cdot \cos(\varphi)} = -f \cdot \tan(\varphi)$$

- Fluchtpunkt: $(\frac{f}{m \cdot \cos \varphi}, -f \cdot \tan(\varphi))$

Daraus folgt für den Fluchtpunkt von horizontalen Geraden:

- ▶ Die Bilder von Geraden in der xz-Grundebene haben den gemeinsamen Fluchtpunkt $(\frac{f}{m \cdot \cos \varphi}, -f \cdot \tan(\varphi))$ gdw. sie parallel sind.
- ▶ Alle Fluchtpunkte der Geraden in der Grundebene liegen im Bild auf der Geraden $y' = -f \cdot \tan(\varphi)$). Diese Gerade heißt der **Horizont**.
- ▶ Der Fluchtpunkt liegt im Unendlichen, wenn
 - ▶ die Geraden parallel zur x-Achse sind ($m = 0$) oder
 - ▶ die Kamera senkrecht zur Grundebene steht ($|\varphi| = \pi/2$).

Konjugierte Fluchtpunkte

Satz

Seien $g_1 \perp g_2$ Geraden in der Grundebene mit den Steigungen m_1 und $m_2 = -\frac{1}{m_1}$ und den Fluchtpunkten $(x'_1, -f \cdot \tan(\varphi)), (x'_2, -f \cdot \tan(\varphi))$. Dann gilt:

$$m_1 \cdot m_2 = -1$$

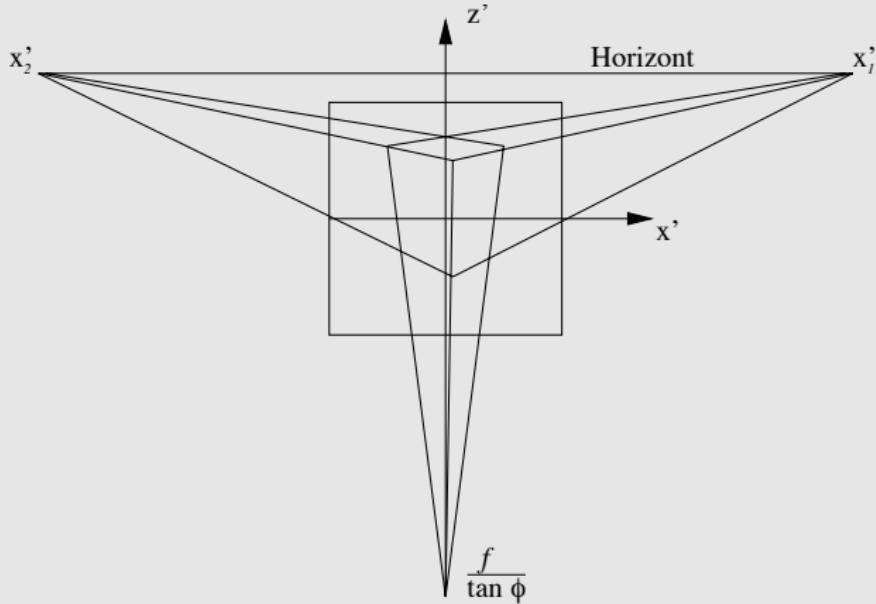
$$x'_1 \cdot x'_2 = \frac{f}{m_1 \cdot \cos \varphi} \cdot \frac{f}{m_2 \cdot \cos \varphi} = -\frac{f^2}{\cos \varphi^2}$$

Konjugierte Fluchtpunkte

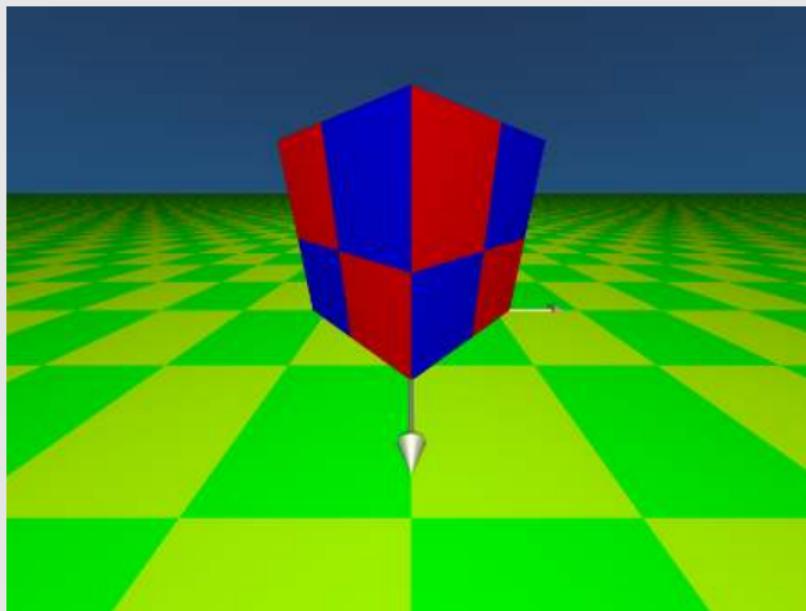
Definition (Konjugierte Fluchtpunkte)

Die Fluchtpunkte x_1 und x_2 der zueinander senkrechten Geraden $g_1 \perp g_2$ parallel zur Grundebene heißen konjugierte Fluchtpunkte.

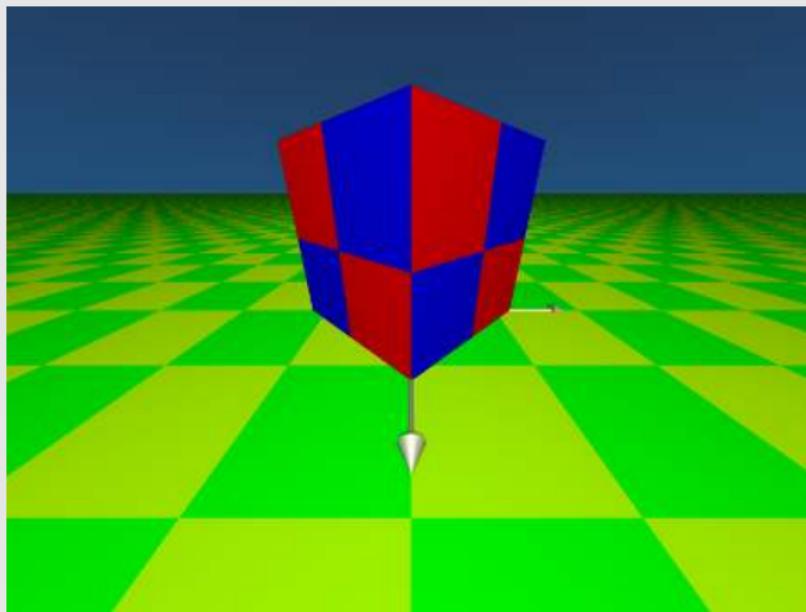
Konjugierte Fluchtpunkte eines Quaders



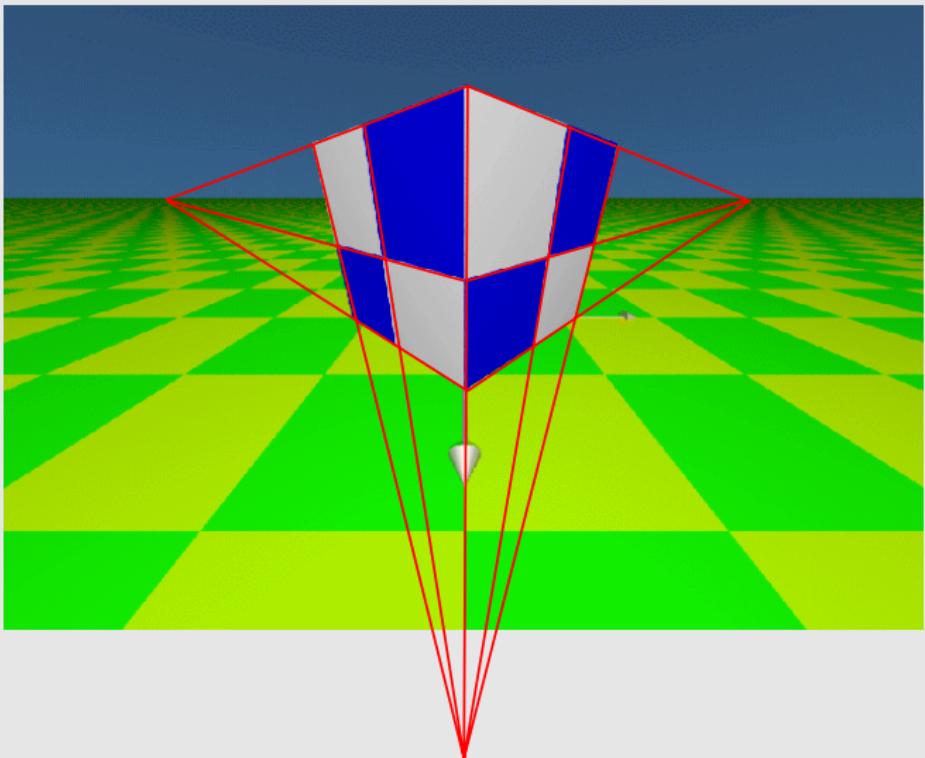
Zweipunktprojektion oder Dreipunktprojektion?



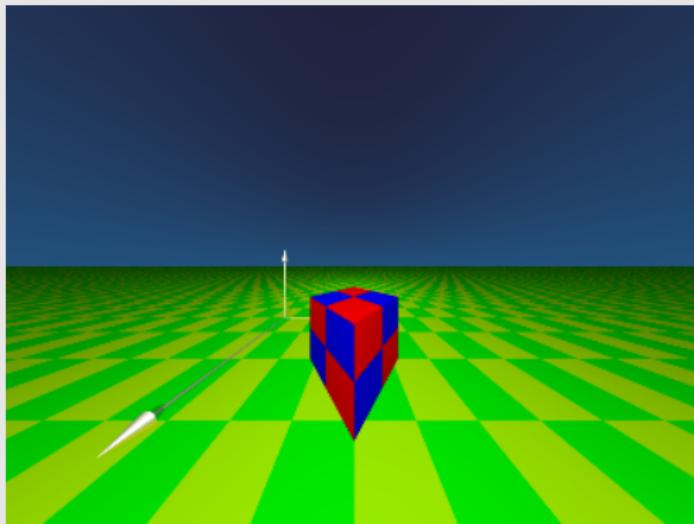
Zweipunktprojektion oder Dreipunktprojektion?



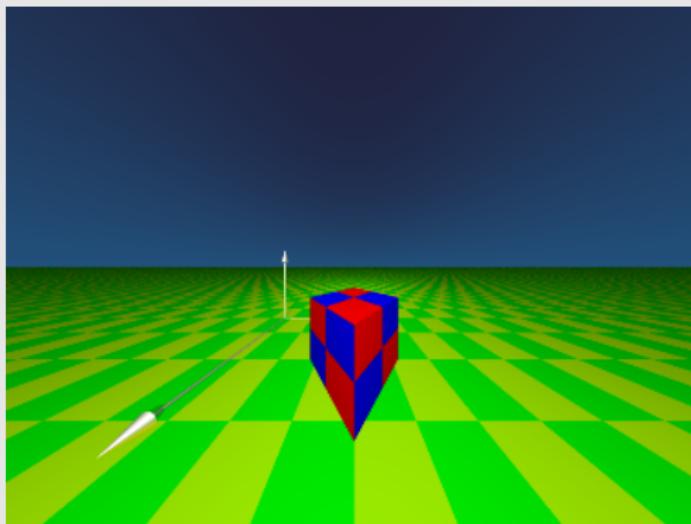
Zweipunktprojektion! Die Parallelen zur X-Achse bilden keinen Fluchtpunkt im Endlichen.



Einpunktprojektion oder Zweipunktprojektion?

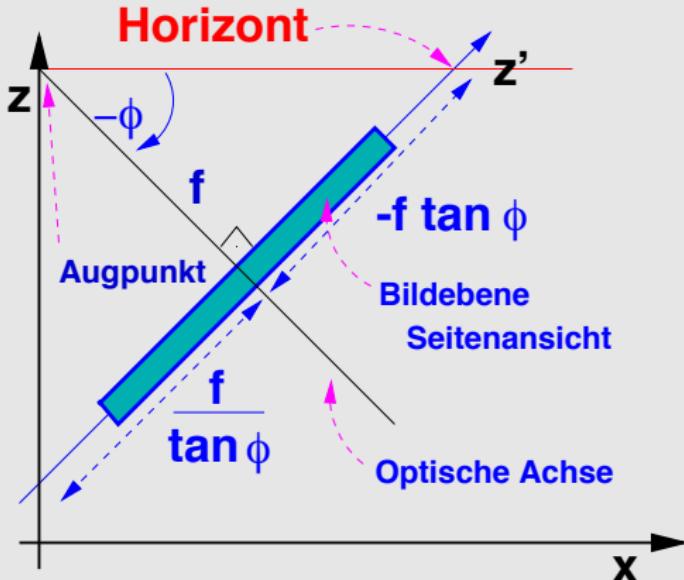


Einpunktprojektion oder Zweipunktprojektion?



Einpunktprojektion! Nur die Parallelen zur Z-Achse bilden einen Fluchtpunkt im Endlichen.

Der Horizont



Die Horizontlinie ist die Schnittlinie einer Ebene parallel zur Grundebene mit der Bildebene.

Gesetze der Perspektive: Fluchtpunkte
Illusionen
Virtuelle und reale Kameras
Höherdimensionale Räume

Projektionsgleichung für Weltkoordinaten
Hauptfluchtpunkte
Perspektivische Elemente in der Malerei

Teil V

Perspektivische Effekte

Perspektivische Elemente in der Malerei

12

Gesetze der Perspektive: Fluchtpunkte

- Projektionsgleichung für Weltkoordinaten
- Hauptfluchtpunkte
- Perspektivische Elemente in der Malerei

13

Illusionen

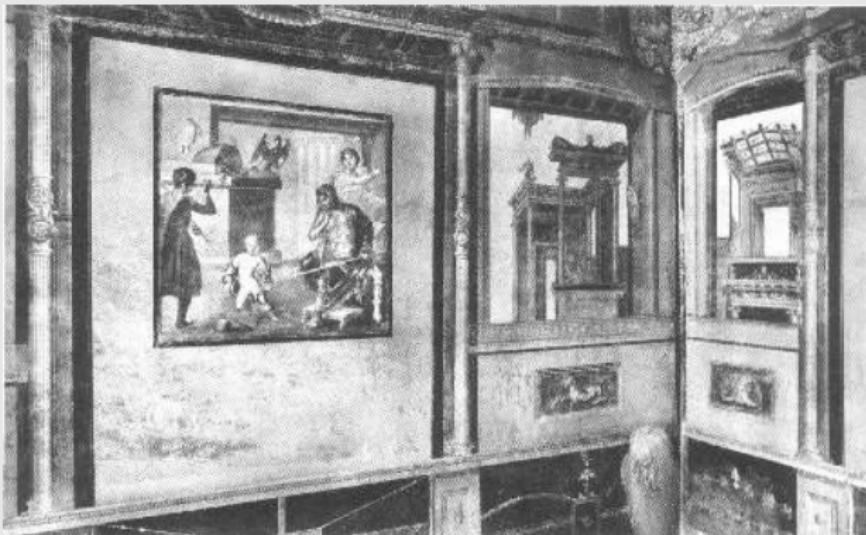
14

Virtuelle und reale Kameras

15

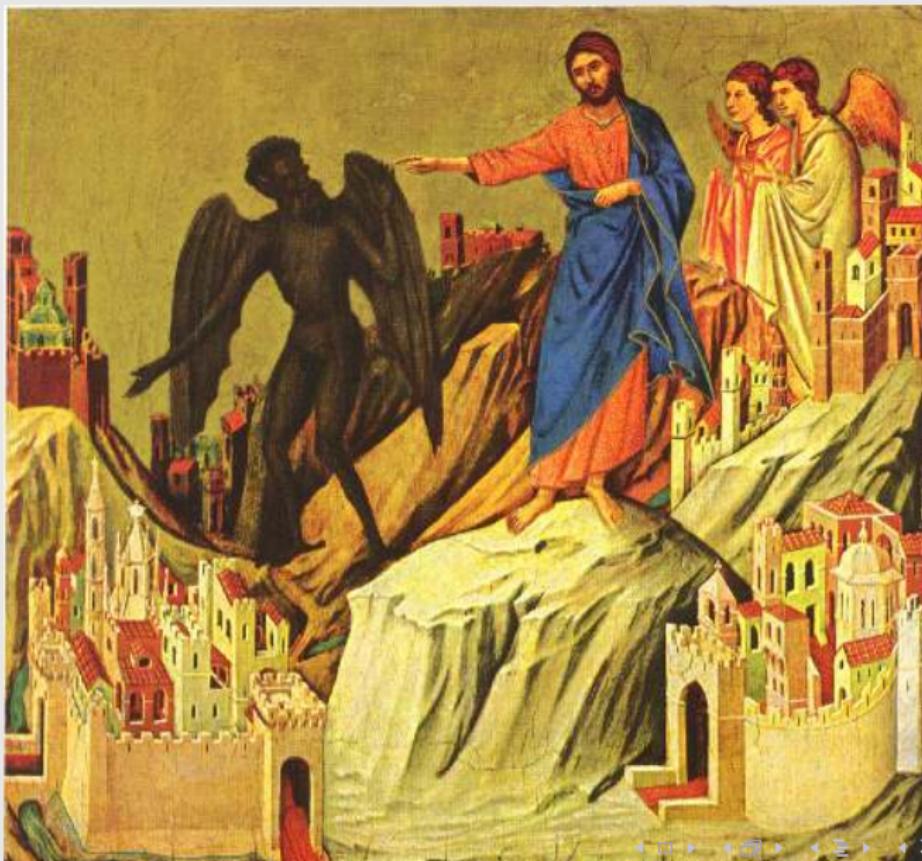
Höherdimensionale Räume

Pompeii (Casa dei Vettii), ca. 70 n.Chr.



Duccio di Buonisegna

„Die Versuchung Christi“

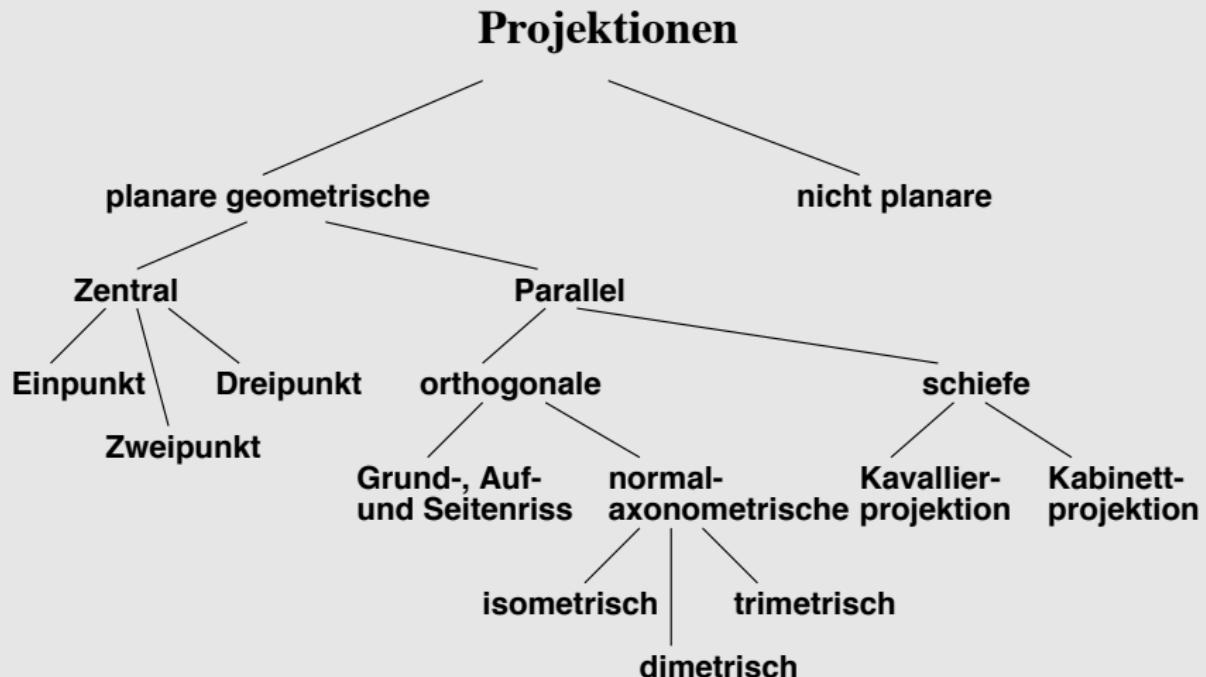


Raffael von Urbino (1483–1520)

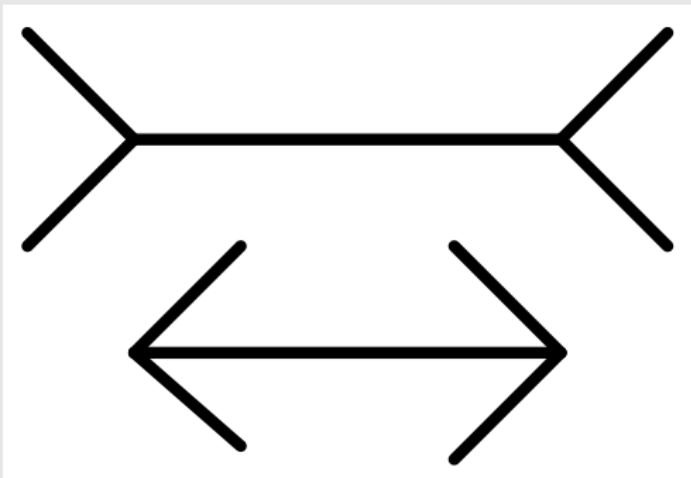
„Schule der Philosophen“



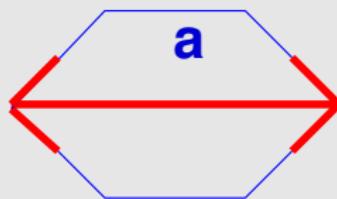
Übersicht: Projektionen



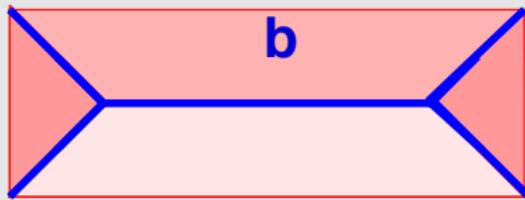
Welche Strecke ist länger? a oder b?



Erklärungsversuch: Konvexe und konkavе Kanten

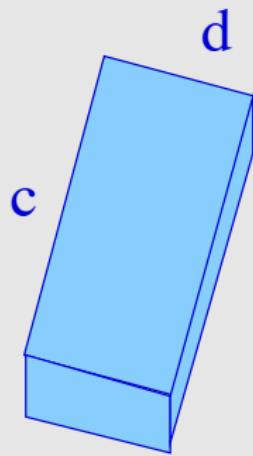
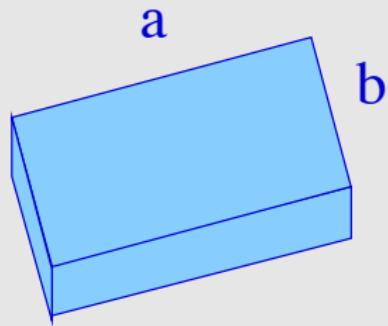


"a" ist konvex
und liegt vorne.



"b" ist konkav
und liegt weiter
hinten.

Welche Strecke ist länger? a oder c? b oder d?



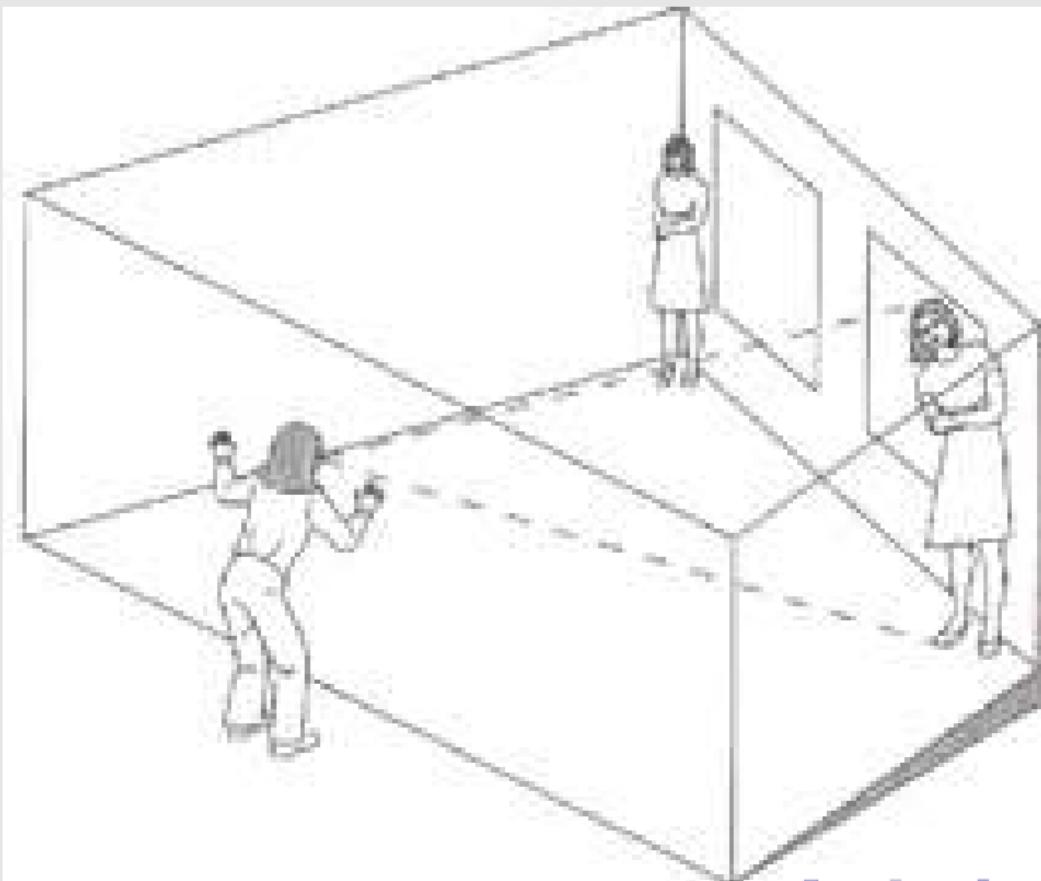
Illusion: Perspektivische Verkürzung von Strecken

Quader a und b

Illusion: Perspektivische Verkürzung von Strecken

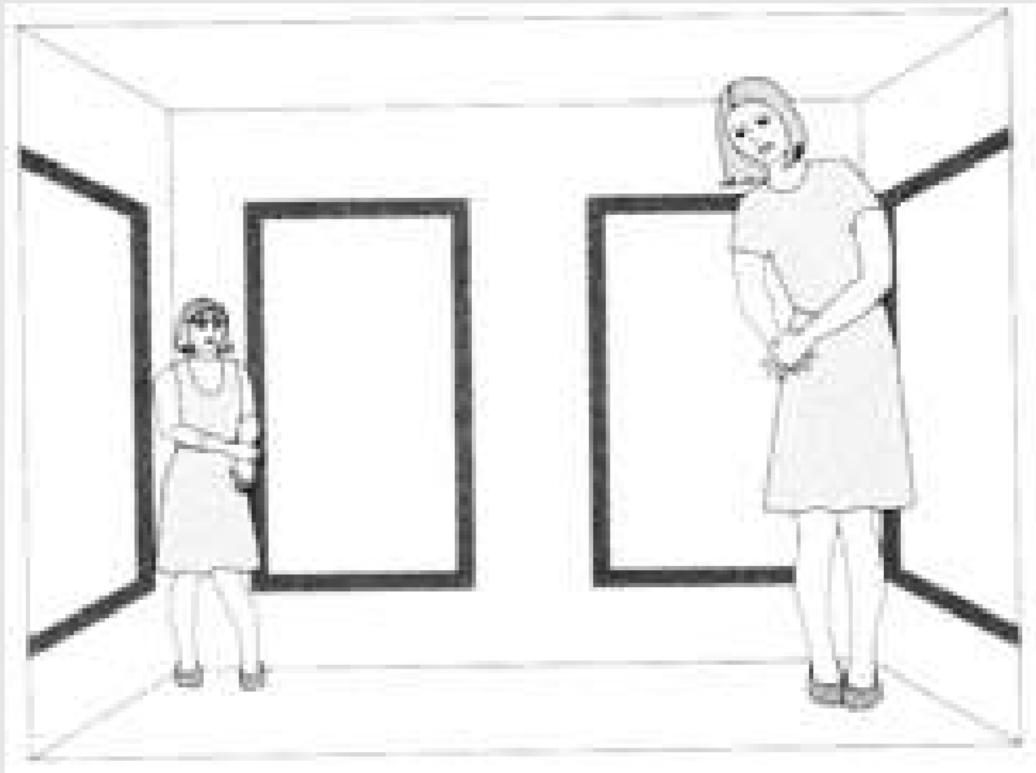
Illusion: Quader a und b

Der Ames-Raum: Blick von außen



Der Ames-Raum:

Speziellen Betrachterstandort



Der Ames-Raum in spezieller Ansicht

Der Ames-Raum in spezieller Ansicht

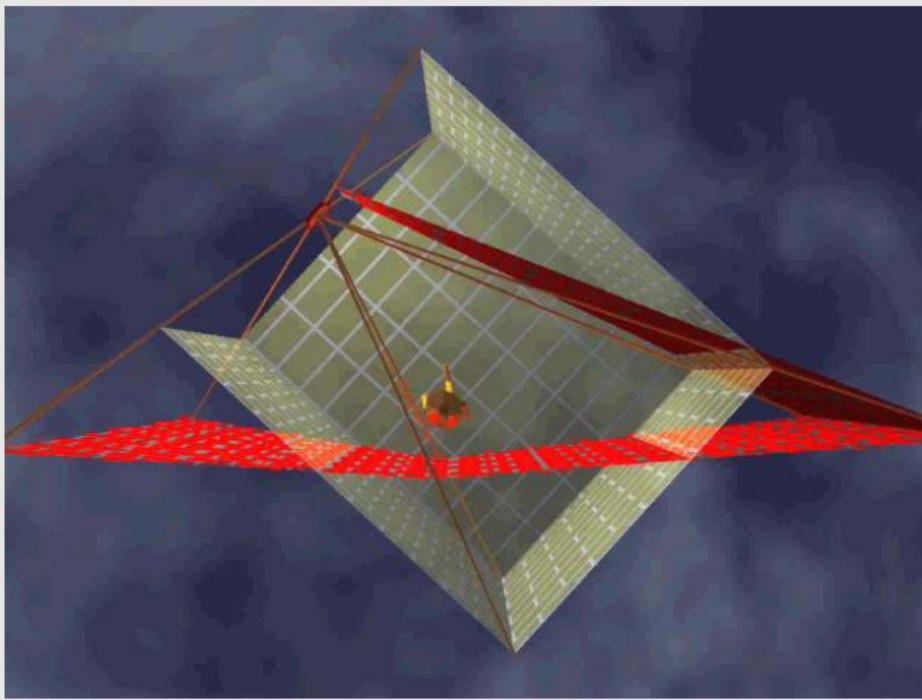
Der Ames-Raum mit verräterischer Textur

Der Ames-Raum mit verräterischer Textur

Der Ames-Raum von oben

Der Ames-Raum von oben gesehen

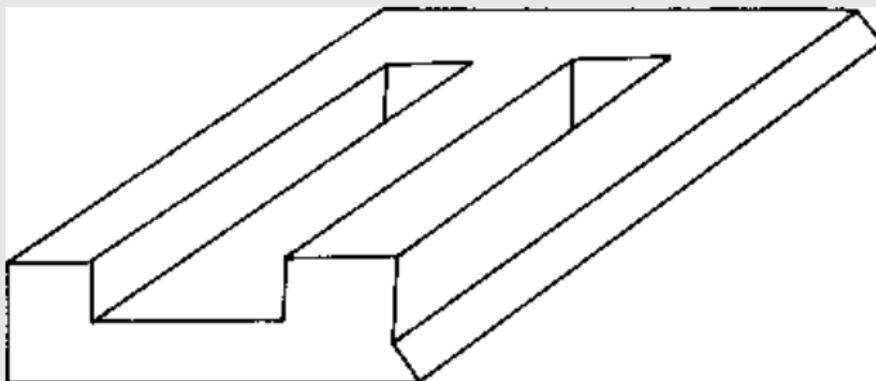
Der Ames-Raum: Wahre und scheinbare Wände



Der Ames-Raum: Konstruktionslinien

Movie: Der Ames-Raum – Konstruktionslinien

Bilder unmöglicher Objekte



Bilder unmöglicher Objekte

Bilder unmöglicher Objekte entstehen, wenn wir Winkel zeichnen und mit Geraden verbinden, deren lokale perspektivische Interpretationen widersprüchlich sind und nicht das Bild eines einzigen dreidimensionalen Objektes sein können: beispielsweise eine konvexe Kante die in eine konkave Kante übergeht.

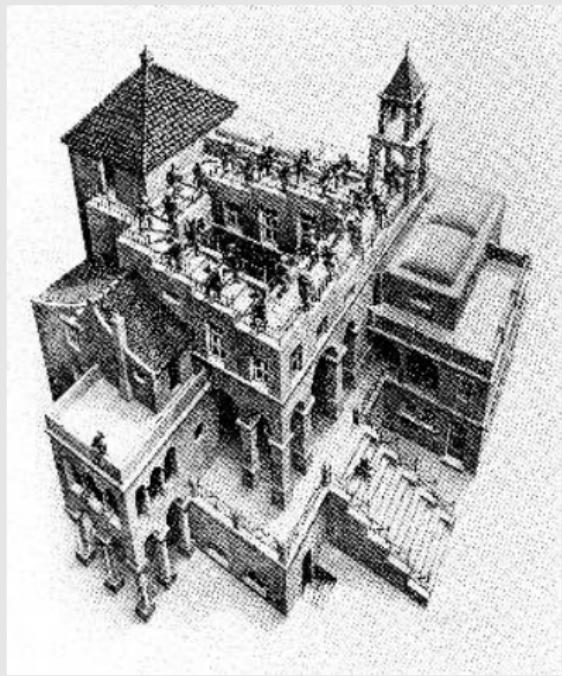
Meister im Konstruieren solcher Bilder waren

William Hogarth: Kupferstecher, 1697–1764

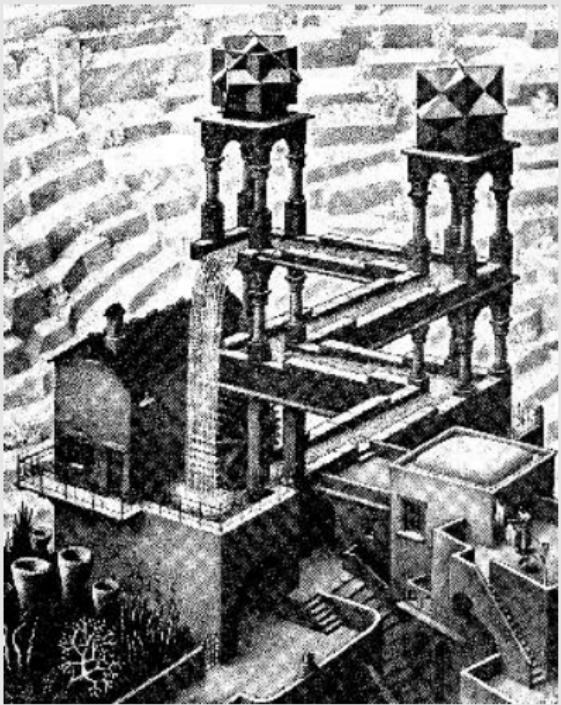
M.C. Escher: Grafiker, 1898–1972



M.C. Escher



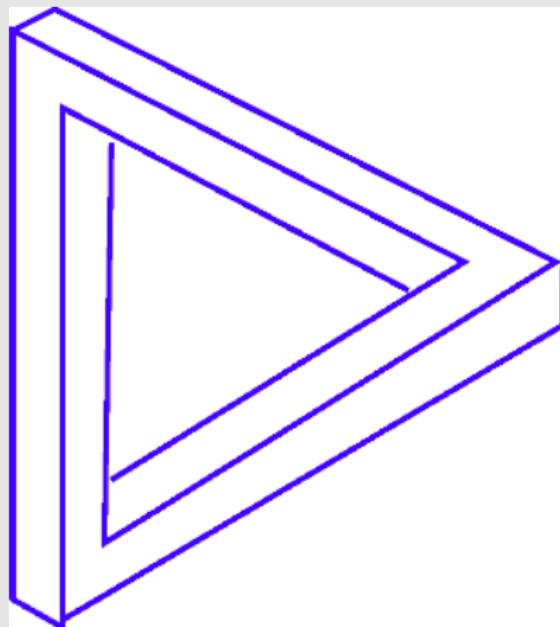
Treppauf und treppab, 1960



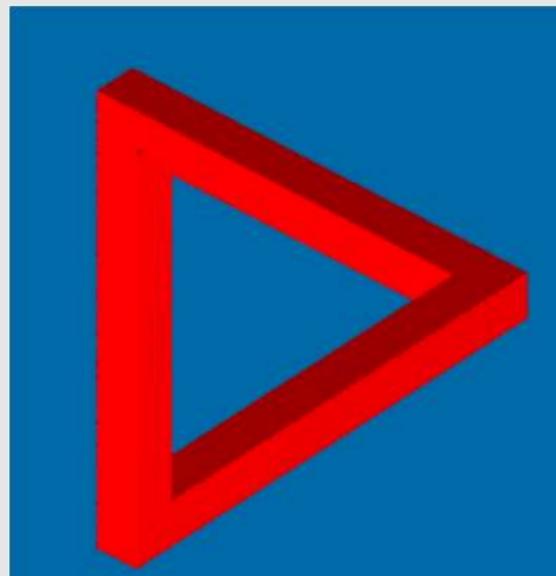
Wasserfall, 1961

Das „Penrose triangle“

Ein unmögliches Objekt?



Dieses unmögliche Objekt gibt es wirklich.



Das Penrose triangle in spezieller Ansicht

Movie: Das unmögliche Objekt in **spezieller** Ansicht

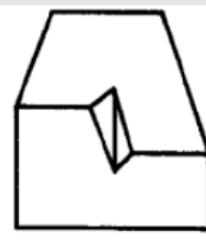
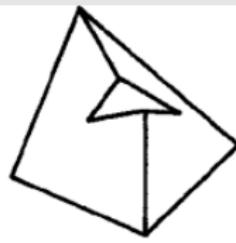
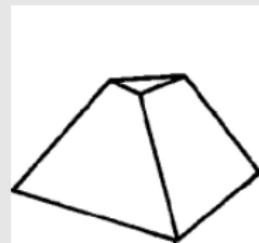
Das unmögliche Objekt:

In allgemeiner Ansicht

Movie: Das unmögliche Objekt in **allgemeiner Ansicht**



Linienzeichnungen, die keine 3D-Polyeder darstellen:



Kulissenmalerei



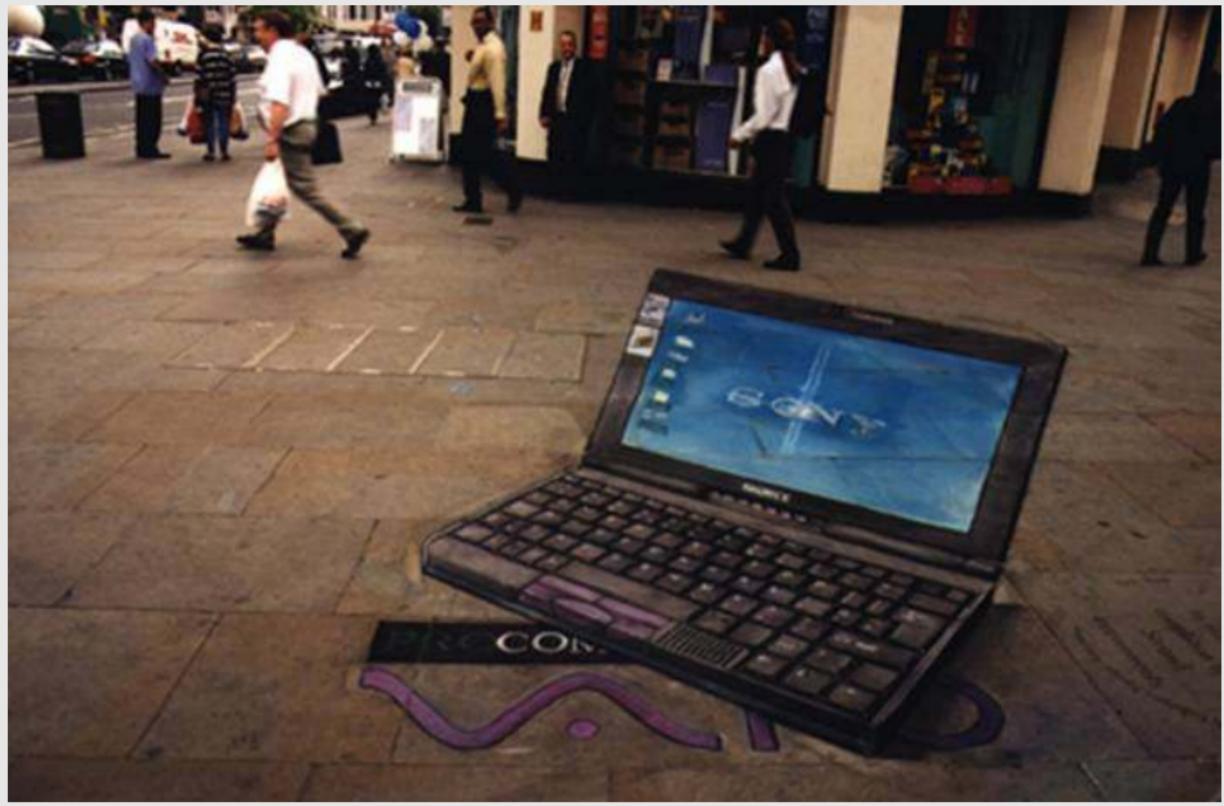
Kulissenmalerei



Kulissenmalerei



Kulissenmalerei



Kulissenmalerei



Kulissenmalerei



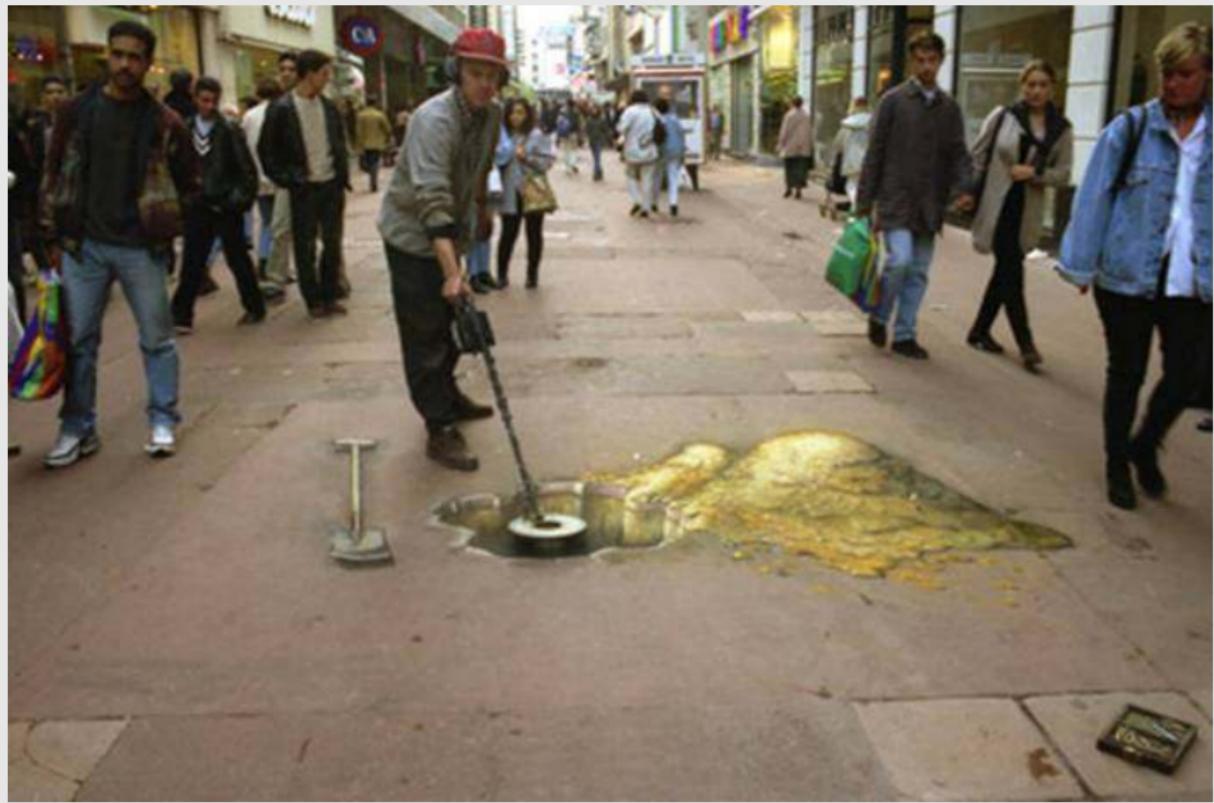
Kulissenmalerei



Kulissenmalerei



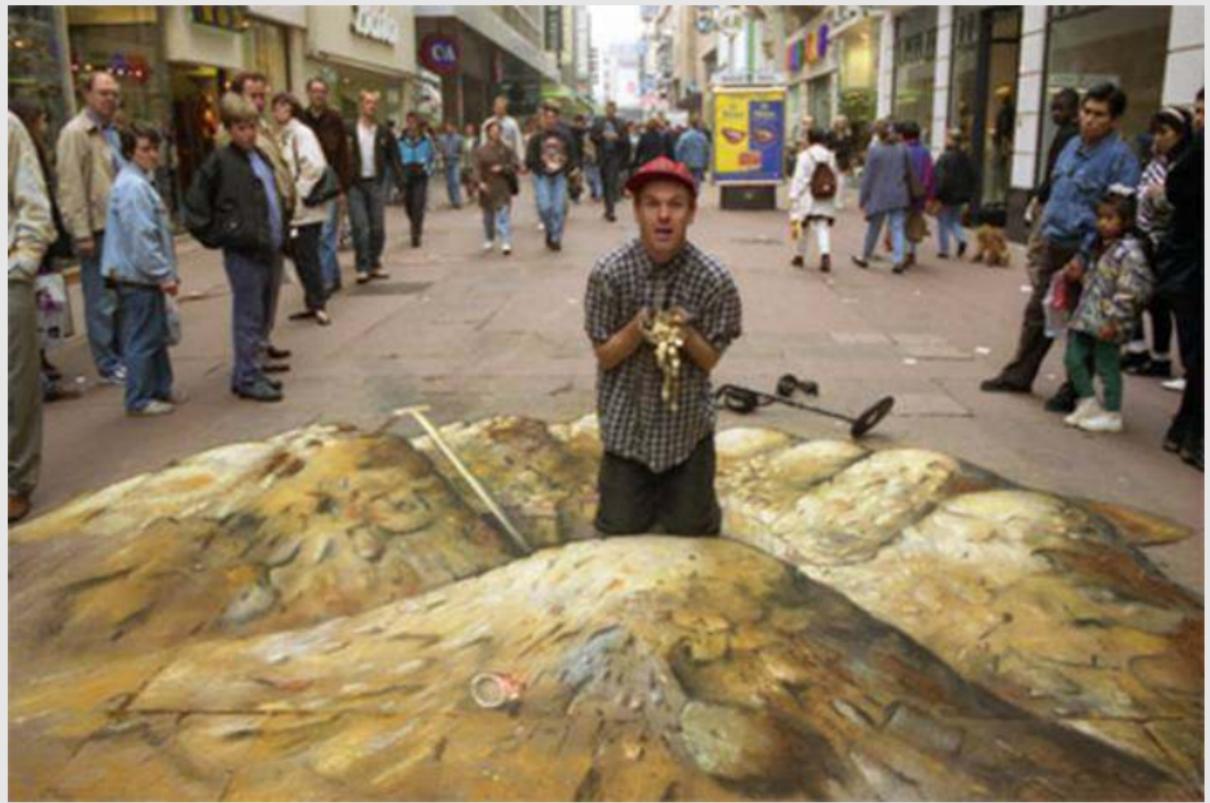
Kulissenmalerei



Kulissenmalerei



Kulissenmalerei



Kulissenmalerei



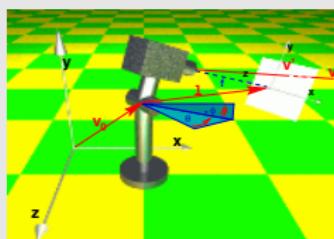
Kulissenmalerei



Virtuelle und reale Kameras

12 Gesetze der Perspektive:
Fluchtpunkte

13 Illusionen



14 Virtuelle und reale Kameras

- Die Linsenkamera
- Virtuelle Kameras in POV-Ray
- Kamerakalibrierung
- Betrachtungstransformation,
3D-Clipping

15 Höherdimensionale Räume

Linsenkamera

- ▶ Die Lochkamera erzeugt nur scharfe Bilder, wenn das Loch sehr klein ist. Daher ist die Bildhelligkeit nur sehr gering. Wird das Loch vergrößert, werden die Bilder allerdings unscharf.
- ▶ Lösung: Das Loch durch eine Linse ersetzen.
- ▶ Eine Linse erzeugt ein scharfes Bild bei (nahezu) beliebig großer Öffnung.
- ▶ Die Entfernung, in der das Bild eines Objektes scharf erscheint (Bildweite), ist eine Funktion der Entfernung des Objektes von der Linse (Gegenstandsweite).

Abbildungsgleichung für eine **dünne** Linse:

s = Gegenstandsweite

s' = Bildweite

f = Brennweite

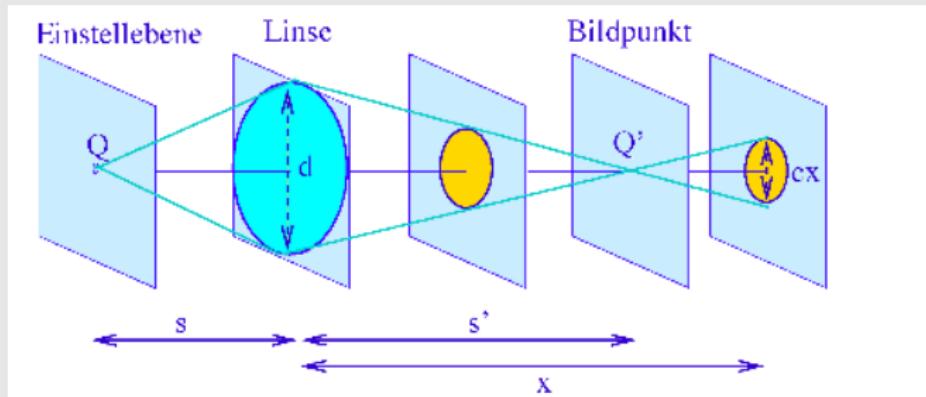
$$\frac{1}{s} + \frac{1}{s'} = \frac{1}{f}$$

- ▶ Nachteil: Es können nur solche Objekte gleichzeitig scharf abgebildet werden, die in der **Einstellebene** liegen.
- ▶ Die **Einstellebene** ist diejenige Ebene senkrecht zur optischen Achse, deren Punkte bei einer gegebenen Bildweite s' auf der Bildebene scharf abgebildet werden.

Nur die Blüte in der Einstellebene erscheint scharf.



Wovon hängt die Größe des Zerstreuungskreises $c(x)$ ab?



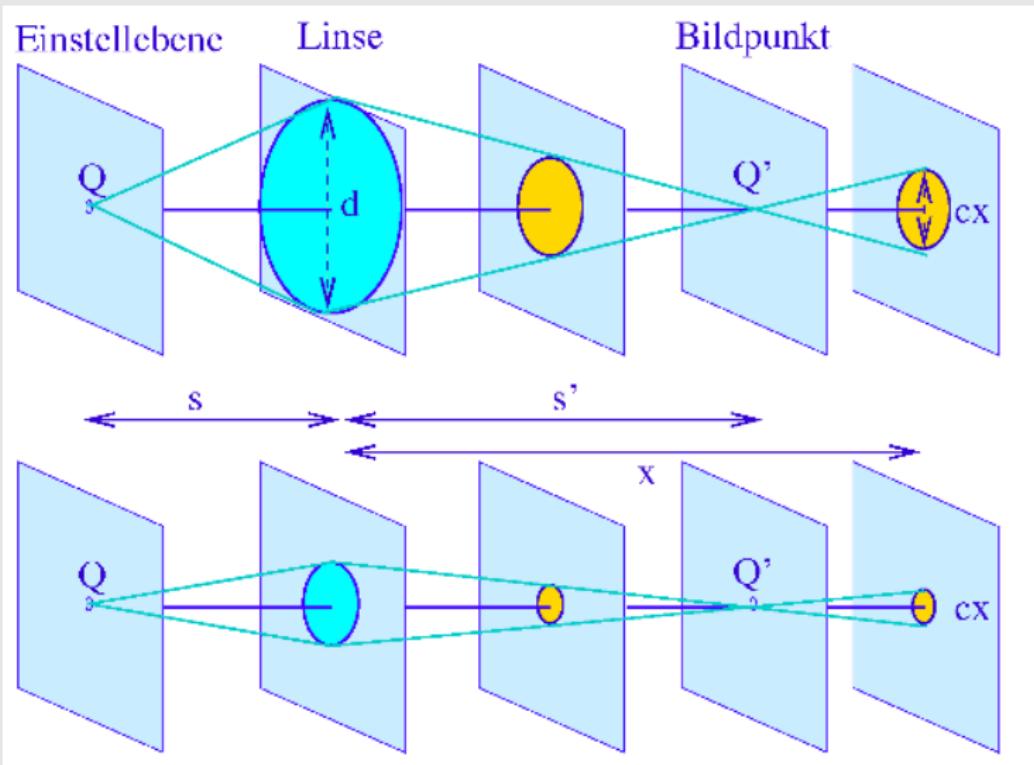
$$\frac{c(x)}{d} = \frac{x - s'}{s'} = \frac{\Delta_x}{s'}$$

$$c(x) = d \cdot \frac{x - s'}{s'} = d \cdot \frac{\Delta_x}{s'}$$

Durchmesser des Zerstreuungskreises

- ▶ Der Durchmesser des Zerstreuungskreises ist
 - ▶ umgekehrt proportional zur Bildweite und damit zum Abbildungsmaßstab und
 - ▶ proportional zum Durchmesser der Blende (Apertur).
- ▶ Je weiter wir die Blende öffnen, desto größer werden bei gegebener Einstellung die Zerstreuungskreise.

Große und kleine Apertur, Zerstreuungskreise



Beachte:

Die Unschärfe kann nur wahrgenommen werden, wenn die Zerstreuungskreise größer sind als die Auflösung des Bildes (Abtastabstand bei Digitalbildern, Abstand des Filmkorns).

Nah- und Fernpunkt

Definition (Nah- und Fernpunkt)

Die Ebenen, die den Bereich begrenzen, in dem bei einer gegebenen Einstellebene die Zerstreuungskreise kleiner sind als die Auflösung des Bildes, heißen **Nahpunktsebene** und **Fernpunktsebene**, die Schnittpunkte dieser Ebenen mit der optischen Achse heißen entsprechend **Nahpunkt** t_N und **Fernpunkt** t_F .

Definition (Tiefenschärfe)

Der Abstand $t_F - t_N$ zwischen Fernpunkt und Nahpunkt.

Wovon hängt die Tiefenschärfe ab?

Sei δ der größte zulässige Durchmesser eines Streukreises.

Für den Nahpunkt und den Fernpunkt gilt dann:[dtv Lexikon, 1970]

$$t_N = \frac{s \cdot \delta}{d + \delta}$$

$$t_F = \frac{s \cdot \delta}{d - \delta}$$

- ▶ Die Tiefenschärfe hängt bei gegebener Einstellebene und gegebener Auflösung nur von der Blendenöffnung ab und ist umso größer, je kleiner die Blende ist.
- ▶ Die Tiefenschärfe kann wirksam zur **Bildgestaltung** eingesetzt werden.

Freistellen eines Objektes durch geringe Tiefenschärfe



Geringe Tiefenschärfe ist beliebt für Porträts



Landschaftsbild mit großer Tiefenschärfe



Gravitationslinsen

Jedes Medium, das den Lichtweg krümmt, kann als Linse wirken, nicht nur Glas.



Virtuelle Kameras in POV-Ray

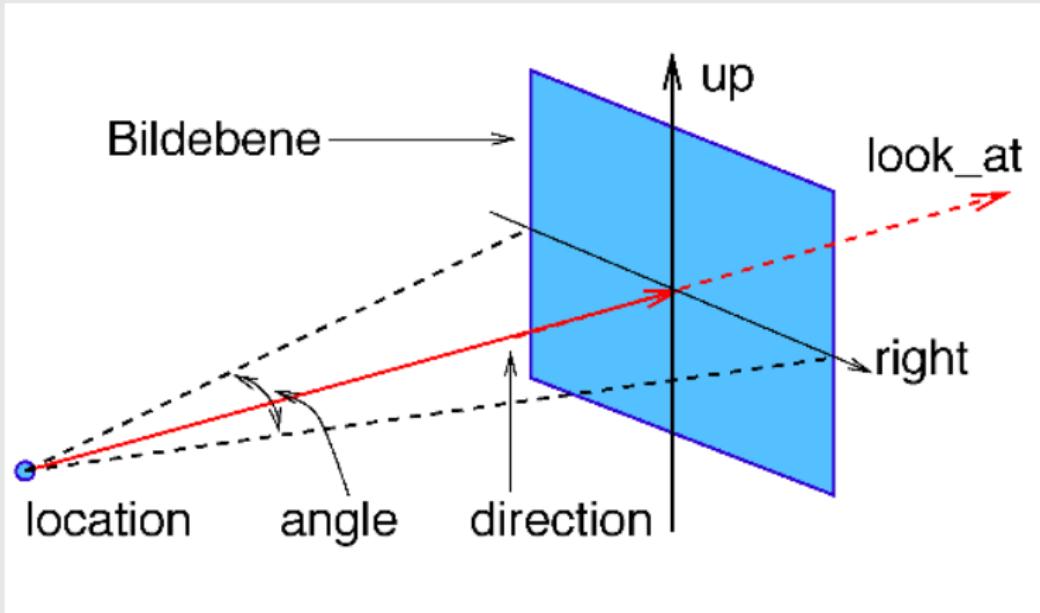
```
camera {
[ perspective | orthographic | fisheye |
 ultra_wide_angle | omnimax | panoramic |
 cylinder FLOAT ]
location <vector>
look_at <vector>
angle FLOAT
aperture FLOAT
focal_point ... }
```

Beispiel (Kamera):

- ▶ Eine Kamera mit Zentralprojektion
- ▶ um vier Einheiten in y-Richtung verschoben,
- ▶ die auf den Ursprung ausgerichtet ist,
- ▶ mit einem Blickfeld von 60°.

```
camera {  
    perspective  
    location 4*y  
    look_at 0  
    angle 60 }
```

Parameter der Kamera in POV-Ray



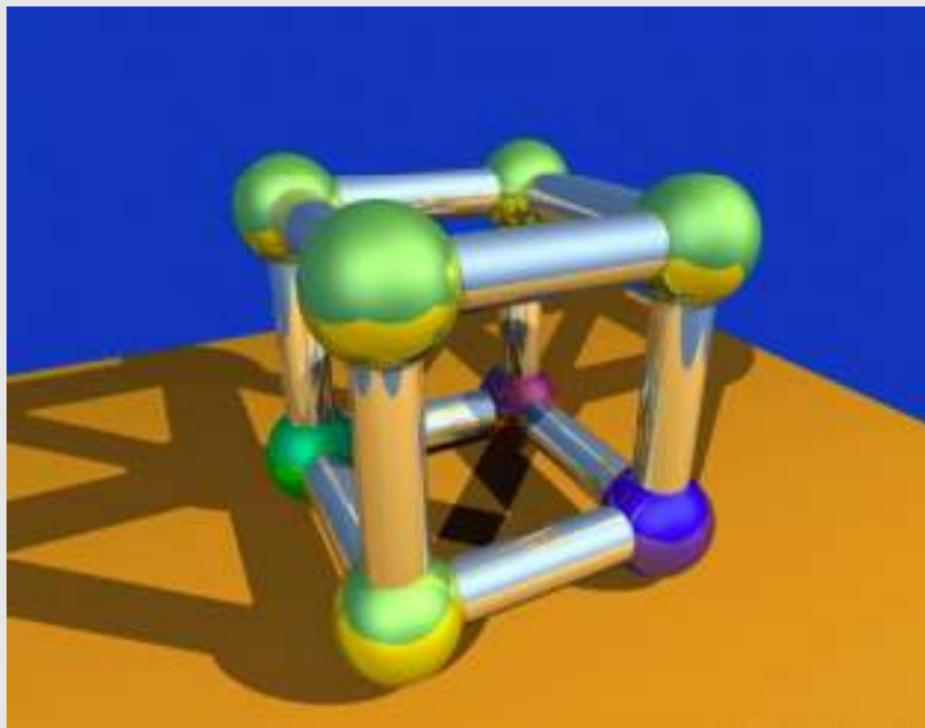
Zentralprojektion

Beispiel

Eine Kamera, Zentralprojektion, Öffnungswinkel 70°

```
camera {  
perspective  
location <-20, 30, -40>  
look_at <0, 10, 0>  
    angle 70  
}
```

Das erzeugte Bild (Beisp. 24)



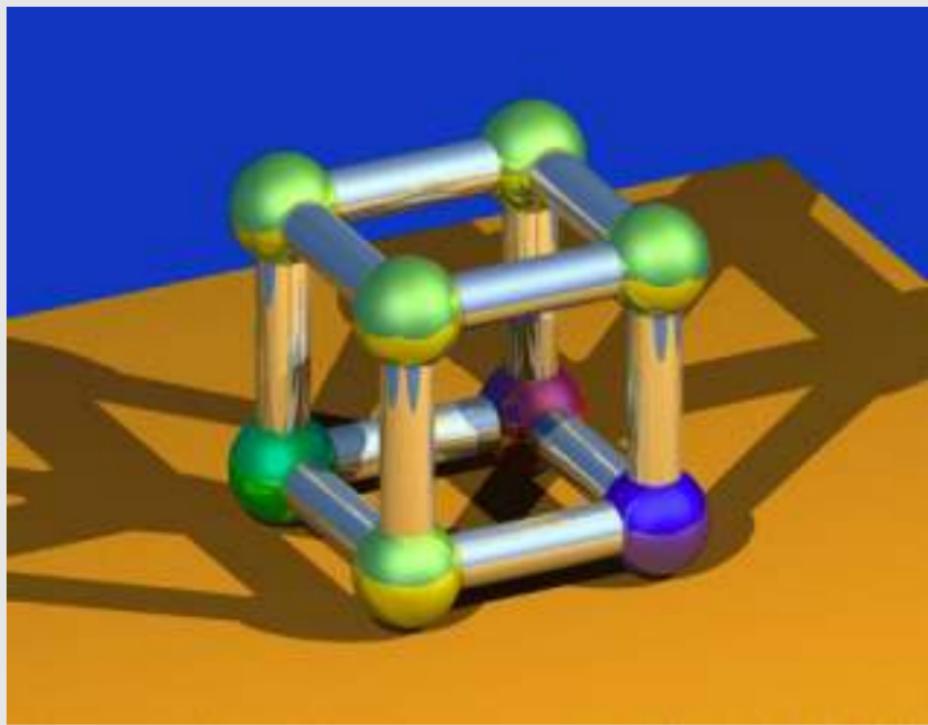
Orthogonale Parallelprojektion

Beispiel

Orthogonale Parallelprojektion, die Lage der Bildebene wird durch up and right-Vektoren beschrieben.

```
camera {  
    orthographic  
    location <-20, 30, -40>  
    right <50*4/3, 0, 0>  
    up <0, 50, 0>  
  
    look_at <0, 10, 0>  
    angle 70  
}
```

Das erzeugte Bild (Beisp. 25)



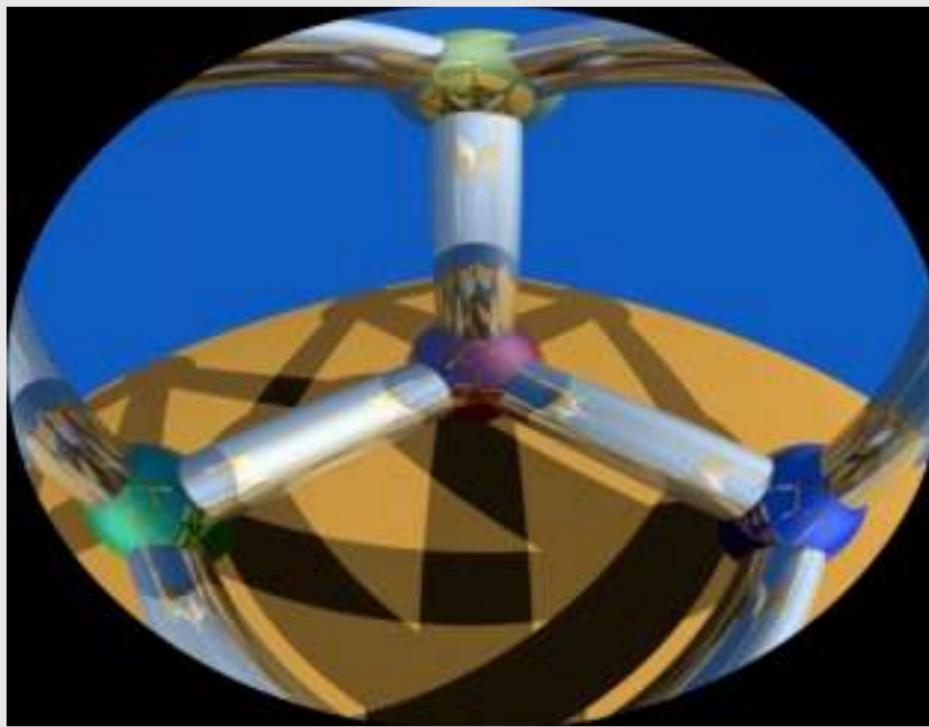
Fischauge mit 180°

Beispiel

Ein extremes Weitwinkelobjektiv: Fischauge mit 180° Rundumsicht

```
camera {  
    fisheye  
    angle 180  
    location <0, 10, 0>  
    right <4/3, 0, 0>  
    up <0, 1, 0>  
    look_at <10, 0, 10>  
}
```

Das erzeugte Bild (Beisp. 26)



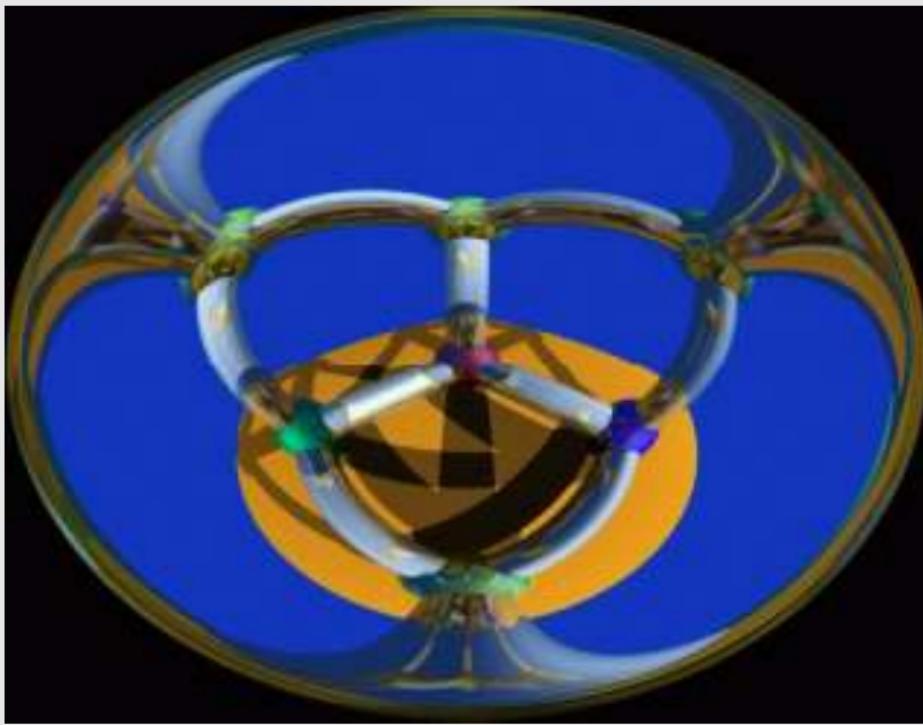
Fischauge mit 360°

Beispiel

Ein extremes Weitwinkelobjektiv: Fischauge mit 180° Rundumsicht

```
camera {  
    fisheye  
    angle 360  
    location <0, 10, 0>  
    right <4/3, 0, 0>  
    up <0, 1, 0>  
    look_at <10, 0, 10>  
}
```

Super-Fischauge 360°



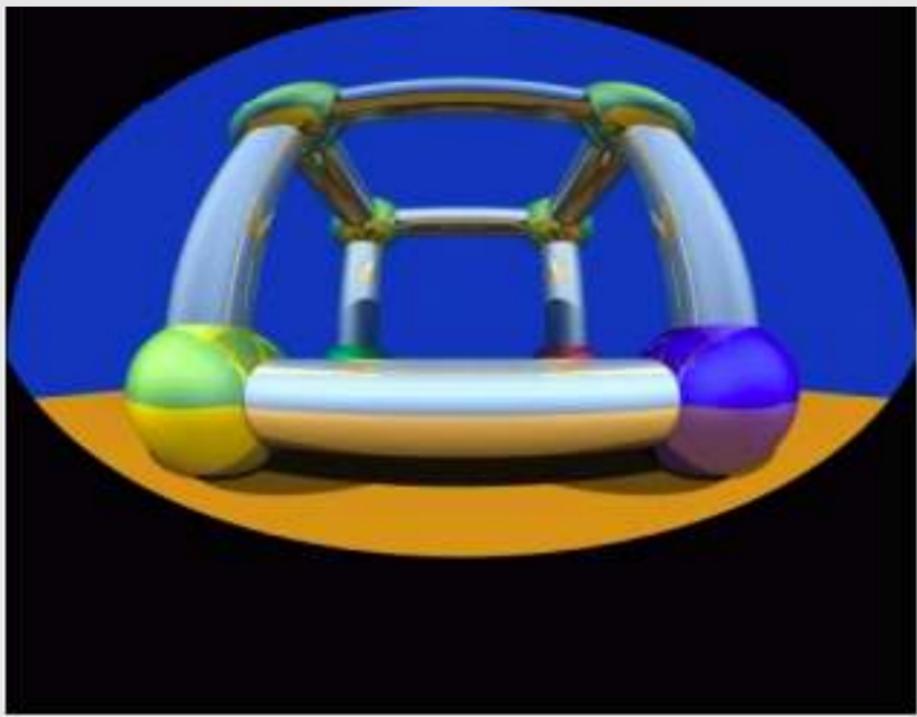
Omnimax-Projektion 180°

Beispiel

Projektion an eine Kuppel mit 180° Rundumsicht

```
camera {  
    omnimax  
    angle 360  
    location <0, 2, -20>  
    right <4/3, 0, 0>  
    up <0, 1, 0>  
    look_at <0, 2, 10>}
```

Das erzeugte Bild (Beisp. 28)

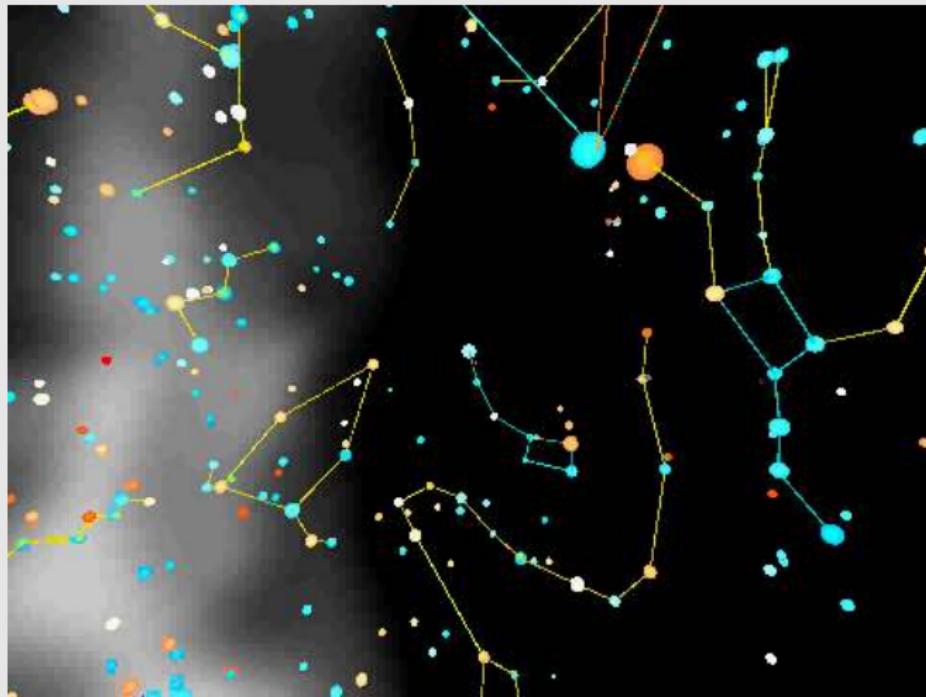


Bewegte Kamera

Space Rollercoasters

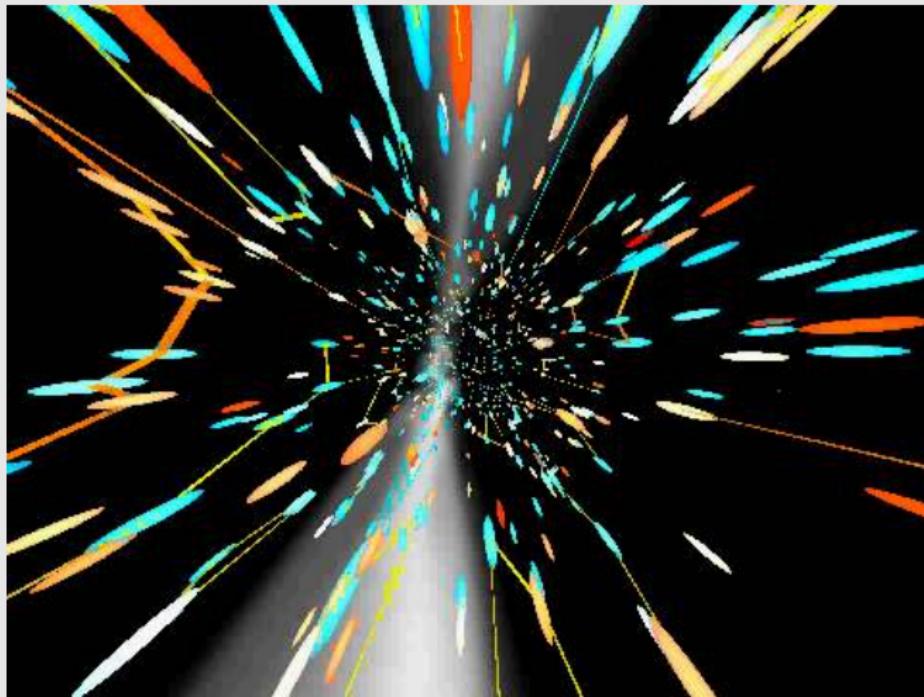
Verzerrung: Zentralprojektion

Zentralprojektion, Öffnungswinkel 90° , leichte elliptische Verzerrung



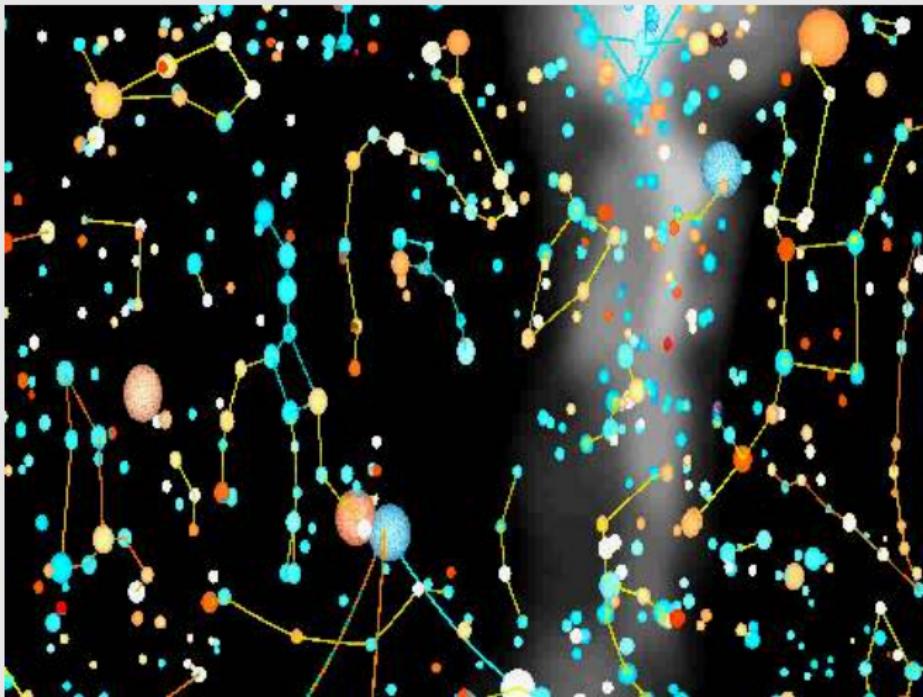
Verzerrung: Zentralprojektion

Zentralprojektion, Öffnungswinkel 170°, starke elliptische Verzerrung



Besser: Sphärische Projektion

Sphärische Projektion, Öffnungswinkel 170°, nur leichte elliptische Verzerrung



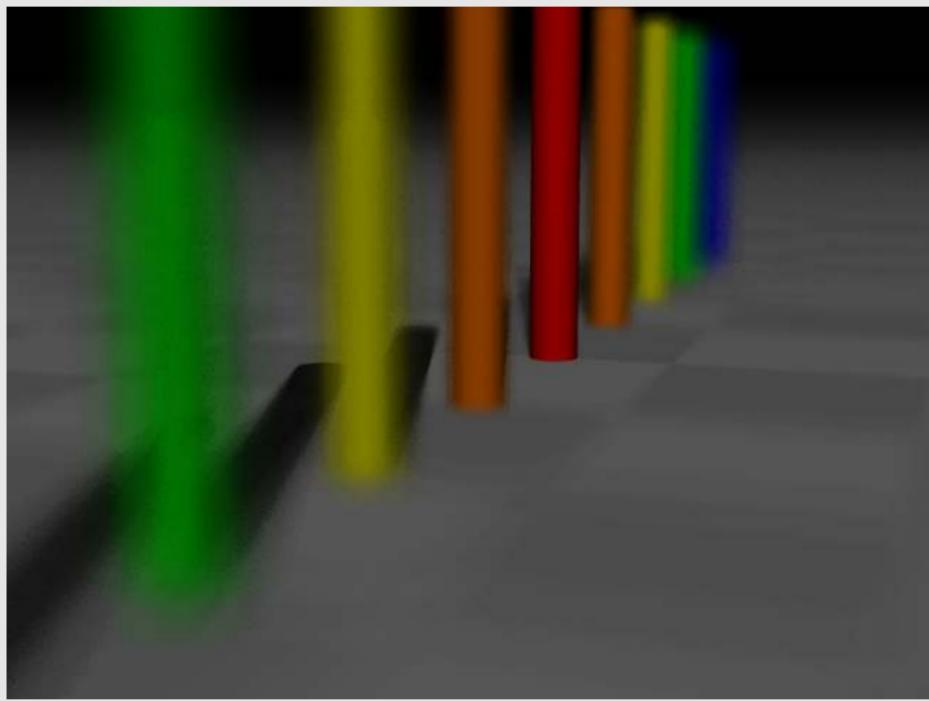
Fokussierungsunschärfe in POV-Ray

Beispiel

Fokussierungsunschärfe kann in POV-Ray über den Apertur-Parameter spezifiziert werden.

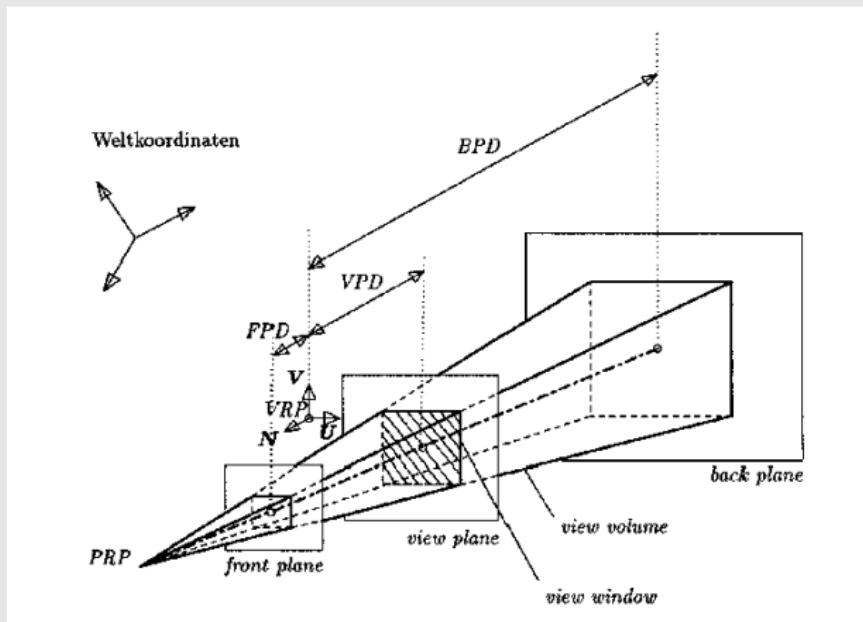
```
location <2, 2, -11 >
angle 37
look_at <-1.75,0,0>
aperture 2
focal_point <0, 0, 0>
}
```

Das erzeugte Bild (Beisp. 29)



Betrachtungstransformation, 3D-Clipping

Der Bildraum: View-Reference Coordinate System



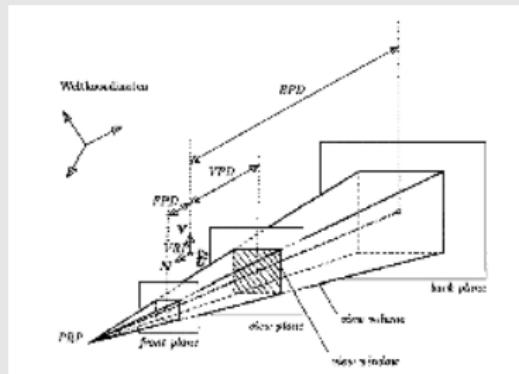
Festlegung des view coordinate systems durch view reference point (VRP), view plane normale (N), view plane distance (D) und view up vector (VUV).

View Coordinate System

- ▶ View reference point VRP:
Ein Referenzpunkt im Weltkoordinatensystem, auf den die Pseudo-Kamera fokussiert.
- ▶ View plane VP:
Die Bildebene, auf der die Szene projiziert wird, angegeben durch einen Normalvektor.
- ▶ View plane normal N:
Normalabstand vom VRP.

View Coordinate System

- ▶ View plane distance D:
- ▶ View reference plane:
Eine Ebene parallel zur Bildebene, in der der VRP liegt.
- ▶ View up vector VUV:
Bestimmt die Länge des Bildes in der Bildebene.



View Coordinate System

Z: parallel zu N

Y: Projektion des VUV parallel zu N in die VRP

X: senkrecht zu z und y, Rechtssystem

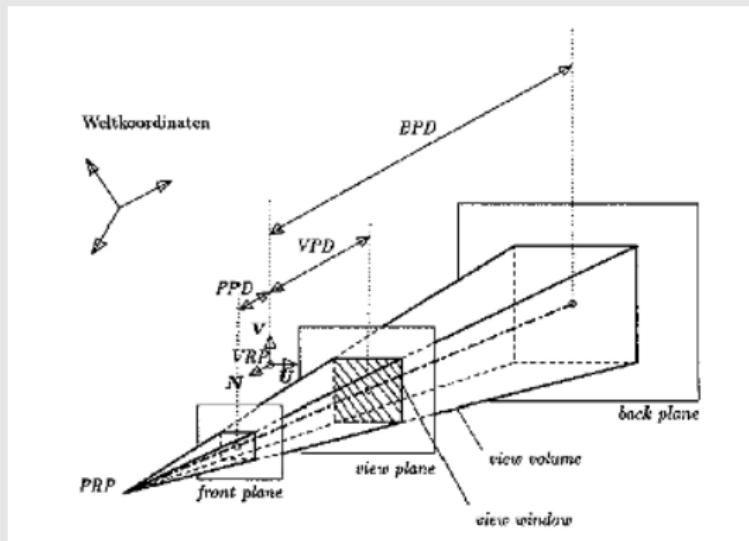
Läßt sich leicht durch Rotation und Translation aus dem Weltkoordinatensystem herleiten.

View Window und View Volume

Festlegung des abzubildenden Ausschnitts der Szene

- ▶ durch die Wahl des Objektives (reale Welt),
- ▶ durch das Window (virtuelle Welt).

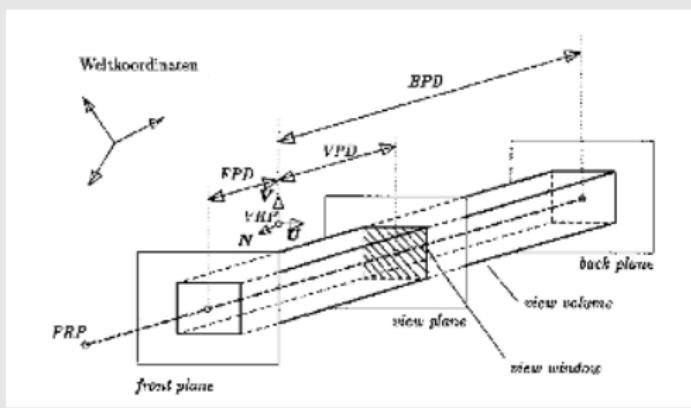
Durch das view window definierter Bildraum bei der Zentralprojektion



View Window und View Volume

- ▶ Front- und Backplane sind nötig, damit
 - ▶ nahe Objekte nicht die ganze Szene verdecken,
 - ▶ entfernte, zu kleine Objekte nicht stören.
- ▶ Form des View-Volume:
 - ▶ Zentralprojektion: Pyramidenstumpf
 - ▶ Parallelprojektion: Prisma (Quader)

Durch das view window definierter Bildraum bei Parallelprojektion



Betrachtungstransformation

Naiver Ansatz:

- ▶ Transformation des Weltkoordinatensystems in das VRC-System,
- ▶ Clipping an den Bildraumgrenzen,
- ▶ Projektionen des Bildrauminhalts auf die Bildebene,
- ▶ Darstellung des View-Windows auf dem Endgerät.

Betrachtungstransformation

Parallelprojektion

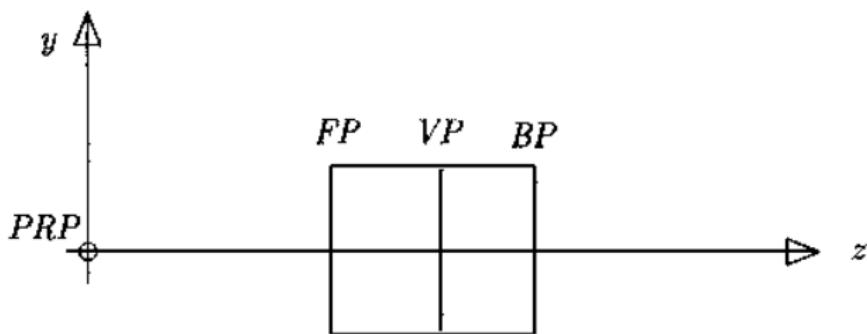
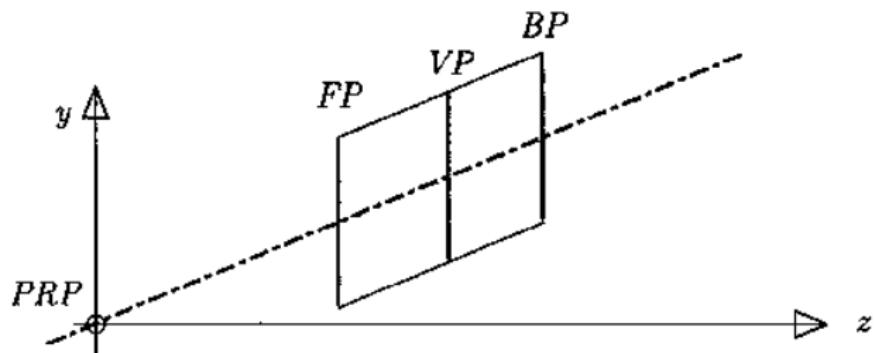
Für das Clipping müssen die Objekte mit 6 Ebenen geschnitten werden (2 parallel zur Bildebene, 4 beliebig).

⇒ Transformation des Bildraumes achsenparallel.

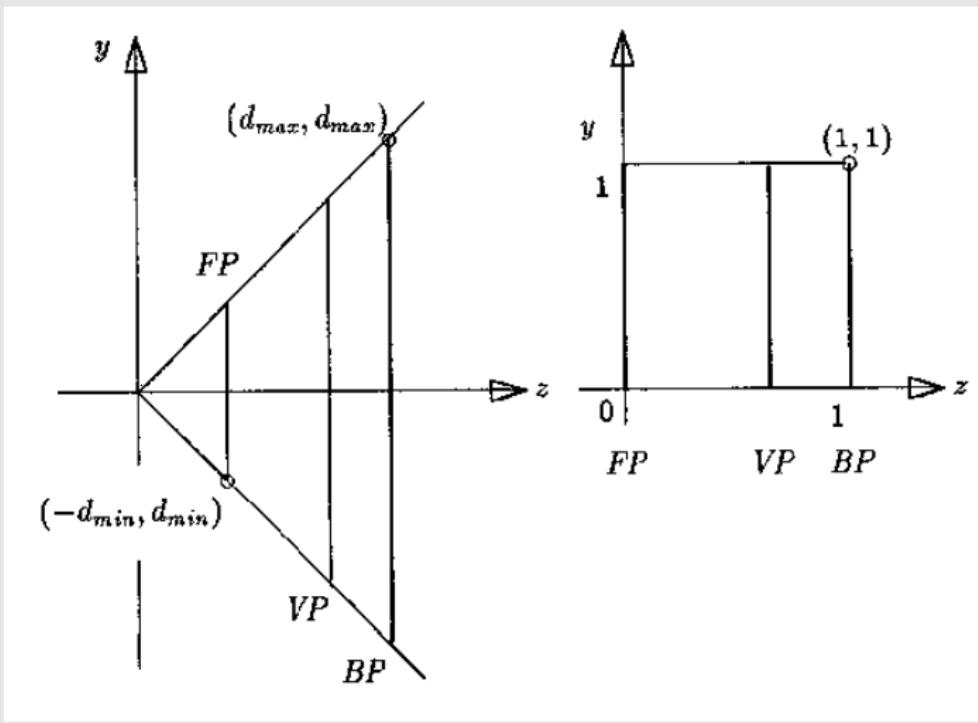
[Fellner, 1992, S. 263]

Im transformierten Raum ist die Zentralprojektion ein **Riß**, und die Bildkoordinaten sowie das Clipping sind triviale Operationen.

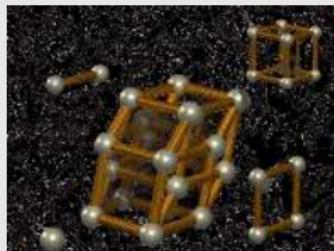
Scherung des Bildraums bei Schrägriß



Transformation des regelmäßigen Pyramidenstumpfes in den Einheitswürfel



Höherdimensionale Räume



12 Gesetze der Perspektive:
Fluchtpunkte

13 Illusionen

14 Virtuelle und reale Kameras

15 Höherdimensionale Räume

- Reduktion der Anzahl der Dimensionen
- Abbildung der räumlichen Dimensionen auf andere Parameter
- Projektion in Unterräume

Problem:

Wie kann man Räume mit mehr als drei Dimensionen veranschaulichen?

Solche Visualisierungen sind wichtig für

- ▶ Mustererkennung
- ▶ Naturwissenschaftliche Anwendungen, z.B. Simulationen
- ▶ Tests, Fehlersuche,
- ▶ Dokumentation, Veröffentlichungen,
- ▶ Verständnis.

Mehrdimensionale Skalierung:

- ▶ Transformation der Daten in einen Raum mit weniger Dimensionen, so daß die *Abstände zwischen den Bildern der Proben* der Unähnlichkeit der Proben entsprechen.
- ▶ Es sind nicht immer exakte Lösungen möglich, aber eventuell eine gute Approximation.

Gegeben seien:

n Proben $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ und ihre

Abbildungen $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_n$,

Abstände δ_{ij} zwischen \vec{x}_i und \vec{x}_j ,

d_{ij} zwischen \vec{y}_i und \vec{y}_j .

Gesucht: Die optimale Konfiguration der \vec{y}_i ,

so daß die $\frac{n(n-1)}{2} d_{ij}$ möglichst ähnlich den δ_{ij} sind.

Kriteriumsfunktionen:

$$J_{ee} = \frac{1}{\sum_{i < j} \delta_{ij}^2} \sum_{i < j} (d_{ij} - \delta_{ij})^2$$

$$J_{ff} = \sum_{i < j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$

$$J_{ef} = \frac{1}{\sum_{i < j} \delta_{ij}} \sum_{i < j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}$$

- ▶ Die Kriterien sind normalisiert und nur von den relativen Abständen abhängig.
- ▶ Sie sind invariant gegen starre Verschiebungen.

Optimierung durch Gradientensuche:

$$d_{ij} = \|\vec{y}_i - \vec{y}_j\|$$

$$\vec{\nabla}_{\vec{y}_k} J_{ee} = \frac{1}{\sum_{i < j} \delta_{ij}^2} \sum_{i \neq k} (d_{kj} - \delta_{kj}) \frac{\vec{y}_k - \vec{y}_j}{d_{kj}}$$

$$\vec{\nabla}_{\vec{y}_k} J_{ff} = 2 \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \frac{\vec{y}_k - \vec{y}_j}{d_{kj}}$$

$$\vec{\nabla}_{\vec{y}_k} J_{ef} = \frac{2}{\sum_{i < j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \frac{\vec{y}_k - \vec{y}_j}{d_{kj}}$$

Initialisierung: Zufällig gewählte \vec{y} oder Projektion der \vec{x} -Dimensionen mit der größten Varianz in den y-Raum.

Beispiel: 30 Punkte einer Helix

Beispiel

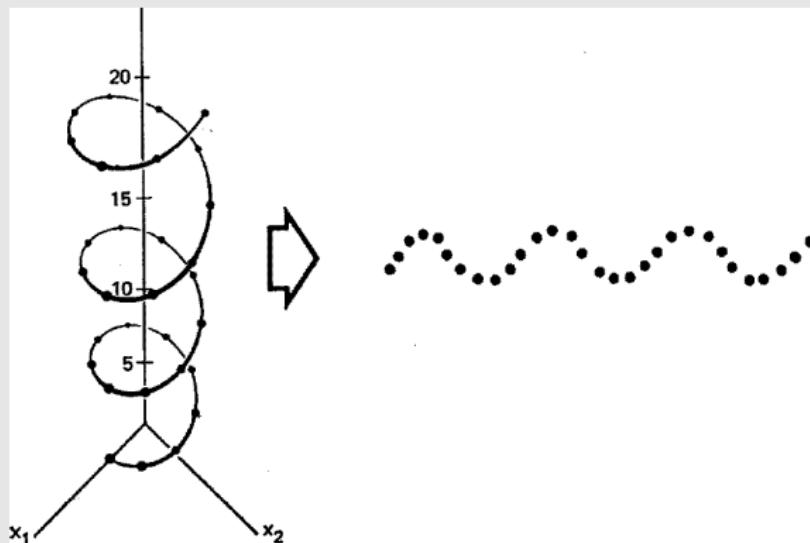
$$x_1(k) = \cos x_3(k)$$

$$x_2(k) = \sin x_3(k)$$

$$x_3(k) = \frac{k}{\sqrt{2}}, \quad k = 0, 1, \dots, 29.$$

Mit dem J_{ef} -Kriterium wurde in 20 Iterationsschritten eine Sinus-förmige 2D-Repräsentation gefunden.

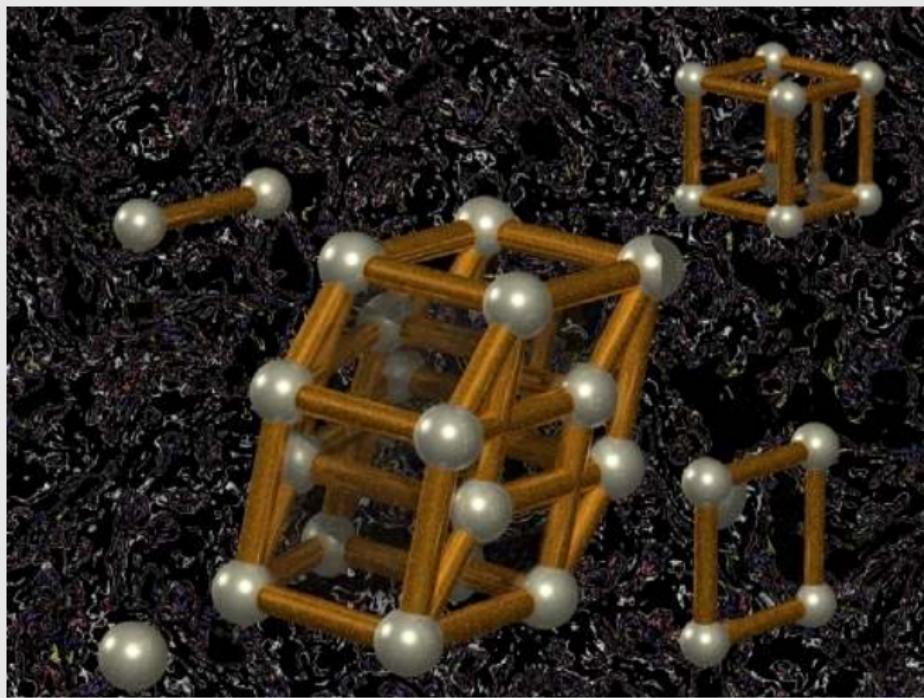
Reduktion der Dimensionalität von 3D auf 2D



Visualisierung von Objekten mit mehr als 3 Dimensionen

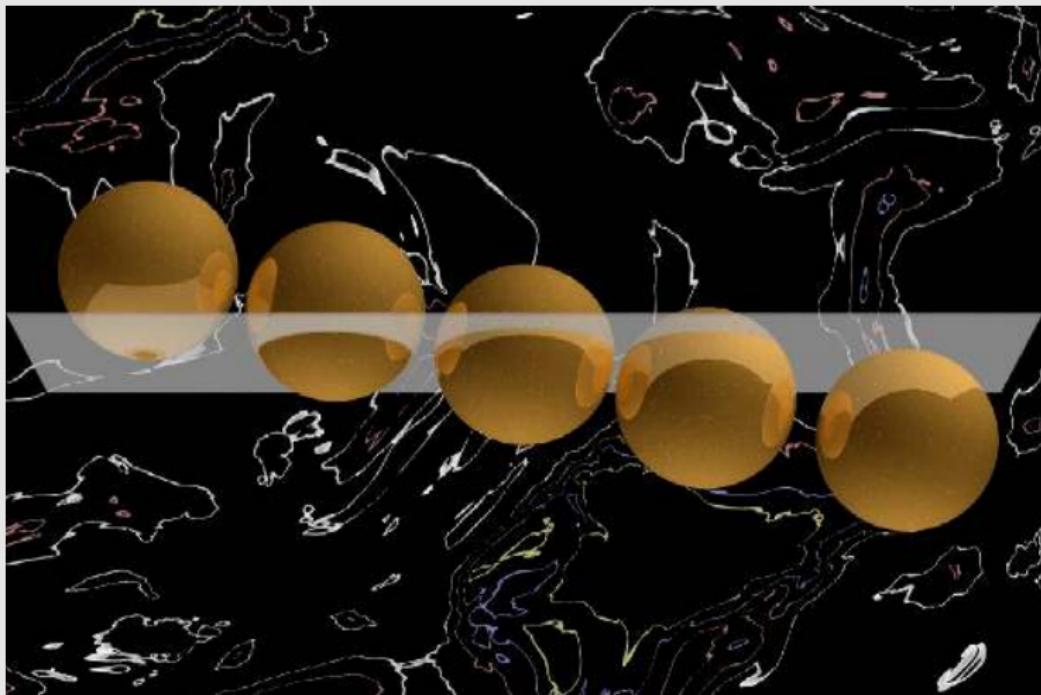
- ▶ Eine oder mehr Dimensionen auf andere Parameter abbilden: *Farbe, Textur, Symbole, Zeit, Isolinien, Vektorfelder* (siehe beispielsweise die geographische Farbskala auf Landkarten oder Seekarten).
- ▶ Teile der Objekte in Unterräume projizieren.

Würfel in 1, 2, 3 und 4 Dimensionen



Ein Gedankenexperiment

Eine dreidimensionale Kugel durchdringt eine Ebene.



Ein Gedankenexperiment:

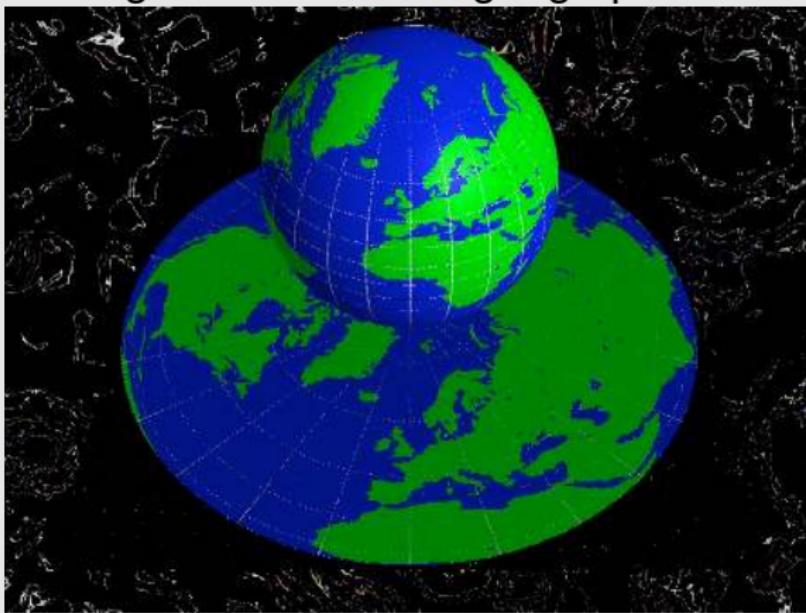
Wir stellen uns vor, wir wären zweidimensionale Wesen und lebten in einer Ebene.

Wir erleben gerade, wie eine dreidimensionale Kugel unsere Welt durchstößt.

Was sehen wir?

- ▶ Zunächst sehen wir nur einen Punkt.
- ▶ Es folgt ein Kreis, der zunächst immer größer wird und dann wieder schrumpft.
- ▶ Die Menge der Kreise kann als zweidimensionale Visualisierung der Kugel verwendet werden.

Ganz analog: Die Kreise der geographischen Breite



Ein zweidimensionales Abbild der Kugeloberfläche:

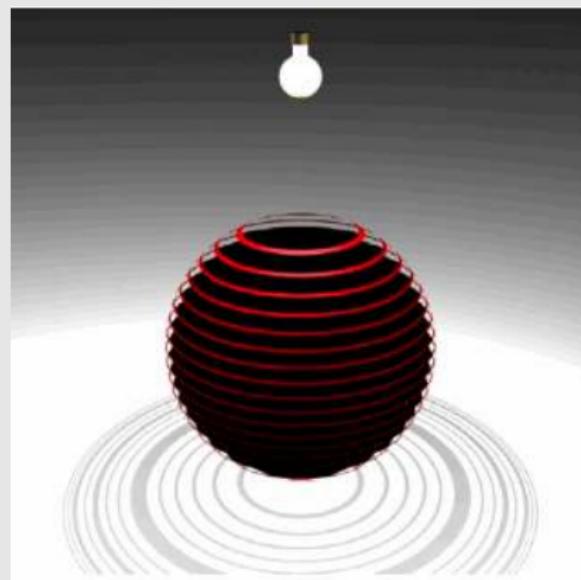
Eine ähnliche Folge von Kreisen können wir als zweidimensionales Abbild der Kugeloberfläche erzeugen, wenn wir parallele Streifen aus der Oberfläche herausschneiden.

- ▶ Die Kugeloberfläche wird einer Schar von parallelen Ebenen zum Schnitt gebracht.
- ▶ Die Schnittlinien werden auf eine Ebene projiziert.
- ▶ Wenn die Projektionsebene parallel zu den Schnittebenen liegt, dann sind die Projektionen konzentrische Kreise, sonst sich schneidende Ellipsen.

Projektion senkrecht zu den Schnittebenen

Zentralprojektion der Kreise auf eine Ebene parallel zu den Schnittebenen:

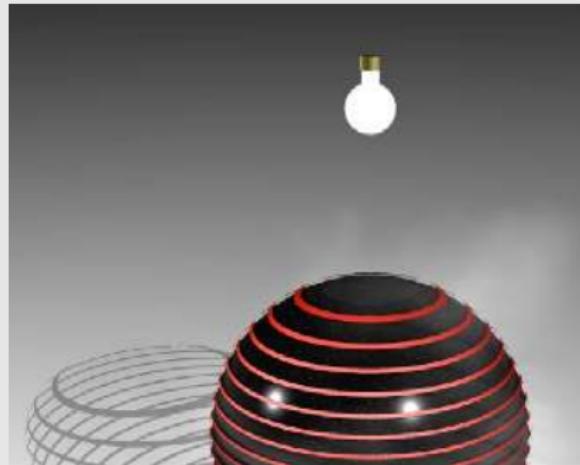
Die Breitenparallele werden als konzentrische Kreise abgebildet.



Projektion schräg zu den Schnittebenen

Zentralprojektion der Kreise auf eine Ebene schräg zu den Schnittebenen:

Die Breitenparallele werden als Ellipsen abgebildet.

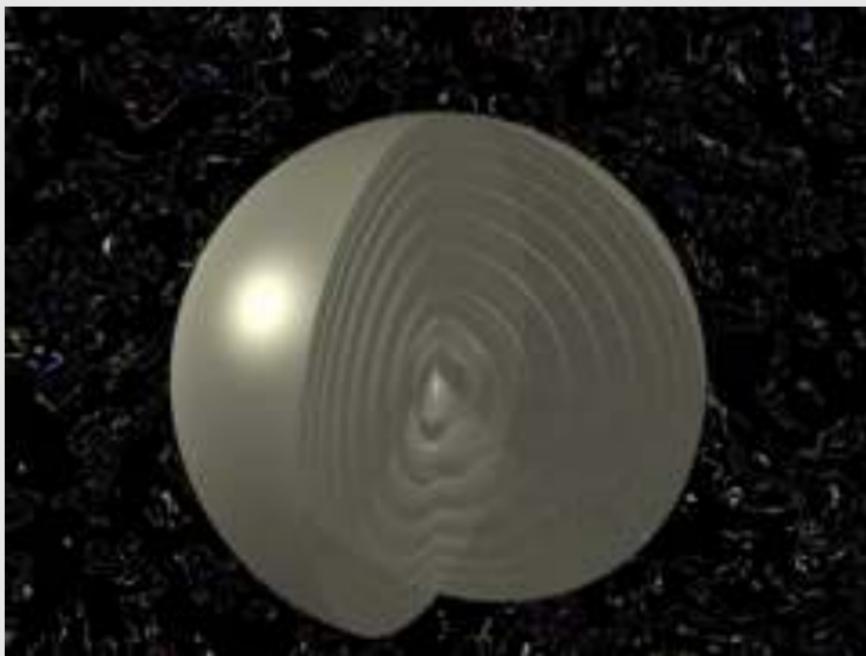


Eine 4D-Einheitskugel im Ursprung:

$$\begin{aligned}x_1^2 + x_2^2 + x_3^2 + x_4^2 &= 1 \\x_1^2 + x_2^2 + x_3^2 &= 1 - x_4^2\end{aligned}$$

- ▶ Für konstante Werte von x_4 erhalten wir 3D-Kugeln, deren Radien zwischen 0 und 1 variieren und von x_4 abhängen.
- ▶ Wenn eine 4D-Kugel parallel zur x_4 -Richtung senkrecht unser 3D-Universum durchstößt, dann sehen wir eine Folge von Kugeln, die die Oberfläche der 4D-Kugel bilden.
- ▶ Die Oberfläche der 4D-Kugel besteht also aus unendlich vielen Kugeln mit unterschiedlichen Radien, genauso wie die Oberfläche der 3D-Kugel aus unendlich vielen Kreisen besteht.

Eine 4D-Einheitskugel im Ursprung:



Eine 4D-Kugel geschnitten von einer 3D-Hyperebene

Eine 4D-Einheitskugel im Ursprung

geschnitten mit einer Hyperebene in beliebiger Lage

Einheitskugel: $x_1^2 + x_2^2 + x_3^2 = 1 - x_4^2$

Hyperebene: $Ax_1 + Bx_2 + Cx_3 + x_4 + \delta_i = 0$

einsetzen für x_4^2 : $(Ax_1 + Bx_2 + Cx_3 + \delta_i)^2 = x_4^2$

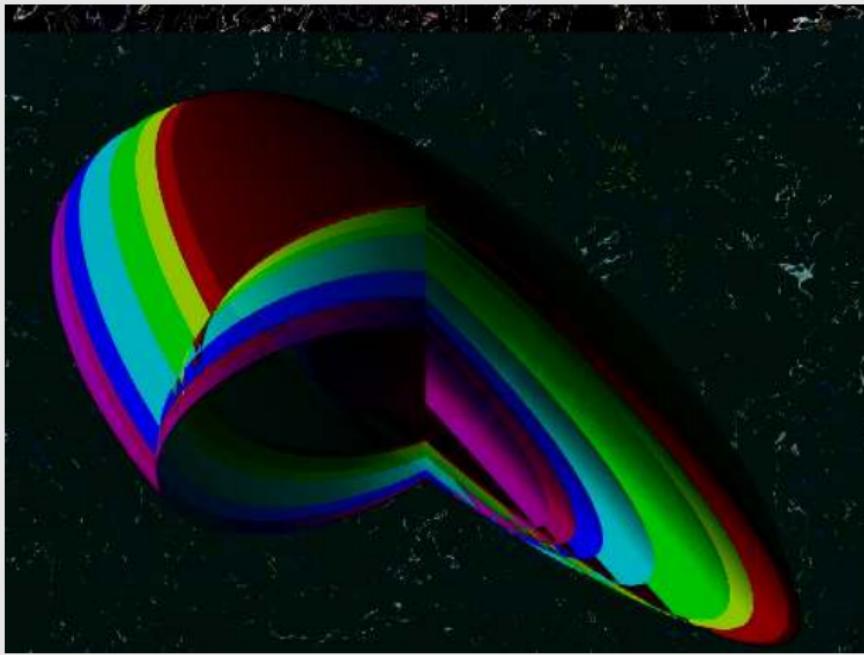
$$(1 + A^2)x_1^2 + (1 + B^2)x_2^2 + (1 + C^2)x_3^2 + \delta_i^2$$

$$+ 2ABx_1x_2 + 2ACx_1x_3 + 2BCx_2x_3$$

$$+ 2A\delta_i x_1 + 2B\delta_i x_2 + 2C\delta_i x_3 = 1$$

Die Projektionen der Schnitte sind also gescherte Ellipsoide.

Ellipsoide auf der Oberfläche einer 4D-Kugel



Eine 4D-Kugel, geschnitten von einer schrägen 3D-Hyperebene

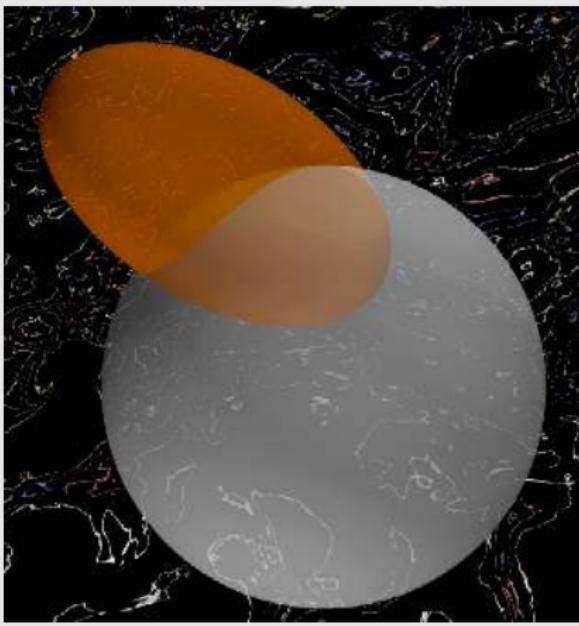
Eine 4D-Kugel am Hauptgebäude

Movie: Blue Screen: Eine 4D-Kugel am Hauptgebäude

Eine 4D-Kugel am Informatikum

Movie: Blue Screen: Eine 4D-Kugel am Informatikum

Ein Ellipsoid durchdringt eine gekrümmte Fläche



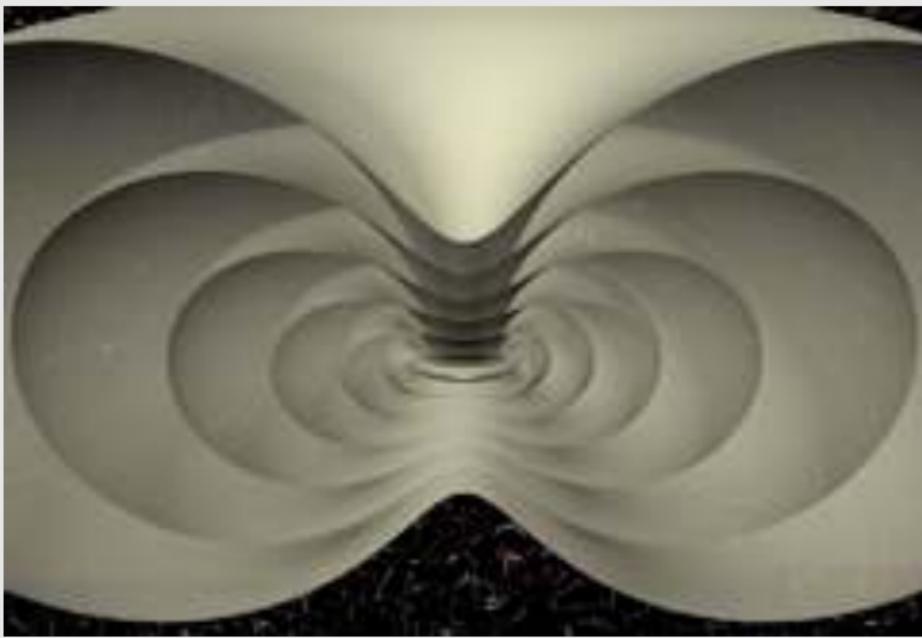
Noch ein Gedankenexperiment:

Wir stellen uns wieder vor, wir wären zweidimensionale Wesen in einer zweidimensionalen Welt, aber diesmal leben wir in einer Welt, die gegenüber der dritten Dimension gekrümmmt ist, beispielsweise in einer Kugeloberfläche.

Unsere Welt wird von einem Ellipsoid durchdrungen.

- ▶ Die Grenze des Durchdringungsbereichs projiziert sich jetzt als Kurve vierten Grades auf die x_1, x_2 -Ebene.

Projizierte Hopf-Kreise



Teil VI

Das Licht

Das Licht



16

Elektromagnetische Strahlung

- Das Elektromagnetisches Spektrum
- Interferenz und Brechung
- Strahlungsgesetze

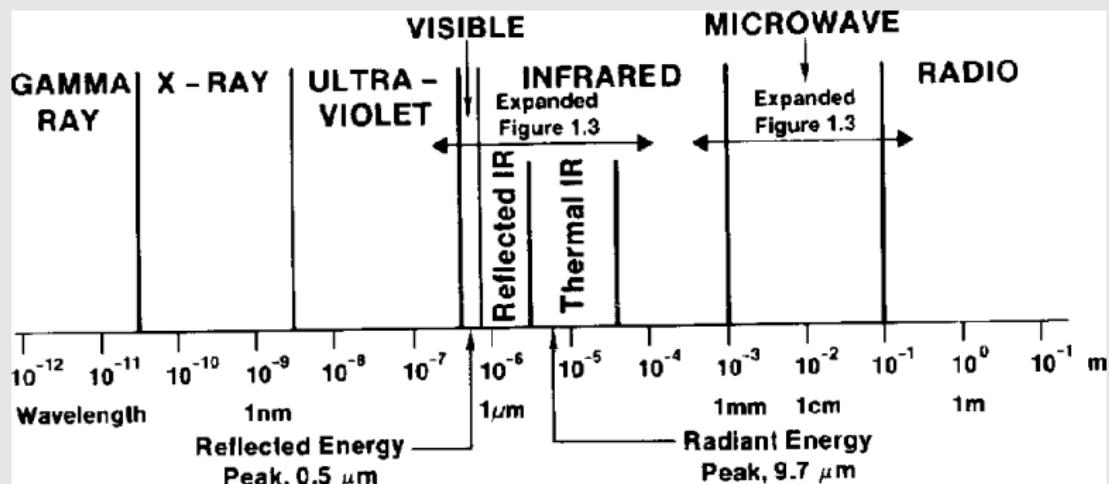
17

Licht und Oberflächen

18

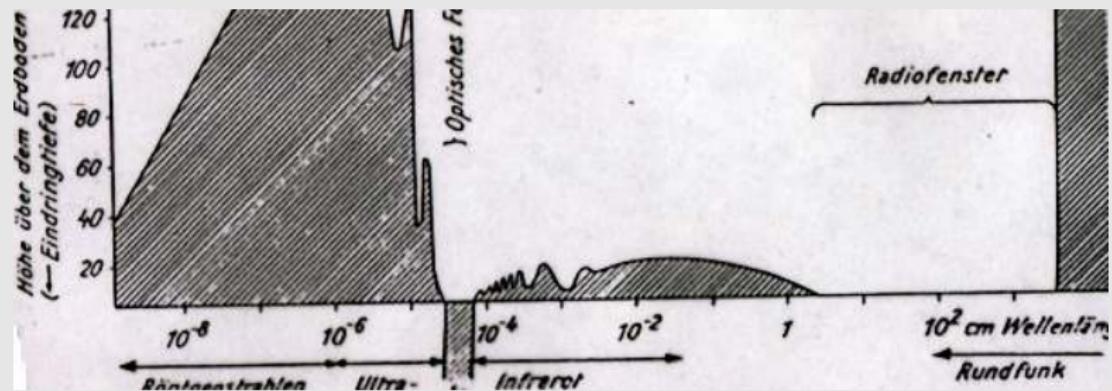
Umgebungseffekte

Das elektromagnetische Spektrum



Das elektromagnetische Spektrum

Die Eindringtiefe in die Atmosphäre, optisches Fenster und Radarfenster



Spektrale Bänder

Gammastrahlen	< 0.03 nm
Röntgenstrahlen	0.03 – 3.0 nm
UV-Strahlen	0.03 – 0.4 μm
Photographisches UV-Band	0.3 – 0.4 μm
Sichtbares Licht	0.4 – 0.7 μm
Infrarot	0.7 – 100 μm
Photographisches IR-Band	0.7 – 3 μm
Thermisches IR-Band	3 – 5 μm, 8 – 14 μm
Mikrowellen	0.1 – 30 cm
Radar	0.1 – 30 cm
Radio	> 30 cm

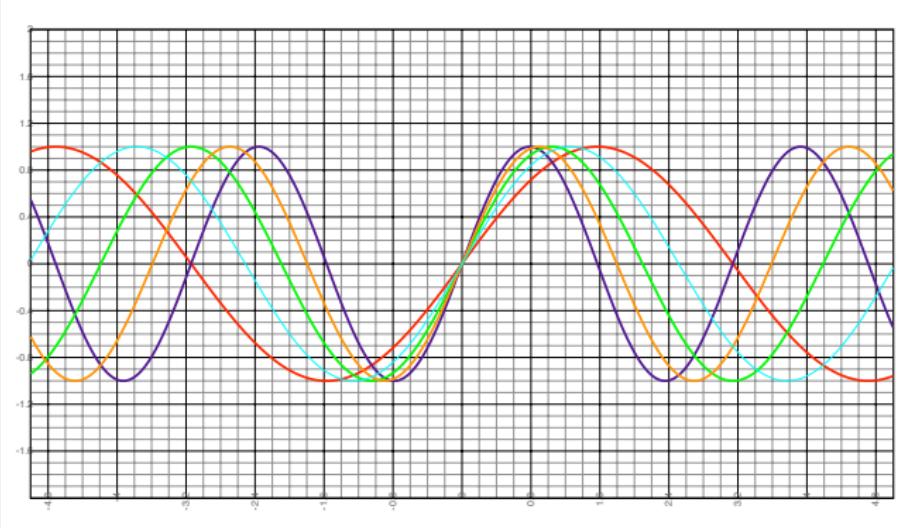
Eine planare elektromagnetische Welle

Die Schwingung

Elektrisches Feld und Magnetfeld

Die Felder

Licht als elektromagnetische Welle



Weißes Licht ist ein Gemisch elektromagnetischer Wellen unterschiedlicher Wellenlängen.

Rotes Licht

Rotes Licht

Weiβes Licht

Maske

Modelle für das Phänomen „Licht“

Zur Erinnerung: Zwei Modelle für das Phänomen „Licht“:

Licht als elektromagnetische Welle: Wenn Licht als elektromagnetische Welle modelliert wird, lassen sich Phänomene wie Beugung oder Brechung erklären.

Lichtquanten (Photonen): Wenn Licht als Strahlung von Teilchen unterschiedlicher Energie modelliert wird, lassen sich Phänomene der Wechselwirkung zwischen Materie und Licht (Absorption, Streuung, Lumineszenz usw.) erklären.

Wellenlänge, Frequenz und Geschwindigkeit

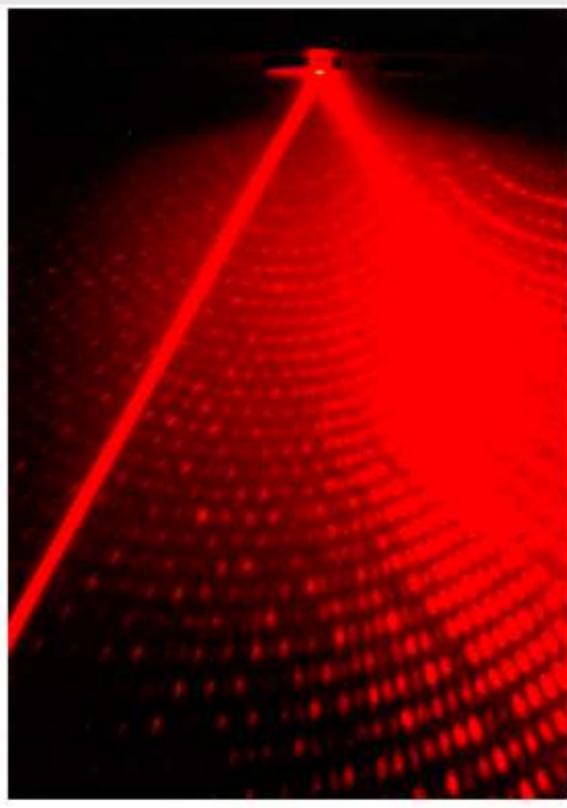
- ▶ Licht hat eine Wellenlänge λ , eine Frequenz ν und eine Geschwindigkeit c .

$$c = \lambda \cdot \nu$$

- ▶ Licht verschiedener Wellenlängen wird als verschiedenfarbig wahrgenommen.
- ▶ Die Lichtgeschwindigkeit c ist von dem Medium abhängig, durch das sich das Licht bewegt, und am größten im Vakuum: $c = 299790\text{km/s}$.
- ▶ Ein Lichtquant hat eine Energie E

$$E = h \cdot \nu = \frac{h \cdot c}{\lambda}, \text{ wobei } h = 6,6260693 \cdot 10^{-34} \text{ J s}$$

Interferenz



Lichtbrechung beim Übergang Luft / Wasser



Beispiel: Diamanten

Maske

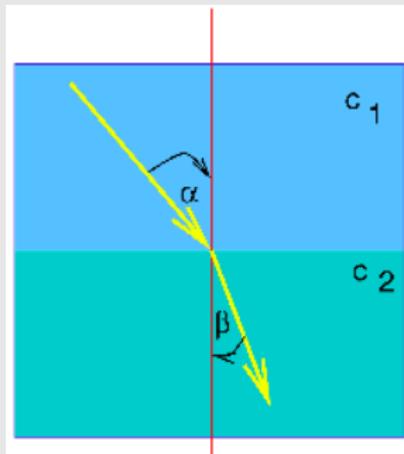
Die Brechzahl (der Brechungsindex) n

α : Austrittswinkel im Medium 1

β : Eintrittswinkel im Medium 2

c_1, c_2 : Geschwindigkeit der Welle in den Medien 1 und 2

n : Brechzahl



Satz (Brechungsgesetz)

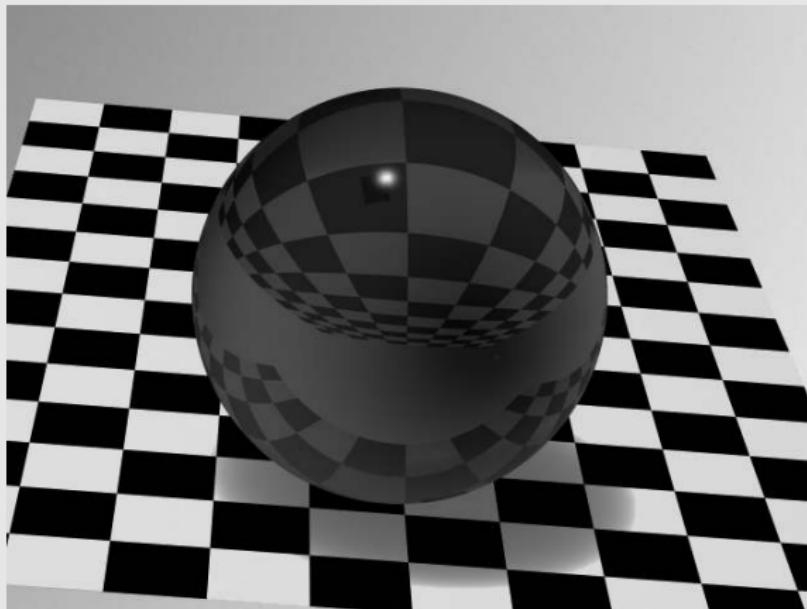
$$\frac{\sin \alpha}{\sin \beta} = \frac{c_1}{c_2}$$
$$n = \frac{c_1}{c_2} \quad (1)$$

Lichtbrechung in POV-Ray

- ▶ Der Brechungsindex n von transparenten Objekten wird seit der POV-Ray-Version 3.5 in einem **interior**-statement spezifiziert.
- ▶ Vordefinierte Brechungsindizes und weitere Konstanten finden Sie im file consts.inc.
- ▶ Vordefinierte transparente Materialien finden Sie in glass.inc.
- ▶ Beispiel: Eine Kugel mit dem Brechungsindex eines Diamanten:

```
#include "consts.inc"
sphere // transparent sphere, radius 0, made of
{ 0, 1 pigment {White filter 0.95}
  hollow
  interior { ior Diamond_Ior }
}
```

Eine Diamantkugel



*Persistence Of Vision raytracer version 3.5 sample file
(crystal.pov). A synthetic photograph by Dan Farmer.*

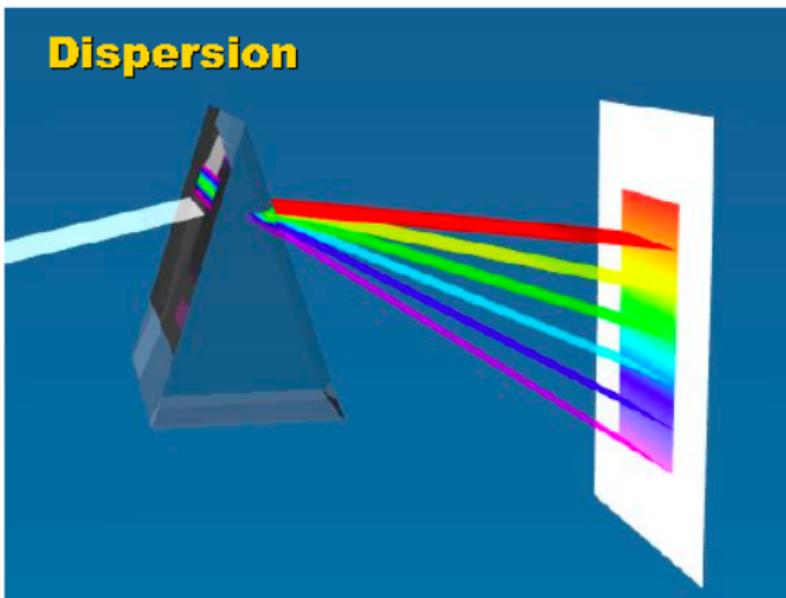
Beachte:

- ▶ Die Lichtgeschwindigkeit in einem bestimmten Material ist eine *Funktion der Wellenlänge des Lichtes*.
- ▶ Daher hängt auch die Brechzahl n von der Wellenlänge ab.
- ▶ Bei sichtbarem Licht nimmt die Brechzahl von transparenten Materialien mit zunehmender Wellenlänge ab.
- ▶ **Blaues Licht** wird daher von Glas stärker gebrochen als **rotes Licht**.
- ▶ Bei der Angabe einer Brechzahl ist daher auch die Wellenlänge anzugeben, für die diese Brechzahl gilt. Als Referenzwellenlängen werden die Wellenlängen der Fraunhoferlinien C, D, F des Wasserstoffs verwendet.

Dispersion

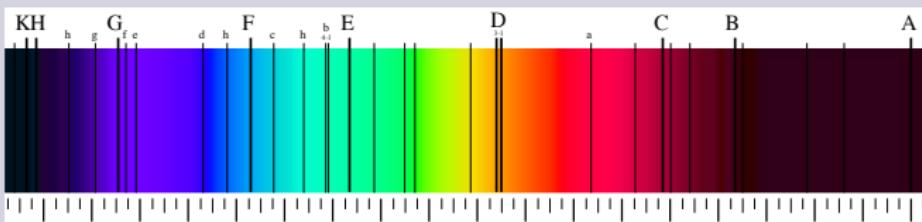
- ▶ Die Komponenten eines Gemischs von Licht unterschiedlicher Wellenlängen werden unterschiedlich stark gebrochen.
- ▶ Beim Durchgang von weißem Licht durch ein Prisma tritt daher eine Dispersion ein.
- ▶ Die Brennpunkte von Linsen aus nur einem Material sind für jede Wellenlänge anders (Chromatische Aberration).
- ▶ Zur Korrektur von solchen Farbfehlern werden Objektive aus mehreren Linsen mit unterschiedlicher Brechzahl hergestellt.

Dispersion von weißem Licht beim Durchgang durch ein Prisma



Definition (Dispersion)

Unter Dispersion versteht man die Wellenlängenabhängigkeit der Fortpflanzungsgeschwindigkeit einer Wellenbewegung, und damit auch der Brechzahl eines beliebigen Materials. Seien n_C , n_D , n_F die Brechzahlen eines Materials für Licht der C, D, F-Fraunhoferlinien:



Mittlere Dispersion D : $D = n_F - n_C$.

Relative Dispersion ω : $\omega = \frac{n_F - n_C}{n_D - 1}$.

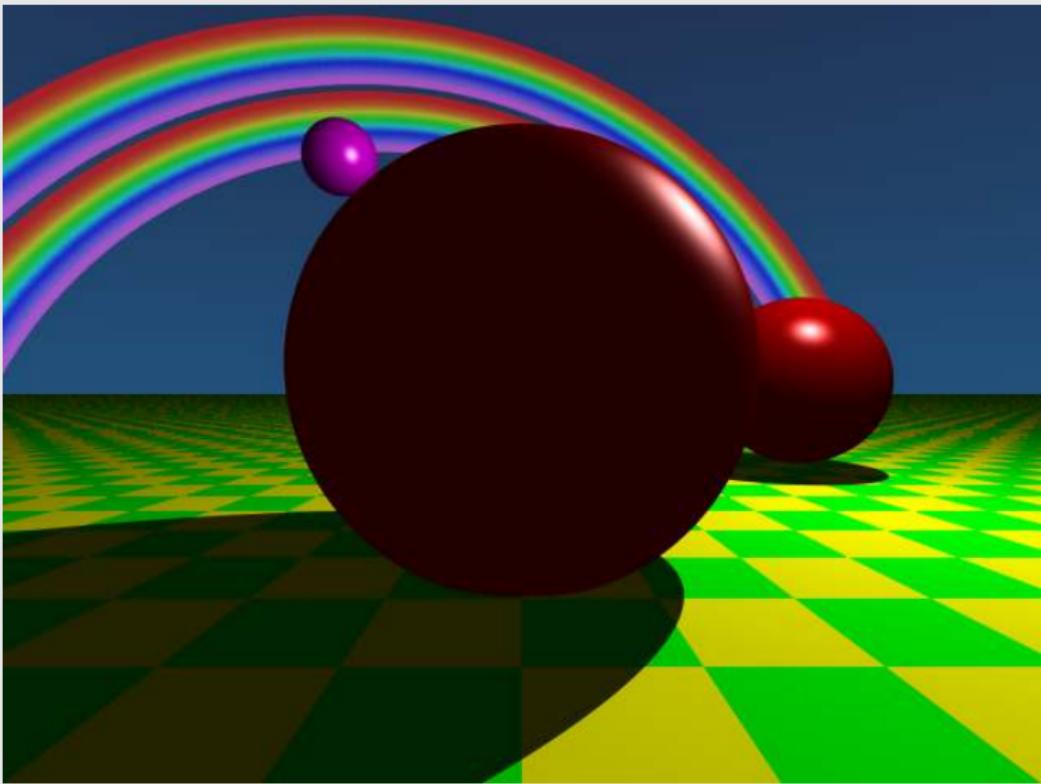
Abbesche Zahl ν : $\nu = \frac{1}{\omega}$.

Ein Regenbogen



Hast-Paß, Neuseeland

Ein Regenbogen in POV-Ray (Beispiel: 33)



Regenbogen in POV-Ray

Ein Regenbogen wird mit folgenden Parametern spezifiziert:

angle: Der Winkeldurchmesser des Bogens

width: Die Breite des Ringes; angle und width bestimmen die Form des Regenbogens

direction: Die Richtung zur Lichtquelle, bestimmt die Position des Regenbogens am Himmel

distance: Die Entfernung

colour_map: Die Tabelle der Regenbogenfarben, kann beliebig sein

Beispiel

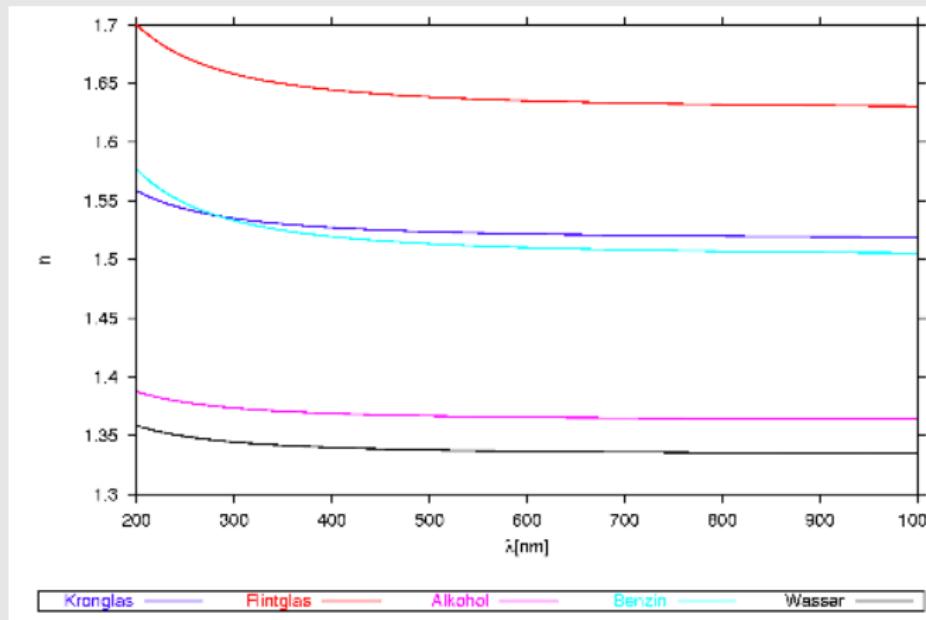
```
rainbow {  
  angle 42.5  
  width 5  
  distance 1.0e7  
  direction <-0.2, -0.2, 1>  
  jitter 0.01  
  colour_map {  
    [0.000 colour r_violet1]  
    [0.100 colour r_violet2]  
    [0.214 colour r_indigo]  
    [0.328 colour r_blue]  
    [0.442 colour r_cyan]  
    [0.556 colour r_green]  
    [0.670 colour r_yellow]  
    [0.784 colour r_orange]  
    [0.900 colour r_red1]  } }
```

Brechzahlen für transparente Materialien

Brechzahl n

Fraunhoferlinie $\lambda[\text{\AA}]$	C	D	F	Relative Dispersion ω
Kronglas	1.514	1.517	1.524	0.0193
Flintglas	1.622	1.627	1.639	0.0271
Alkohol	1.361	1.363	1.367	0.0165
Benzin	1.497	1.503	1.514	0.0338
Wasser	1.332	1.334	1.338	0.0180

Sellmeierformel: Die Brechzahl in Abhangigkeit von der Wellenlange



Dispersion in POV-Ray

- ▶ In der **interior**-Spezifikation kann neben dem Brechungsindex auch die Dispersion spezifiziert werden.
 - ▶ dispersion: Das Verhältnis der Brechungsindizes von Violett to Rot.
 - ▶ dispersion_samples: Zahl der Farbstufen, mit denen das Spektrum gerechnet wird,
 - ▶ caustic: Simulierte Kaustiken (Lichtbündelungseffekte)
 - ▶ **fade_distance** Die Entfernung, in der das Licht auf die Hälfte abgefallen ist, der voreingestellte Wert ist sieben.
 - ▶ **fade_power** die Geschwindigkeit, mit der das Licht abfällt.
 - ▶ fade_color die Farbe, die das Licht durch die Absorption erhält.

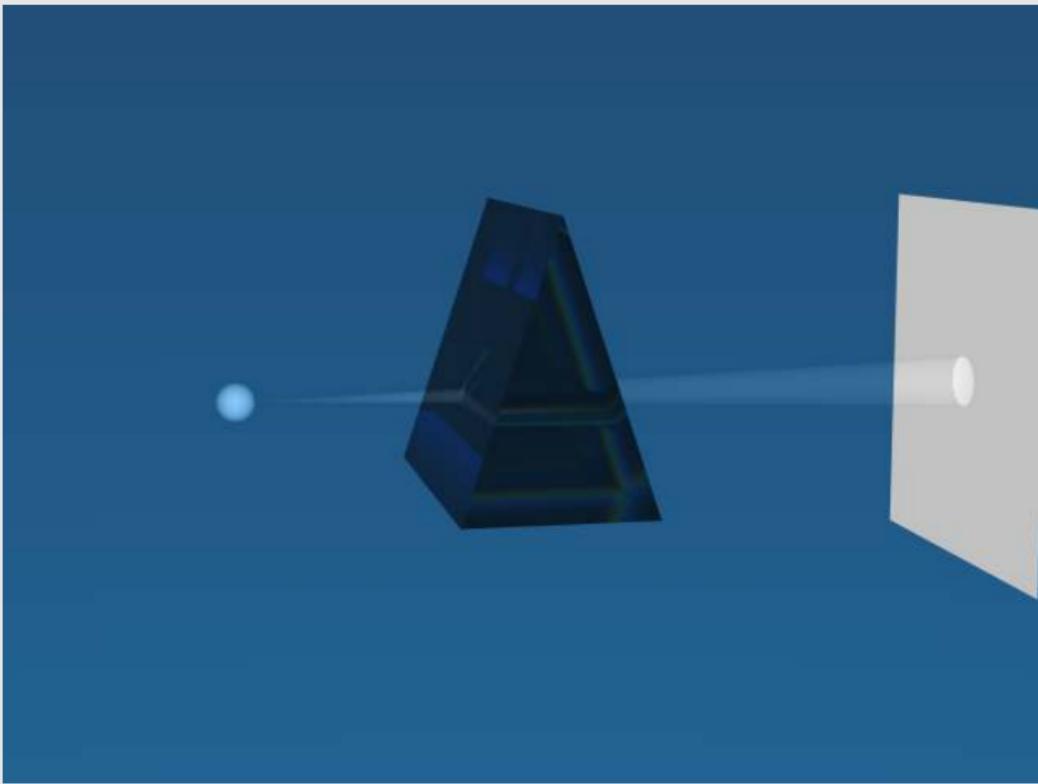
Beispiel: Das Innere eines Prismas

Beispiel

Brechung mit Dispersion: Eine **interior**-Spezifikation, um auch die Dispersion sichtbar zu machen.

```
interior { ior Diamond_ior  
          dispersion 2 //Diamond Dispersion =  
          fade_distance 1  
          fade_power 2  
          caustics 1.0  
          dispersion_samples 100 }
```

Der Lichweg durch ein Prisma (Beispiel: 34)



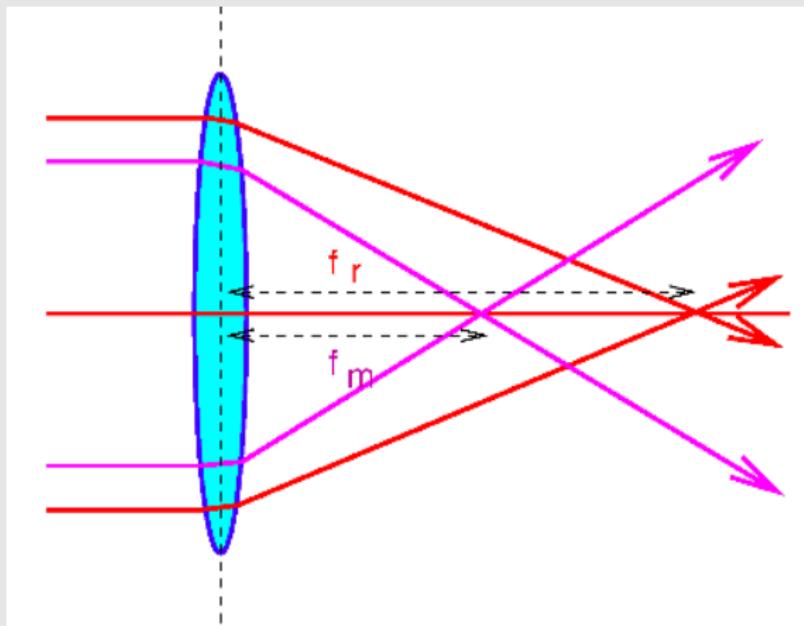
Diamanten mit Dispersion und Kaustiken

Beispiel

Interior-Spezifikation mit dispersiver Transmission

```
#declare diamondInterior = interior {
    ior Diamond_ior
    dispersion 2 //Diamond Dispersion = 1.035
    fade_distance 1
    fade_power 2
    caustics 1.0
    dispersion_samples 100}
```


Chromatische Aberration bei einer Sammellinse



Chromatische Aberration von Linsen

Definition (Chromatische Aberration von Linsen)

Seien f_C, f_D, f_F die Brennweiten einer Linse für Licht der C, D, F-Fraunhoferlinien und n_c, n_D, n_F die entsprechenden Brechzahlen.

Die **chromatische Aberration A** einer Linse ist die Differenz

$$A = f_C - f_F = \frac{n_F - n_C}{n_D - 1} f_D = \omega f_D.$$

Die **relative Dispersion ω** ist nur vom Material der Linse abhängig, nicht von ihrer Form:[Alonso and Finn, 1972]

$$\omega = \frac{A}{F_D} = \frac{f_C - f_F}{f_D} = \frac{n_F - n_C}{n_D - 1}.$$

Lichtquellen und Strahlungsgesetze

Lichtquellen, die aufgrund einer hohen Temperatur Licht abstrahlen, lassen sich häufig als **schwarze Körper** modellieren.

Definition (**Schwarze Strahlung**)

Idealisierter Grenzfall der Strahlung eines Körpers, wobei die Energieverteilung durch das **Plancksche Strahlungsgesetz** beschrieben wird.

Definition (**Schwarzer Körper**)

Ein Körper, der eine schwarze Strahlung aussendet.

Wiensches Verschiebungsgesetz:

Satz (Wiensches Verschiebungsgesetz:)

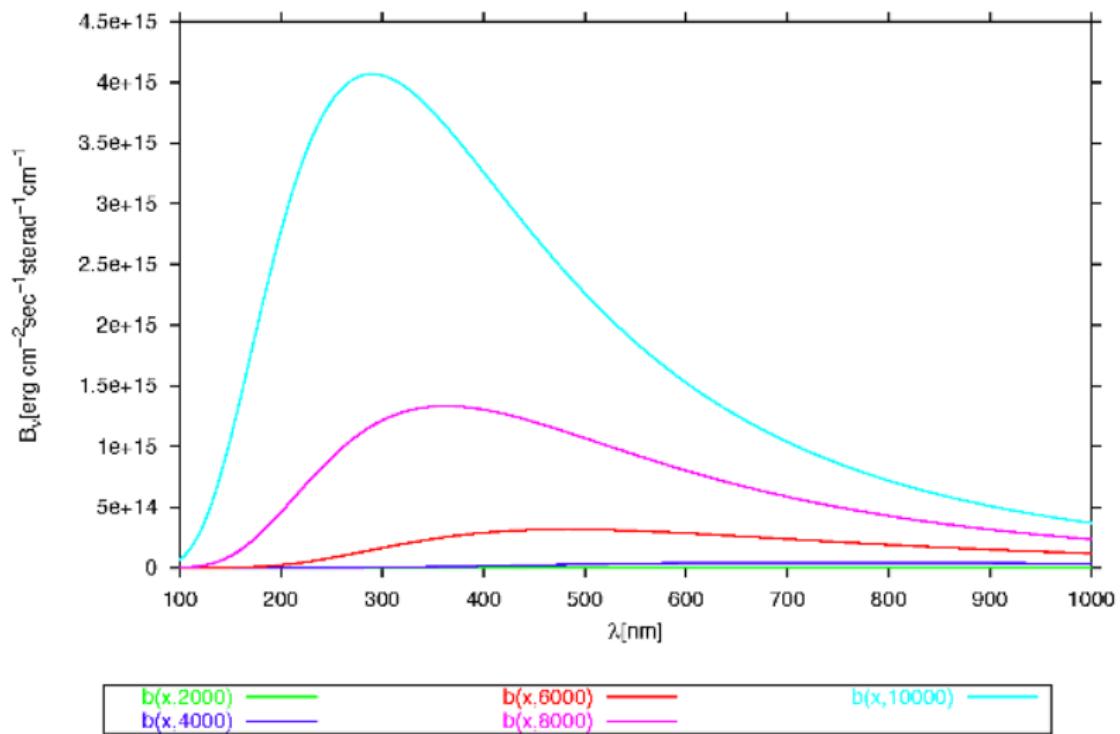
Je heißer ein schwarzer Körper ist, desto mehr verschiebt sich das Maximum λ_{max} der Strahlung zu den kürzeren Wellenlängen:

$$\lambda_{max} \cdot T = const = 0.2897 \text{ cm grad}$$

Schwarzer Körper: $T=273\text{K}$ bis $T=25000\text{K}$

Maske

Energiedichte der Strahlung



Temperatur und Strahlungsleistung von schwarzen Körpern

Strahlungsleistung von schwarzen Körpern

Temperatur [K]	λ_{max} [Å]	Farbe	Strahlungsleistung [$\frac{\text{Watt}}{\text{cm}^2}$]
1 000	29 000	infrarot	5,8
4 000	7 200	rot	1 500
7 000	4 120	violett	14 000
10 000	2 900	UV	58 000
1 000 000	29	Röntgenstrahlung	$5.8 \cdot 10^{12}$

Farbtemperatur

- ▶ Nach dem Wienschen Verschiebungsgesetz kann man jeder Lichtquelle entsprechend ihres Strahlungsmaximums eine Farbtemperatur zuordnen.
 - ▶ Licht niedriger Farbtemperatur mit einem Strahlungsmaximum bei den **gelben** oder **roten** Wellenlängen wird als **warmes Licht** empfunden,
 - ▶ Licht hoher Farbtemperatur mit dem Maximum im **Blau-Grün-Bereich** wird als **kalt** empfunden.
- ▶ Das Sonnenlicht entspricht einem schwarzen Strahler der Temperatur 5785 K. Die spektrale Zusammensetzung des Sonnenlichts empfinden wir als weißes Licht.

Weißabgleich

- ▶ Beim Photographieren sollte eine Kamera immer auf die Lichtverhältnisse eingestellt werden.
- ▶ Der Weißabgleich stellt die spektrale Empfindlichkeit auf das Umgebungslicht ein.
- ▶ Ohne Weißabgleich werden Gegenstände, die wir als weiß wahrnehmen, farbig abgebildet.
- ▶ Die Farbtemperatur von Blitzlicht entspricht dem Tageslicht.

Farbtemperatur

Definition (Farbtemperatur)

Die Farbtemperatur einer Lichtquelle ist nach Hyde (1910) die Temperatur des schwarzen Körpers, bei der dieser ein Licht gleicher Farbe wie die betreffende Lichtquelle aussendet.

- ☞ Die Farbtemperatur soll nur die Farbe einer Lichtquelle beschreiben, nicht deren spektrale Zusammensetzung.
- ☞ Lichtquellen, deren Farbe nicht genau der einer schwarzen Strahlung gleicht, wird eine ähnliche Farbtemperatur zugeordnet. (DIN 5033, Bl.1, 1953)

Farbe

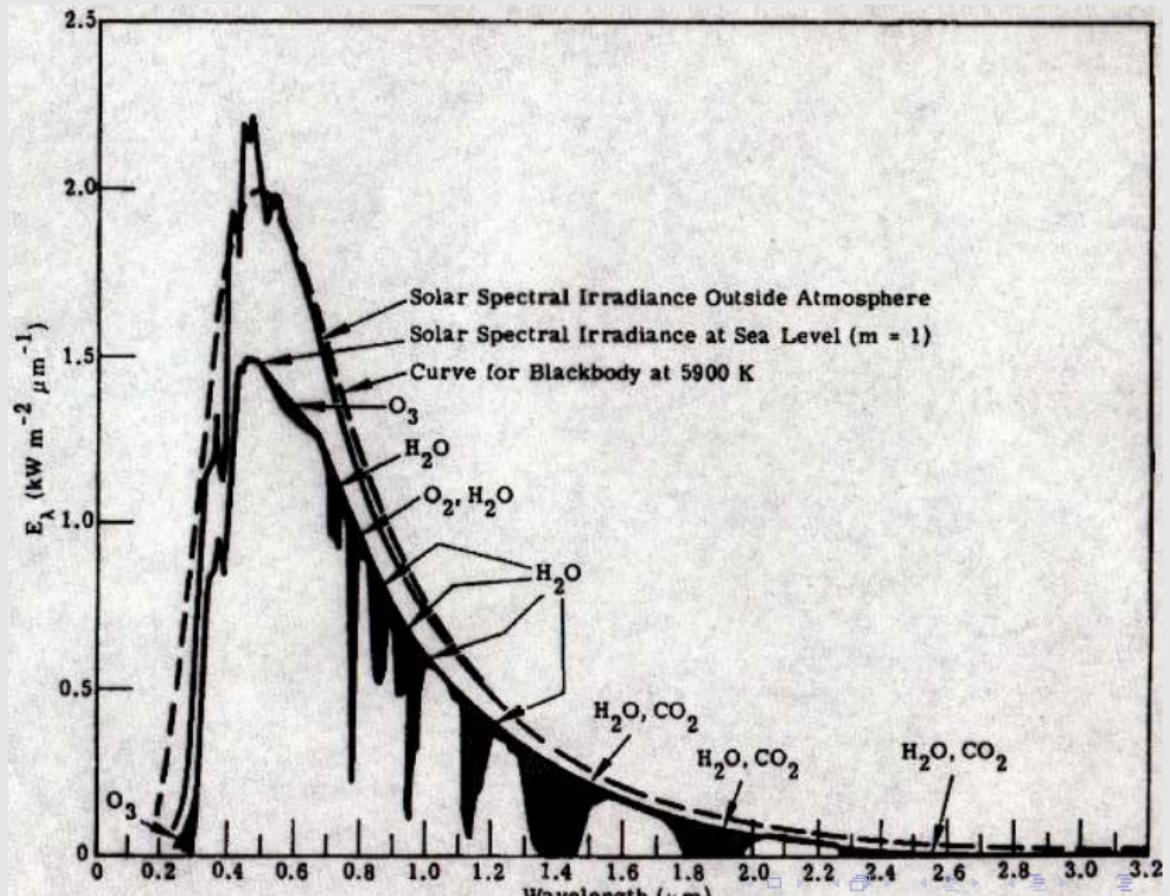
Definition (**Farbe**)

ist eine Empfindung, die durch sichtbare Strahlung über das Sinnesorgan „Auge“ ausgelöst wird.

Lichtquelle und Farbtemperatur

Lichtquelle	Temperatur
Kohlefadenlampe Goebel, Edison (1879)	2 100 °C
Wolframglühlampe	2 400 °C
Wolframlampe (Argon, Stickstoff) nach Langmuir	2 750 °C
Bogenlampe	4 000 – 5 000 °C
Sonne	≈ 5600 °C
Wega, Sirius	≈ 10 000 °C
Spica	≈ 37 000 °C

Die Sonne als schwarzer Strahler



Die Farbe der Sterne

Die Farben der Sterne sind ein Maß für deren Oberflächentemperatur:



Die Farbe von Objekten

Die Farbe, in der wir einen Gegenstand wahrnehmen, hängt ab:

- ☞ von der spektralen Zusammensetzung der Beleuchtung,
- ☞ von den Pigmenten in der Oberfläche, die einen Teil des einfallenden Lichtes absorbieren,
- ☞ von der Auswertung in unserem visuellen System.

Um die Farbe eines Gegenstandes beurteilen und mit anderen Farben vergleichen zu können, müssen wir ihn daher bei einer standardisierten Beleuchtung betrachten, typischerweise bei Tageslicht.

Farbe des Umgebungslichtes

An der Wasseroberfläche: Alle Farben leuchten.



Farbe des Umgebungslichtes

An der Wasseroberfläche: Alle Farben leuchten.

1 m Wassertiefe: Rot fängt an zu verblassen.



Farbe des Umgebungslichtes

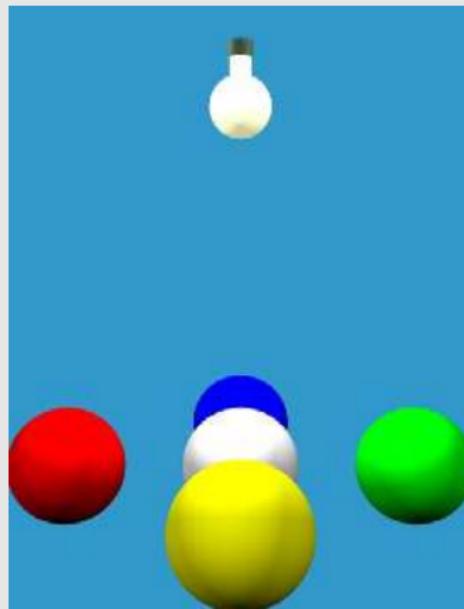
An der Wasseroberfläche: Alle Farben leuchten.

1 m Wassertiefe: Rot fängt an zu verblassen.

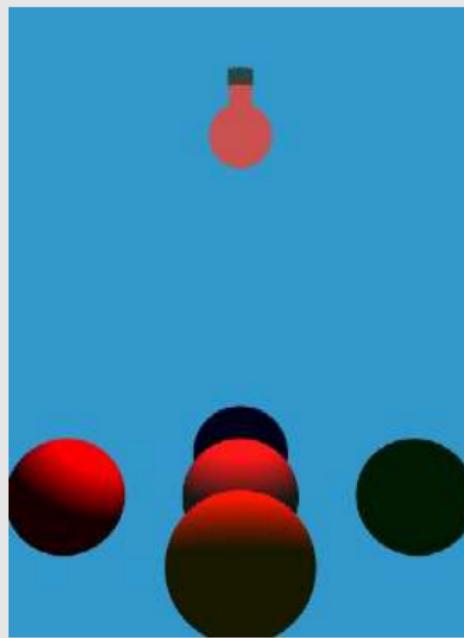
4 m Wassertiefe: Die Farben sind fast völlig verschwunden.



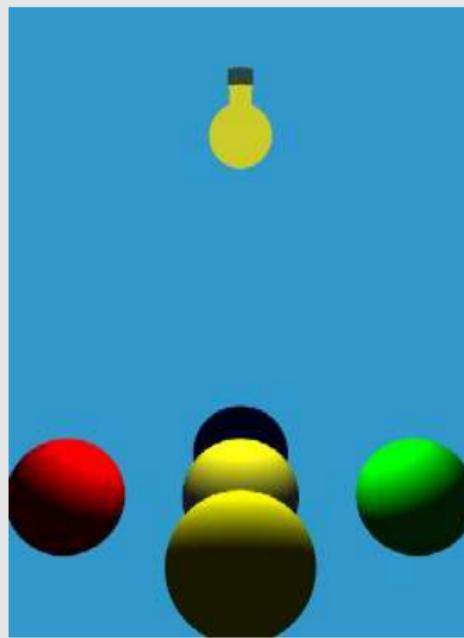
Beispiel: Vier Kugeln in weißem Licht



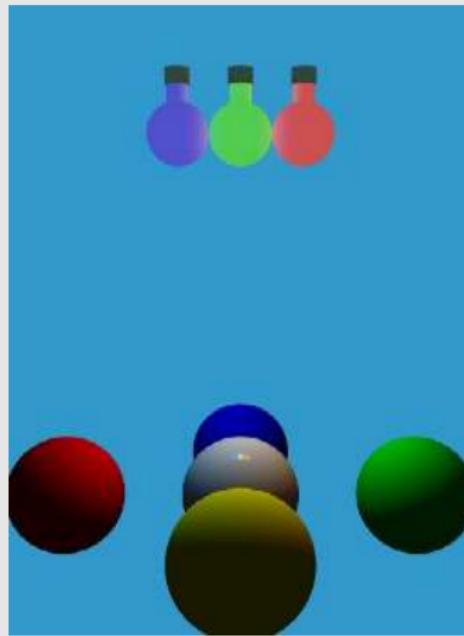
Beispiel: Vier Kugeln in rotem Licht



Beispiel: Vier Kugeln in gelbem Licht



Beispiel: Vier Kugeln in rot-grün-blau Licht



Dem Weißmacher

*Newtonisch Weiß den Kindern vorzuzeigen,
Die pädagogischem Ernst sogleich sich neigen,
Trat einst ein Lehrer auf, mit Schwungrads
Possen,
Auf selbem war ein Farbenkreis geschlossen.*

*Das dorlte nun. „Betracht es mir genau!
Was siehst du, Knabe?“ Nun, was seh ich? Grau!
„Du siehst nicht recht! Glaubst du, daß ich das
leide?
Weiß, dummer Junge, Weiß! So sagt's
Mollweide!?”*

Johann Wolfgang von Goethe

Umgebungseffekte



16 Elektromagnetische Strahlung

17 Licht und Oberflächen

18 Umgebungseffekte

- Lichtquellen
- Transmission, atmosphärische Effekte
- Atmosphärische Effekte in POV-Ray

Punktlichtquellen

Punktlichtquellen strahlen von einem Punkt aus isotrop in alle Richtungen.

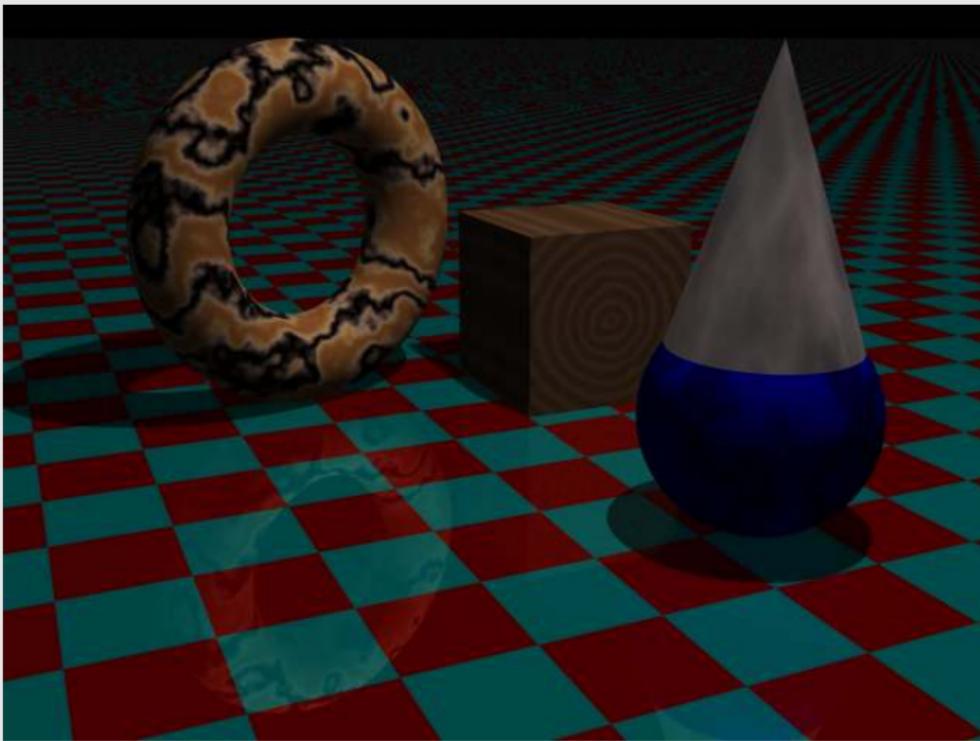
- ▶ Die Lichtstärke ist in allen Raumwinkeln gleich.
- ▶ Die Beleuchtungsstärke nimmt mit dem Quadrat der Entfernung von der Lichtquelle ab.
- ▶ Eine Punktlichtquelle erzeugt scharf begrenzte Schlagschatten.

Punktlichtquellen in POV-Ray

```
light_source {
    <2, 10, -3> /* Position */
    color White /* Farbe */
}
```

Diese Lichtquelle ergibt scharf begrenzte Schlagschatten, aber die Beleuchtungsstärke wird nicht mit der Entfernung schwächer.

Eine Punktlichtquelle



Beispiel: Punktlichtquelle



Originalversteinerung eines Labyrinthzähners (Labyrinthodon)

PANZELURCH

Schneckenhaus-Museum Goldkronach

Alter ca. 210 Mio.

aus dem Kreide-Zeit

Fundort: Nürnberg

Punktlichtquellen mit abnehmende Beleuchtungsstärke

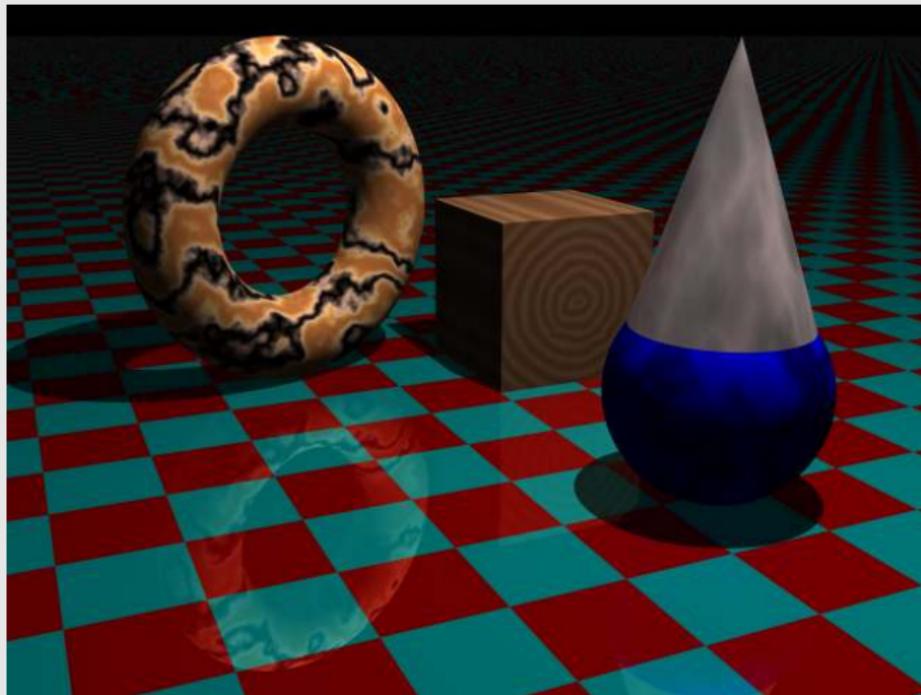
```
light_source {  
    <2, 10, -3> /* Position */  
    color White /* Farbe */  
    fade_distance 10  
    fade_power 2}
```

`fade_distance` ist die Entfernung mit maximaler Beleuchtungsstärke,

`fade_power` ist ein Exponent, der angibt, wie schnell das Licht schwächer wird — `fade_power=1` wäre eine lineare Abnahme.

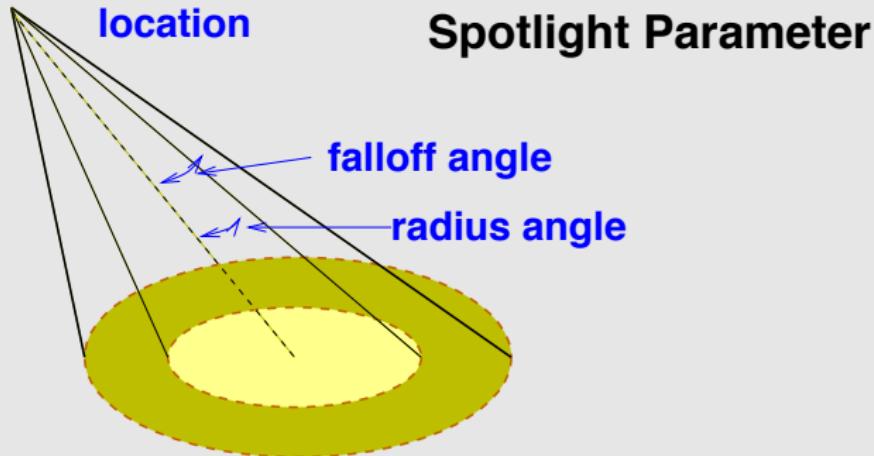
Punktlichtquelle mit abnehmender Beleuchtungsstärke

fade_distance 10, fade_power 2



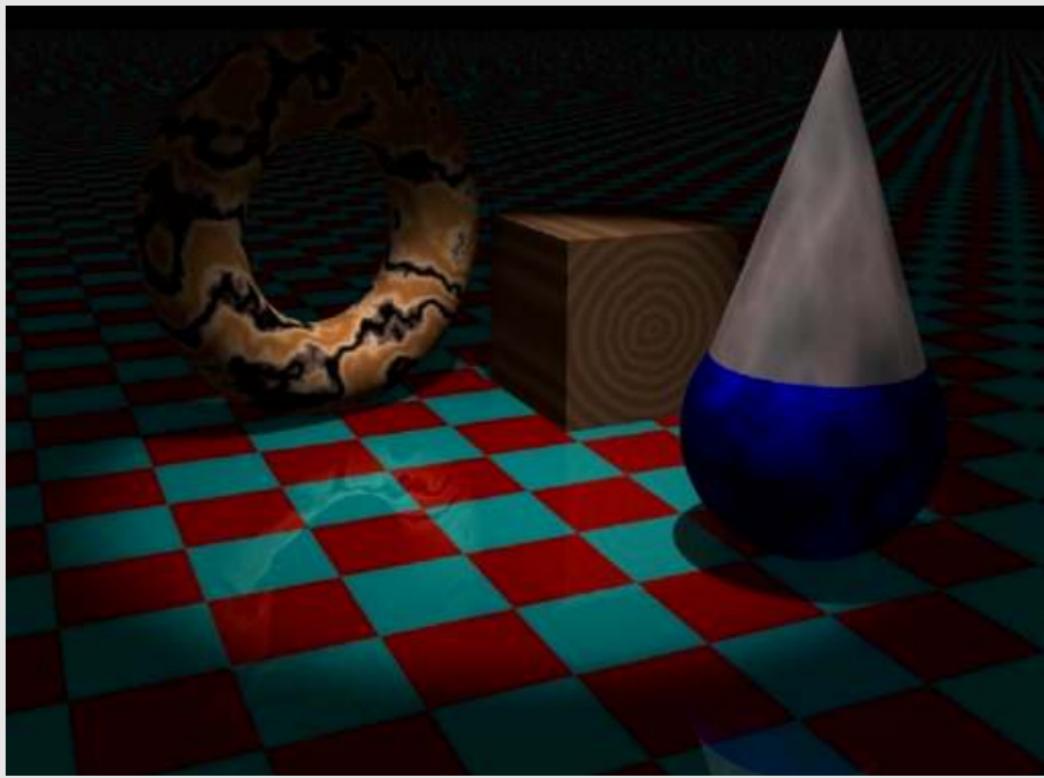
Scheinwerfer: spotlight

Scheinwerfer leuchten nur einen begrenzten Raumwinkel aus und werden in den Randbereichen schwächer.



```
light_source {
    <0,10,-3>
    color White // Weisses Licht
    spotlight // Scheinwerfer
    radius 15 // heller Kegel 15 Grad
    falloff 20 // 20 Grad Umbra
    tightness 10 // Buendelung 10
    point_at <0,0,0> // Angestrahlter Punkt
}
```

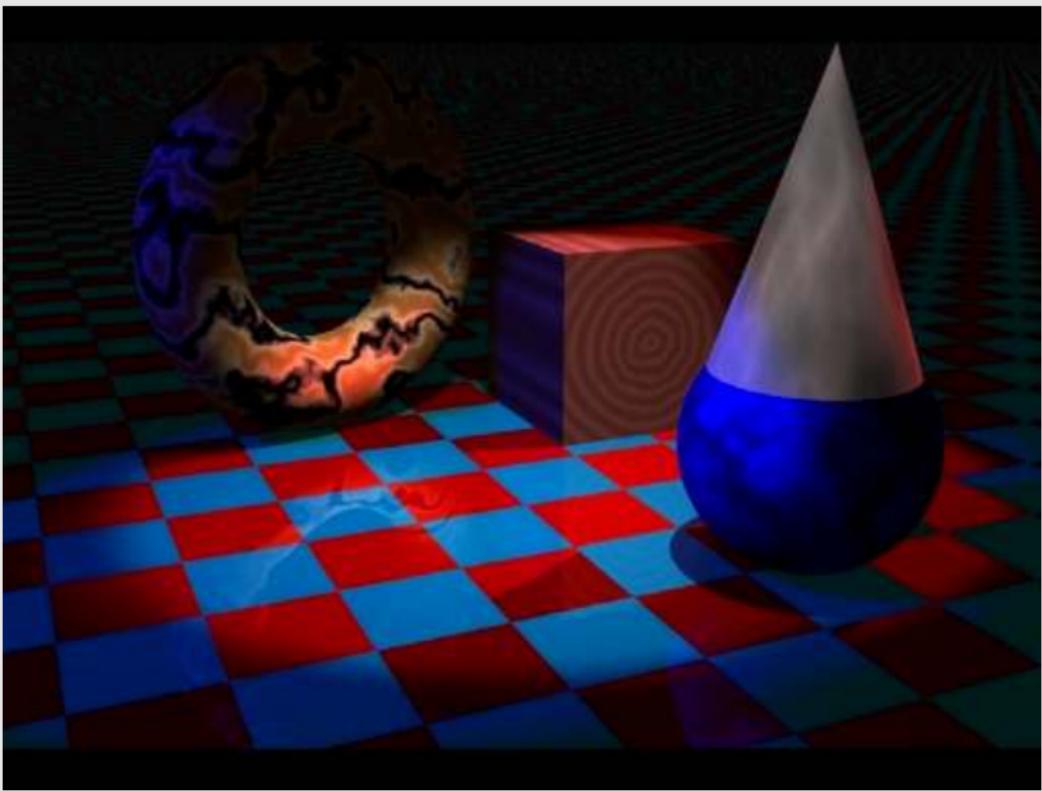
Die Szene im Scheinwerferlicht



Zwei Scheinwerfer

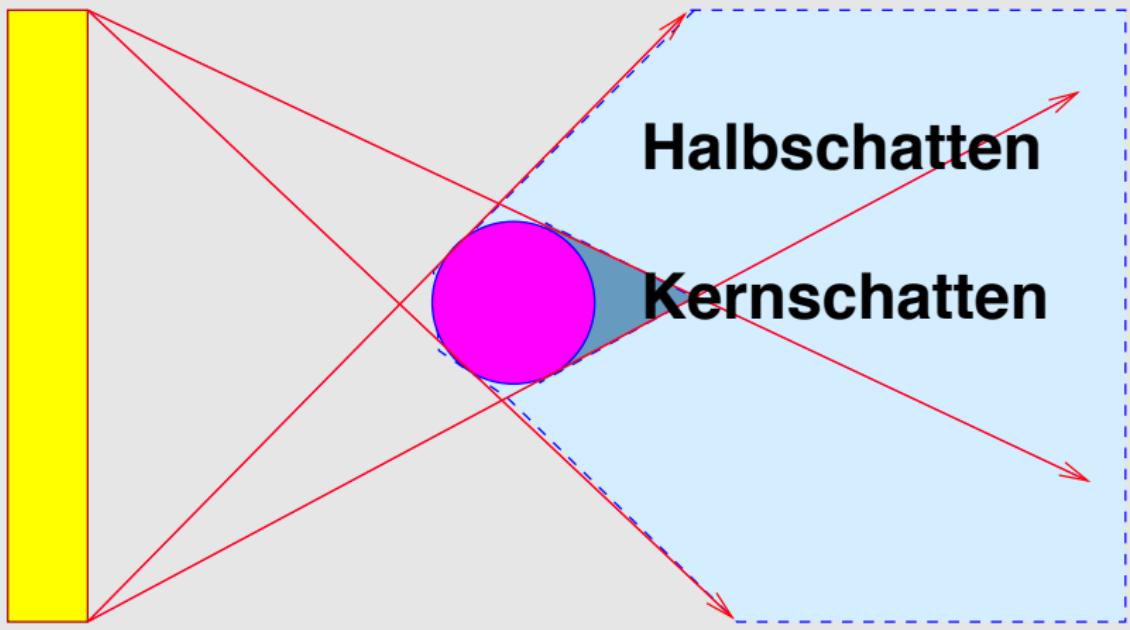
```
light_source { // spot 2
    <10,10, -1>
    color Red
    spotlight radius 12 falloff 14
    tightness 10
    point_at <2,0,0>}
light_source { // spot 3
    <-12, 10, -1>
    color Blue
    spotlight radius 12 falloff 14
    tightness 10
    point_at <-2,0,0>}
```

Drei Scheinwerfer: Weiß, rot, blau



Flächenlicht: Ausgedehnte Lichtquellen

Ausgedehnte Lichtquellen verursachen weiche Schattengrenzen, da sie auch partiell verdeckt sein können.

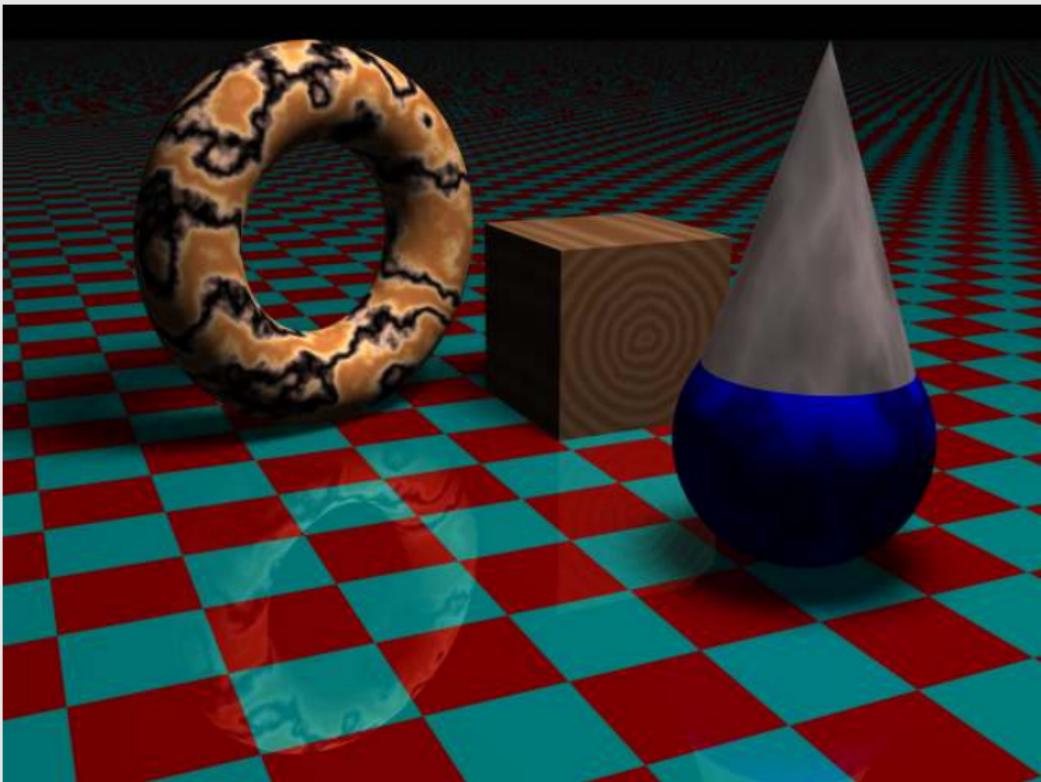


area_light

In POV-Ray gibt es keine echten Flächenstrahler, sondern sie werden als Matrix von Punktlichtquellen modelliert, hier ein 5×5 -Feld:

```
light_source { // Flaechenlicht
    <2,10,-3>
    color White
    area_light
    <5, 0, 0><0, 0, 5> 5, 5
    adaptive 1
    jitter }
```

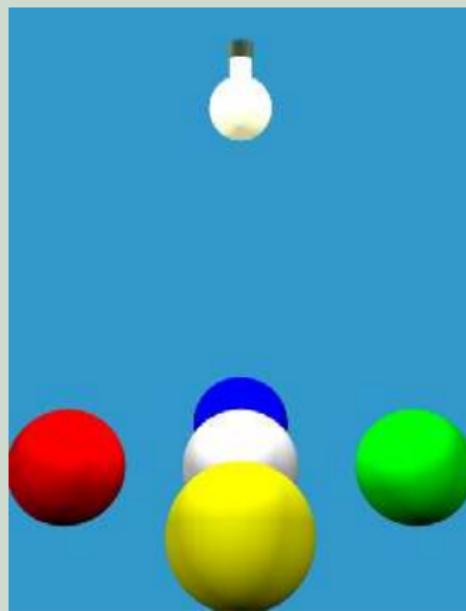
Ein Flächenlicht (area-light)



Beispiel: Eine sichtbare Lichtquelle: looks_like

Beispiel

Eine sichtbare Lichtquelle



Modellierung der Glühlampe

```
#declare Lightbulb = union {
    merge {
        sphere { 0, 1}
        cylinder { z, 0, 1
                    scale <0.35, 0.35, 1.0>
                    translate 0.5*z }
        texture {
            pigment {color rgb <1,1,0.8>}
            finish {ambient .8 diffuse .6} }
        cylinder {z, 0, 1
                    scale <0.4, 0.4, 0.5>
                    texture {Brass_Texture }
                    translate 1.5*z }
        rotate -90.0*x
        scale 1}
```

Die Glühlampe als Lichtquelle

```
light_source { y*4*sphradius color White  
looks_like { Lightbulb }}
```

Molekülstreuung: Das blaue Himmelslicht



Lhasa, Jokhang

Molekülstreuung: Rotfärbung bei tief stehender Sonne



Waikiki-Beach, Hawaii

Aerosolstreuung: Sichtbare Lichtstrahlen

Dicht am Horizont ist das Sonnenlicht roter als im Zenit.



Tahiti, Blick auf Moorea

Aerosolstreuung: Kontrastminderung bei weit entfernten Objekten



Marlborough Sound, New Zealand

Aerosolstreuung: Kontrastminderung bei weit entfernten Objekten



Huangshan-Gebirge

Aerosolstreuung: Kontrastminderung bei weit entfernten Objekten



Huangshan-Gebirge

Aerosolstreuung: Kontrastminderung bei weit entfernten Objekten



Española, Galapagosinseln

Trübung durch Staub



In der Atacama-Wüste, Nordchile

Trübung durch Staub



Die Dünens von Huacachina, Peru

Trübung durch Staub



Die Dünens von Huacachina, Peru

Nebel: fog

Beispiel

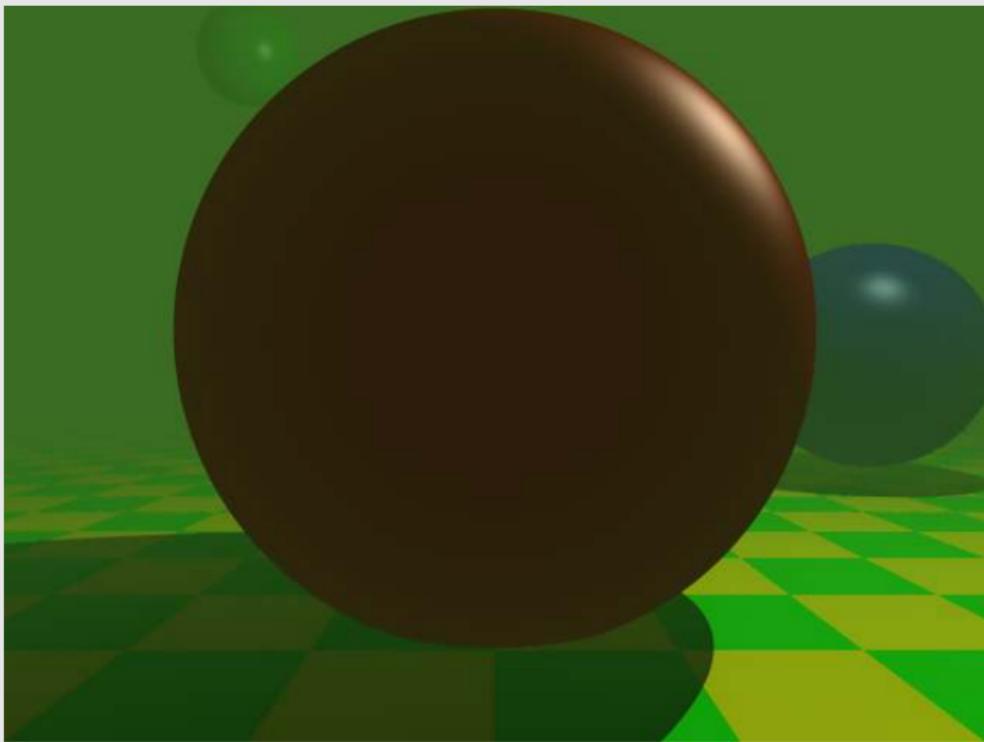
Nebel Ein konstanter Nebel wird über ein **fog**-statement in die Szene eingefügt. Parameter sind:

color: Die Farbe des Nebels.

distance: Die Sichtweite (die Entfernung, in der noch 36.8% des Hintergrunds zu sehen sind).

```
fog {  
    distance 150  
    color rgb<0.3, 0.5, 0.2> }
```

Ein einfacher Nebel



Bodennebel: ground fog

Bodennebel

Ein Nebel, der am Boden eine konstante Dichte hat und dann nach oben langsam dünner wird.

Parameter sind:

`fog_type 2`: wählt „ground fog“ aus.

`fog_offset`: Die Höhe der Bodennebelschicht mit konstanter Dichte.

`fog_alt`: Die Höhe, in der der Nebel deutlich dünner geworden ist.

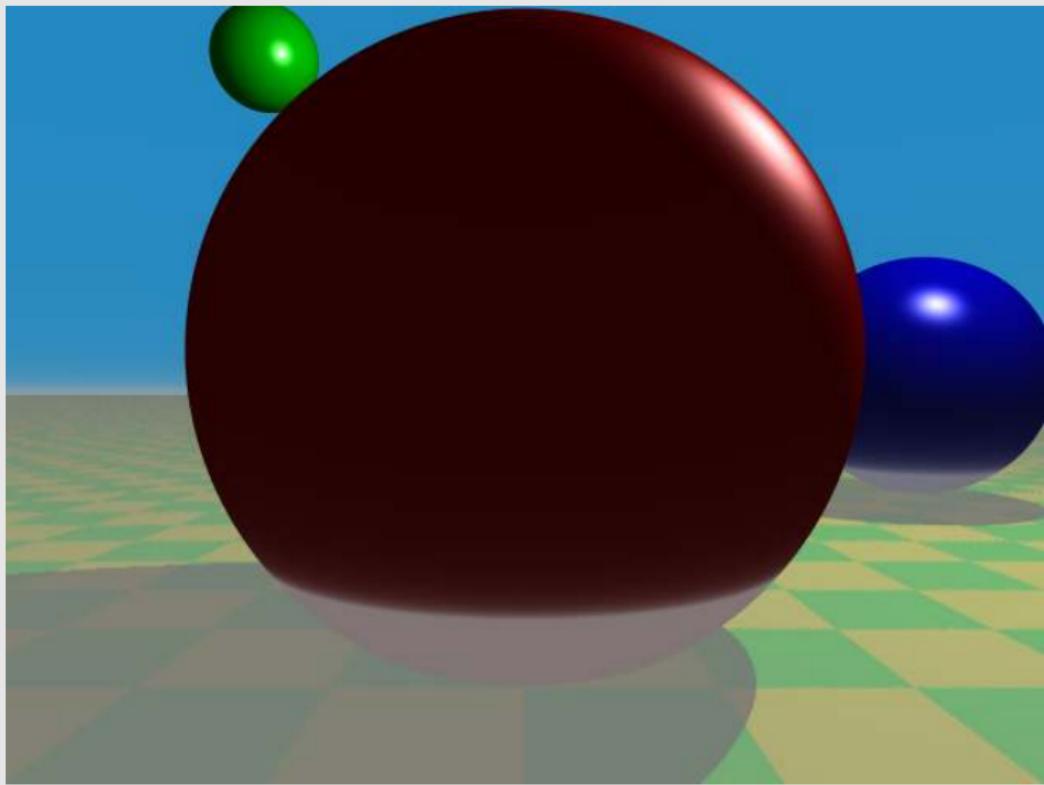
In der Höhe **`fog_offset+fog_alt`** ist die Dichte des Nebels auf 25% gesunken.

Bodennebel in POV-Ray

Beispiel

```
fog{ color rgbt <.7,.7,.7,.25>
    fog_type 2
    fog_alt 0.5
    fog_offset 0
    distance 1.5
    turbulence <.15, .15, .15> }
```

Bodennebel, das Bild zu Beispiel 44



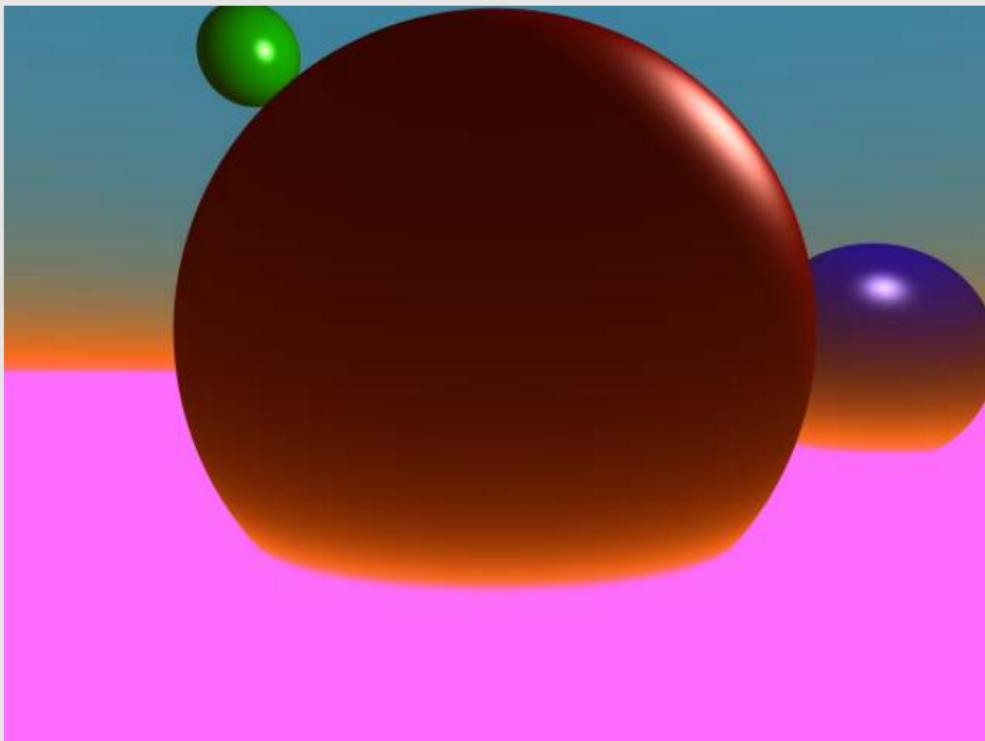
Geschichteter Nebel: layered fog

Beispiel

Geschichteter Nebel: layered fog In einer Szene können mehrere verschieden gefärbte Nebelschichten gleichzeitig verwendet werden.

```
fog{color Blue
    fog_type 2
    fog_alt 0.35
    fog_offset 0
    distance 1.5
    turbulence <.15, .15, .15>}
fog{color Orange
    fog_type 2
    fog_alt 1.45
    fog_offset 0
    distance 1.5
    turbulence <? ? ?>}
```

Geschichteter Nebel (Beispiel: 45)



Scheinwerfer und Nebel



Media-Spezifikation

- ▶ Nicht-feste Körper, wie Gas, Staub, Rauch oder Flüssigkeiten können Licht
 - ▶ absorbieren (**absorption**),
 - ▶ emittieren (**emission**), oder
 - ▶ streuen (**scattering**),..
- ▶ Die Effekte hängen von der Dichte (**density**) des Mediums ab.
- ▶ Diese Eigenschaften können im **interior**-statement in einer **media**-Spezifikation beschrieben werden.

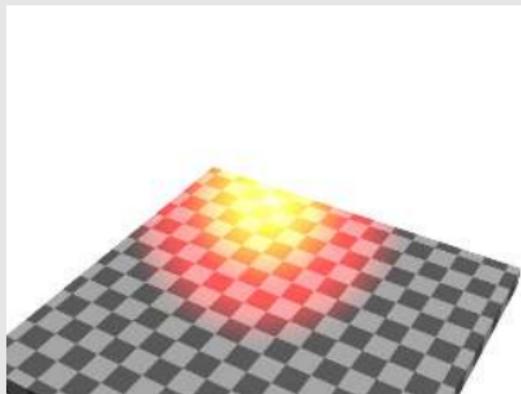
Ein Beispiel: Ein emittierendes Medium

Beispiel

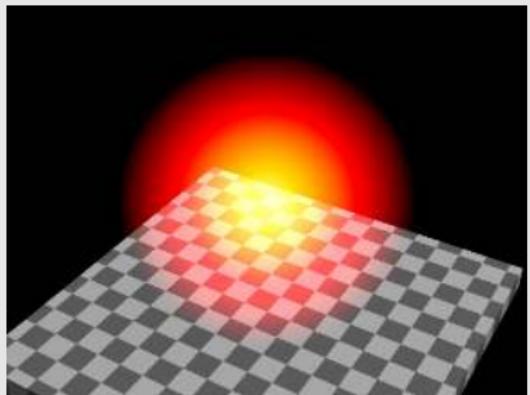
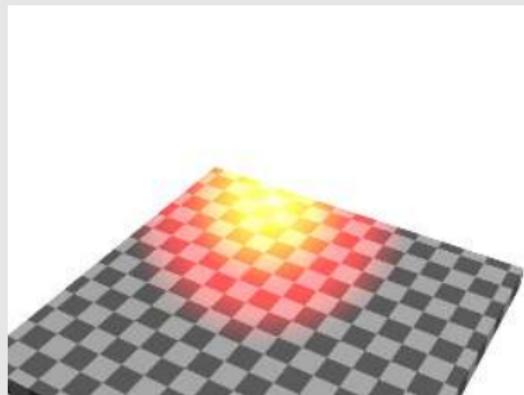
Eine leuchtende Kugel: Dieses Skript erzeugt eine Kugel mit transparenter weißer Hülle und sphärischer DichteVerteilung mit variierender Farbe. So können Flammen und leuchtende Gase modelliert werden.

```
sphere // transparent sphere containing media
{ 0,1 pigment { rgbt 1 } hollow
  interior
  { media
    { emission 1
      density
      { spherical density_map
        { [0 rgb 0]
          [0.4 rgb <1,0,0>]
          [0.8 rgb <1,1,0>]
          [1 rgb 1]}}}}}
```

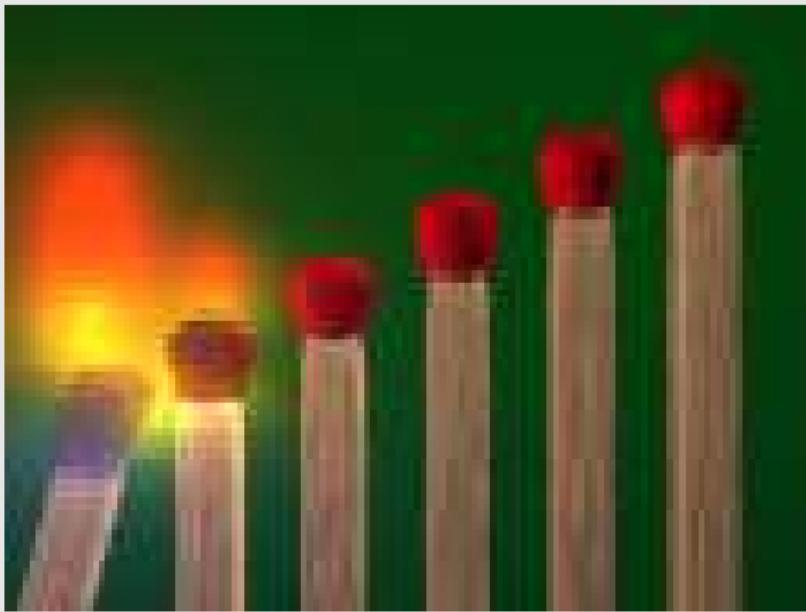
Die leuchtende Kugel vor weißem und schwarzem Hintergrund



Die leuchtende Kugel vor weißem und schwarzem Hintergrund



Ein komplexeres Beispiel für emittierende Pigmente



// By Dan Farmer 1996

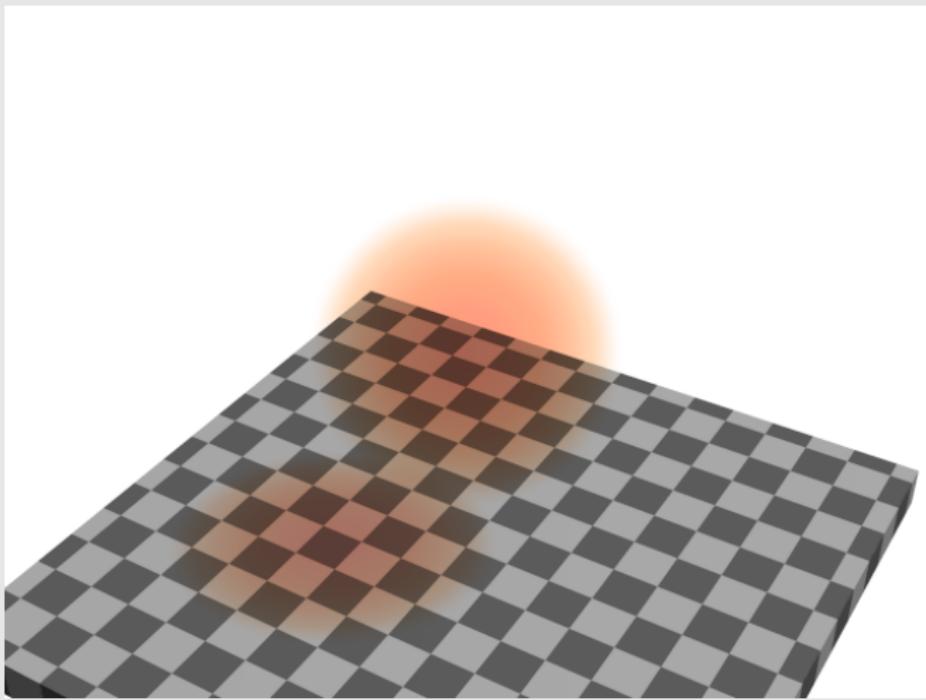
Ein absorbierendes Medium

Beispiel

Ein absorbierendes Medium Eine Kugel aus absorbierendem Material, mit sphärischer Dichteverteilung.

```
media
{ absorption 1
  density
  { spherical density_map
    { [0 rgb 0]
      [0.4 rgb 0]
      [0.5 rgb <0,0.5,1>]
      [1 rgb <0,1,1>]}}}
```

Ein absorbierendes Medium



Ein isotrop streuendes Medium

Beispiel

Ein streuendes Medium: Ein Medium, das isotrop in alle Richtungen streut, Farbe grau

box

```
{ -1,1 pigment { rgbt 1 } hollow
```

```
interior
```

```
{ media
```

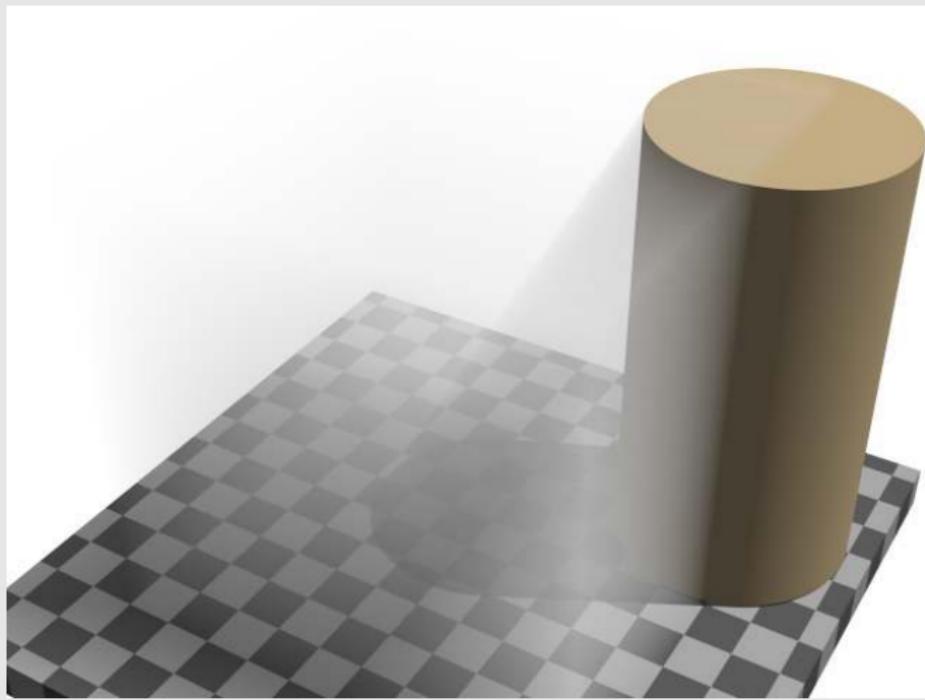
```
    { scattering { 1, 0.5 }}}} //isotrop , grau
```

cylinder

```
{ <0.9, -1, 0.7>, <0.9, 0.9, 0.7>, 0.5
```

```
pigment { rgb <1, 0.8, 0.5> }}
```

Ein isotrop streuendes Medium



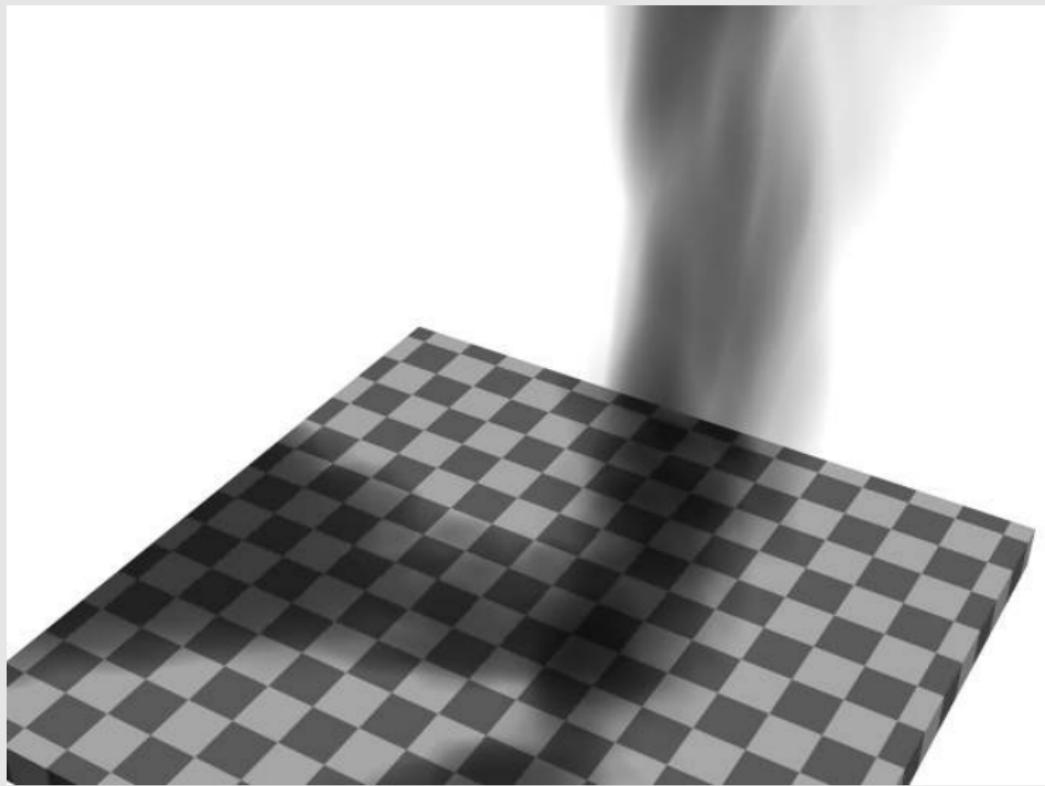
Typen der Streuung in POV-Ray

Typ	Streumodell	Anwendung
1	isotrop	richtungsunabhängig
2	Mie haze	Aerosolstreuung, gerichtet
3	Mie murky	Aerosolstreuung, stark gerichtet
4	Rayleigh scattering	Molekülstreuung, gerichtet, λ -abhängig Achtung! POV-Ray ignoriert die Wellenlängenabhängigkeit
5	Henyey- Greenstein scattering	Ein parametrisches Modell zur Approximation einer Vielzahl von Streutypen

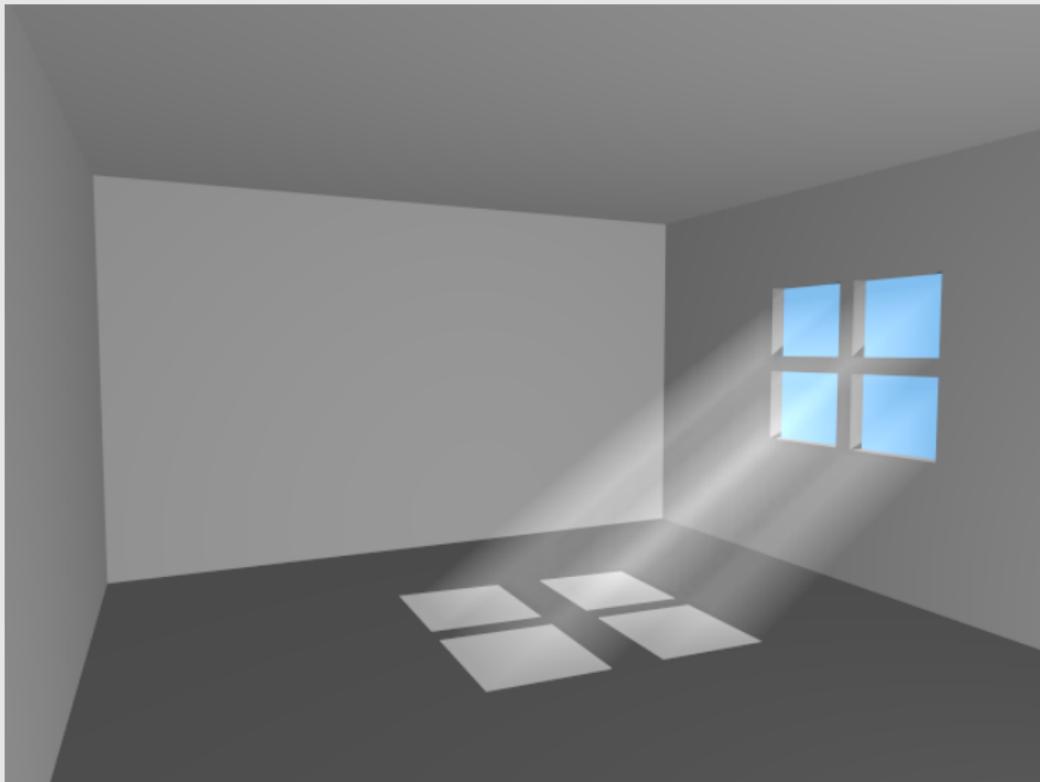
Eine Rauchsäule: Skalierung und Turbulenz

```
sphere
{ 0,1.5 pigment { rgbt 1 } hollow
  interior
  { media
    { absorption 7
      density
      { spherical density_map
        { [0 rgb 0]
          [0.5 rgb 0]
          [0.7 rgb .5]
          [1 rgb 1]}
        scale 1/2
        warp { turbulence 0.5 }
        scale 2}}}
    scale <1.5,6,1.5> translate y}
```

Beispiel: Eine Rauchsäule



Lichtstrahlen durch ein Fenster



Rechenzeit Intel: 40 Sekunden

Beispiel: Anwendung der Media, Kometenschweif

Beispiel: Anwendung der Media, Stern Mira

Mira

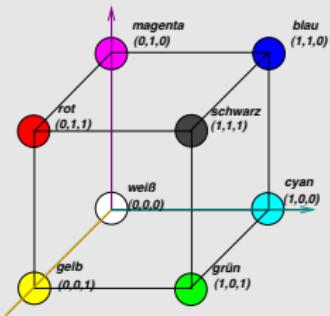
Lichtstrahlen durch ein Fenster

mit Photonen gerechnet



Rechenzeit Intel: ca. 20 Minuten

Farbsysteme



19

Farbmodelle

- Farbparameter
- Mischen von Farben
- Farbsysteme

20

Physiologie des Farbsehens

21

Farbgestaltung

Farbparameter

Die Farbe einer Lichtquelle kann durch drei Parameter beschrieben werden:

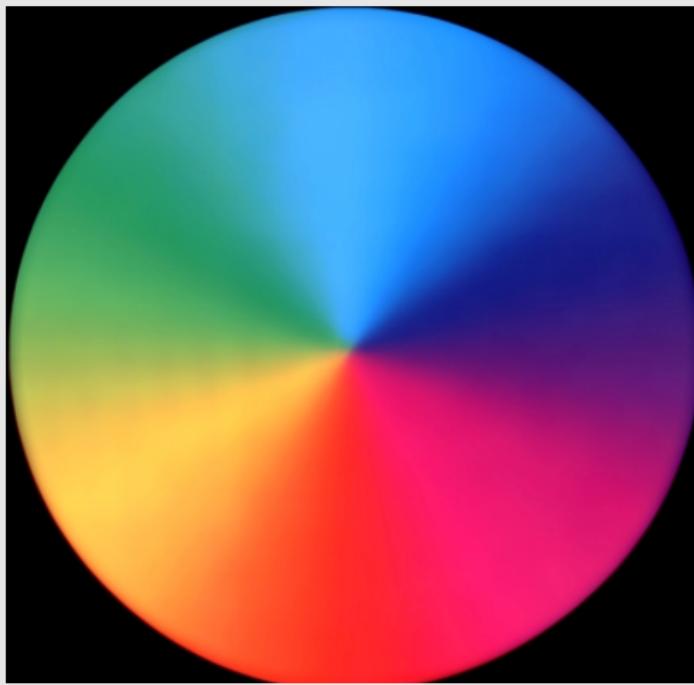
- ▶ Farbton
- ▶ Sättigung (Reinheit)
- ▶ Helligkeit

Definition (Farbton)

Der Farbton (hue) einer Farbe ist eine der unmittelbar empfundenen Eigenschaften. Er kann durch eine farbtongleiche Wellenlänge oder den Bezug auf einen Farbenkreis angegeben werden.

Der Farbton ist nur für **bunte** Farben definiert. **Unbunte Farben** (schwarz, grau, weiß) haben keinen Farbton.

Spektralfarben und Farbenkreis



Farbton und äquivalente Wellenlänge

Farbton	Wellenlänge [nm]
magenta	380–400
blau	450–480
grün	500–550
gelb	570–580
rot	> 630

Sättigung

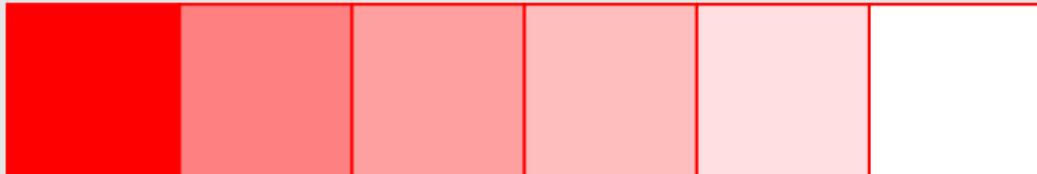
Definition (Sättigung)

Die Sättigung oder Reinheit (saturation, purity) einer Farbe ist die zweite der unmittelbar empfundenen Eigenschaften:

- ▶ Die Sättigung beschreibt, wie rein oder „ausgewaschen“ eine Farbe erscheint. Eine Farbe erscheint umso reiner, je weniger unterschiedliche Längenwellen zu ihr beitragen.

Spektralfarben: maximale Sättigung,

Pastellfarben: wenig gesättigt.



Eindeutige Spezifikation von Farben:

Farben lassen sich eindeutig angeben, indem sie als Mischung (Linearkombination) ausgewählter, definierter Grundfarben beschrieben werden:

Additive (optische) Farbmischung: Die additive

Farbmischung ist eine gleichzeitige oder rasch periodisch wechselnde Beleuchtung derselben Netzhautstelle durch verschiedene Farbreize. *Es wird das Licht verschiedener Lichtquellen vereinigt (addiert).*

Subtraktive (substantielle) Farbmischung: Eine Folge von spektralen Veränderungen am ursprünglichen Farbreiz; *Es werden absorbierende Pigmente kombiniert.*

Trichromatische Theorie

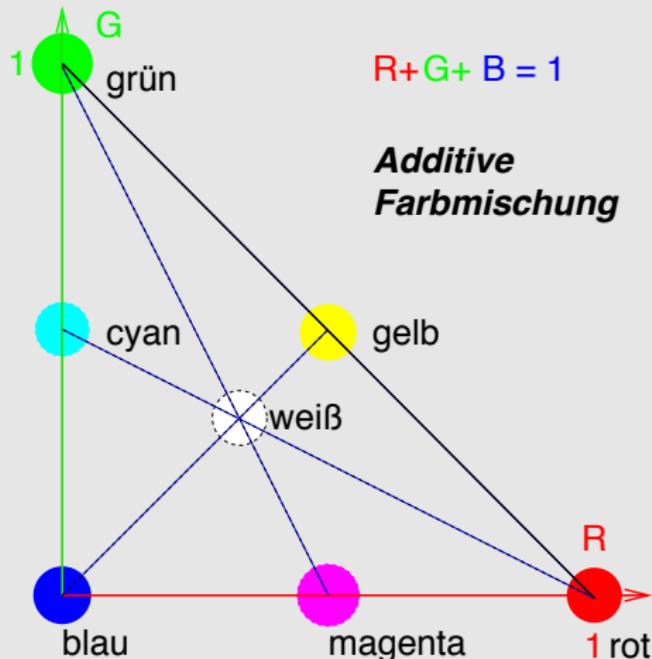
Trichromatische Theorie

Nach Young-1802, Helmholtz-1852 und Maxwell-1860 lässt sich jede beliebige (vom Menschen wahrnehmbare) Farbe durch additives Mischen von nur drei monochromatischen Lichtern erzeugen.

- ▶ Für Menschen mit normaler Farbtüchtigkeit lassen sich alle Farbtöne selbstleuchtender Farben durch die drei Primärfarben (Rot, Grün, Blau) hinreichend und eindeutig beschreiben.

Beispiel: Additive Farbmischung durch Licht

Additive Farbmischung

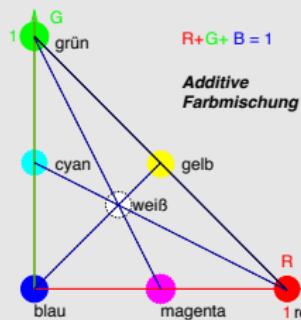


Komplementärfarben

Definition (Komplementärfarben)

Komplementärfarben sind Paare von Farben,

- ☞ deren additive Mischung weiß ergibt,
- ☞ bzw. schwarz bei subtraktiver Mischung,



z. B.:

- ▶ rot + cyan
- ▶ blau + gelb
- ▶ grün + magenta

Komplementärfarben im Farbdiagramm

Satz (Komplementärfarben)

Komplementärfarben (Gegenfarben) liegen im Farbdiagramm auf einer Geraden durch den Weißpunkt oder Schwarzpunkt.

Das CIE-System (Commission Internationale de l'éclairage)

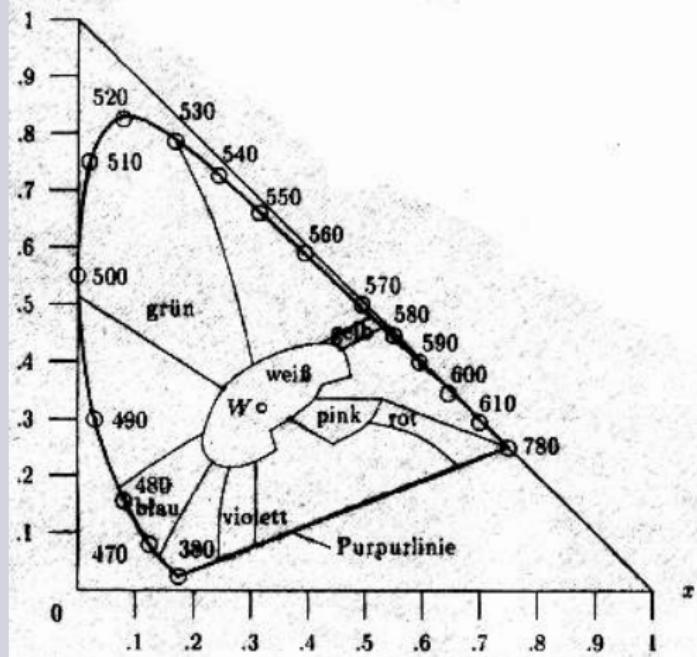
Das CIE-System ist seit 1931 ein internationaler Standard.

- ▶ Das System geht von drei künstlichen Grundfarben aus, die durch ihr Spektrum beschrieben werden.
- ▶ Seien X, Y, Z die Anteile einer Farbe an den drei Grundfarben. Dann sind die x, y, z -Komponenten im CIE-Farbraum:

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}, \quad z = \frac{Z}{X + Y + Z}$$

- ▶ Der Weißpunkt W approximiert das Tageslicht.

Das CIE-System

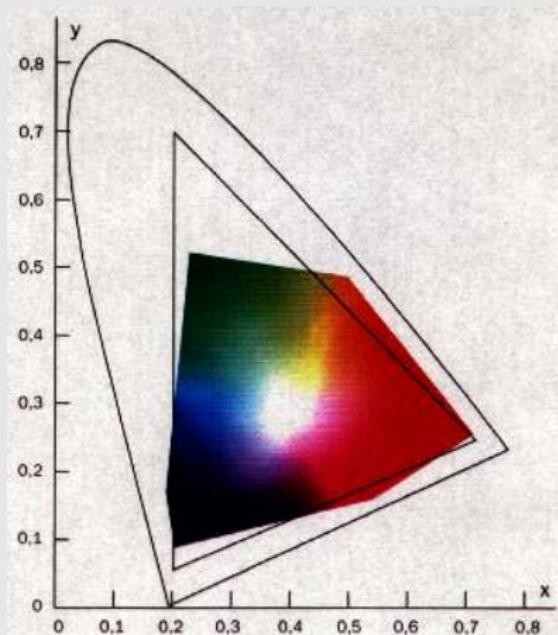


Das CIE-System

Das CIE-System beschreibt mehr Farben, als der Mensch sehen kann.

- ▶ Die äußere, gebogene Linie des Diagramms umschließt die sichtbaren Farben.
- ▶ Nicht alle für das bloße Auge sichtbaren Farben lassen sich technisch erzeugen, weil man für die Grundfarben an die verfügbaren Pigmente gebunden ist.
- ▶ Eine Teilmenge dieser Farben lässt sich durch Kombination von Pigmenten auf Papier drucken.
- ▶ Eine andere Teilmenge lässt sich mittels Fluoreszenz auf Bildschirmen (Kathodenstrahlröhren) erzeugen.

Darstellbare Farben: Drucker und Bildschirm



Uniformes CIE-System

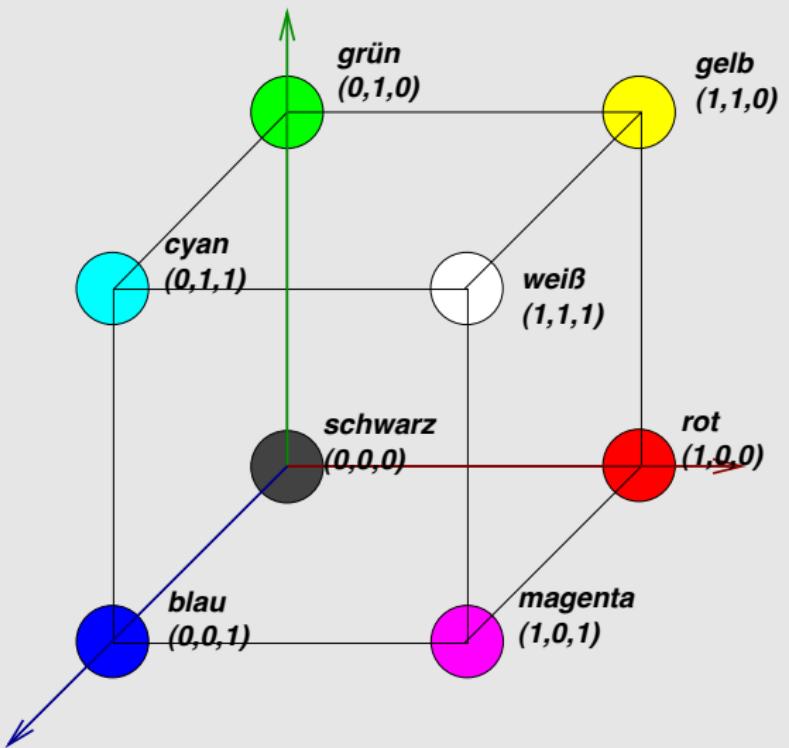
- ☞ Der Abstand zweier Farben im CIE-Raum sagt nichts über den *empfundenen Farbunterschied* aus.
- ☞ Der *uniforme CIE-Farbraum* von 1976 ist gegenüber dem Original-CIE-Raum so verzerrt, daß die Abstände der Farben proportional zu empfundenen Farbunterschieden sind. Die Kordinaten in diesem Raum sind u' , v' .

$$u' = \frac{2x}{6y - x + 1.5} \quad v' = \frac{4.5y}{6y - x + 1.5}$$

RGB-System

- ▶ Das **RGB**-System ist ein weitverbreitetes System zur **additiven** Mischung von Farben, insbesondere zur Farberzeugung auf Bildschirmen.
- ▶ Grundfarben: **Rot**, **Grün** und **Blau**.
- ▶ Auf einem Farbbildschirm werden farbige Bilder erzeugt, indem **rote**, **grüne** und **blaue** Phosphorpunkte entsprechend ihrem Anteil an einer darzustellenden Mischfarbe zum Leuchten angeregt werden (Fluoreszenz).
- ▶ Die von eng benachbarten fluoreszierenden Phosphor-Punkten abgestrahlten Farben addieren sich auf der Netzhaut zu Mischfarben.

Der **RGB**-Einheitswürfel



Der **RGB**-Einheitswürfel

- ▶ Im RGB-Würfel können — anders als im CIE-Diagramm — auch Intensitäten dargestellt werden.
- ▶ Eine Farbe erscheint umso heller, je weiter sie vom Schwarzen Punkt entfernt ist.
- ▶ Auf der Diagonalen liegen die unbunten Farben — Schwarz, Weiß, Grau.
- ▶ Der Abstand von dieser Diagonalen ist ein Maß für die Reinheit der Farbe.

- ▶ Das YIQ-System ist das Farbmodell des amerikanischen NTSC-Systems (1953), NTSC = National Television Standard Committee.
- ▶ Das System verwendet die Parameter Y,I,Q, wobei Y die Intensität der Farbe ist.

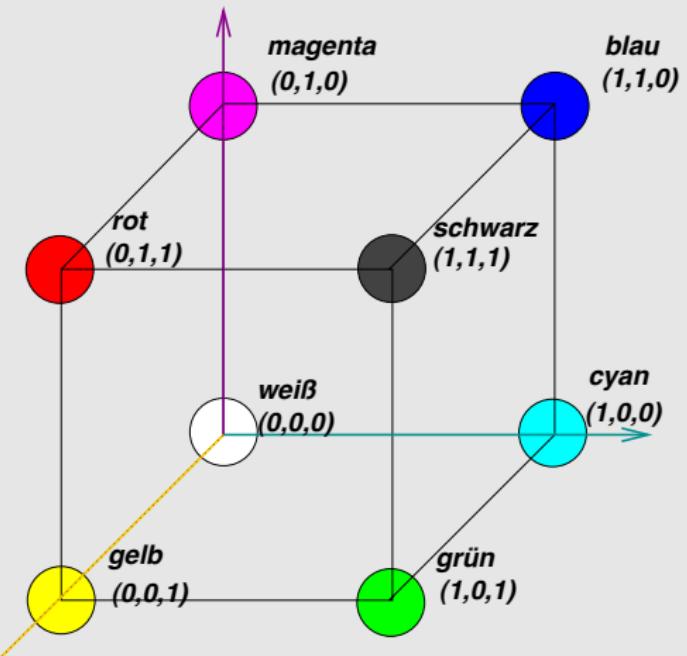
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ .596 & -.274 & -.322 \\ .211 & -.522 & .311 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Das PAL-System verwendet die Komponenten: Y, R-Y, B-Y.

Das CMY-Modell

- ▶ Das CMY-Modell ist ein *subtraktives* Farbmodell für den Farbdruck. Es beschreibt, wie durch Mischen von absorbierenden Pigmenten auf dem Papier Mischfarben entstehen.
- ▶ Die Grundfarben sind Cyan, Magenta, Gelb (und Schwarz).
- ▶ Der Vektor $(0,0,0)$ bedeutet „Weiß“ — keine absorbierenden Pigmente, und $(1,1,1)$ ist die Farbe „Schwarz“— volle Absorption.

Der CMY-Einheitswürfel



Umwandlung zwischen **RGB**-und **CMY**-Werten

Es werden die Farbvektoren der Komplementärfarben berechnet:

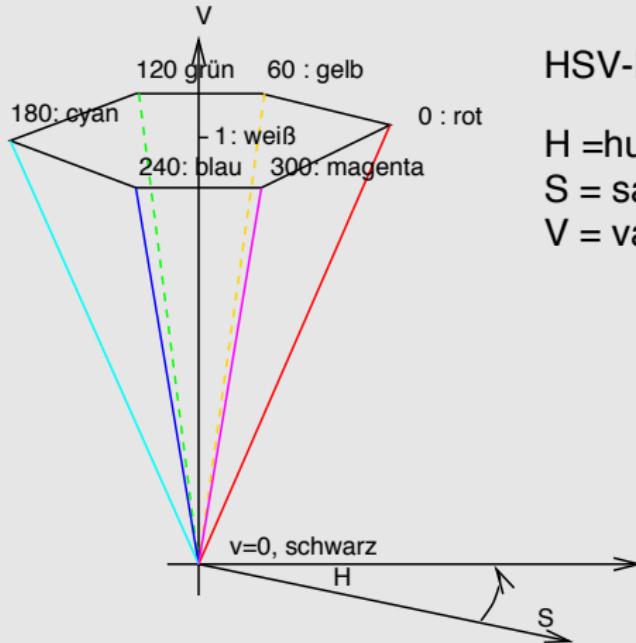
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

HSV-Modell

Das HSV-Modell beschreibt die Farben nicht durch die Mischung von Grundfarben sondern durch die Farbparameter *Farbton (hue)*, *Sättigung (saturation)* und *Helligkeit (value)*.

- ▶ Die Farbe wird durch einen Farbwinkel im HSV-Farbkreis angegeben, die Sättigung durch den Abstand vom Weißpunkt.
- ▶ Der HSV-Farbkreis entsteht durch die Projektion des **RGB**-Würfels entlang der Schwarz-Weiß-Achse in eine Ebene senkrecht zu dieser Achse.

Die HSV-Pyramide



HSV-Pyramide

H = hue

S = saturation

V = value

Die HSV-Parameter:

Farbton: Der Farbton wird durch den Winkel H angegeben, 0 = rot.

Komplementärfarben liegen 180° auseinander.

Sättigung S: 0=ungesättigt (grau), 1=reine Farbe.

Intensität V: 0=schwarz, 1=weiß.

Vor- und Nachteile des HSV-Systems

- ▶ Das System ist gut für die interaktive Komposition von Farben geeignet, aber ungünstig für Mustererkennungsverfahren (ungleiche Wertebereiche für H,S,V, Unstetigkeit bei 360°, unsichere Werte für H und S bei kleinen Intensitäten).
- ▶ Für die numerische Auswertung von Farbwerten sind das **RGB**-System oder das **CMY**-System daher deutlich günstiger.

Das CNS-Modell (color naming system)

1982 Benk, Brownston, Kaufmann: Natürlichsprachliche Farbbezeichnungen, z.B. *very light strong yellowish green*.

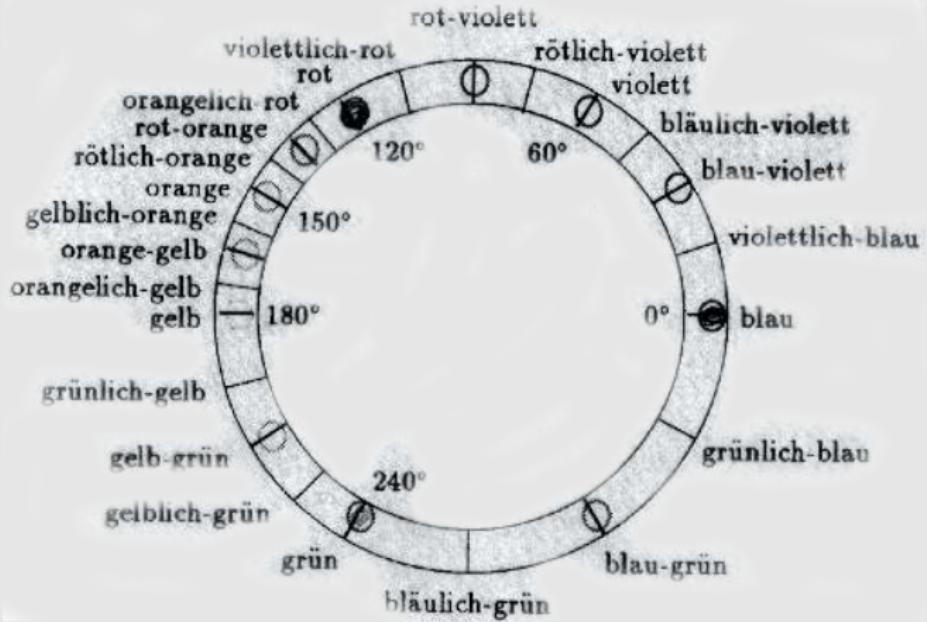
Parameter: Farbe, Sättigung, Intensität.

Helligkeitsstufen: very dark, dark, medium, light, very light.

Sättigung: grayish, moderate, strong, vivid.

chromatische Farbskala: Analog zur Kompaßrose,
7 Primärfarben, Halb- und Vierteltöne.

Der CNS-Farbkreis



Farbskalen

Achromatische Farbskala: black, very dark gray, dark gray, (medium) gray, light gray, very light gray, white.

Deutsche Farbnamen: Im deutschen werden Halb- und Vierteltöne mit Bindestrich und Suffix „lich“ angegeben: grünlich-blau, orangefisch-gelb.

Farben in POV-Ray

Das POV-Ray-System verwendet **RGB**-Farben, die als Vektoren in zwei Weisen angegeben werden können. Zusätzlich können Farbnamen definiert werden.

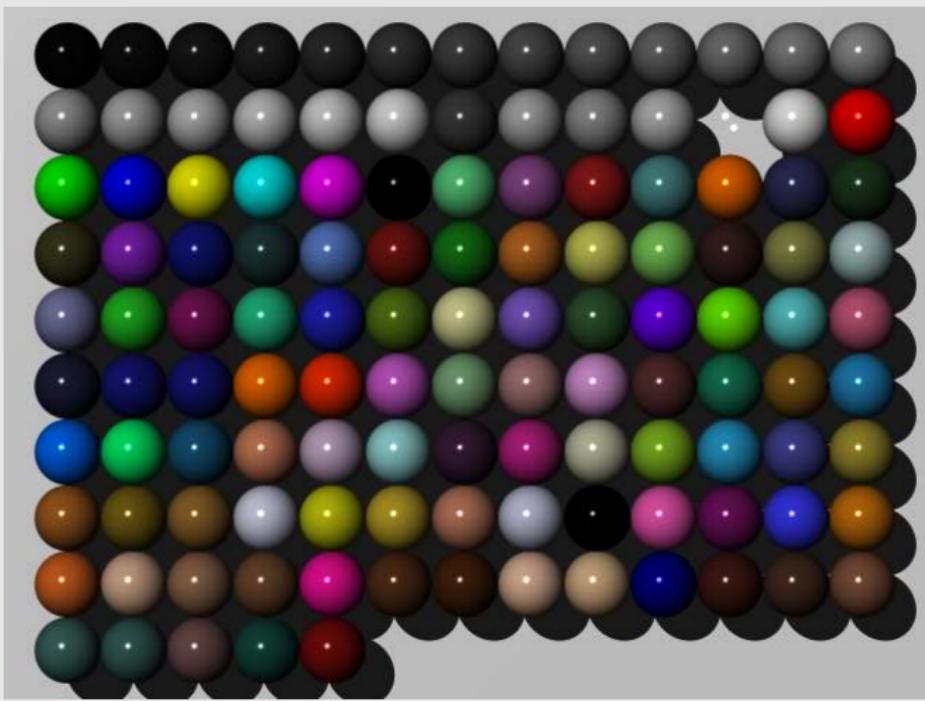
Farbvektoren: `color rgb <1, 0.5, 0.5>`, ein ungesättigtes Rot.

Vordefinierte Farben: Im file `colors.inc` sind rund 150 Farben vordefiniert.

Ein Auszug aus colors.inc

```
#declare Aquamarine = color red 0.439216
          green 0.858824 blue 0.576471
#declare BlueViolet = color red 0.62352
          green 0.372549 blue 0.623529
#declare Brown = color red 0.647059
          green 0.164706 blue 0.164706
#declare CadetBlue = color red 0.372549
          green 0.623529 blue 0.623529
#declare CornflowerBlue = color red 0.258824
          green 0.258824 blue 0.435294
#declare HuntersGreen = color red 0.13
          green 0.37 blue 0.31
```

Die Farben aus colors.inc



Teil VII

Farbmodelle

Physiologie des Sehens



19

Farbmodelle

20

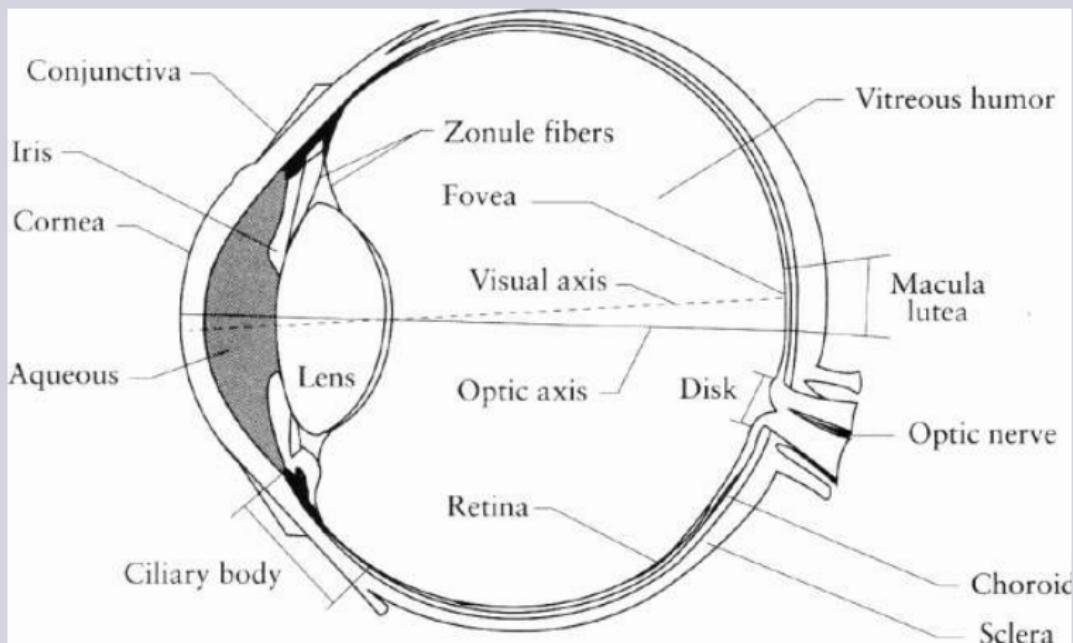
Physiologie des Farbsehens

- Anatomie des Auges
 - Die Linse
 - Die Netzhaut
 - Farbwahrnehmung
- Chromatec-Verfahren

21

Farbgestaltung

Ein Querschnitt durch das Auge



Literatur zur Physiologie des Sehens

-  Glassner, A. S. (1995).
Principles of Digital Image Synthesis, volume 2 of *The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling*.
Morgan Kaufmann Publ. Inc., Los Altos/CA.
-  Fellner, W. (2004).
Computergrafik, volume 58 of *Reihe Informatik*.
BI Wissenschaftsverlag, Mannheim – Wien – Zürich,
3. edition.
-  Erb, C. and Fahle, M. (2006).
Farbensehen und erworbene Farbsinnstörungen.
Der Ophthalmologe, 4:349–361.

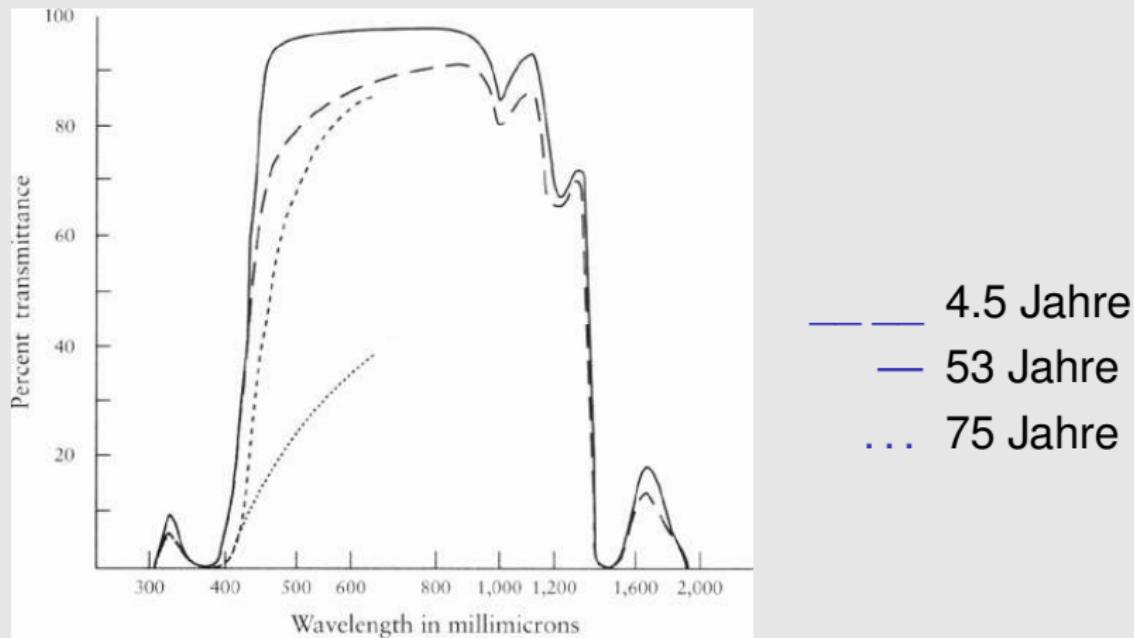
Die Linse

- ▶ Die Linse kann über Muskeln verformt werden, um die Brennweite der Sehentfernung anzupassen.
- ▶ Die Linse ist nicht gegen die chromatische Aberration korrigiert.
- ▶ Das Auge kann daher nur auf eine Wellenlänge zur Zeit fokussieren.
- ▶ Befinden sich ein **blaues Objekt** und ein **rotes Objekt** in der gleichen Entfernung vor unserem Auge, kann nur eins von beiden zur Zeit scharf abgebildet werden.
- ▶ Das **blaue Objekt** scheint für uns weiter entfernt zu sein als das **rote**.

Durchlässigkeit der Linse:

- ▶ Die Linse absorbiert im Bereich **cyan-blau** doppelt so stark wie im Bereich **gelb-orange**. Daraus ergibt sich eine **Blauschwäche** des Auges.
- ▶ Altersbedingt verstärkt sich diese Blauschwäche noch durch eine zunehmende Trübung der Linse. Die Linse verfärbt sich gelblich und wirkt dann wie ein Gelbfilter.

Altersbedingte Transmission der Linse



Direkte Transmission

Die Brennweite der Linse

- ▶ Bei Kurzsichtigkeit oder Weitsichtigkeit ist die Netzhaut in der falschen Entfernung zur Linse. Solche Sehfehler können mit Brillen oder Kontaktlinsen korrigiert werden.



Trichromatischen Theorie

Die peripheren Mechanismen des Farbensehens folgen der trichromatischen Theorie von Young-1802, Helmholtz-1852 und Maxwell-1860, die zeigt, dass sich jede beliebige Farbe durch additive Mischung von drei monochromatischen Lichtern erzeugen lässt.

- ▶ Alle Farbtöne selbstleuchtender Farben sind durch die drei **Primärfarben** (Rot, Grün, Blau) hinreichend und eindeutig beschreibbar.
- ▶ Die Mehrheit der Bevölkerung verwendet zur Mischung eines vorgegebenen Farbtons annähernd identische Anteile der Primärfarben.

Diese Tatsachen haben zu der Auffassung geführt, daß die Netzhaut des Auges über drei unterschiedliche farbempfindliche Sensorentypen verfügt.

Photometrische Auflösung des Auges

Das Auge kann ca.

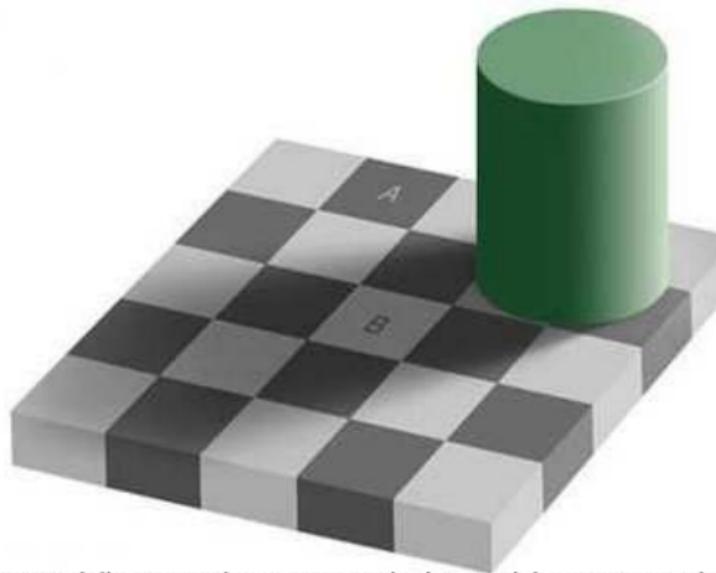
- ▶ 500 Helligkeitsstufen unterscheiden,
- ▶ 200 Farbtöne unterscheiden,
- ▶ 26 Sättigungsstufen unterscheiden.

Beim achromatischen Sehen (nachts) können nur die ca. 500 Helligkeiten unterschieden werden, aber beim Farbsehen bis zu $500 \times 200 \times 26 = 2600000$ Farbstufen.

Farbkontrast

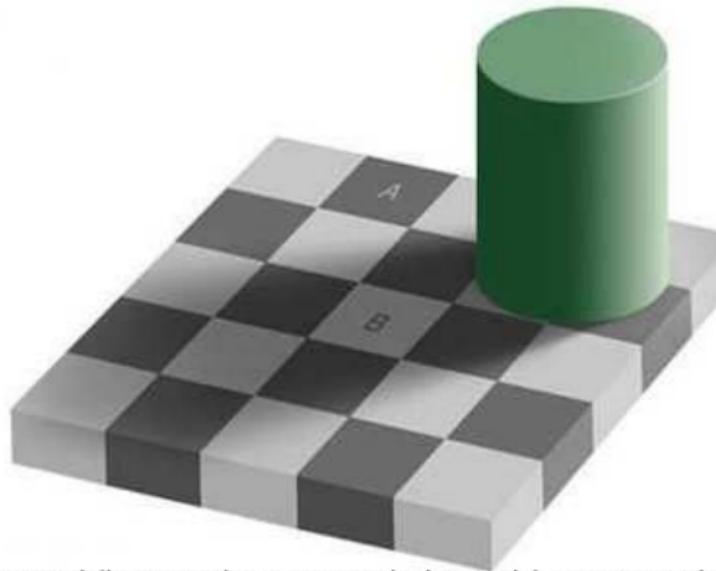
- ▶ Wir können räumlich benachbarte Farben unterscheiden, wenn sie einen ausreichenden Kontrast haben, aber wir können Farben schlecht absolut wiedererkennen.
- ▶ In den Ganglienzellen wird vor allem der Farbkontrast ausgewertet, nicht die absoluten Farbwerte.

Schachbrettillusion



Optical Illusion 5: The square marked A, and the square marked B are the same shade of grey. (It's true, I checked.)

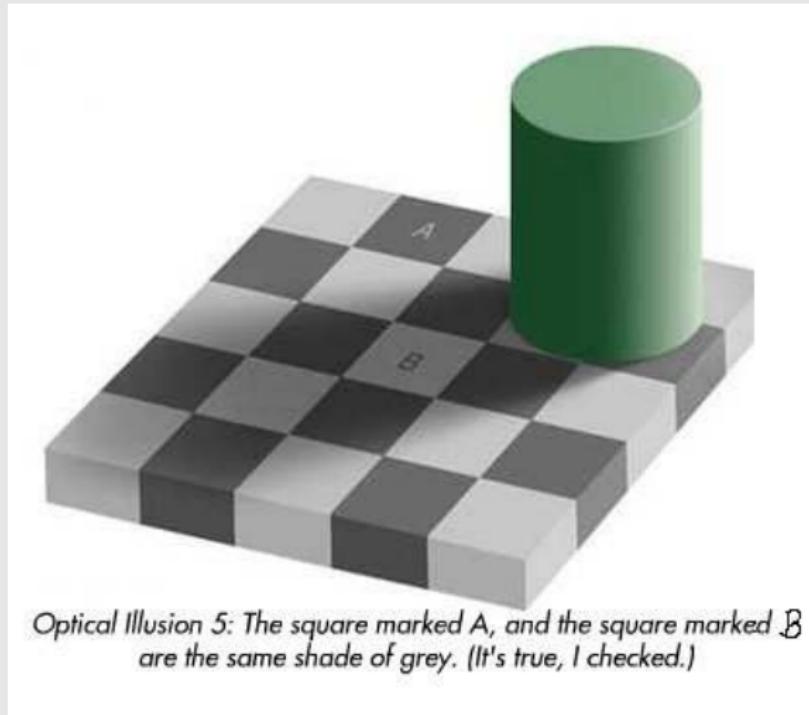
Schachbrettillusion



Optical Illusion 5: The square marked A, and the square marked B are the same shade of grey. (It's true, I checked.)



Schachbrettillusion



Aufbau der Netzhaut:

6 Mio. Zapfen: Photorezeptoren für Licht hoher Intensität. Sie dienen der Wahrnehmung von Farben bei Tageslicht oder anderem hellen Licht. Es gibt drei unterschiedliche Arten von Zapfen, die sich in den Photopigmenten unterscheiden:

S-Zapfen: (420 nm)

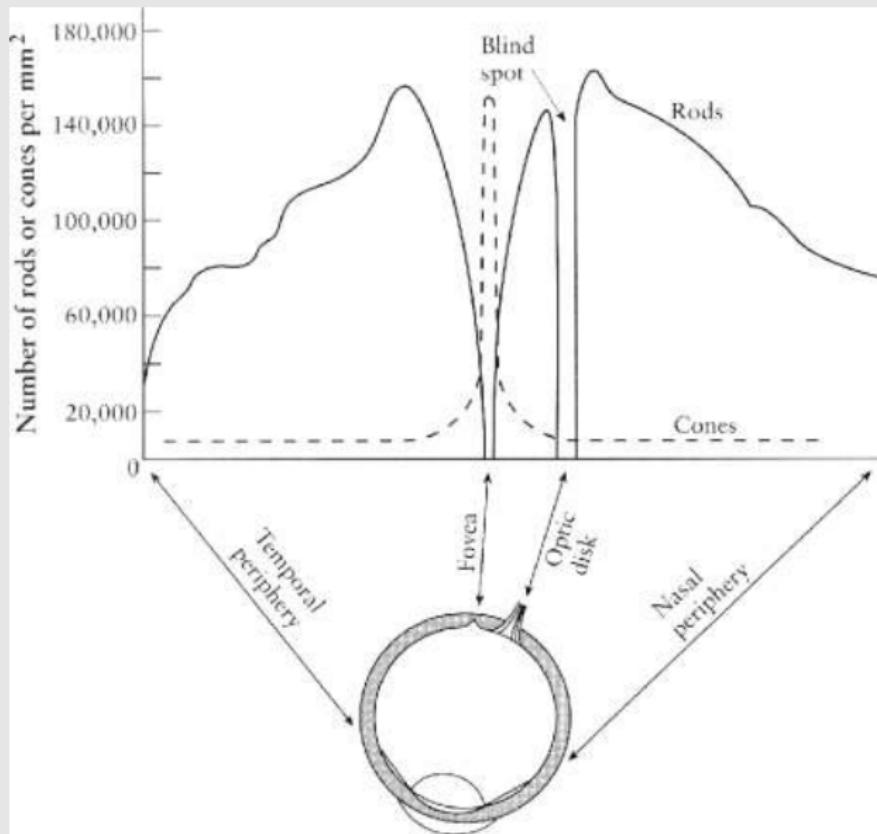
M-Zapfen: (530 nm)

L-Zapfen: (560 nm)

120 Mio. Stäbe: Photorezeptoren für Licht geringer Intensität. Die Stäbe eignen sich für das Sehen bei Nacht, vermitteln aber keine Farbinformation.

Das Photopigment ist Rhodopsin, mit einer maximalen Empfindlichkeit bei 500 nm.

Verteilung der Rezeptoren auf der Netzhaut



Geometrische Auflösung der Netzhaut



Verteilung der Zapfen auf der Netzhaut:

(S) blau: ca. 2% der Zapfen sind **blau-empfindlich**. Sie liegen überwiegend im äußeren Ring der Netzhaut.

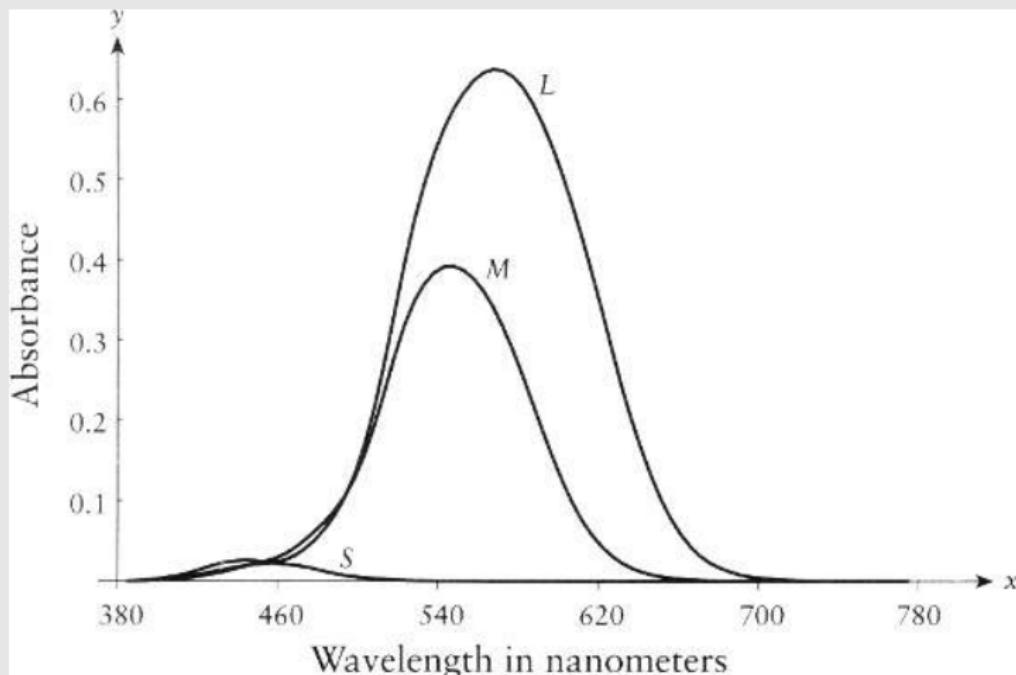
Kleine blaue Objekte können unsichtbar werden, wenn das Auge sich darauf konzentriert.

(M) grün: ca. 32% der Zapfen, mittlerer Ring.

(L) gelb: ca. 64%, in der Mitte der Retina.

In der Fovea finden sich 147000 Zapfen pro Quadratmillimeter.

Absorptionsfähigkeit der Zapfen



Wie entsteht der Farbeindruck?

- ▶ Die Zapfen favorisieren zwar bestimmte Frequenzen, reagieren aber auf einen sehr breiten Bereich des sichtbaren Lichts.
 - ▶ Ein einzelnes Photon kann daher zwar in einem Zapfen eine Reiz auslösen, aber das reicht nicht aus, um die Frequenz des Photons zu bestimmen.
 - ▶ Erst die Auswertung einer Folge solcher Reize in den nachgeschalteten Ganglienzellen ergibt einen Farbeindruck.
- ☞ *Der Farbeindruck entsteht nicht am Photorezeptor sondern erst durch die Auswertung im visuellen System.*
- Das ist ähnlich der Farberzeugung durch „dithering“.

Farbempfindung

- ▶ Die Farbempfindung entspricht in erster Näherung der Verteilung der Wellenlängen des eintreffenden Lichtes.
- ▶ Sie wird über die Rezeptoren der Retina und die nachgeschalteten Ganglienzellen vermittelt.
- ▶ Dabei erscheinen zwei Farbreize mit unterschiedlichen Spektren farblich identisch, wenn sie die Rezeptoren im gleichen Verhältnis erregen und daher dieselbe neuronale Reizantwort auslösen.
- ▶ Andererseits kann das menschliche Auge Farbnuancen unterscheiden, die sich physikalisch sehr ähnlich sind.

Metamerie

- ▶ Mit nur drei wellenlängenspezifischen Lichtrezeptoren können die Wellenlängenmuster durch das menschliche Auge nur grob erfasst werden.
- ▶ Daher gibt es Wellenlängenmuster, die sich physikalisch deutlich unterscheiden und trotzdem denselben Farbeindruck hinterlassen.
Dieses Phänomen wird als **Metamerie** bezeichnet.
- ▶ Dabei erscheinen zwei Farbreize mit unterschiedlichen Spektren farblich identisch, wenn sie die Rezeptoren im gleichen Verhältnis erregen und daher dieselbe neuronale Reizantwort auslösen.

Ein guter Kompromiß

Warum wird das Spektrum nicht genauer durch spezifischere Photorezeptoren abgetastet?

- ☞ Das menschliche Auge ist ein guter Kompromiß bzgl. Farbauflösung und räumlicher Auflösung:
 - ▶ Eine Abtastung an mehr Wellenlängen würde mehr Rezeptoren erfordern.
 - ▶ Die Augen müßten größer sein oder wir würden räumliche Auflösung verlieren.

Gegenfarbentheorie

Die Gegenfarbentheorie basiert auf phänomenologischen Beobachtungen:

Satz

Die Farben Rot und Grün so wie Blau und Gelb sind perzeptuell je zu einem Gegensatzpaar verbunden.

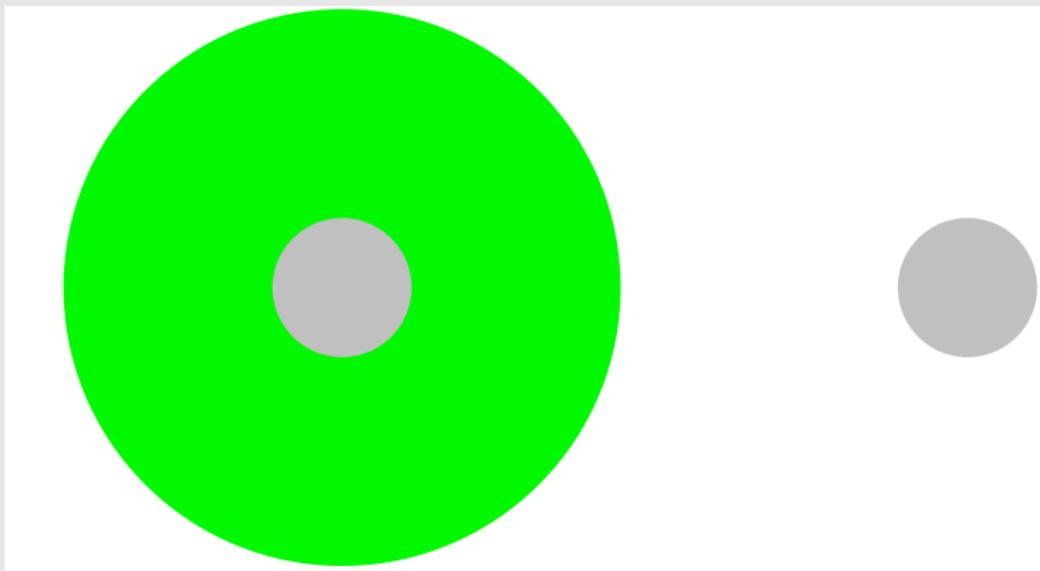
Beweis.

Durch Beobachtungen

Farbiger Sukzessivkontrast: Farbige Nachbilder.

Farbiger Simultankontrast: Scheinfarben in der Nachbarschaft.

Farbiger Simultankontrast



Beide grauen Kreise haben dieselbe Farbe.

Farbenfehlsichtigkeit

Ist eine Gruppe der Photorezeptoren krankhaft verändert, so können weniger Farben unterschieden werden. Abhängig davon, welche Gruppe betroffen ist, werden mehrere Formen der Farbenfehlsichtigkeit unterschieden.

Beispiel: Drei Farbkomponenten eines Farbbildes



Zwei Formen der Fehlsichtigkeit



Deutanopie

Protanopie

Zwei Formen der Fehlsichtigkeit

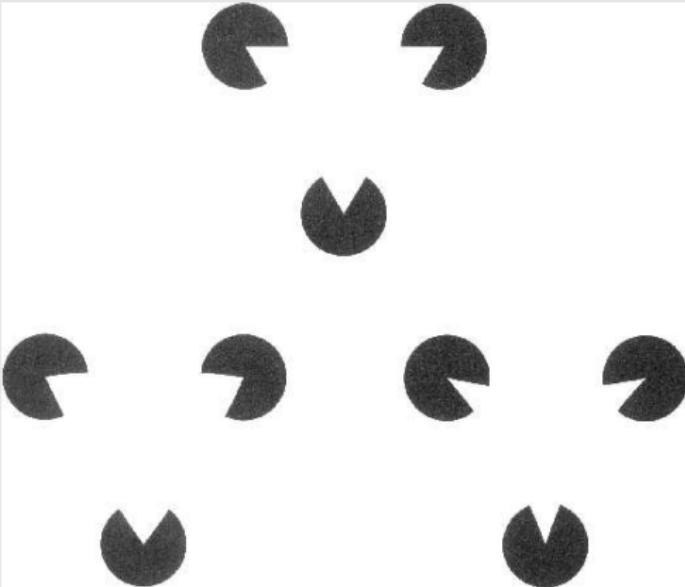


Deutanopie

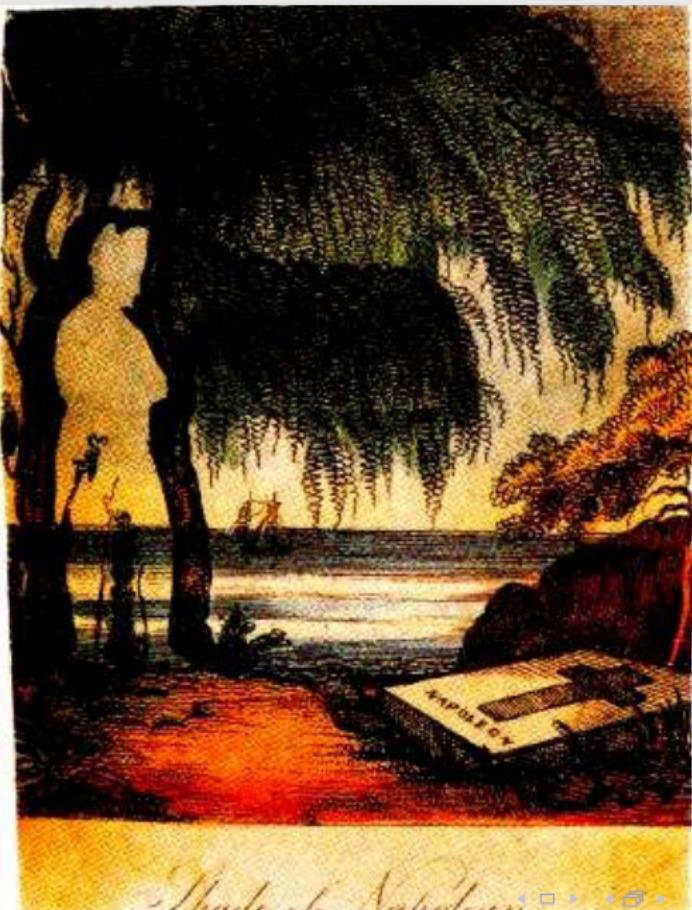


Protanopie

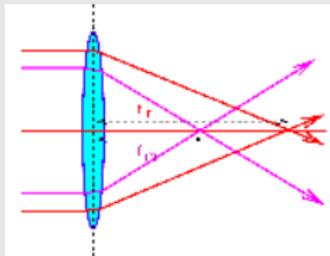
Gestaltwahrnehmung



Napoleon an seinem Grab



Chromatec-Verfahren



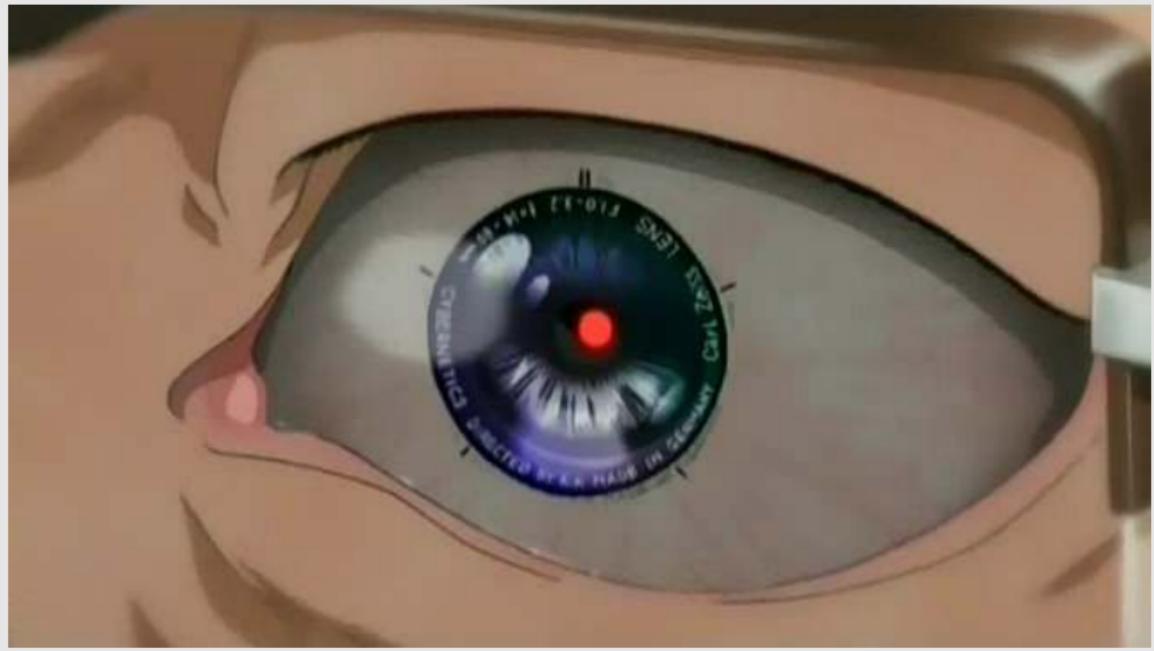
Chromatische
Aberration

- ▶ Die Linse des Auges ist nicht gegen chromatische Aberration korrigiert.
- ▶ Rote Objekte erscheinen bei gleicher Entfernung näher als blaue Objekte.

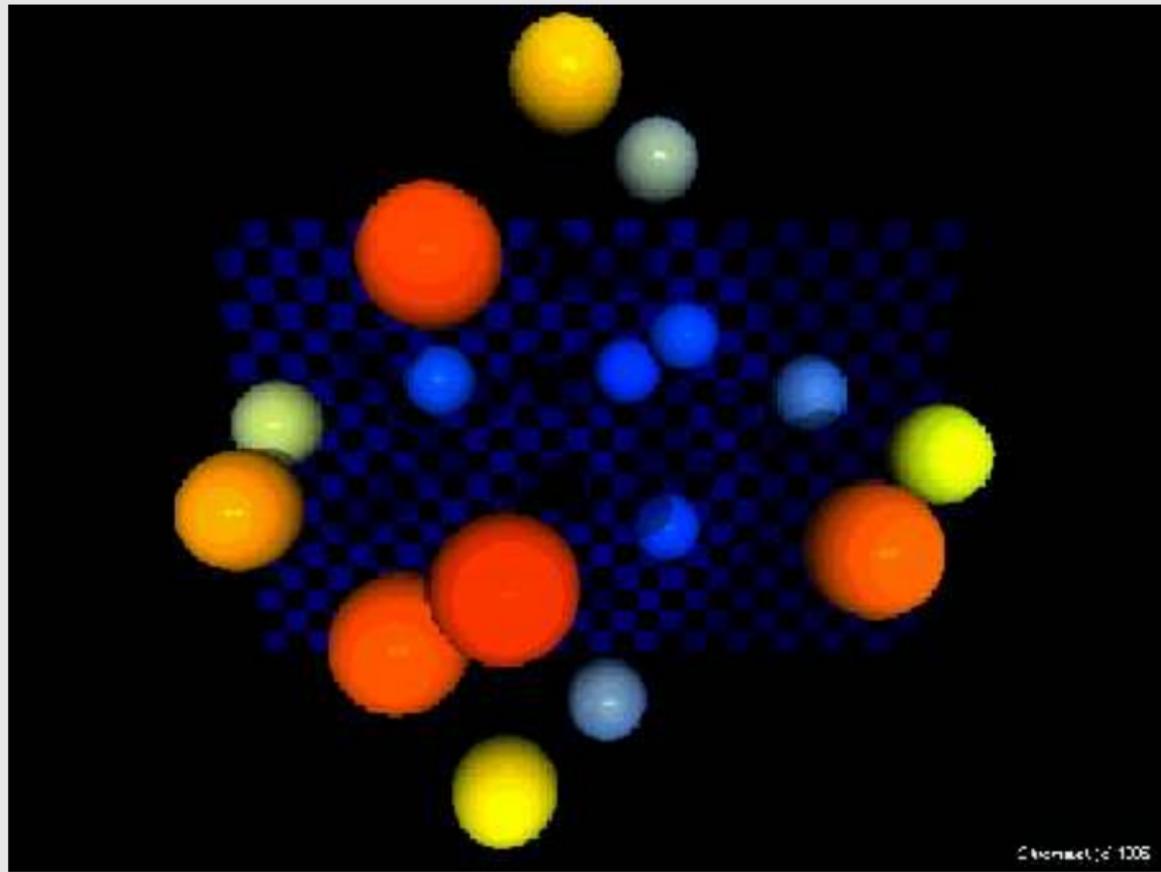
☞ **Idee:** Erzeuge einen Tiefeneindruck, indem weit entfernte Objekte **blau** dargestellt werden und nahe Objekte in **rot**.

Augen von Zeiss?

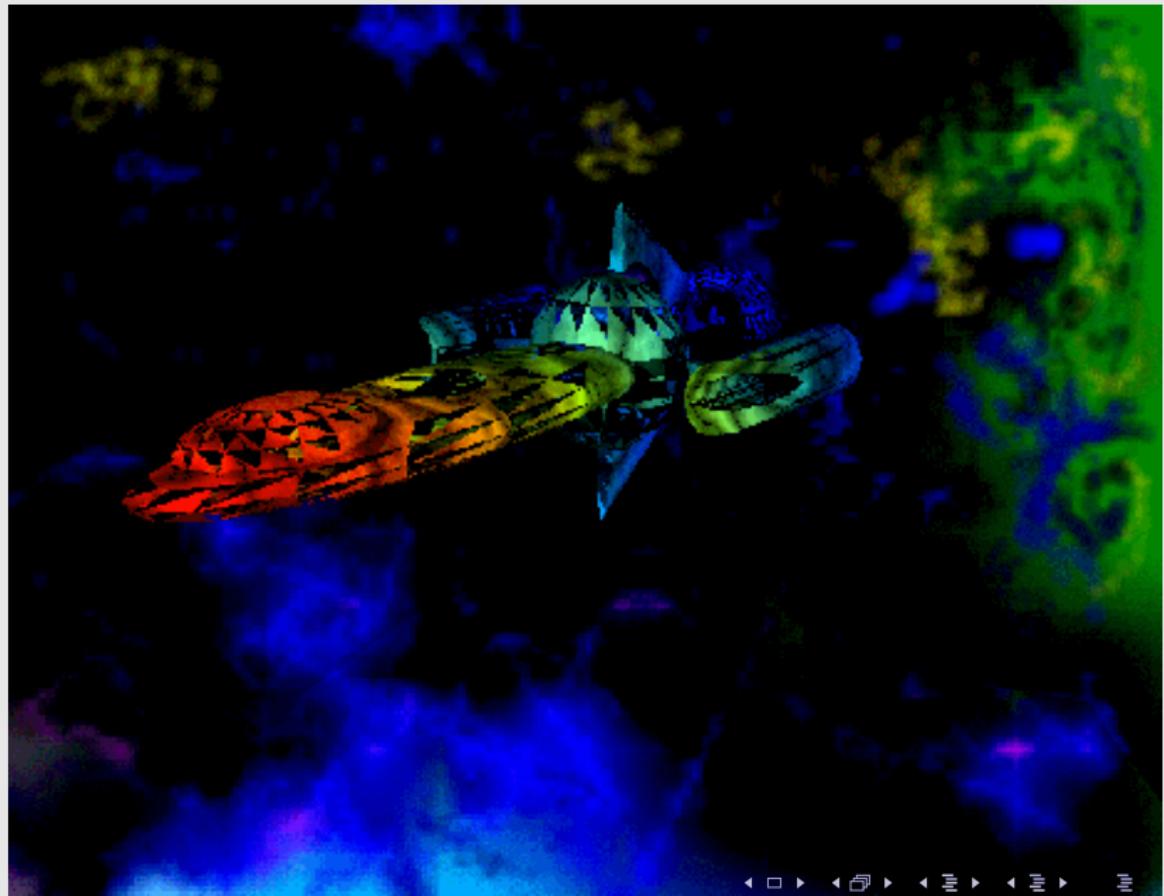
Aus: Ghost in the Shell: Stand Alone Complex



Beispiele:



Beispiele:



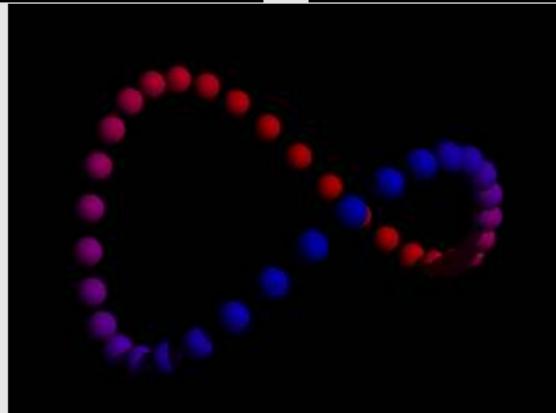
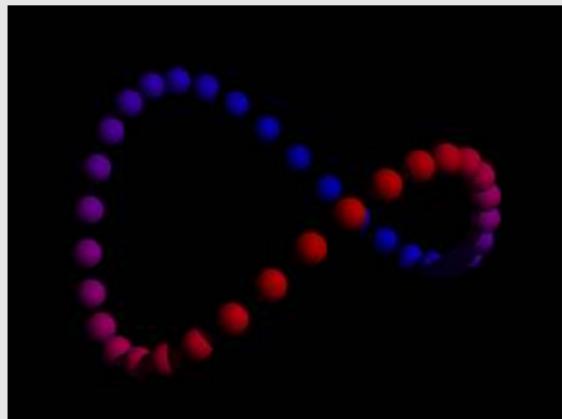
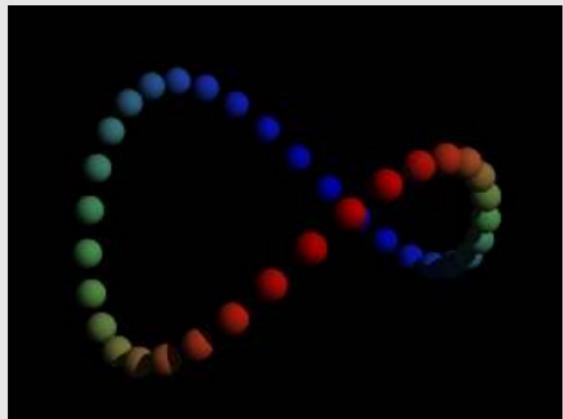
Beispiele:



Beispiele:



Welche Farbverteilung wirkt plastischer?



Farbgestaltung:

- ▶ Reines Blau sollte für Text, dünne Linien und kleine Objekte vermieden werden.
- ▶ Aneinander grenzende Farben sollten sich nicht nur im **Blauanteil** unterscheiden, also sollte man beispielsweise nicht **Cyan** und **Grün** nebeneinandersetzen.
- ▶ Alte Menschen benötigen ein höheres Helligkeitsniveau, um Farben unterscheiden zu können.
- ▶ **Rot** und **Grün** sollten in den Randbereichen von großen Grafiken vermieden werden.
- ▶ **Zuviele Farben (UND FONTS)**, die Unterschiedliches bedeuten, überfordern die Aufnahmefähigkeit.

Farbgestaltung:

- ▶ Zusammengehörige Objekte sollten auf einem gleichen farbigen Hintergrund dargestellt werden.
- ▶ Ähnliche Farben sollten eine ähnliche Bedeutung signalisieren.
- ▶ Farben sind ähnlich, wenn sie im uniformen CIE-Diagramm benachbart sind.
- ▶ Helligkeit und Sättigung eignen sich sehr gut, um die Aufmerksamkeit zu erregen.
- ▶ **Kalte Farben** (kurze Wellenlänge) eignen sich für Statusinformation.
- ▶ **Warme Farben** (lange Wellenlänge) eignen sich für dringende Nachrichten, Eingabeaufforderungen usw..

Teil VIII

Texturen und Fraktale



22

Modellierung von Texturen in POV-Ray

- Pigmente und Muster
- Modifikatoren
- Normalentexturen

23

Materialien und Transparenz

Texturen natürlicher Oberflächen

Natürliche Oberflächen sind in der Regel texturiert:

- ▶ Sie haben eine regelmäßige, zufällig unregelmäßige Struktur.

Texturen natürlicher Oberflächen

Natürliche Oberflächen sind in der Regel texturiert:

- ▶ Sie haben eine regelmäßige, zufällig unregelmäßige Struktur.
- ▶ Die Textur geht entweder auf eine *Struktur der Pigmentierung* oder auf eine *Rauigkeit der Oberfläche* oder beides zurück.

Texturen natürlicher Oberflächen

Natürliche Oberflächen sind in der Regel texturiert:

- ▶ Sie haben eine regelmäßige, zufällig unregelmäßige Struktur.
- ▶ Die Textur geht entweder auf eine *Struktur der Pigmentierung* oder auf eine *Rauigkeit der Oberfläche* oder beides zurück.
- ▶ **Texturen sind wichtig für die Erzeugung photorealistischer Bilder.**

Texturen natürlicher Oberflächen

Natürliche Oberflächen sind in der Regel texturiert:

- ▶ Sie haben eine regelmäßige, zufällig unregelmäßige Struktur.
- ▶ Die Textur geht entweder auf eine *Struktur der Pigmentierung* oder auf eine *Rauigkeit der Oberfläche* oder beides zurück.
- ▶ Texturen sind wichtig für die Erzeugung photorealistischer Bilder.
- ▶ Texturgradienten sind wichtig für den räumlichen Eindruck.

Beispiel für Texturen: Wolken, Wasser, Vegetation



Tahiti, Hotel Beachcomber Royal

Beispiel für Texturen: Gestein



Pancake Rocks, Punakaiki, Neuseeland

Modellierung von Materialeigenschaften

In POV-Ray wird bei der Modellierung der photometrischen Materialeigenschaften zwischen der Modellierung der Oberfläche und der Modellierung des Inneren unterschieden:

Sichtbare Oberfläche: Im **texture**-statement werden die Farbverteilung, die Rauigkeit und die Oberflächennormalen definiert.

Das Innere: Die Transmissionseigenschaften eines Objektes (Brechung, Absorption, Streuung) werden im **interior**-statement beschrieben.

Material: Eine **material**-Definition deklariert ein Material als Einheit aus **texture** und **interior**.

Beispiel: Material

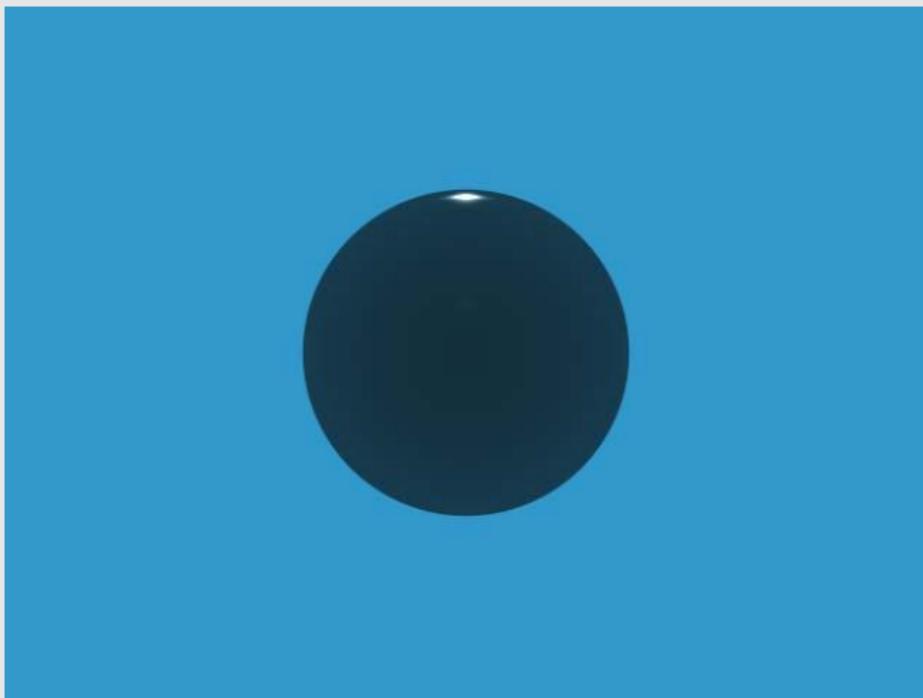
Beispiel

Ein Kristallglass-Material

```
#include "Glass.inc"
#declare MyGlass=
material{
    texture{T_Glass3} // Textur
    interior{I_Glass1}} // Das Innere

object {sphere {0,1
    material{MyGlass}}}}
```

Das Bild zum Beispiel 54



Modellierung von Texturen

In der Computergrafik werden im wesentlichen drei Ansätze verfolgt:

- ▶ Statistische Variation eines Grundmusters (periodische Texturen, wie beispielsweise Holzmaserung),
- ▶ fraktale Strukturen für selbstähnliche Texturen (Vegetation, Gestein, Kristalle),
- ▶ Texturkarten (Bilder natürlicher Texturen, wenn nichts anderes mehr hilft).

Texturen in POV-Ray:

POV-Ray Attribute von Oberflächenstrukturen:

Pigmente: Die Verteilung der Pigmente im Material,

- ▶ entweder eine uniforme Verteilung,
- ▶ ein Muster und eine Farbtabelle
- ▶ oder eine Dichtefunktion.

Modifikatoren: Die Grundmuster können mit Modifikatoren deterministisch oder statistisch variiert werden.

Transformationen: Alle affinen Transformationen sind auf Texturen anwendbar.

Struktur der Oberfläche: Die Rauigkeit der Oberfläche kann modelliert werden, indem die Orientierung der Oberflächennormalen lokal variiert wird. Weitere Parameter beziehen sich auf die Reflektanz: matt, spiegelnd usw.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.

gradient: Verlauf zwischen mehreren Farben.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.

gradient: Verlauf zwischen mehreren Farben.

granite: Gesteinstextur.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.

gradient: Verlauf zwischen mehreren Farben.

granite: Gesteinstextur.

leopard: Regelmäßiges Punktemuster.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.

gradient: Verlauf zwischen mehreren Farben.

granite: Gesteinstextur.

leopard: Regelmäßiges Punktemuster.

marble: Ein Schlierenmuster.

Ein Auszug aus dem Pigment-Musterkatalog:

brick: Ein Muster für Steinmauern.

checker: Ein Schachbrettmuster.

crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.

gradient: Verlauf zwischen mehreren Farben.

granite: Gesteinstextur.

leopard: Regelmäßiges Punktemuster.

marble: Ein Schlierenmuster.

onion: Konzentrische Kugeln, wie Zwiebelschalen.

Ein Auszug aus dem Pigment-Musterkatalog:

- brick: Ein Muster für Steinmauern.
- checker: Ein Schachbrettmuster.
- crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.
- gradient: Verlauf zwischen mehreren Farben.
- granite: Gesteinstextur.
- leopard: Regelmäßiges Punktemuster.
- marble: Ein Schlierenmuster.
- onion: Konzentrische Kugeln, wie Zwiebelschalen.
- spiral1,2: Konzentrische Spiralen.

Ein Auszug aus dem Pigment-Musterkatalog:

- brick: Ein Muster für Steinmauern.
- checker: Ein Schachbrettmuster.
- crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.
- gradient: Verlauf zwischen mehreren Farben.
- granite: Gesteinstextur.
- leopard: Regelmäßiges Punktemuster.
- marble: Ein Schlierenmuster.
- onion: Konzentrische Kugeln, wie Zwiebelschalen.
- spiral1,2: Konzentrische Spiralen.
- wood: Konzentrische Ringe, wie Holzmaserung.

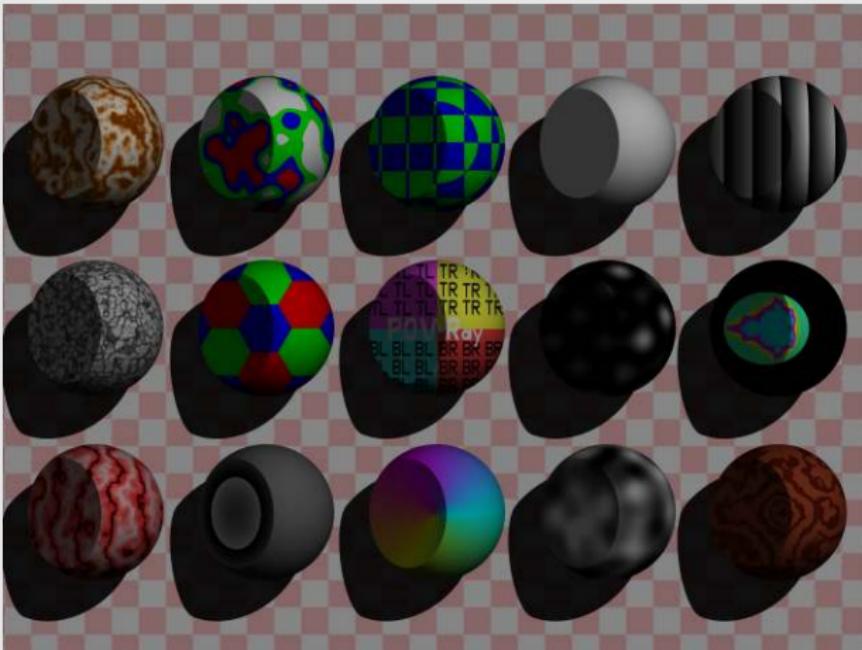
Ein Auszug aus dem Pigment-Musterkatalog:

- brick: Ein Muster für Steinmauern.
- checker: Ein Schachbrettmuster.
- crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.
- gradient: Verlauf zwischen mehreren Farben.
- granite: Gesteinstextur.
- leopard: Regelmäßiges Punktemuster.
- marble: Ein Schlierenmuster.
- onion: Konzentrische Kugeln, wie Zwiebelschalen.
- spiral1,2: Konzentrische Spiralen.
- wood: Konzentrische Ringe, wie Holzmaserung.
- mandel: Die Mandelbrot-Menge, ein fraktales Muster.

Ein Auszug aus dem Pigment-Musterkatalog:

- brick: Ein Muster für Steinmauern.
- checker: Ein Schachbrettmuster.
- crackle: Sieht wie ein Voronoi Diagramm aus, viele unterschiedlich große Dreiecke.
- gradient: Verlauf zwischen mehreren Farben.
- granite: Gesteinstextur.
- leopard: Regelmäßiges Punktemuster.
- marble: Ein Schlierenmuster.
- onion: Konzentrische Kugeln, wie Zwiebelschalen.
- spiral1,2: Konzentrische Spiralen.
- wood: Konzentrische Ringe, wie Holzmaserung.
- mandel: Die Mandelbrot-Menge, ein fraktales Muster.

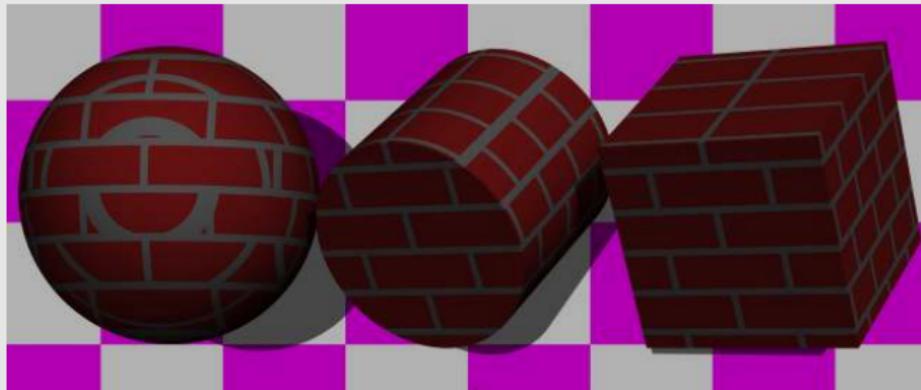
Der Musterkatalog



agate	bozo	checker	color	White	gradient	x
granite	hexagon	image_map	leopard		mandel	
marble	onion	radial	spotted		wood	

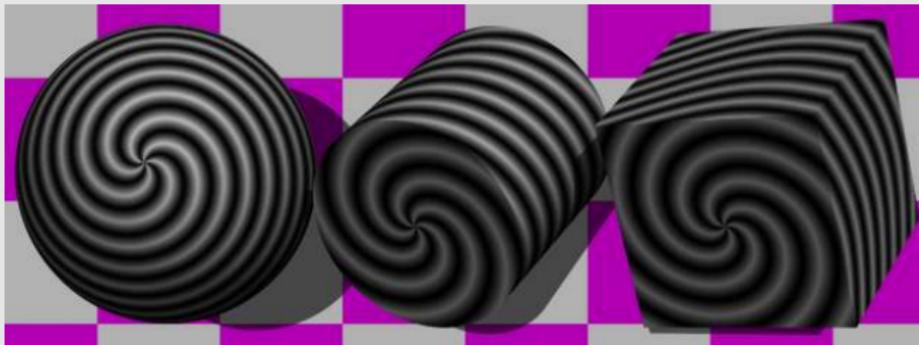
Beispiel: brick

```
#declare T1=  
texture{  
pigment{  
brick  
scale 0.1}}
```



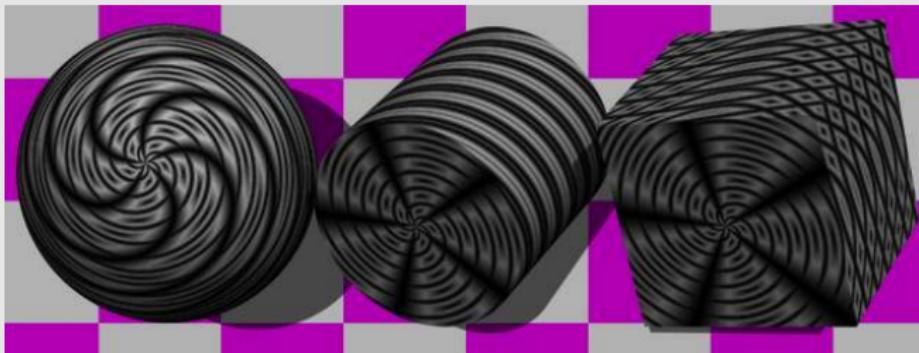
Beispiel: spiral1

```
#include "colors.inc"
#declare T1= // Das Muster als Pigment
texture{
    pigment{
        spiral1 5 // 5 Spiralarme
        color_map{[0.0 Black][1.0 White]}
        scale 0.24}}
```



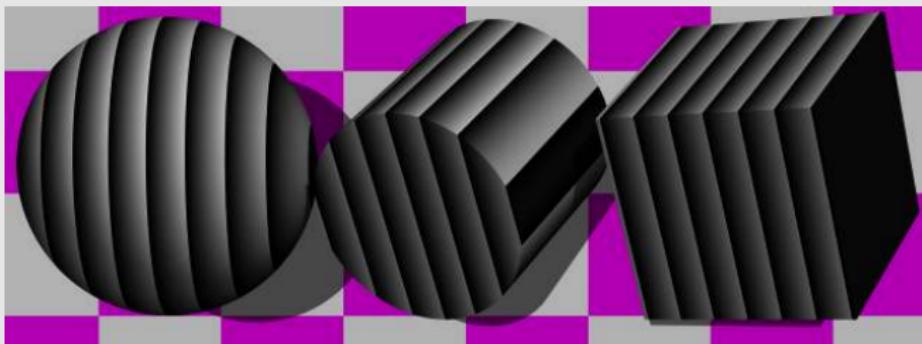
Beispiel: spiral2

```
#declare T1=
texture{
pigment{
    spiral2 5
color_map {[0.0 Black][1.0 White]}
scale 0.24}}
```



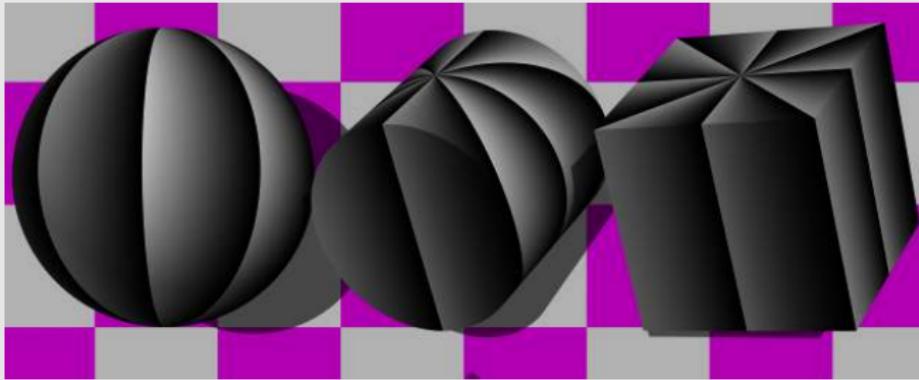
Beispiel: gradient

```
#declare T1=
texture{
pigment{
gradient x
color_map {[0.0 Black][1.0 White]}
scale 0.24}}
```



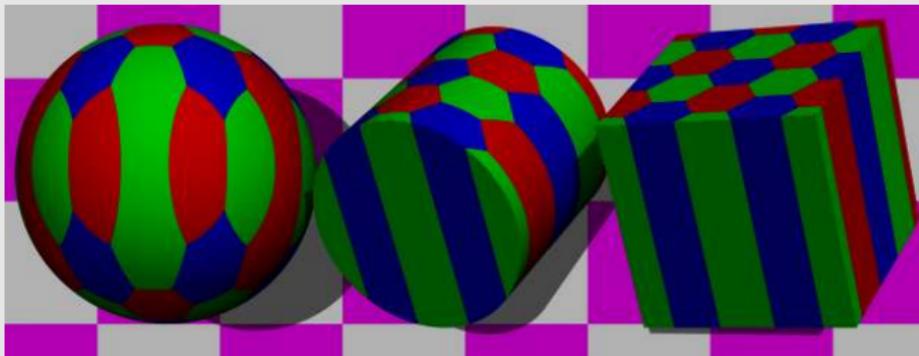
Beispiel: radial

```
#declare T1=
texture{
    pigment{
        radial frequency 8
        color_map {[0.0 Black][1.0 White]}
        scale 0.24}}
```



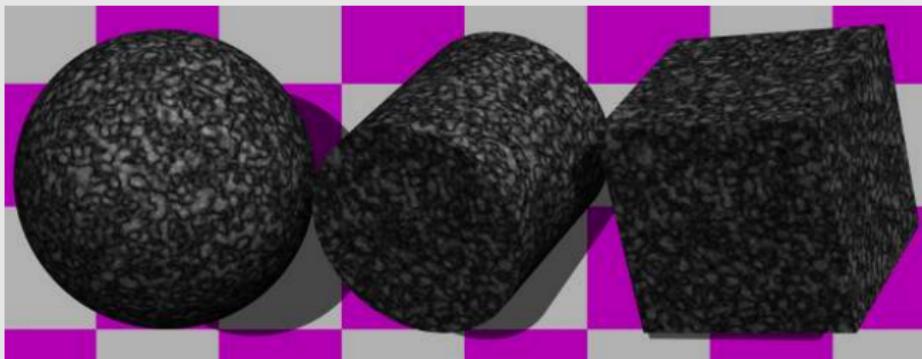
Beispiel: hexagon

```
#declare T1=  
texture{  
    pigment{  
        hexagon  
        scale 0.24}}}
```



Beispiel: granite

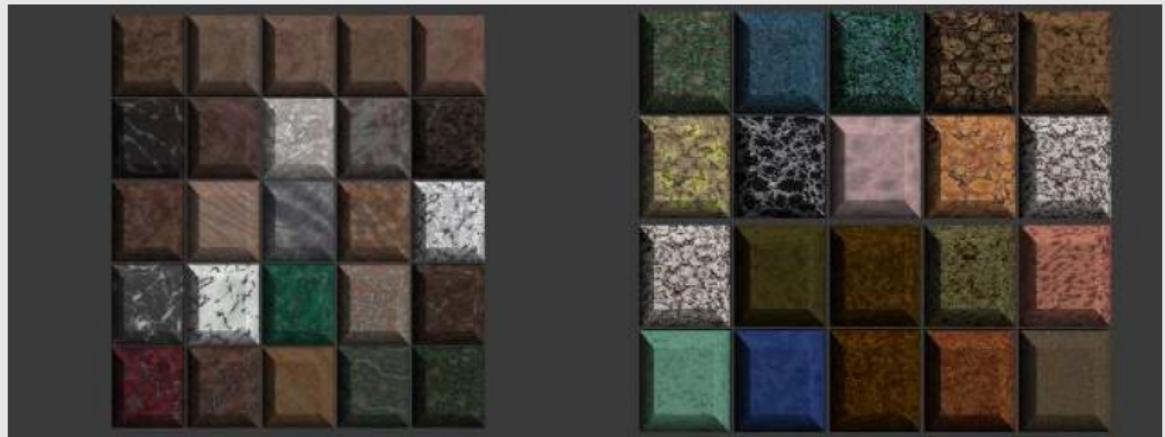
```
#declare T1=
texture{
pigment{
    granite color_map{[0.0 Black][1.0 White]}
    scale 0.24}}
```



Holztexturen in woods.inc



Gesteinstexturen in stones1.inc und stones2.inc



Modifikatoren

Jedes Muster kann mit Modifikatoren verändert werden:

Affine Transformationen: Alle affinen Transformationen, wie Skalierung und Rotation, sind zulässig.

warp: Systematische Verdrehung des Musters um eine Achse.

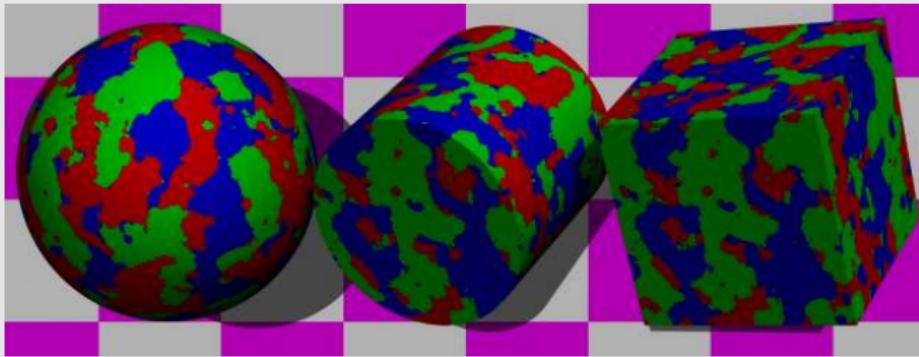
frequency, phase: Systematische Veränderung der Periode und Phasenlage.

turbulence: Zufällige Modifikationen:

turbulence <turbulence vector>

Beispiel: hexagon-Muster mit Turbulenz

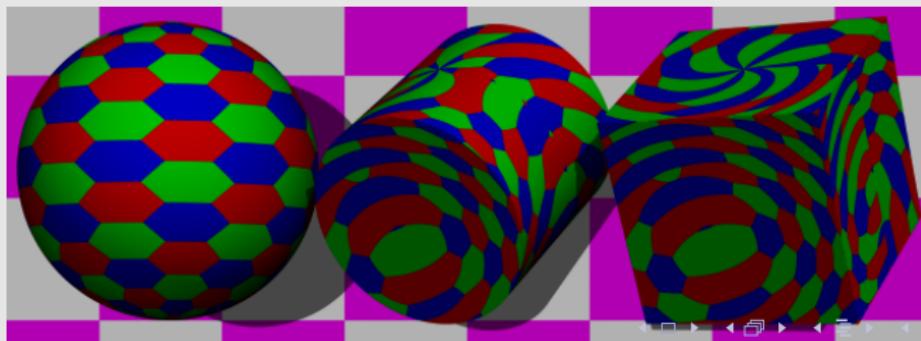
```
#declare T1=
texture{
pigment{
hexagon
scale 0.24
turbulence 1.5}}
```



Beispiel: hexagon-Muster mit warp

Verdrehen um die Y-Achse:

```
#declare T1=
texture{
pigment{
hexagon
scale 0.048
warp {
spherical
orientation y
dist_exp 1 }}}}
```



Wolken als Texturkarte

Wolken als Texturkarte

Das Windsystem der Erde: Warp und Turbulence

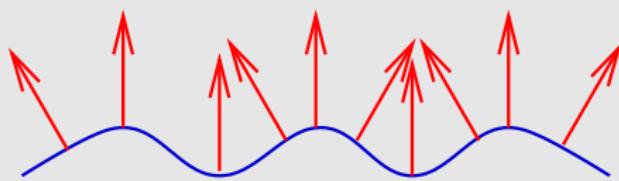
Normalen-Texturen: Rauhe Oberflächen

Die bisher vorgestellten Texturen (checker, hexagon, leopard usw.) basieren auf einer inhomogenen Verteilung der Pigmente.

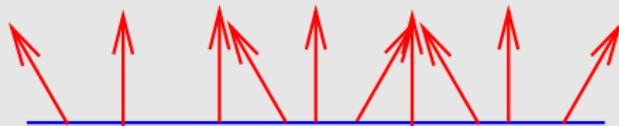
- ☞ Eine Oberfläche kann auch texturiert aussehen, obwohl die Pigmentierung homogen ist, nämlich dann, wenn die Oberfläche nicht glatt ist und ihre Orientierung im Raum lokal variiert.
- ☞ Beispiele: Wellen, Dünen usw.
- ☞ Diese Texturen können in POV-Ray dadurch modelliert werden, daß die Richtung der Oberflächennormalen systematisch und/ oder zufällig variiert wird.

Variation der Oberflächennormalen

Variation der Oberflächennormalen
Eine rauhe Oberfläche, Normalenvektoren



Modellierung in POV-Ray

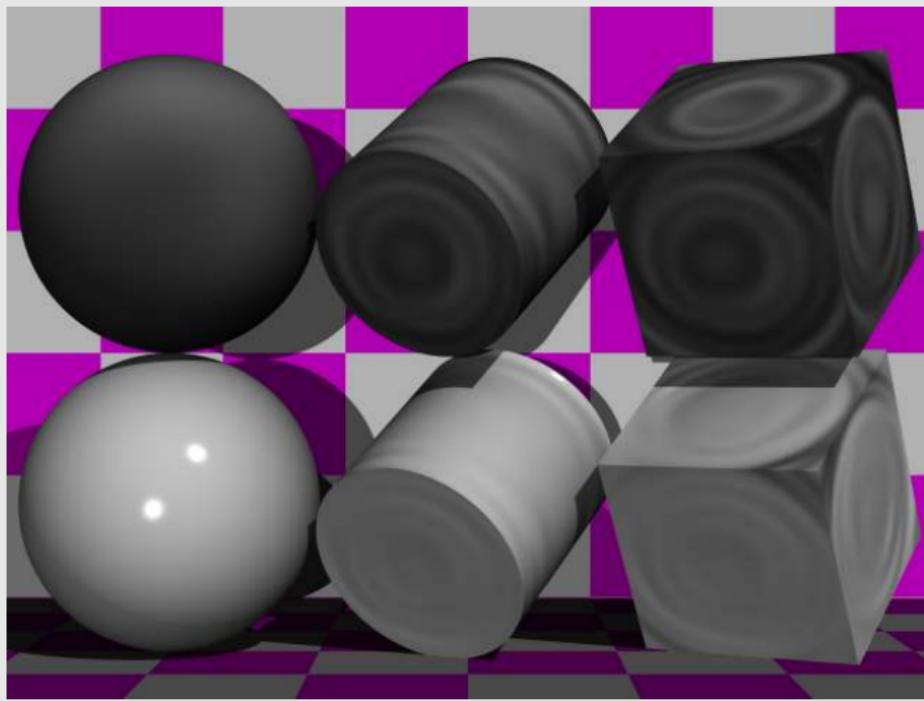


Das waves-Muster als Pigment und als Normalentextur

```
#declare T1=
texture{
    pigment{
        waves
        color_map{
            [0.0 Black]
            [1.0 White]}
        scale 0.1}}
```

```
#declare T2=
texture{
    pigment{White}
    normal{waves 0.4
          scale 0.1}
    finish{phong 0.8 phong_size 200}}
```

waves: als Pigment und als Normalentextur



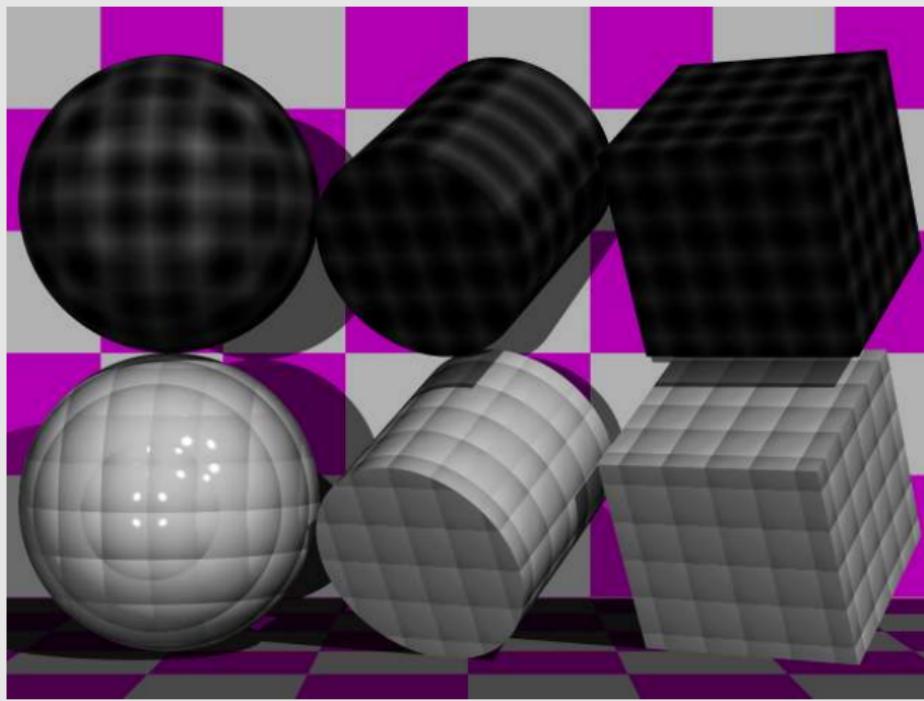
quilted: als Pigment und als Normalentextur

```
#declare T1=
texture{
    pigment{
        quilted color_map{[0.0 Black][1.0 White]}
        control0 1 control1 1
        scale 0.3}}
```

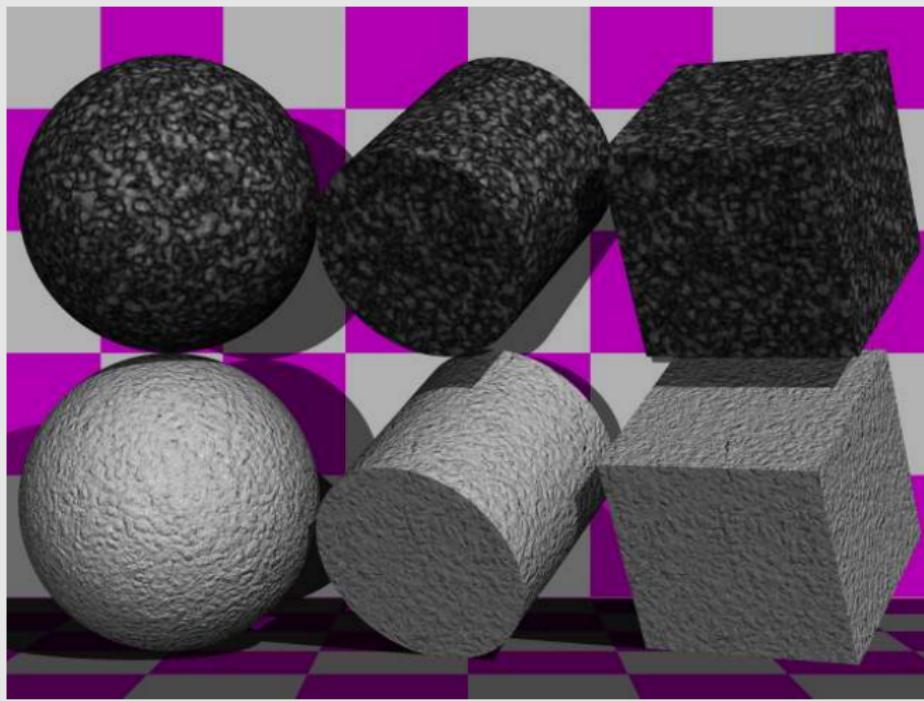


```
#declare T2=
texture{
    pigment{White}
    normal{
        quilted 0.4
        control0 1 control1 1
        scale 0.3}
    finish{phong 0.8 phong_size 200}}
```

quilted: als Pigment und als Normalentextur



granite: als Pigment und als Normalentextur



Beispiel

Abendstimmung auf einer virtuellen tropischen Insel



Beispiel: Haie im Wasser

Beispiel

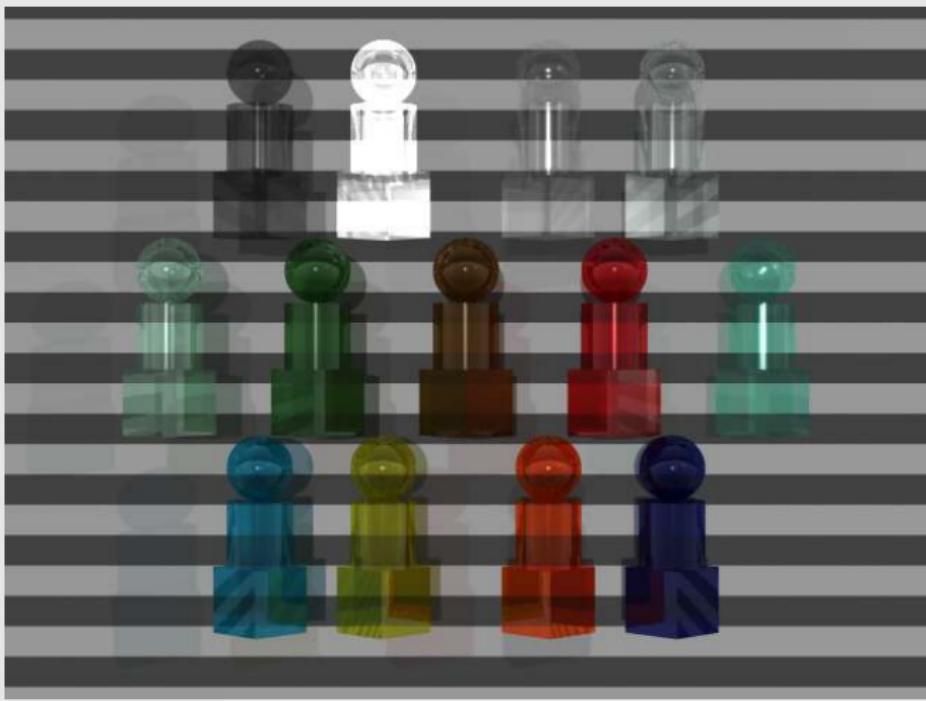
Texturkarten mit transparenten Pigmenten

```
#declare shark=
box {0, <1,1, 0.001> // a shark
pigment {
    image_map {
        gif "shark.gif"
        transmit 10 1.0 }} // cyan ist transparent
    scale <1,0.5,1>}
object {shark
    scale 20
    translate -80*z-6*y}
```

Texturkarten mit Transparenz (Beispiel: 56)



Verschiedene Gläser



Lichtdurchlässige Materialien

Bei der Modellierung von Lichtdurchlässigkeit wird in POV-Ray unterschieden, ob eine Wellenlängenabhängigkeit vorliegt oder nicht:

Filterung: Licht unterschiedlicher Frequenzen wird **unterschiedlich** stark absorbiert, wie beispielsweise bei buntem Glas oder Wasser.

Transparenz: Licht unterschiedlicher Frequenzen wird **gleich** stark absorbiert, wie bei Materialien mit kleinen, lichtdurchlässigen Poren, beispielsweise Staubschichten oder dünnen Textilien.

Filterung und Transparenz kann über den Farbvektor im pigment-statement angegeben werden:

```
pigment {colour rot, grün, blau, Filter,  
Transparenz}
```

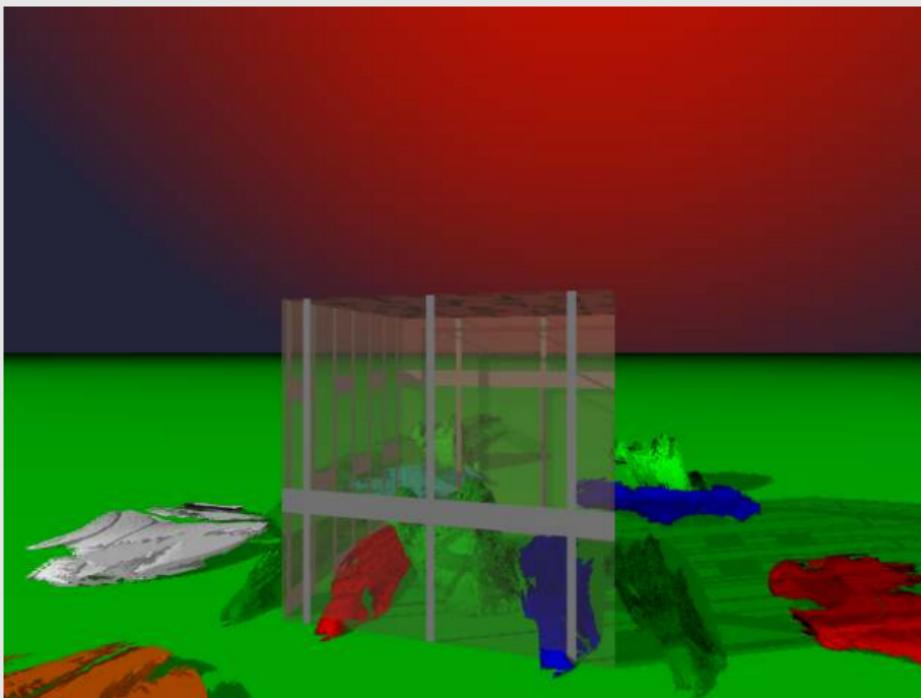
Ein virtuelles Treibhaus

Beispiel

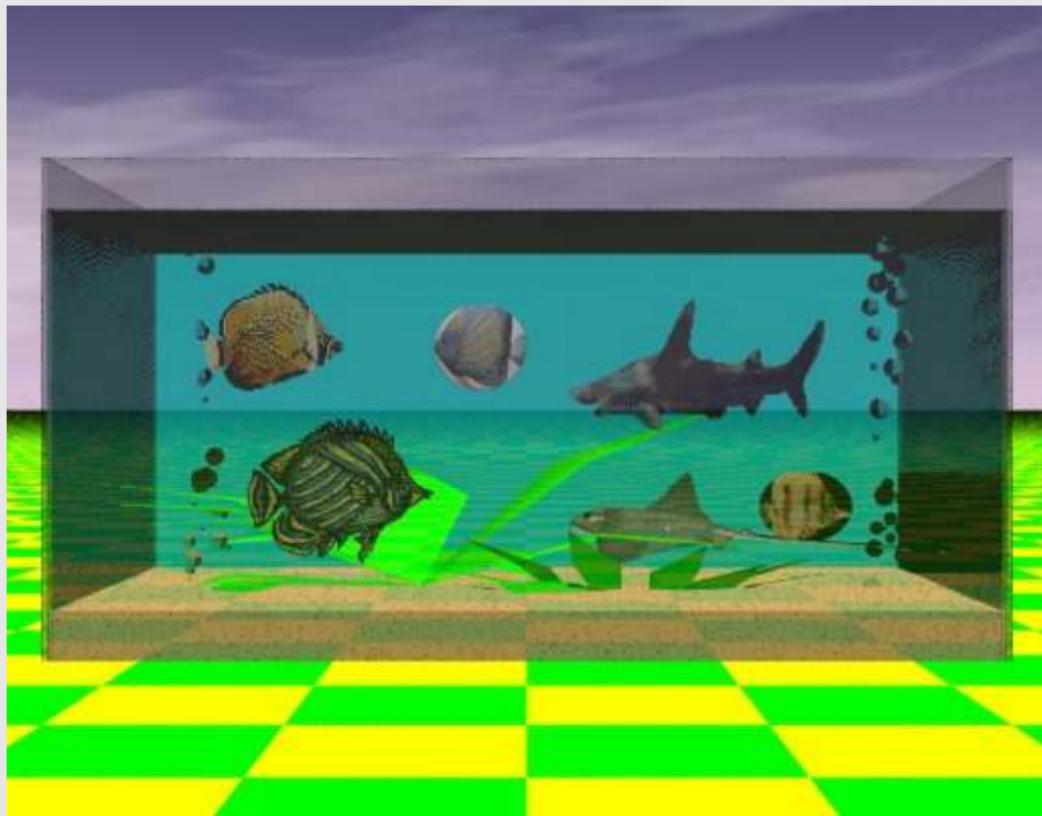
Ein Glashaus mit grauen Fensterrahmen und trüben Glasscheiben, die leicht blau-grün getönt sind.

```
box { -3*(x+z) , 3*x+6*y+3*z  
      pigment { brick  
                  colour Gray  
                  colour <1, 0.9, 0.8, 0.7>  
                  rotate 90*x  
                  scale 0.3*(3*y+x+z) }  
      hollow }
```

Ein Beispiel für transparente Pigmente



Transparente Pigmente, Wasser und Glas



Normalentextur: Wellen auf den Ozeanen

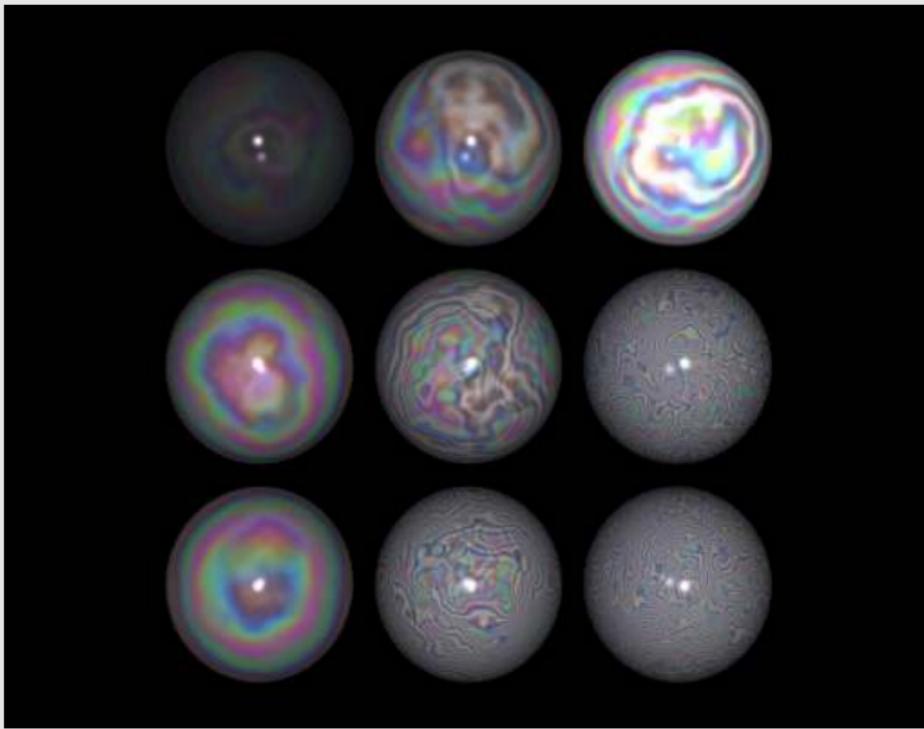
Meer mit animierten Wellen

Normalentextur: Flutberge und Wellen auf den Ozeanen

Material mit Dichtefunktion: Erdatmosphaere

Erdatmosphaere

Seifenblasen



Ein dünner Ölfilm auf den Wogen

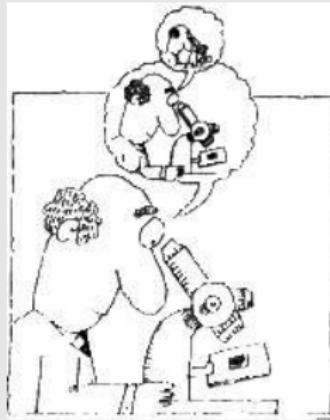


```
finish {  
    irid {0.5 thickness 2 turbulence 0.1}}}
```

Teil IX

Fraktale

Fraktale und Textur



24

Fraktale

- Diskrete Fraktale
- Die Hausdorff-Dimension
- Anwendung fraktaler Strukturen

25

Chaotische Folgen

26

Fraktale Objekte und Texturen

Eigenschaften fraktaler Objekte

- ▶ Kontinuierliche Modelle versagen bei vielen natürlichen Objekten:
Küstenlandschaften, Bäume, Gebirge, Blätter.
- ▶ **Mandelbrot 1977:** „Wie lang ist die Küste von Großbritannien?“
- ▶ Die Antwort ist abhängig von der Auflösung, mit der die Küstenlinie betrachtet wird.

Küstenlinie der U.S.A.

Ostküste der U.S.A.

Wie gemessen?	Länge ca.
Auf dem Globus	2.000 - 3.000 M
Atlas	4.000 - 5.000 M
Auf der Seekarten	10.000 - 12.000 M
Als Fußgänger	\sim 15.000 M
Weg einer Ameise	\sim 30.000 M
\Rightarrow	∞

Küstenlinie der U.S.A.

Ostküste der U.S.A.

Wie gemessen?	Länge ca.
Auf dem Globus	2.000 - 3.000 M
Atlas	4.000 - 5.000 M
Auf der Seekarten	10.000 - 12.000 M
Als Fußgänger	\sim 15.000 M
Weg einer Ameise	\sim 30.000 M
\Rightarrow	∞

Länge ist bedeutungslos für fraktale Kurven.

Erzeugung fraktaler Objekte

Fraktaler Kurven und Objekte werden erzeugt, indem wiederholt eine Transformation f auf eine Punktmenge angewendet wird.

- ▶ Dabei kann ein linearer oder baumrekursiver Prozeß entstehen.

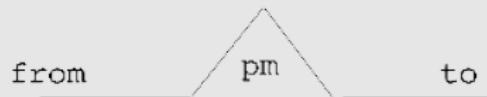
Erzeugung fraktaler Objekte

Fraktaler Kurven und Objekte werden erzeugt, indem wiederholt eine Transformation f auf eine Punktmenge angewendet wird.

- ▶ Dabei kann ein linearer oder baumrekursiver Prozeß entstehen.
- ▶ Je mehr Iterationen ausgeführt werden, desto detailreicher wird das fraktale Objekt.

Die Koch'sche Schneeflocke:

Helge Koch 1904



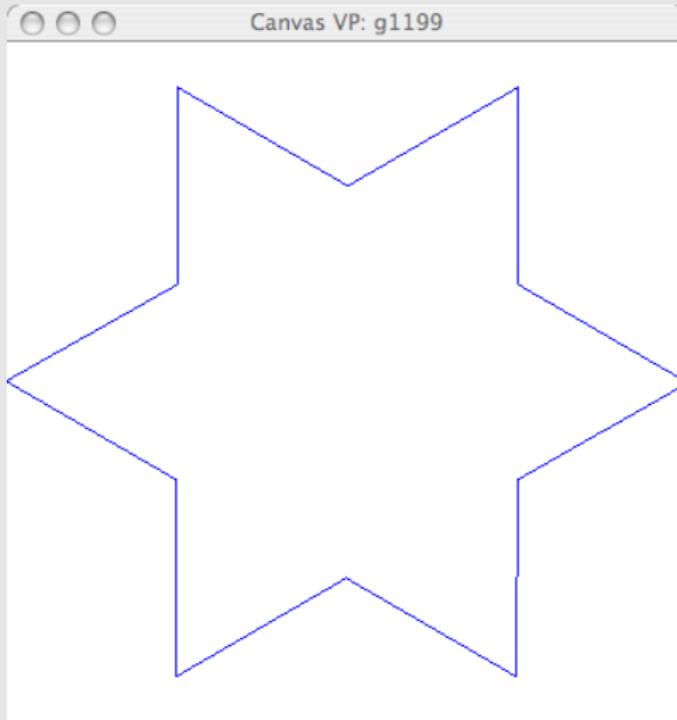
Eigenschaften der Kurve:

Selbstähnlichkeit: Wird ein beliebiger Abschnitt um den Faktor 3 vergrößert, so erhält man das Original.

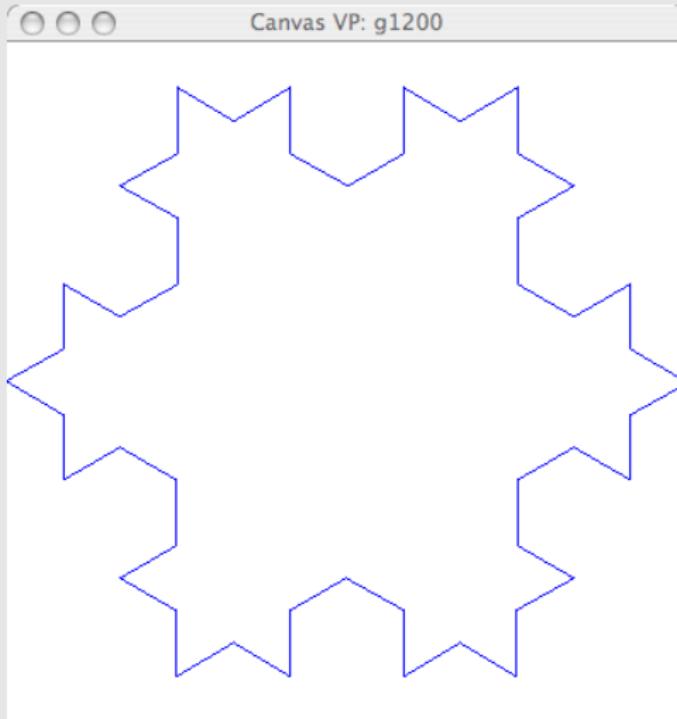
Stetigkeit: Die Kurve ist überall stetig, aber nicht überall glatt.

Tangenten: Es gibt nirgendwo eine Tangente, da es zwischen zwei beliebigen Punkten beliebig viele „zick-zack“-Kurven gibt.

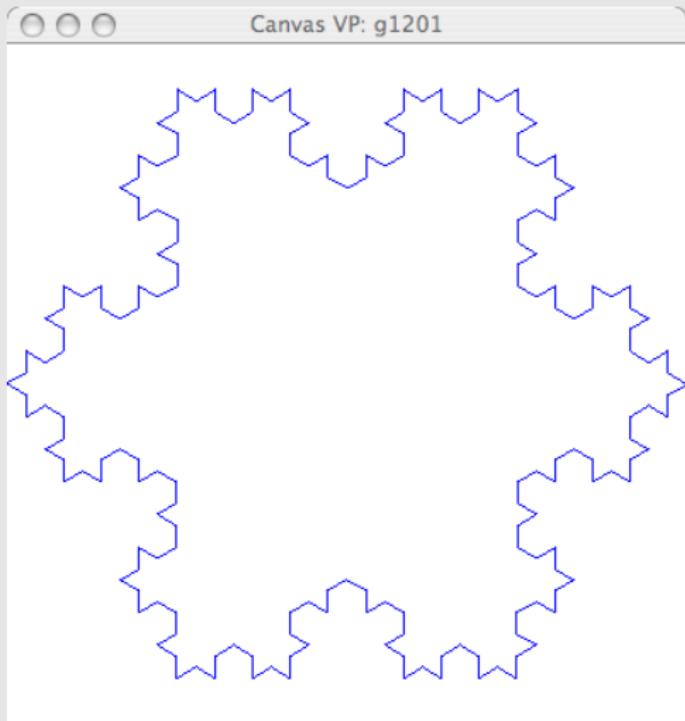
∞ -Länge zwischen zwei Punkten: Die Länge der Kurve geht gegen ∞ , da die Kurve bei jeder Iteration um den Faktor $\frac{4}{3}$ verlängert wird.



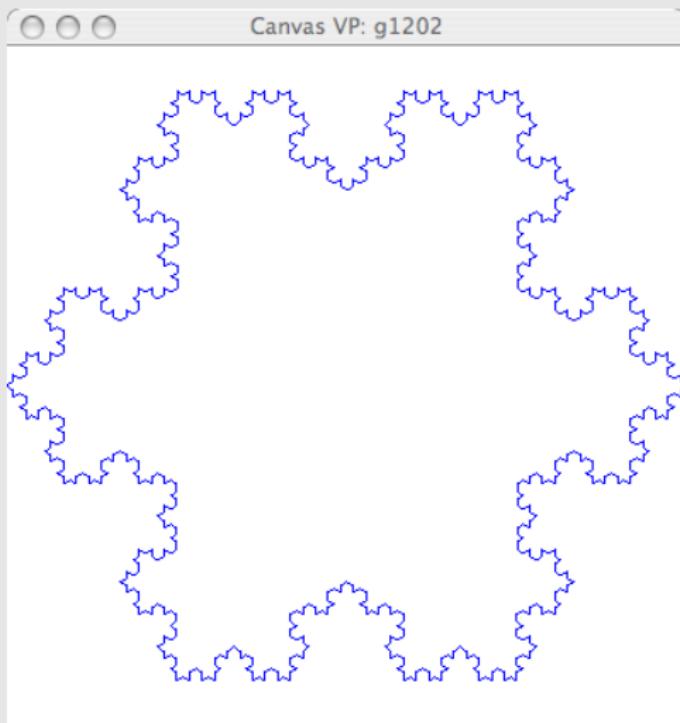
Iteration 1, 12 Linien



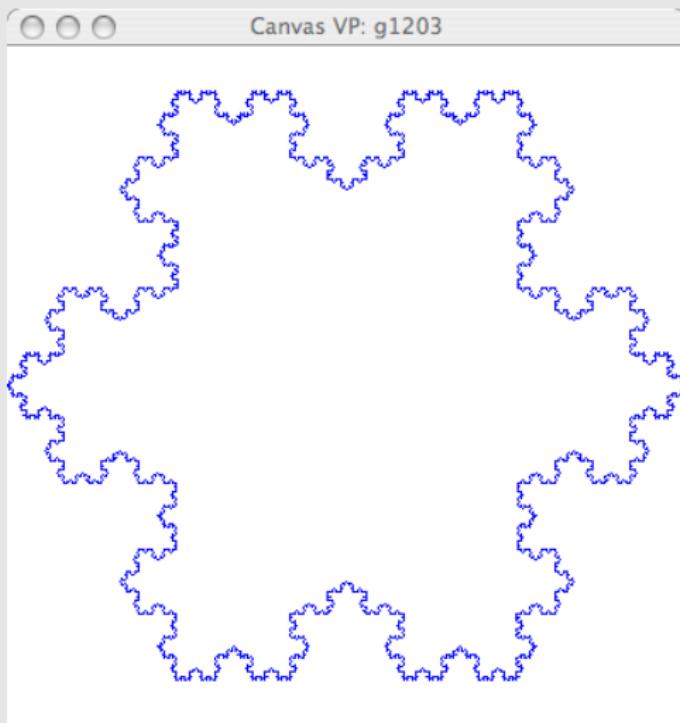
Iteration 2, 48 Linien



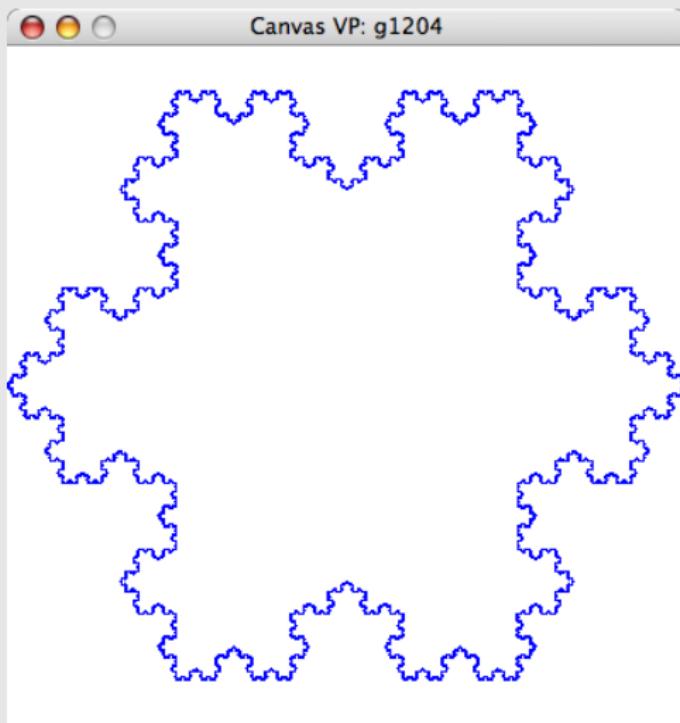
Iteration 3, 192 Linien



Iteration 4, 768 Linien



Iteration 5, 3072 Linien



Iteration 6, 12288 Linien

Schneeflöckchen

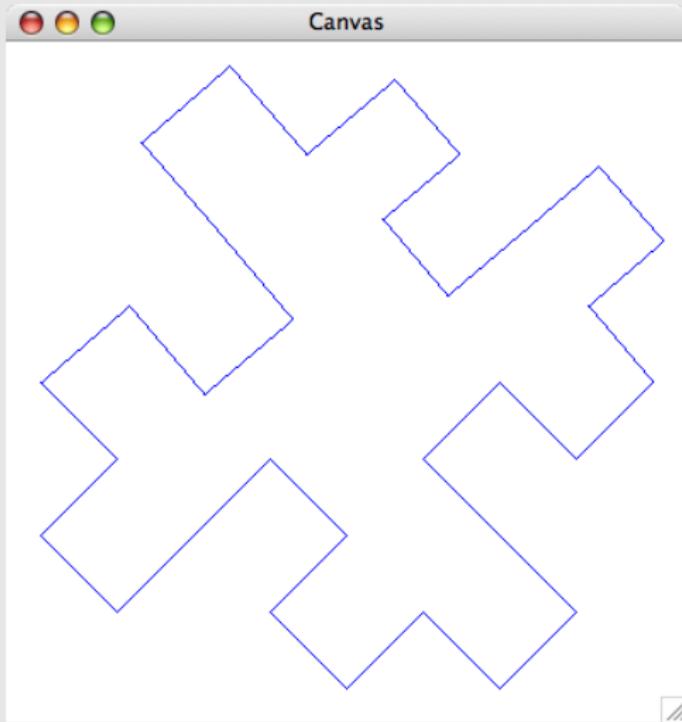
Schneegestöber

Quadratische Koch-Kurve

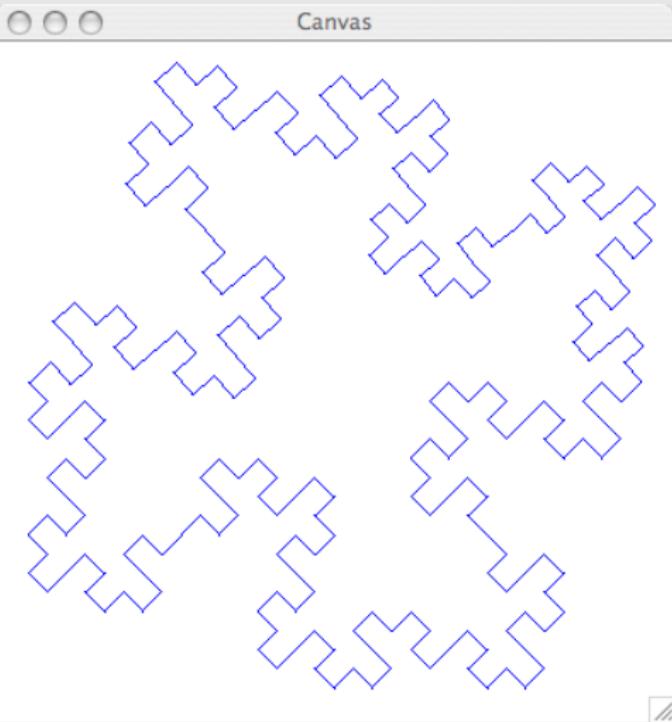
Der Generator:



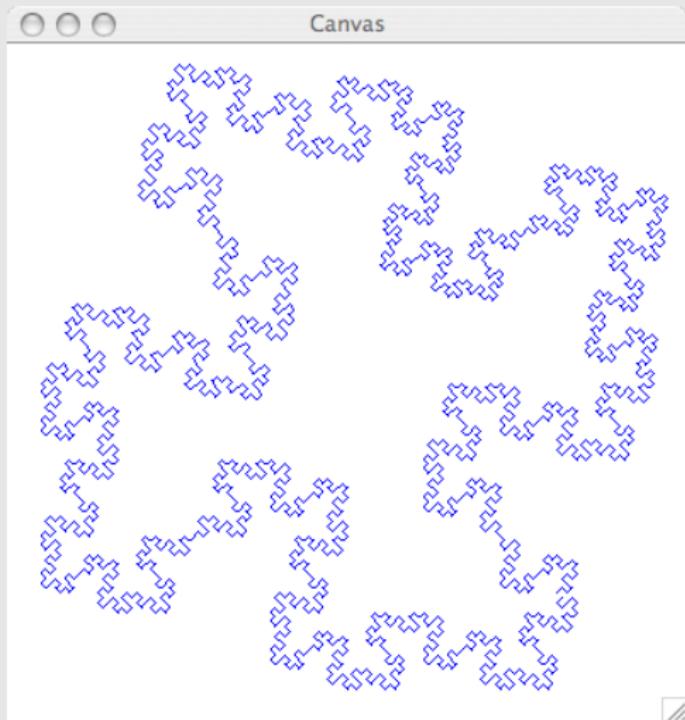
Die Länge der Kurve geht ebenfalls gegen ∞ , da der Generator die Länge bei jeder Iteration verdoppelt.



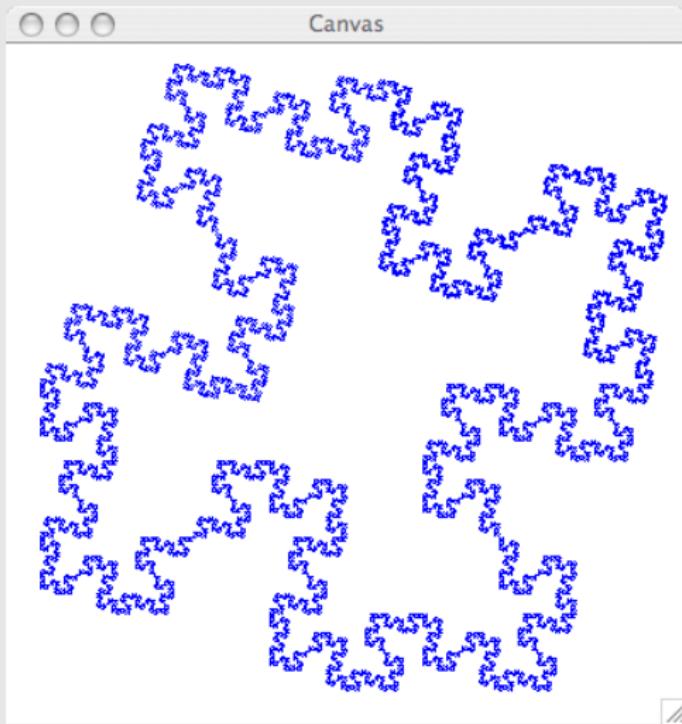
Iteration 1



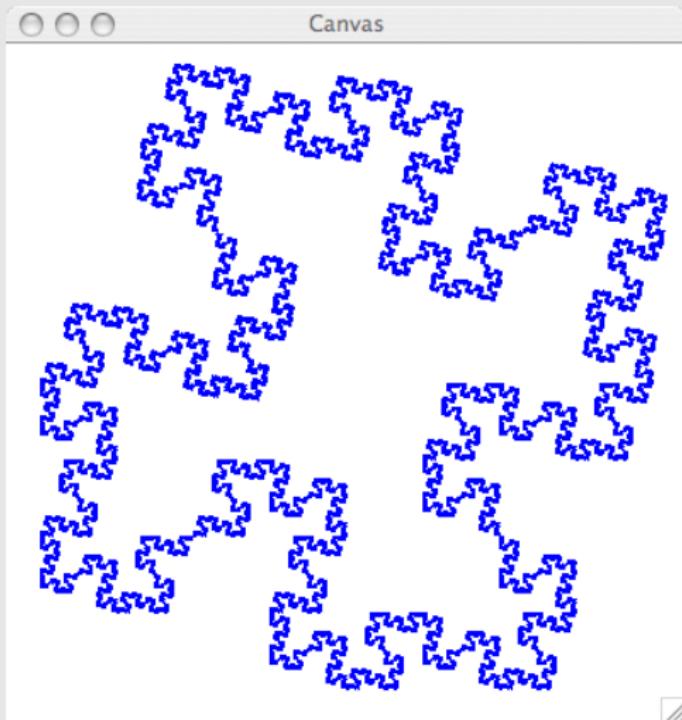
Iteration 2



Iteration 3

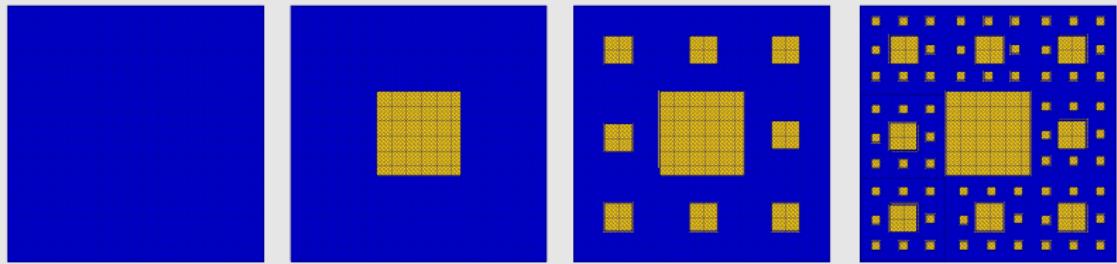


Iteration 4



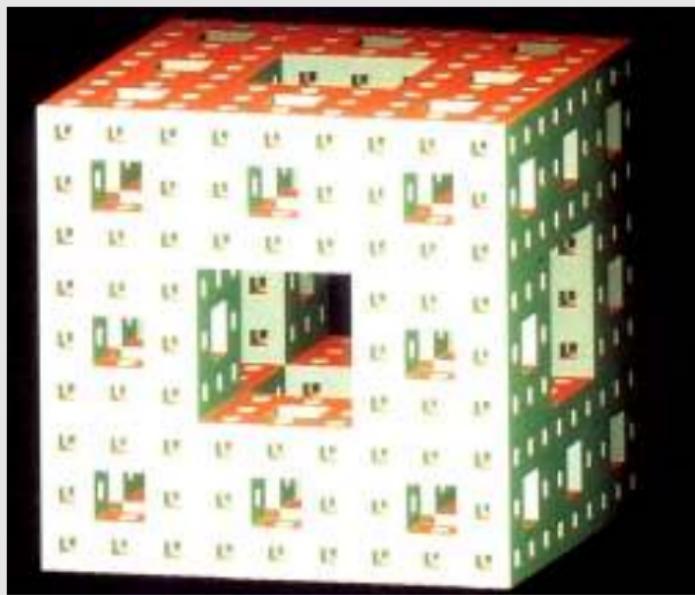
Iteration 5

Der Sierpinski-Teppich



Der Sierpinski-Teppich beginnt mit einem Quadrat.
Das rekursive Entfernen von quadratischen Flächen
generiert die fraktale Form.

Menger-Schwamm



Ein **Menger-Schwamm** ist das 3-dimensionale Analogon zum Sierpinski-Teppich.

Mengerschwamm

Die Dimension von Objekten:

Wenn ein n -dimensionales Objekt senkrecht zu allen seinen Dimensionen verschoben wird, entspricht der überstrichene Raum einem $n + 1$ -dimensionalen Objekt.

- ▶ Ein Punkt hat die Dimension 0.

Die Dimension von Objekten:

Wenn ein n -dimensionales Objekt senkrecht zu allen seinen Dimensionen verschoben wird, entspricht der überstrichene Raum einem $n + 1$ -dimensionalen Objekt.

- ▶ Ein Punkt hat die Dimension 0.
- ▶ Durch Translation des Punktes entsteht eine Gerade (1-dimensional).

Die Dimension von Objekten:

Wenn ein n -dimensionales Objekt senkrecht zu allen seinen Dimensionen verschoben wird, entspricht der überstrichene Raum einem $n + 1$ -dimensionalen Objekt.

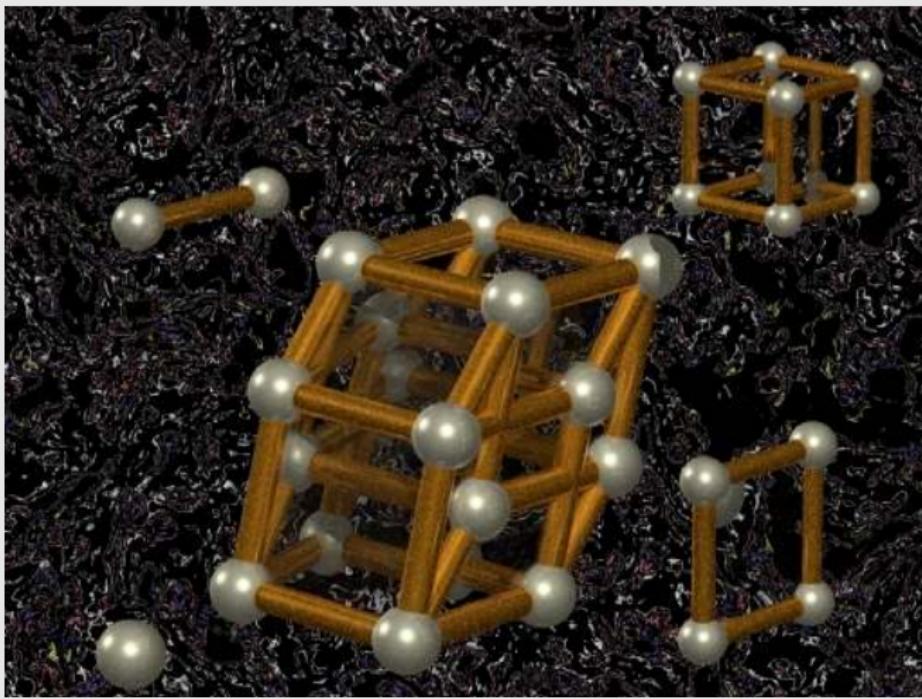
- ▶ Ein Punkt hat die Dimension 0.
- ▶ Durch Translation des Punktes entsteht eine Gerade (1-dimensional).
- ▶ Durch Translation einer Geraden senkrecht zur Geraden entsteht ein Rechteck (2-dimensional).

Die Dimension von Objekten:

Wenn ein n -dimensionales Objekt senkrecht zu allen seinen Dimensionen verschoben wird, entspricht der überstrichene Raum einem $n + 1$ -dimensionalen Objekt.

- ▶ Ein Punkt hat die Dimension 0.
- ▶ Durch Translation des Punktes entsteht eine Gerade (1-dimensional).
- ▶ Durch Translation einer Geraden senkrecht zur Geraden entsteht ein Rechteck (2-dimensional).
- ▶ Durch Translation eines Rechtecks senkrecht zur Ebene entsteht ein Quader (3-dimensional), eines Quaders ein vierdimensionaler Hyperquader usw.

Würfel in 1, 2, 3 und 4 Dimensionen



Hausdorff-Dimension

Wachstum der Überdeckungsmenge

Definition (Hausdorff-Dimension)

Ist X eine Menge im m -dimensionalen Raum und $n(\varepsilon)$ die Anzahl der m -dimensionalen Kugeln mit dem Durchmesser ε die ausreichen, um X zu überdecken, dann hat X die **Hausdorff-Dimension D**, wenn für $n(\varepsilon) \rightarrow \infty$ die Anzahl $n(\varepsilon)$ proportional zu ε^{-D} ist.

$$n(\varepsilon) \sim \varepsilon^{-D}$$

$$\text{bzw. } n(\varepsilon) = c \cdot \varepsilon^{-D}$$

$$D = \frac{\log(n(\varepsilon)) - \log(c)}{-\log(\varepsilon)}$$

Beispiele:

Gerade



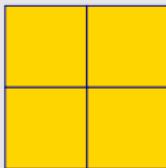
$$\varepsilon = 0,5$$

$$n = 2$$

$$D = 1$$

$$D = \frac{\log(n(\varepsilon)) - \log(c)}{-\log(\varepsilon)}$$

Quadrat

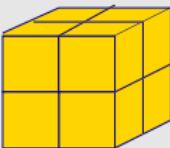


$$\varepsilon = 0,5$$

$$n = 4$$

$$D = 2$$

Kubus



$$\varepsilon = 0,5$$

$$n = 8$$

$$D = 3$$

Die Dimension fraktaler Objekte:



- ▶ In jedem Iterationsschritt ersetzt der Generator Ausschnitte des Objekts durch eine *feste Anzahl N* von skalierten Kugeln.
- ▶ Der Skalierungsfaktor ist bei jedem Iterationsschritt gleich.

Sei S die Zahl der Kugeln, die den *Durchmesser* des Generators überdecken und N die Zahl der Kugeln, die den Generator vollständig überdecken.
Wir wählen $c = 1$.

$$\begin{aligned}\varepsilon &= \frac{1}{S}, \quad n(\varepsilon) = N \\ D &= \frac{\log(n(\varepsilon)) - \log(c)}{-\log(\varepsilon)} = \frac{\log(n(\varepsilon))}{-\log(\varepsilon)} \\ D &= \frac{\log(N)}{-\log(\frac{1}{S})} = \frac{\log(N)}{\log(S)}\end{aligned}$$

Gebrochene Dimension

Für fraktale Kurven ergeben sich **gebrochene Dimensionen** (fraktale Dimensionen):



$$D = \frac{\log(N)}{\log(S)}$$

Für die Koch-Kurve: $N = 8, S = 4$

$$D = \frac{\log 8}{\log 4} = 1,5$$

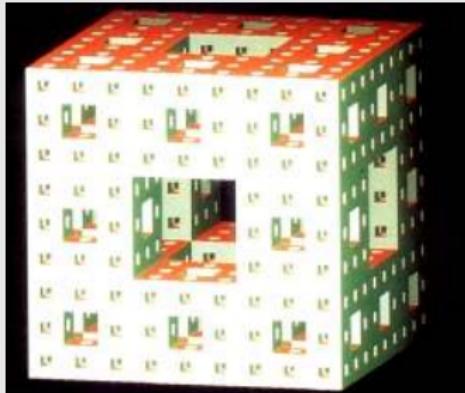
Beispiele für fraktale Dimensionen

Sierpinskigasket



- ▶ Schneeflocke: $N = 4, S = 3, D = 1, 2618$
- ▶ Küstenlinie: $D = 1, 15 - 1, 25$
- ▶ Sierpinskigasket: $N = 3, S = 2, D = 1, 584 \dots$
- ▶ Sierpinski-Teppich: $N = 8, S = 3, D = 1, 89 \dots$

Die Dimension des Menger-Schwamms



- ▶ Unterteilung $3 \times 3 \times 3 = 27$ Würfel.
- ▶ 7 Würfel werden entfernt.
- ▶ $N = 20, S = 3, D = \frac{\log 20}{\log 3} = 2,727$.

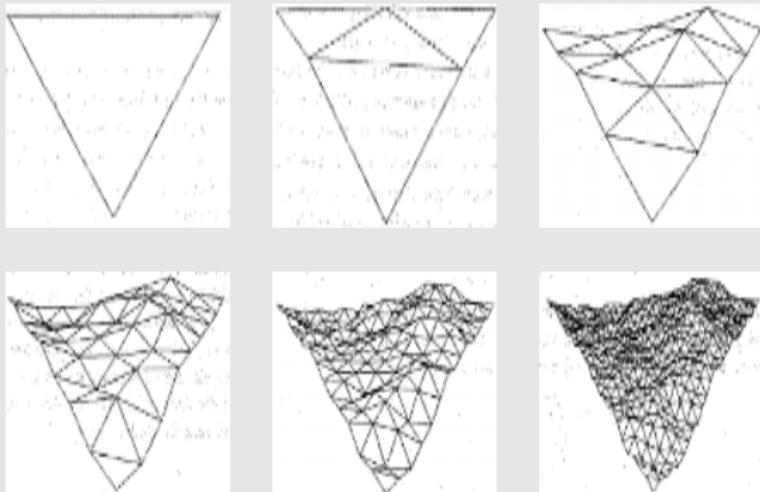
Anwendung fraktaler Strukturen

- ▶ Modellierung natürlicher Objekte mit geringem Speicheraufwand zur Darstellung von
 - ▶ Gebirgen, Flüssen, Küsten,
 - ▶ Wolken,
 - ▶ Pflanzen,
 - ▶ Oberflächenrauhigkeit (Cantor-Mengen).
- ▶ Datenkompression.

Anwendung fraktaler Strukturen

- ▶ Beschreibung physikalischer Prozesse:
 - ▶ Rauschen bei Radioübertragungen,
 - ▶ Diffusion,
 - ▶ Materieverteilung im Weltall (Cantor dust).
- ▶ Umweltschutz.

Modellierung von Gebirgen



Erzeugung einer fraktalen Fläche durch sukzessive Teilung der Dreiecksflächen.

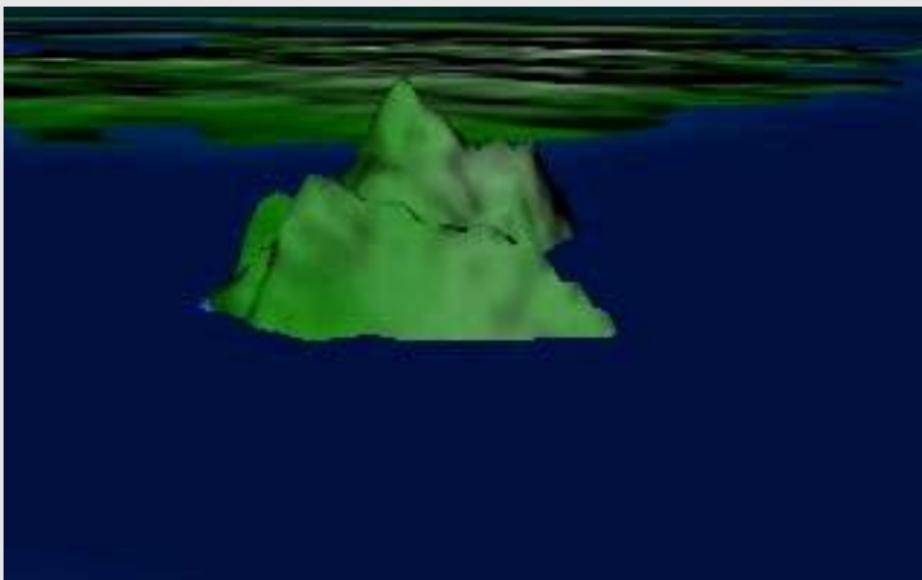
Erzeugung einer fraktalen Fläche durch sukzessive Teilung der Dreiecksflächen.

Fraktale Gebirgslandschaft



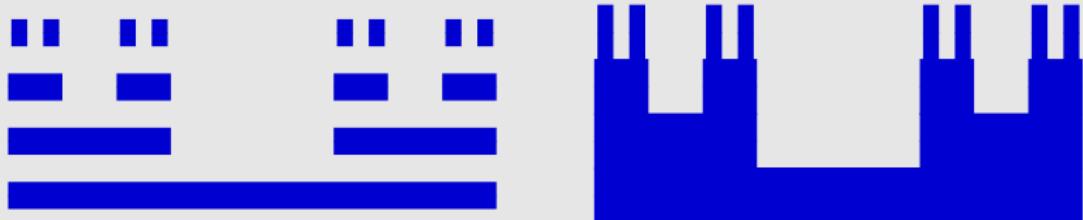
Fraktale Gebirgslandschaft, erzeugt durch sukzessive Dreiecksunterteilung, 8 Unterteilungs-Iterationen

Gebirge in Cinema 4D



Modellierung von Oberflächenrauhigkeiten

- ▶ Cantor-Balken $0 < d < 1, 0$. Die ersten vier Iterationen eines Cantor-Balkens sowie ein Modell für eine rauhe Oberfläche:



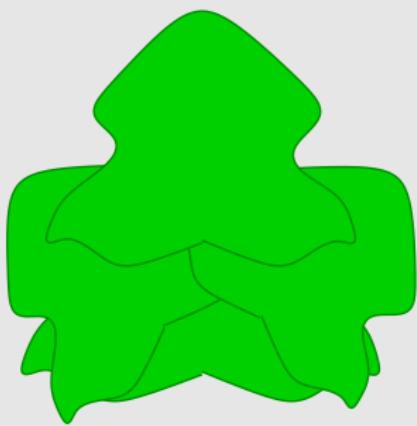
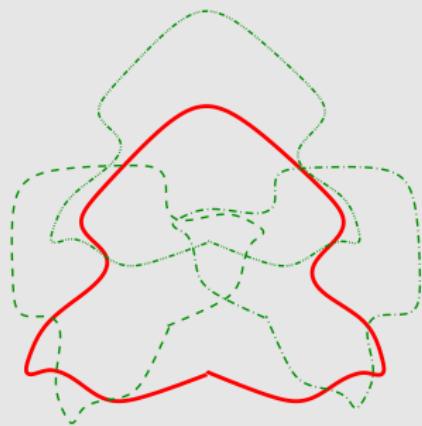
- ▶ 3D-Äquivalent: Cantor-Staub, z.B. Modell für Materieverteilung.

Komposition von höheren Strukturen:Kollagen

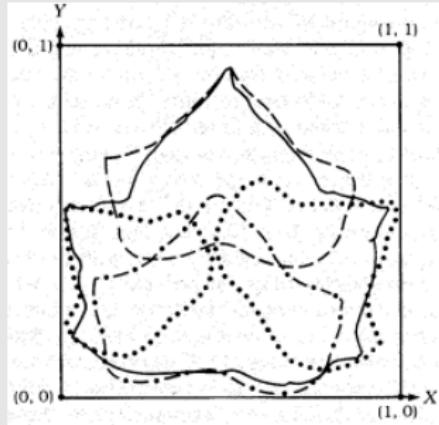
- ☞ Randomisieren,
- ☞ Affine Transformationen
(Rotationen, Scherung, Skalierung, . . .)

Ein fraktales Ahornblatt

Ein Blatt entsteht als Kollage von affin transformierten Kopien der Grundfigur:

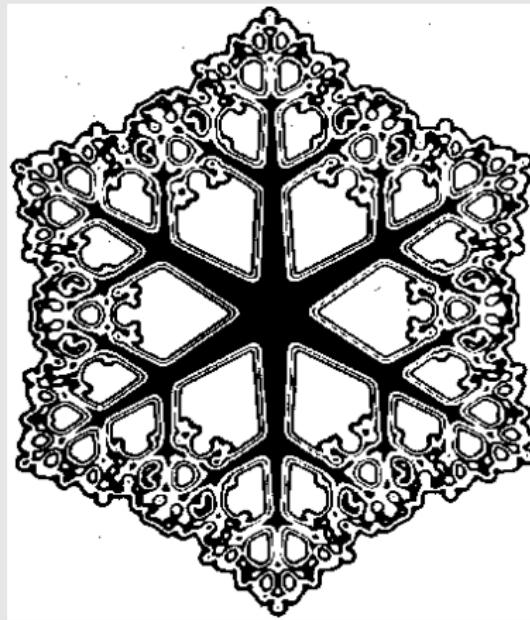


Beispiel: Kollagetechnik



Kollage: Schneeflocke

Die Generierung einer fraktalen Struktur und deren 6-fache Reproduktion resultiert in einer Schneeflocke.



Ein fraktaler Fliederstrauß



Bildkompression:

Die Kollagetechnik kann zur Datenkompression genutzt werden:

- ▶ Gegeben sei eine Grundfigur und eine Folge von affinen Transformationen; dann stellt die Folge der Transformationen (Tabelle der Koeffizienten), die das Bild als Kollage erzeugen, eine vollständige Beschreibung des Bildes dar.
- ▶ Für Bilder, die selbstaffine Strukturen enthalten, kann durch eine solche Codierung eine immense Datenreduktion erreicht werden.

$$x_{n+1} = r \cdot x_n \cdot \cos(A) - s \cdot y_n \cdot \sin(B) + h$$

$$y_{n+1} = r \cdot x_n \cdot \sin(A) + s \cdot y_n \cdot \cos(B) + k$$

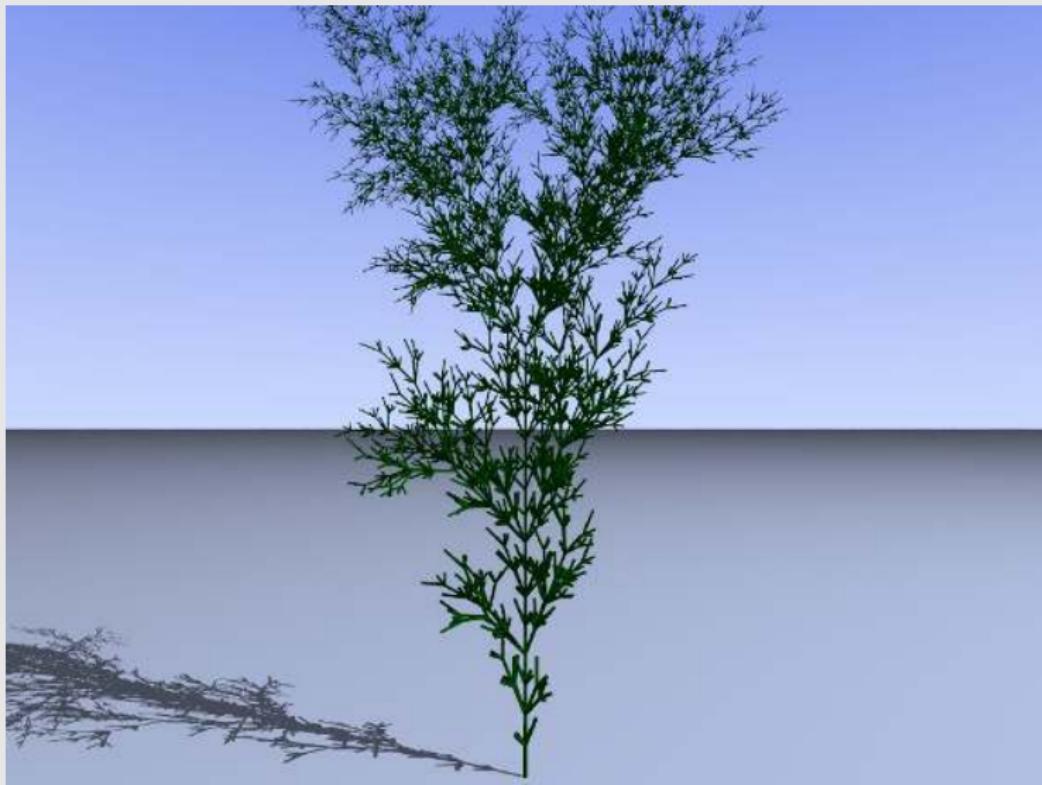
Koeffizienten für den Farn

#	Translationen		Rotationen		Skalierungen		P
	h	k	A	B	r	s	
1	0.0	0.0	0.0	0.0	0.0	1.6	0.005
2	0.0	1.6	-2.5	-2.5	0.85	0	0.8
3	0.0	1.6	49	0.3	0.34	0.975	0.975
4	0.0	0.44	120	-50	0.3	0.37	0.976

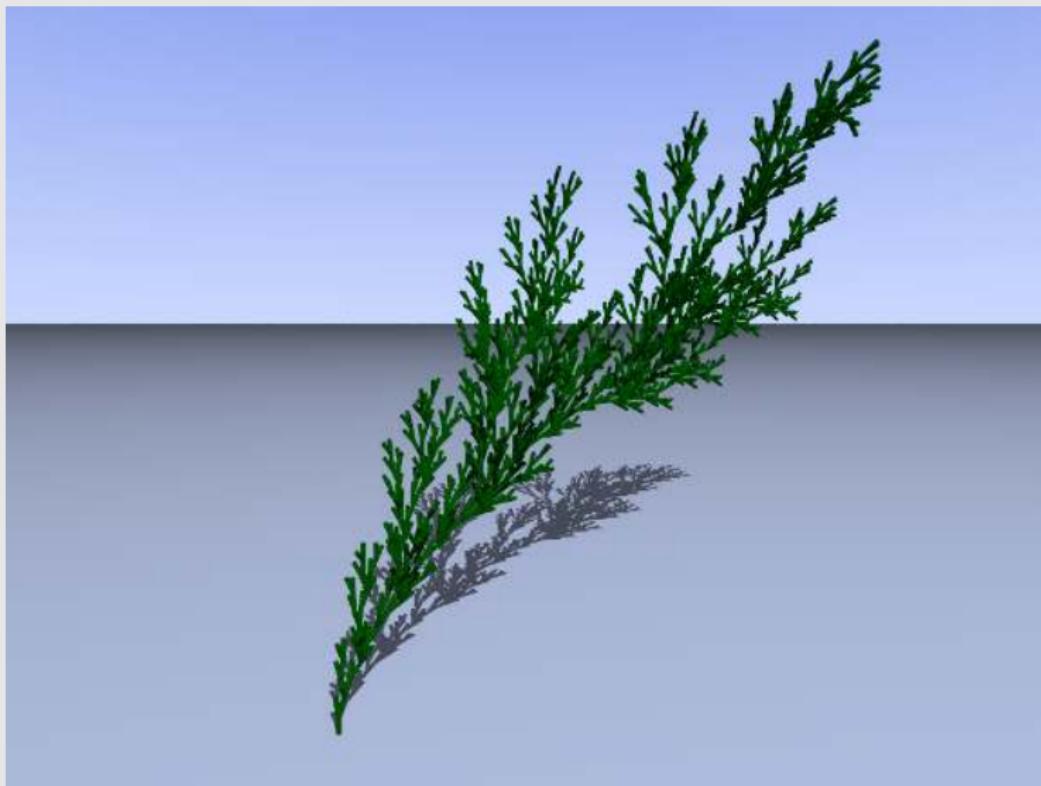
Ein fraktaler Farnwedel und eine Farnpflanze



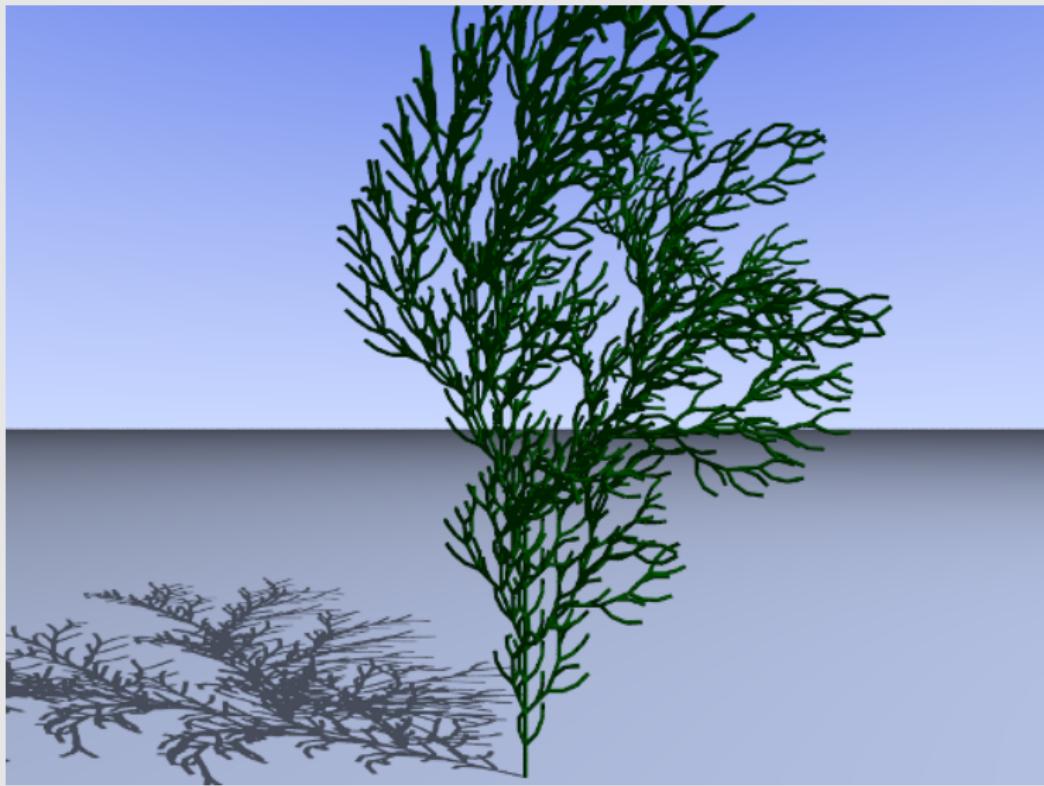
Erzeugung mit Lindenmayer-Systemen



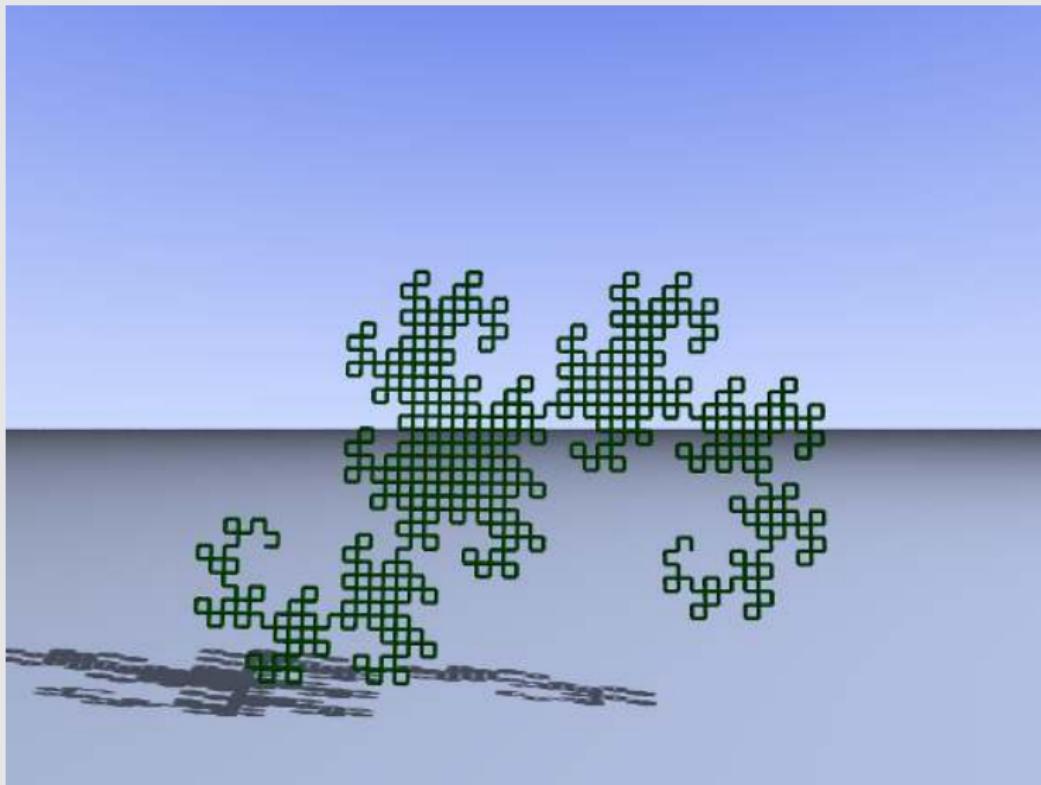
Erzeugung mit Lindenmayer-Systemen



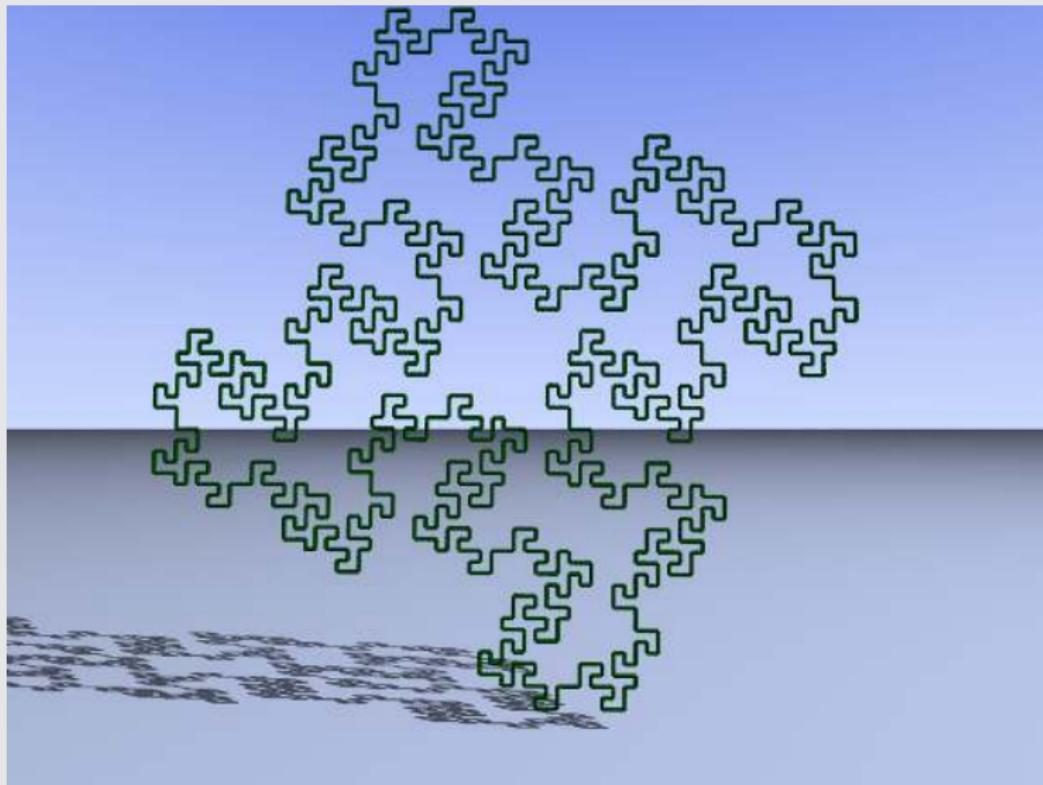
Erzeugung mit Lindenmayer-Systemen



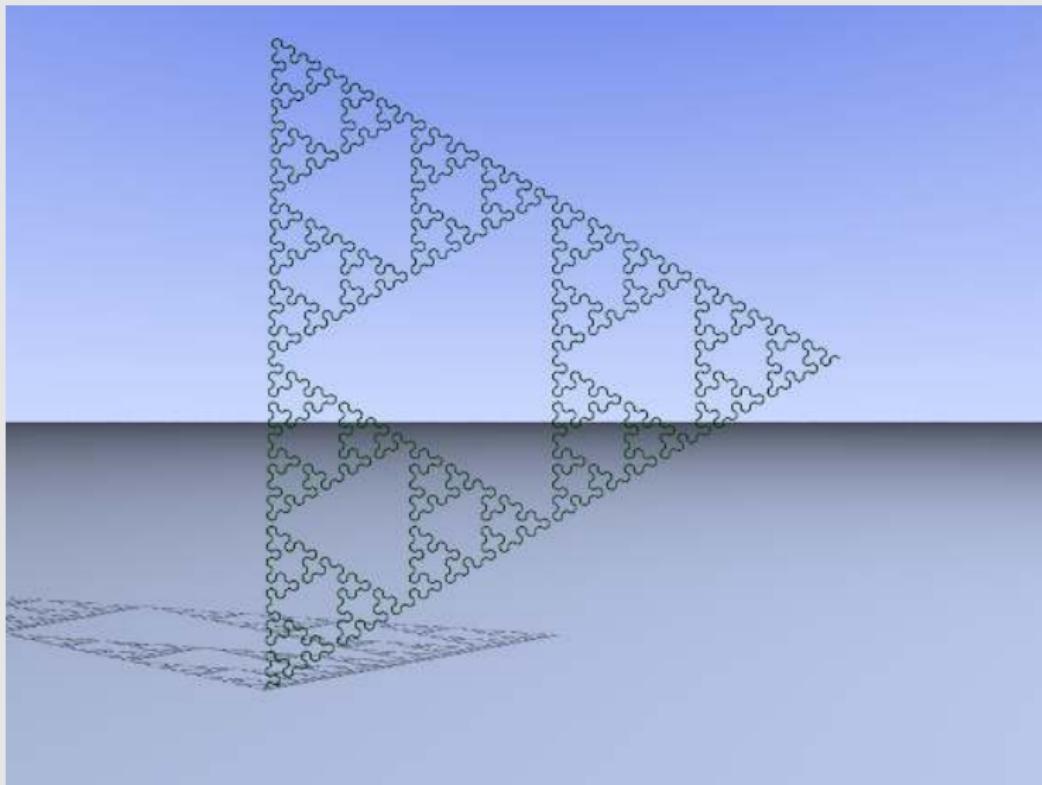
Erzeugung mit Lindenmayer-Systemen



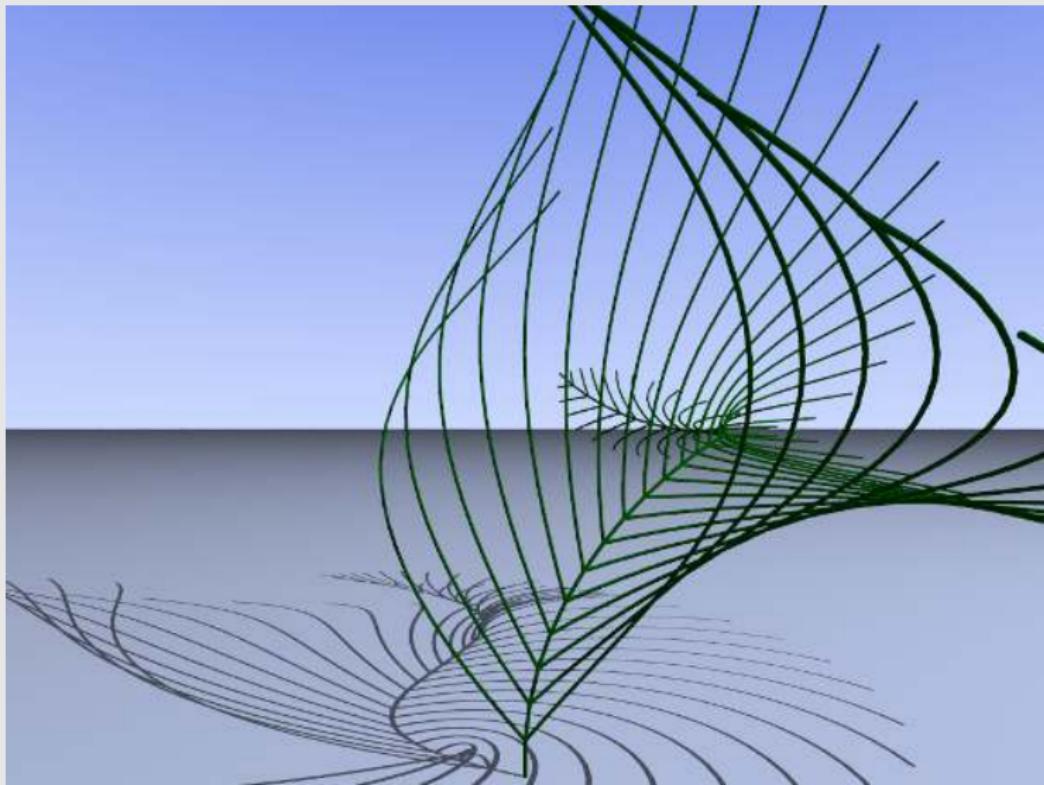
Erzeugung mit Lindenmayer-Systemen



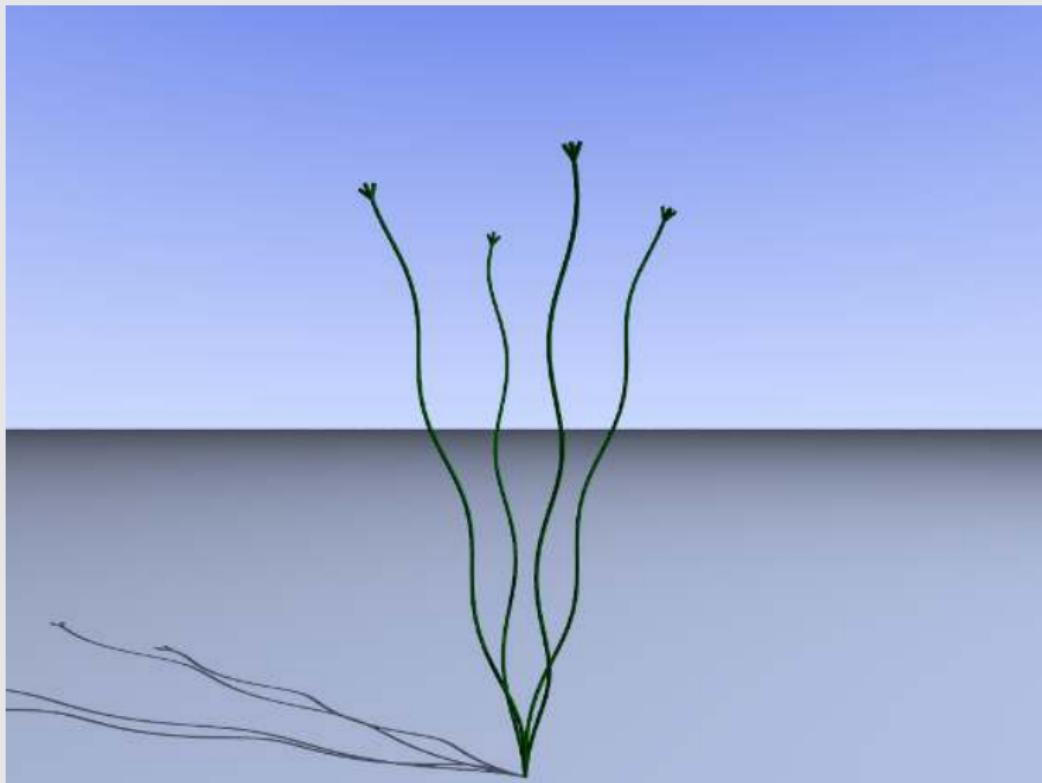
Erzeugung mit Lindenmayer-Systemen



Erzeugung mit Lindenmayer-Systemen



Erzeugung mit Lindenmayer-Systemen



Wie findet man fraktale Modelle?

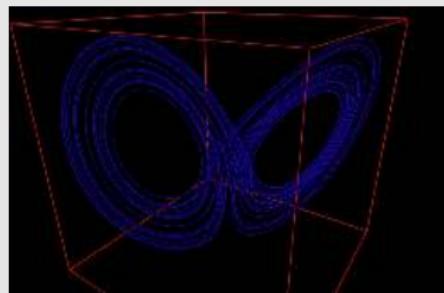
☞ Bisher:

- ▶ Versuch und Irrtum,
- ▶ zufällige Entdeckungen.

☞ Barnsley-Verfahren:

- ▶ liefert eine Folge von affinen Transformationen, aus denen vorgegebene Bilder rekonstruiert werden können.
- ▶ **Vorsicht!** Nicht alle natürlichen Objekte können gut mit Fraktalen repräsentiert werden.

Chaosche Folgen



24

Fraktale

25

Chaosche Folgen

- Topologische Dynamik
- Julia-Mengen
- Die Mandelbrot-Menge

26

Fraktale Objekte und Texturen

Topologische Dynamik

- ☞ Ein neues Teilgebiet der Mathematik, das das Zeitverhalten von Systemen beschreibt, speziell solche Systeme, die durch nicht-lineare Differentialgleichungen modelliert werden können.
- ☞ Auch sehr einfache Systeme von nicht-linearen Differentialgleichungen können unter bestimmten Randbedingungen zu nicht vorhersehbaren Lösungen führen.
- ☞ Für eng benachbarte Parameter ergeben sich völlig andere Prozesse.

Chaotische Folgen

Ein Generator

$$x_{n+1} = 4x_n(1 - x_n)$$

Chaotische Folgen

Ein Generator

$$x_{n+1} = 4x_n(1 - x_n)$$

Vom Generator erzeugte Folgen

Zwei ähnliche Startwerte $x_1 = 0.2$ und $x_1 = 0.21$:

x_1	x_2	x_3	x_4	x_5
0,2	0,64	0,922	0,289	0,596
oder				
0,21	0,664	0,892	0,364	0,962

„logistic equation“

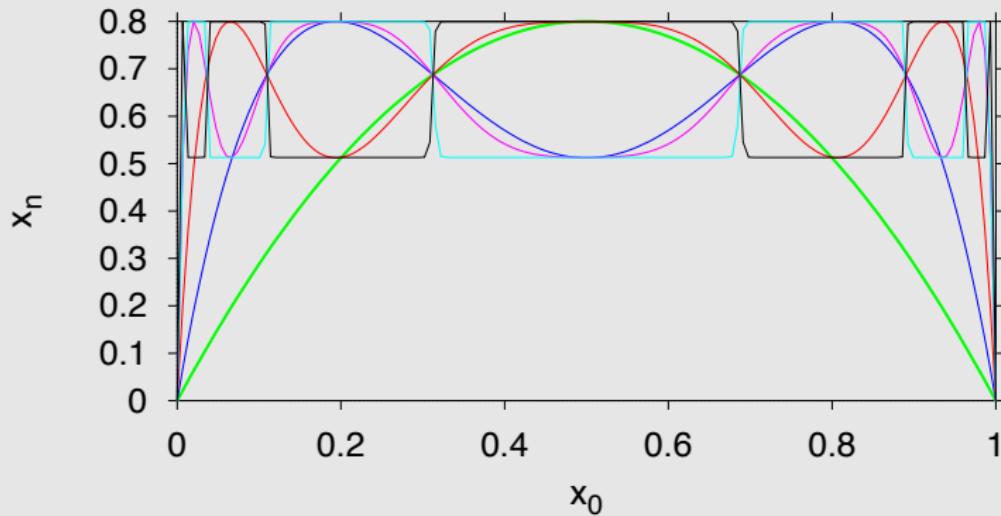
$$x_{n+1} = k \cdot x_n - k \cdot x_n^2$$

Modell für Populationswachstum in Abhängigkeit von der Bevölkerung des Vorjahres;
 k beschreibt das Verhältnis von Geburtenrate zu Sterberate.

- ▶ Für **kleine** Werte von x dominiert der **lineare** Term:
 - $k > 1.0$: Die Bevölkerungszahl wächst exponentiell an.
 - $k < 1.0$: Die Bevölkerungszahl geht gegen Null.
- ▶ Für **große** Werte von x dominiert der **nicht-lineare** Term: Er begrenzt das Wachstum auf einen endlichen Wert.

Iterationen des Generators

Iterationen des Generators $x_{n+1} = k \cdot x_n - k \cdot x_n^2$, $k=3.2$



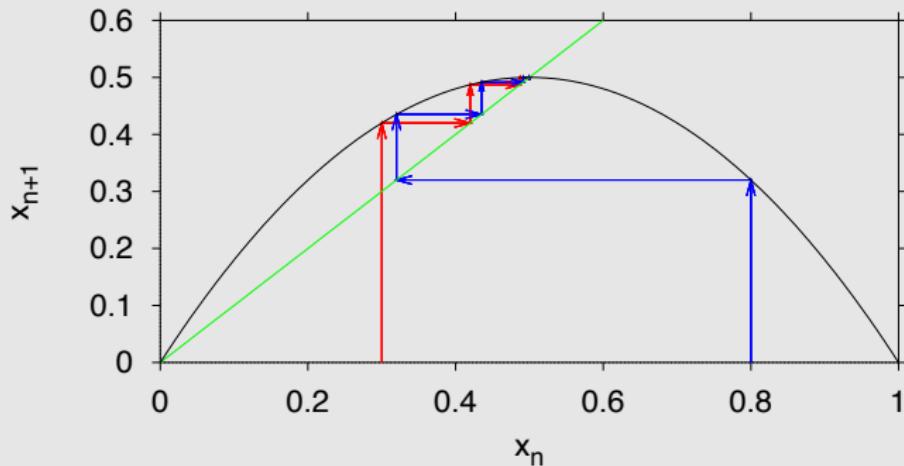
n=1 ———
n=2 ———

n=3 ———
n=4 ———

n=20 ———
n=21 ———

Iterationen des Generators für $k = 2$

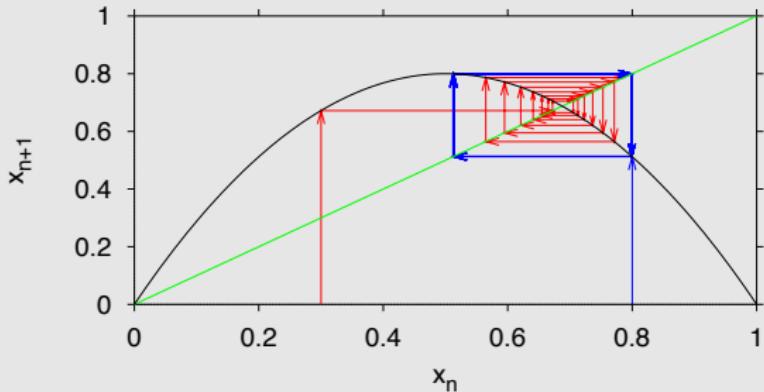
Iterationen des Generators $x_{n+1} = k \cdot x_n - k \cdot x_n^2$, $k=2.0$, f"r $x_0=0.3$ und 0.8



$$x_{n+1} = k \cdot x_n - k \cdot x_n^2$$
$$x_{n+1} = x_n \quad \text{---}$$

Iterationen des Generators für $k = 3.2$

Iterationen des Generators $x_{n+1} = k \cdot x_n - k \cdot x_n^2$, $k=3.2$, für $x_0=0.3$ und 0.8



$$x_{n+1} = k \cdot x_n - k \cdot x_n^2$$

Der Einfluß von k :

$0 \leq k \leq 3$: Die Folge x_0, x_1, \dots konvergiert gegen einen *Fixpunkt*, der nur von k , aber nicht von x_0 abhängt. Dieser Fixpunkt wird *Attraktor* genannt.

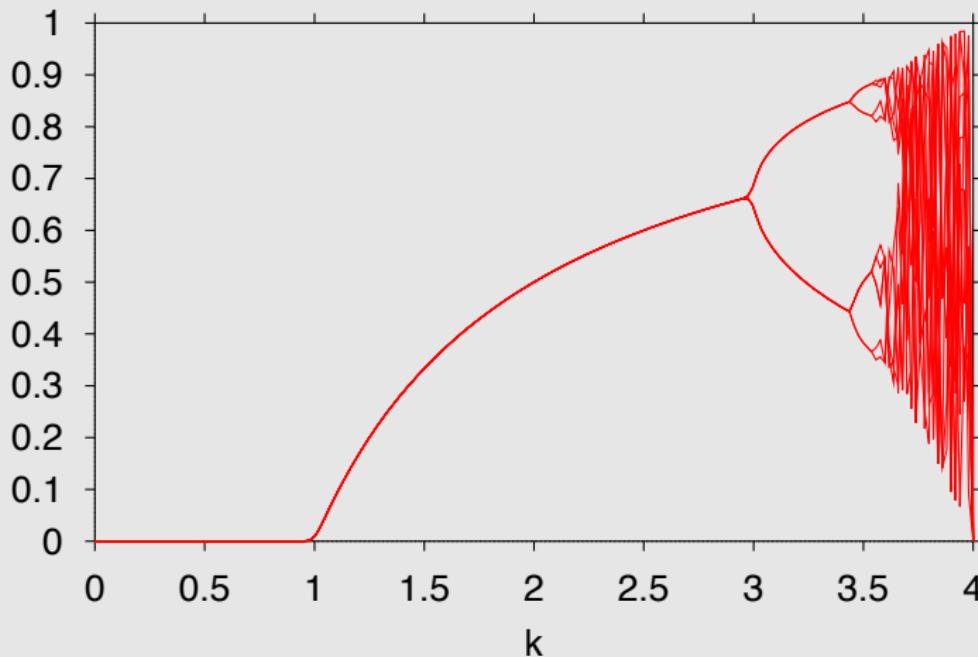
$3.2 \leq k \leq 3.3$: Die Folge hat zwei Fixpunkte, zwischen denen der Wert oszilliert.

$3.3 \leq k \leq 3.5$: Die Folge hat vier Fixpunkte, zwischen denen der Wert oszilliert.

$k = 3.68$: Das Chaos bricht aus ...

Bifurkationsdiagramm

Fixpunkte des Generators $x_{n+1} = k \cdot x_n - k \cdot x_n^2$



Folgen komplexer Zahlen

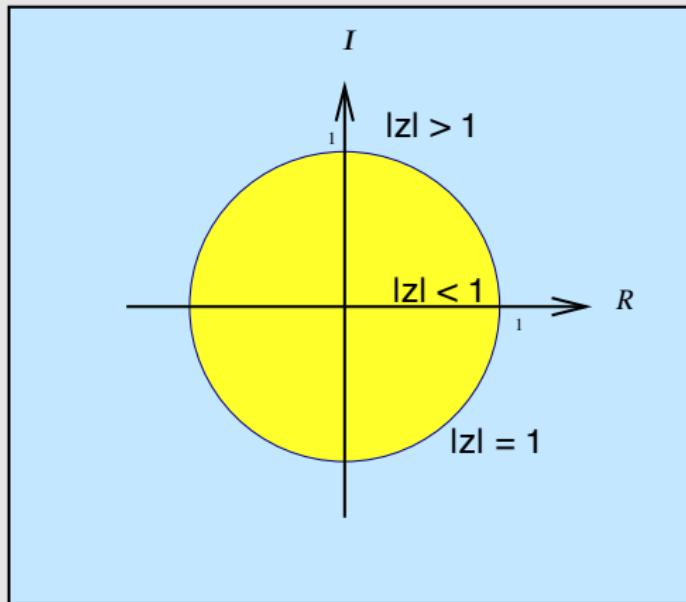
Gegeben sei eine Transformation (Generator-Funktion) f , die iterativ auf die Elemente z der komplexen Zahlenebene angewendet wird.

$$f(z) = z^2 + c \quad c, z \in \mathbb{C}, c = 0$$

Sei $c = 0$. Die Folge $z_0, f(z_0), f(f(z_0)) \dots$ ist dann
divergent für $|z_0| > 1$,
invariant gegen den Generator für $|z_0| = 1$,
konvergent für $|z_0| < 1$.

Folgen komplexer Zahlen: Beispiel für $c = 0$

$$f(z) = z^2 + c \quad c, z \in \mathbb{C}, c = 0$$



Definition (Julia-Menge)

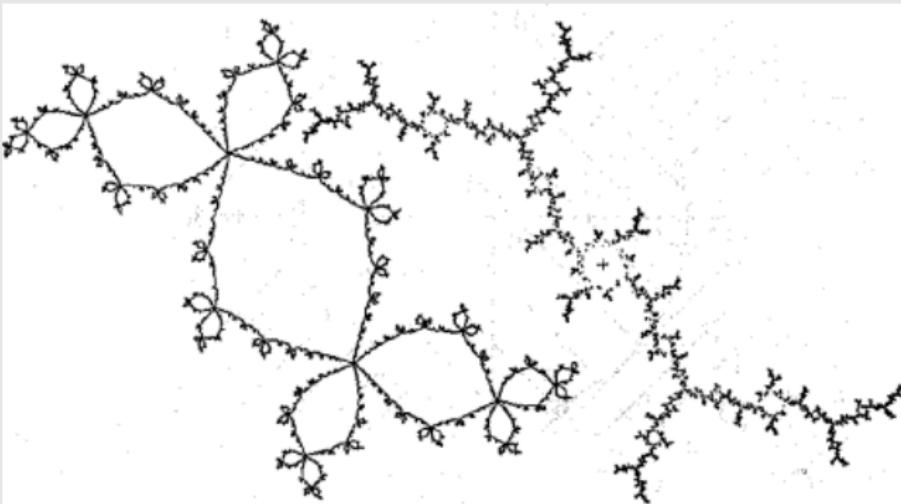
Julia-Menge: Die Menge aller $z \in \mathbb{C}$, die bezüglich des Generators $f(z) = z^2 + c \quad c, z \in \mathbb{C}$ invariant ist.

Gaston Julia, am Anfang dieses Jahrhunderts.

Julia-Kurven, auch Drachenkurven genannt, sind **zusammenhängende** Julia-Mengen.

Cantor-Mengen, auch Cantor-Staub genannt, sind **nicht zusammenhängende** Julia-Mengen.

Julia-Kurven



Julia-Kurven für $f(z) = z^2 + c$ $c, z \in \mathbb{C}$ mit

- (a) $c = -0.12 + 0.74i$ und (b) $c = -0.15652 + 1.03225i$

Julia-Kurven



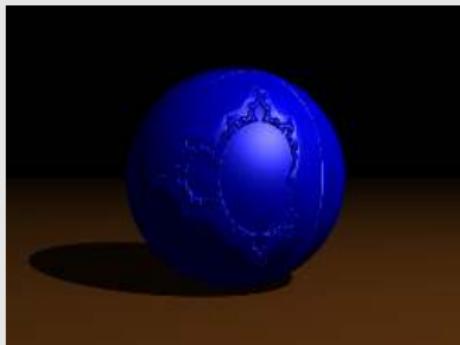
- Julia-Kurven für $f(z) = z^2 + c \quad c, z \in \mathbb{C}$ mit
- (a) $c = 0.32 + 0.043i$ und (b) $c = 0.39054 + 0.58679i$

Zur Nomenklatur

- ▶ Die Funktion $f(z) = z^2 + c$ wird als *squaring generator* bezeichnet.
- ▶ Die erzeugten Kurven werden *self-squared curves* genannt.

Die Mandelbrot-Menge

Julia-Kurven sind Fraktale und zeigen die für Fraktale typische Selbstähnlichkeit.



Problem

Wie lässt sich für einen gegebenen Wert von c vorhersagen, wie die Julia-Menge aussieht?

Lösung

Die Mandelbrotmenge

Mandelbrot-Menge

Definition (Mandelbrot-Menge)

Die Mandelbrot-Menge für die Generatorfunktion

$f(z) = z^2 + c$ ist die Menge aller Punkte $c \in C$ für die gilt:

$$\lim_{n \rightarrow \infty} |z_n| \neq \infty$$

mit $z_n = f(z_{n-1})$, $n \geq 1$ und $z_0 = 0$.

Mandelbrot-Menge

Definition (Mandelbrot-Menge)

Die Mandelbrot-Menge für die Generatorfunktion
 $f(z) = z^2 + c$ ist die Menge aller Punkte $c \in C$ für die gilt:

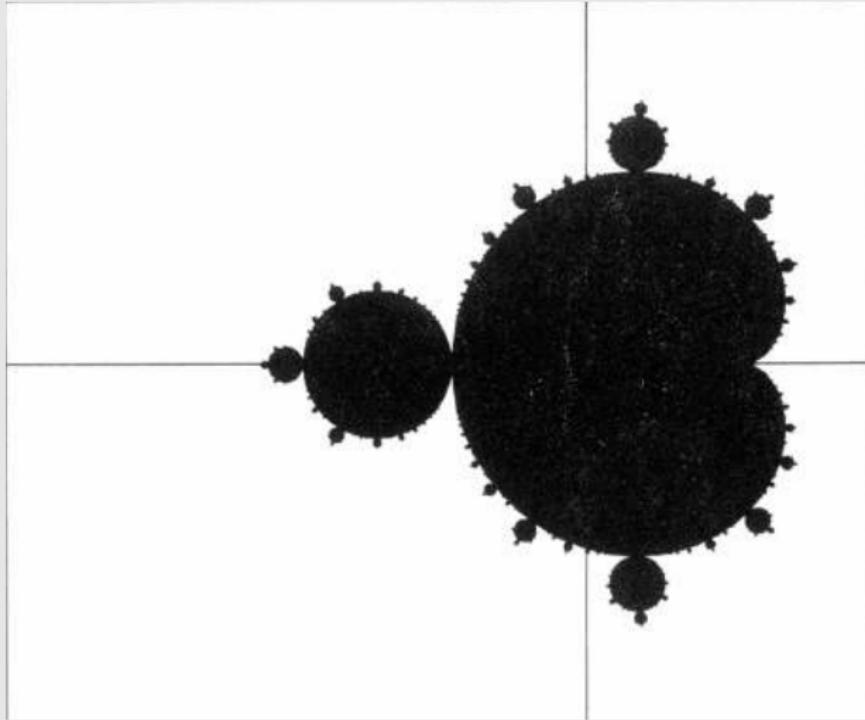
$$\lim_{n \rightarrow \infty} |z_n| \neq \infty$$

mit $z_n = f(z_{n-1})$, $n \geq 1$ und $z_0 = 0$.

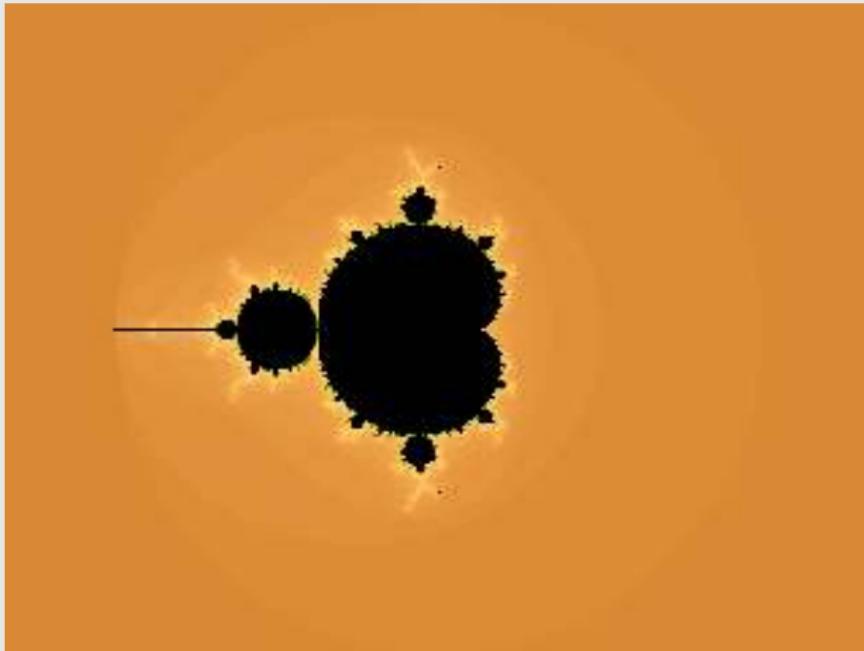
Satz

Julia-Kurven mit der Generatorfunktion $f(z) = z^2 + c$ sind genau dann zusammenhängend, wenn c in der Mandelbrot-Menge liegt.

Die Mandelbrot-Menge: Das Apfelmännchen

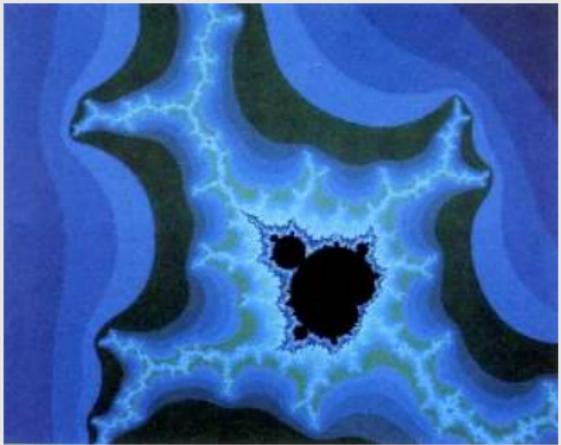
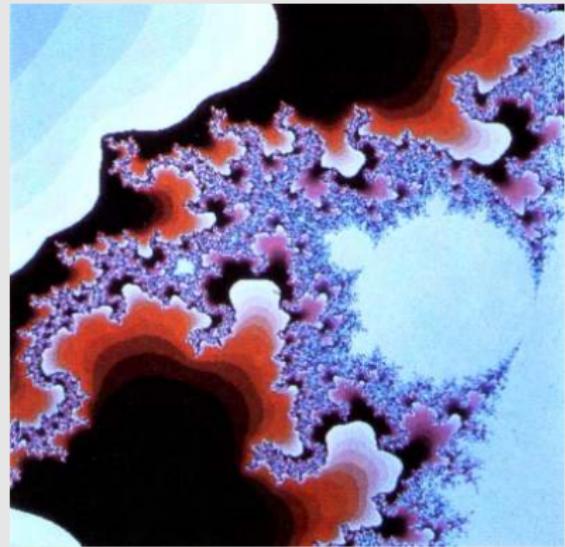


Die Mandelbrot-Menge: Das Apfelmännchen



Eingefärbt und gerechnet mit „Mandelbrot on Cocoa“

Die Mandelbrotmenge: Ausschnitte der Kontur 1



Die Mandelbrotmenge: Nachbarschaft



Berechnung der Mandelbrotmenge

Wie kann die Mandelbrot-Menge berechnet werden?

Für jeden Wert c ist zu prüfen, ob die entsprechende Julia-Menge zusammenhängend ist.

Berechnung der Mandelbrotmenge

Wie kann die Mandelbrot-Menge berechnet werden?

Für jeden Wert c ist zu prüfen, ob die entsprechende Julia-Menge zusammenhängend ist.

Satz (John Hubbard und Adrien Douady)

Die Julia-Menge mit der Generatorfunktion $f(z) = z^2 + c$ ist zusammenhängend gdw. die Folge der z_i für den Startwert $z_0 = 0$ beschränkt bleibt.

Um festzustellen, ob ein Wert c in der Mandelbrot-Menge liegt, ist es also hinreichend zu prüfen, ob die Folge für den Wert 0 beschränkt ist.

Vorgehen:

Lösungsansatz:

Für jeden Punkt der komplexen Zahlenebene ist es möglich zu prüfen, ob er ein Element der Mandelbrot-Menge ist oder nicht, indem die Folge iteriert wird und geprüft wird, ob sie begrenzt bleibt.

Vorgehen:

Lösungsansatz:

Für jeden Punkt der komplexen Zahlenebene ist es möglich zu prüfen, ob er ein Element der Mandelbrot-Menge ist oder nicht, indem die Folge iteriert wird und geprüft wird, ob sie begrenzt bleibt.

Zwei Probleme:

- ① Wie lange muß iteriert werden?
- ② Die Mandelbrot-Menge ist ein Fraktal. Sie ist **nicht bandbeschränkt** und kann daher nicht mit einer endlichen Zahl von Stichproben abgetastet werden.

Vorgehen:

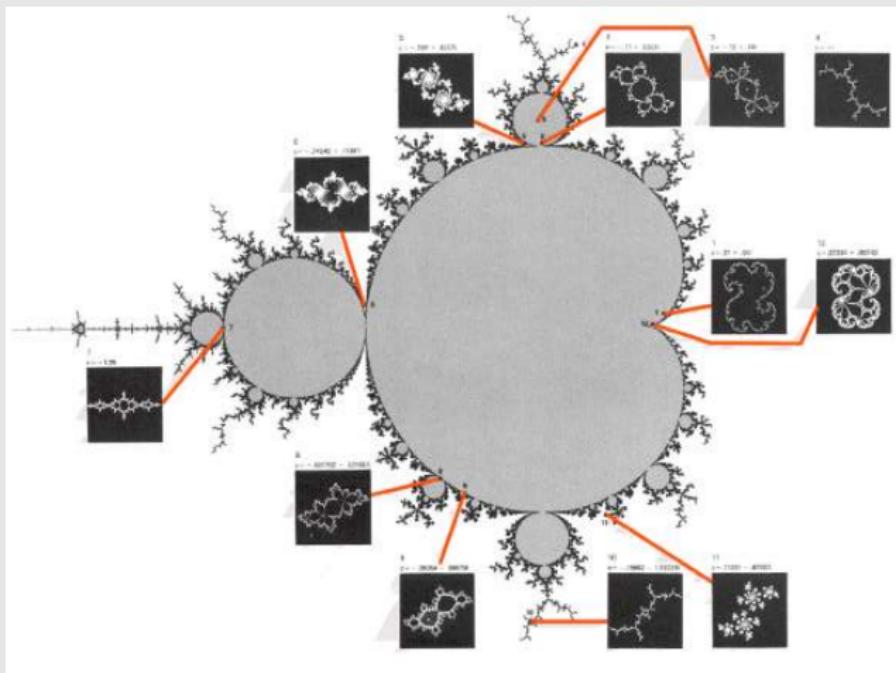
Lösungsansatz:

Für jeden Punkt der komplexen Zahlenebene ist es möglich zu prüfen, ob er ein Element der Mandelbrot-Menge ist oder nicht, indem die Folge iteriert wird und geprüft wird, ob sie begrenzt bleibt.

Zwei Probleme:

- 1 Wie lange muß iteriert werden?
 - 2 Die Mandelbrot-Menge ist ein Fraktal. Sie ist **nicht bandbeschränkt** und kann daher nicht mit einer endlichen Zahl von Stichproben abgetastet werden.
- ☞ Konsequenz: Alle Darstellungen der Mandelbrot-Menge sind **approximativ!**

Ein Wegweiser zu Juliamengen



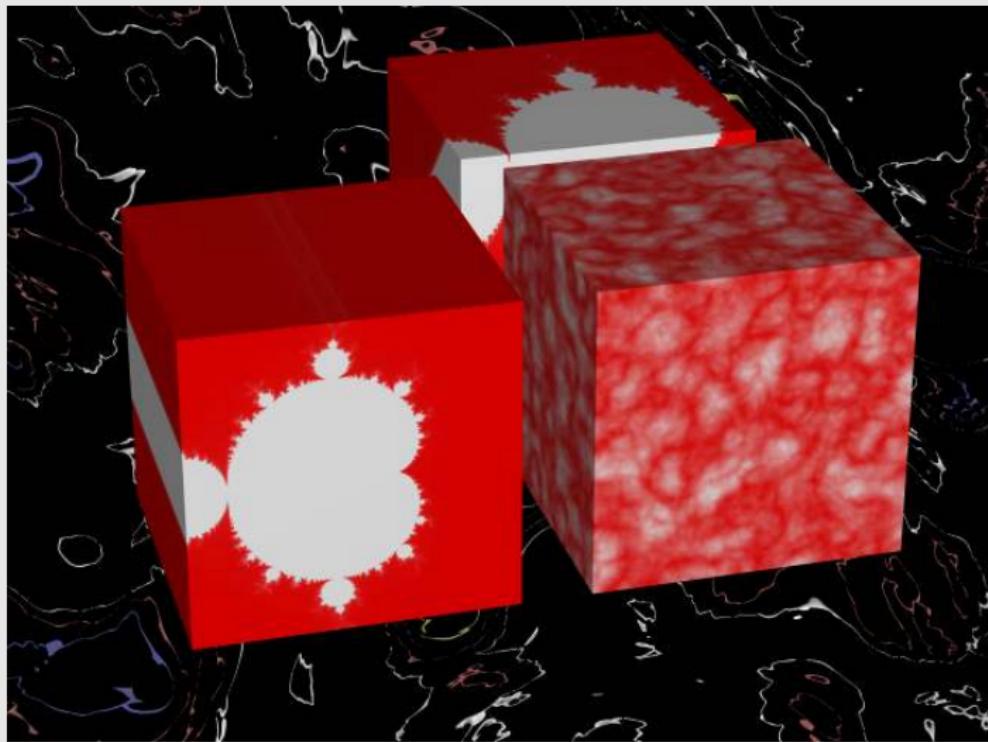
Fraktale Texturen in POV-Ray

Beispiel (Mandelbrotmenge und Granit)

```
#declare Test1 =
    texture
    { pigment // Mandelbrot-Menge
        { mandel 100 //Zahl der Iterationen
            color_map { fourcolors }}}
```

```
#declare Test2 =
    texture { // statistisch erzeugt
        pigment { granite
            color_map { fourcolors }}}
```

Das Bild zu Beispiel 65



4D-Erweiterungen der Algebra der komplexen Zahlen

- ▶ In der komplexen Zahlenebene konnten wir mittels Julia-Mengen fraktale **Linien** beschreiben.
- ▶ Eine Erweiterung der Algebra der komplexen Zahlen auf vier Dimensionen macht es möglich, fraktale **Volumen** zu modellieren. In der Computergrafik werden die beiden folgenden Erweiterungen verwendet;

Quaternionen,
hyperkomplexe Zahlen.

Beiden Erweiterungen ist gemeinsam, daß eine Zahl aus einem Realteil und drei Imaginärteilen i, j, k besteht:

Quaternionen

Definition (Quaternionen)

Ein *Quaternion* q ist eine Linearkombination aus einem reellwertigen Basisvektor 1 und drei imaginären Basisvektoren i, j, k .

$$q = q_0 + q_1 i + q_2 j + q_3 k \quad \text{mit}$$

$$i^2 = j^2 = k^2 = i \cdot j \cdot k = -1 \quad \text{und}$$

$$i \cdot j = k = -j \cdot i \quad \text{und}$$

$$j \cdot k = i = -k \cdot j \quad \text{und}$$

$$i \cdot k = -j, \quad k \cdot i = j$$

Hyperkomplexe Zahlen

Definition (Hyperkomplexe Zahlen)

Eine **hyperkomplexe Zahl** h ist ebenfalls eine Linearkombination aus einem reellwertigen Basisvektor 1 und drei imaginären Basisvektoren i, j, k mit den folgenden Rechenregeln:

$$h = h_0 + h_1 i + h_2 j + h_3 k \quad \text{mit}$$

$$i^2 = j^2 = k^2 = -1, \quad i \cdot j \cdot k = 1 \quad \text{und}$$

$$i \cdot j = k = j \cdot i \quad \text{und}$$

$$j \cdot k = -i = k \cdot j \quad \text{und}$$

$$i \cdot k = -j = k \cdot i$$

4D-Fraktale

Sei $f(h)$ eine Funktion auf den Quaternionen oder den hyperkomplexen Zahlen und sei p eine Konstante:

- ☞ Die Julia-Mengen (Fixpunkte) des Generators

$$h_{n+1} = f(h_n) + p$$

definieren die Oberfläche fraktaler Objekte im vierdimensionalen Vektorraum.

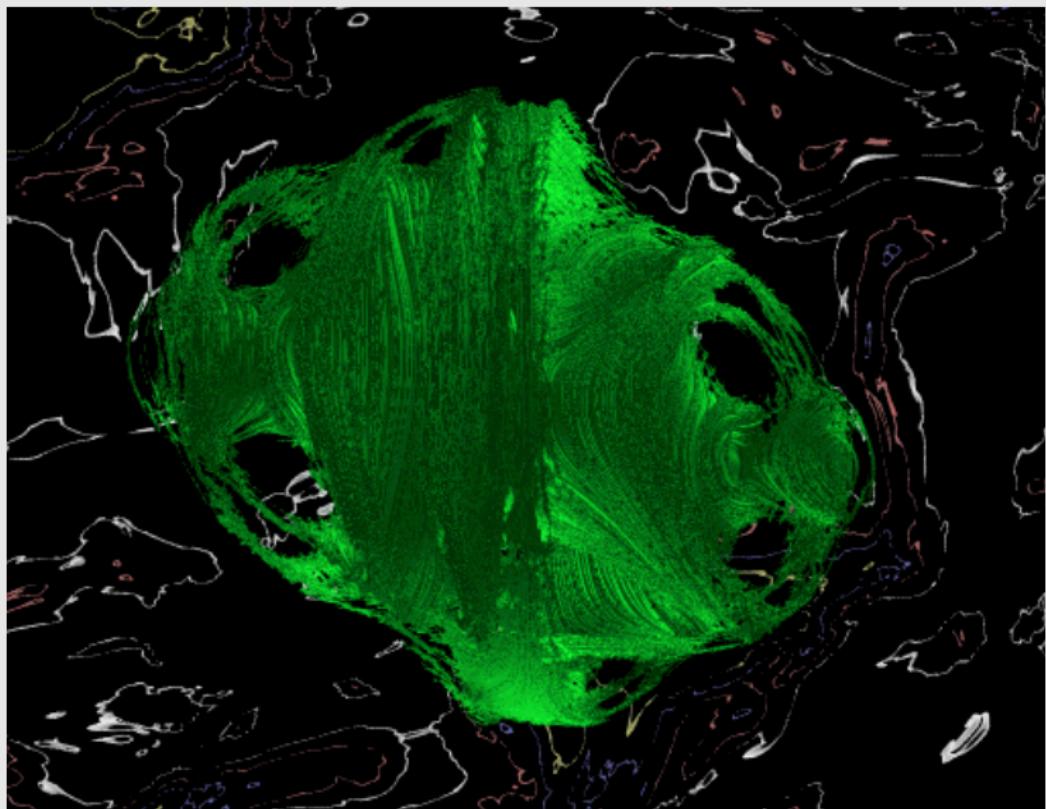
- ☞ Fraktale Objekte im dreidimensionalen Raum erhalten wir, indem wir den Durchschnitt der Objekte mit einer Hyperebene bilden.
- ☞ Die Objekte können zusammenhängend sein oder nicht.

Ein Julia-Fraktal mit Quaternionen

Beispiel (Schnitte durch ein Quaternionen:)

```
#declare color_now
= vnormalize(
    vaxis_rotate(x,x+y+z,clock*360))
julia_fractal {
    <-0.083, 0.0, -.83, -0.025>
    quaternion
    sqr
    max_iteration 20
    precision 20
    slice <0,0,0,1>, 2*clock-1
    texture { pigment { color rgb color_now }
              finish { Metal }}}
```

Das erzeugte Bild (Beisp. 68)



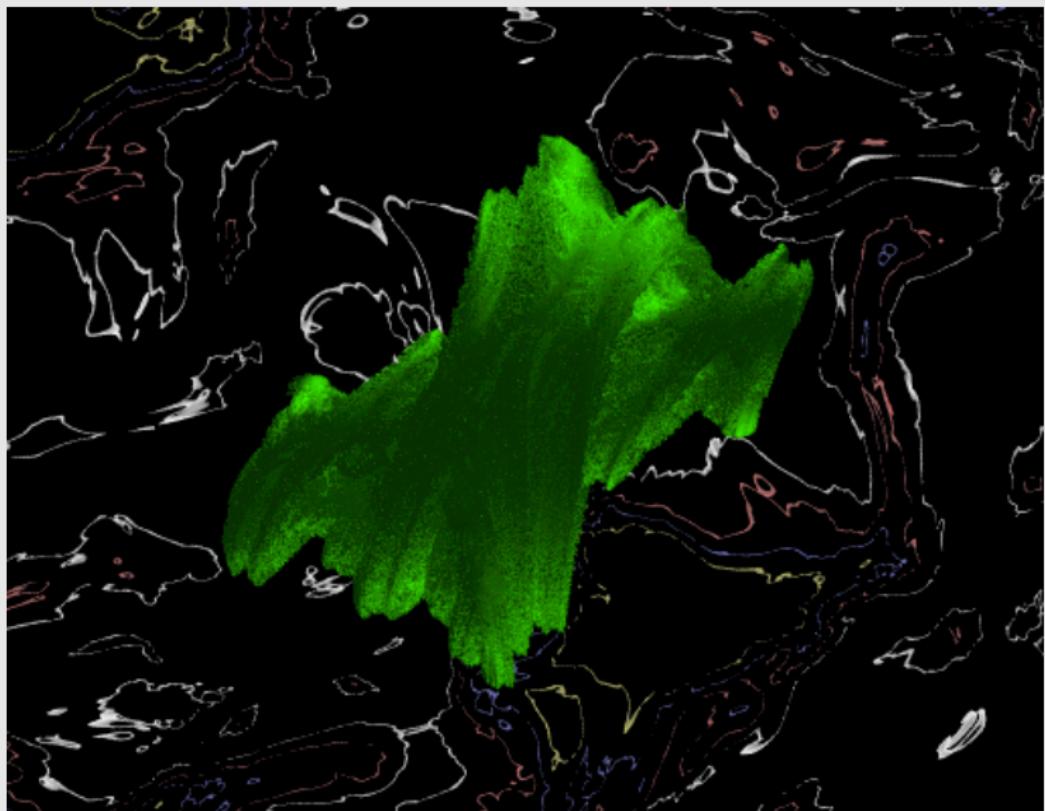
Ein Julia-Fraktal mit Quaternionen

Beispiel (Schnitte durch ein Quaternionen:)

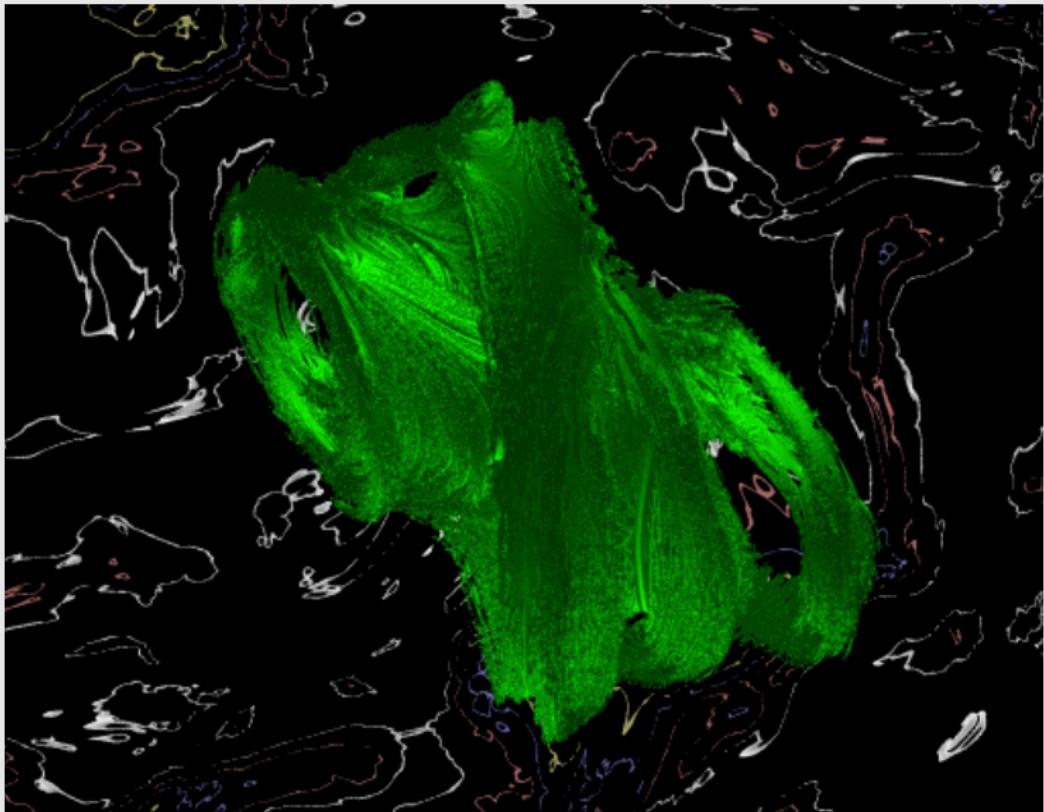
Höhere Zahl von Iterationen (30)

```
#declare color_now=
 vnormalize(vaxis_rotate(x,x+y+z,clock*360))
julia_fractal {
    <-0.39054, -0.58679, 0.025, -0.025>
    quaternion
    sqr
    max_iteration 30
    precision 30
    slice <0,0,0,1>, 2*clock-1
    texture { pigment { color rgb color_now }
              finish { Metal }}}
```

Das erzeugte Bild (Beisp. 69)



Ein weiteres Julia-Fraktal



Eine Fahrt durch den Hyperraum

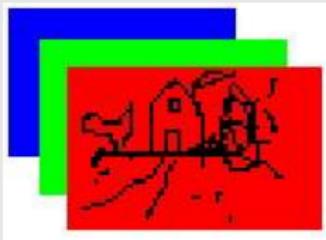
Eine Fahrt durch den Hyperraum

Fraktale intergalaktische Nebel

Teil X

Bilder und Bildfunktionen

Bildfunktionen



27

Analoge Bildfunktionen

28

Digitale Bildfunktionen

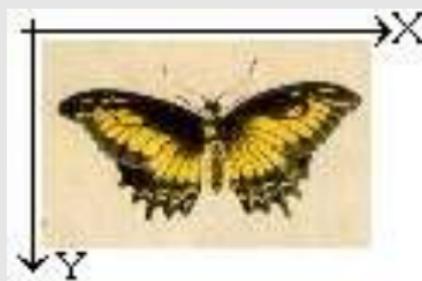
29

Repräsentation von Bildern

Analoge Bildfunktionen

Definition (Bild):

Ein Bild ist ein ebenes Objekt mit variabler Helligkeit oder Farbe.



Bildfunktion

○ Für Grautonbilder:

$$g(x, y) : \mathbb{R} \times \mathbb{R} \Rightarrow \mathbb{R}$$

○ für Farbbilder:

$$g(x, y) : \mathbb{R} \times \mathbb{R} \Rightarrow \mathbb{R} \times \mathbb{R} \times \mathbb{R}$$

Bildfunktion: photometrische Interpretation

Additiv: Helligkeit

Die Werte der Funktion sind

hoch für helle Bildpunkte und

niedrig für dunkle Bildpunkte.

Diese Interpretation entspricht dem additiven Farbmodell für die Darstellung auf einem Bildschirm.

Bildfunktion: photometrische Interpretation

Additiv: Helligkeit

Die Werte der Funktion sind

hoch für helle Bildpunkte und

niedrig für dunkle Bildpunkte.

Diese Interpretation entspricht dem additiven Farbmodell für die Darstellung auf einem Bildschirm.

Subtraktiv: Schwärzung (Grauwert)

Die Werte der Funktion sind

hoch für dunkle Bildpunkte und

niedrig für helle Bildpunkte.

Diese Interpretation entspricht dem subtraktiven Farbmodell für den Druck auf Papier.

Repräsentation durch mehrere Funktionen

Für Farbbilder



Häufig wie 3 Einzelfunktionen
betrachtet...

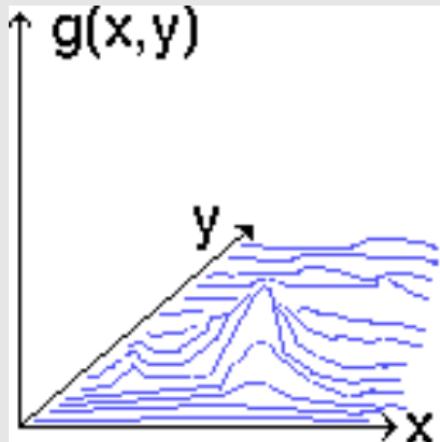
$$\begin{pmatrix} g_{\text{rot}}(x, y) \\ g_{\text{grün}}(x, y) \\ g_{\text{blau}}(x, y) \end{pmatrix} \quad \text{oder} \quad \begin{pmatrix} g_{\text{Intensität}}(x, y) \\ g_{\text{Farbton}}(x, y) \\ g_{\text{Sättigung}}(x, y) \end{pmatrix}$$

Wenn nicht anders angegeben, wird von
Intensitätsbildern ausgegangen.

RGB-Auszüge eines Farbbildes

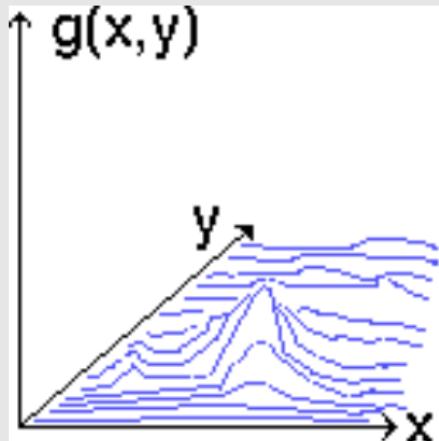


Grauwertgebirge



Anschauliche Vorstellung:
Grauwertgebirge —

Grauwertgebirge



Anschauliche Vorstellung:
Grauwertgebirge —

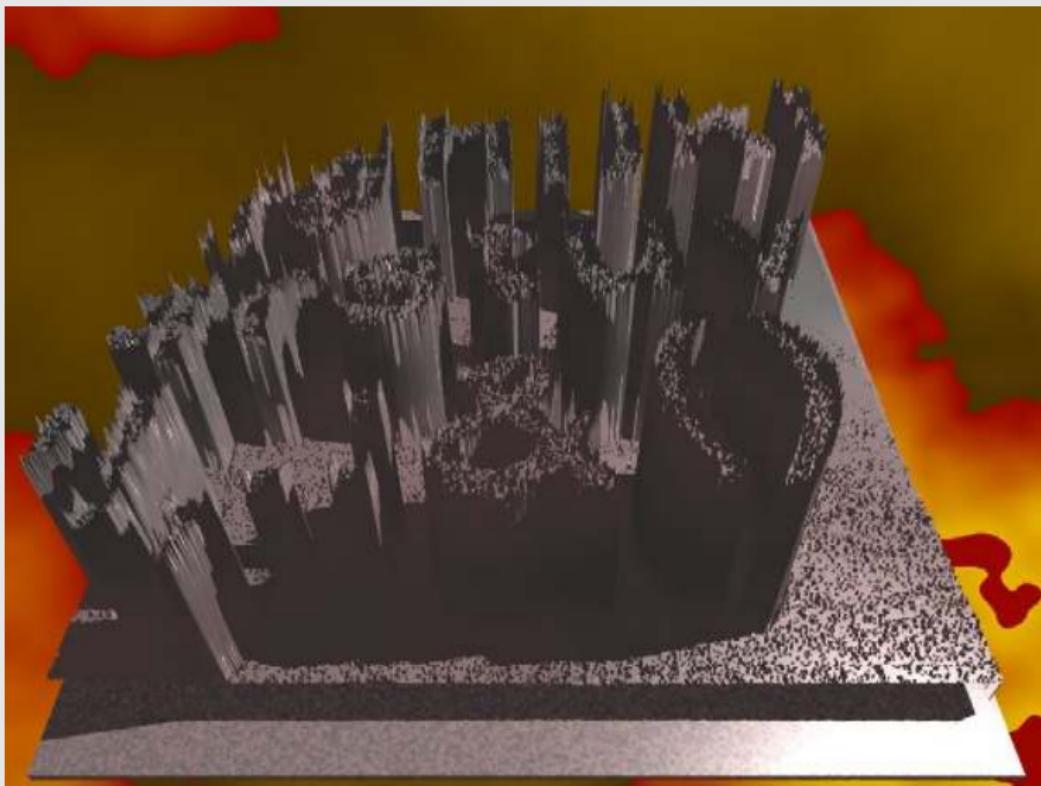
Annahme:

- ▶ $g(x, y)$ ist beschränkt

$$0 \leq g(x, y) \leq g_{\max}$$

- ▶ (und überhaupt mathematisch problemlos).

Das Grauwertgebirge für ein einfaches Bild



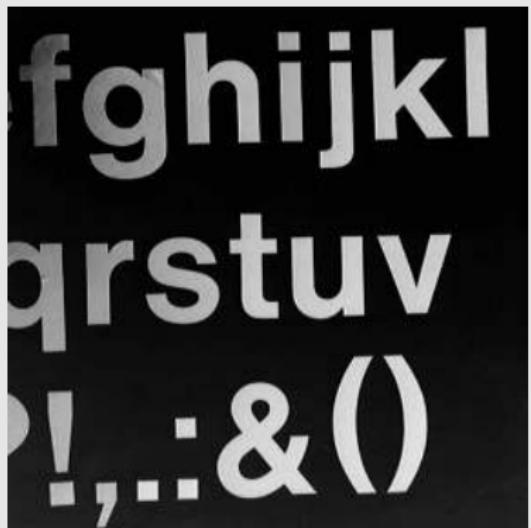
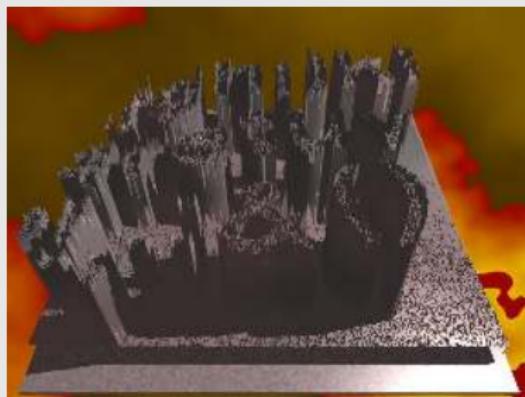
Das Bild zum Grauwertgebirge

fghijkl

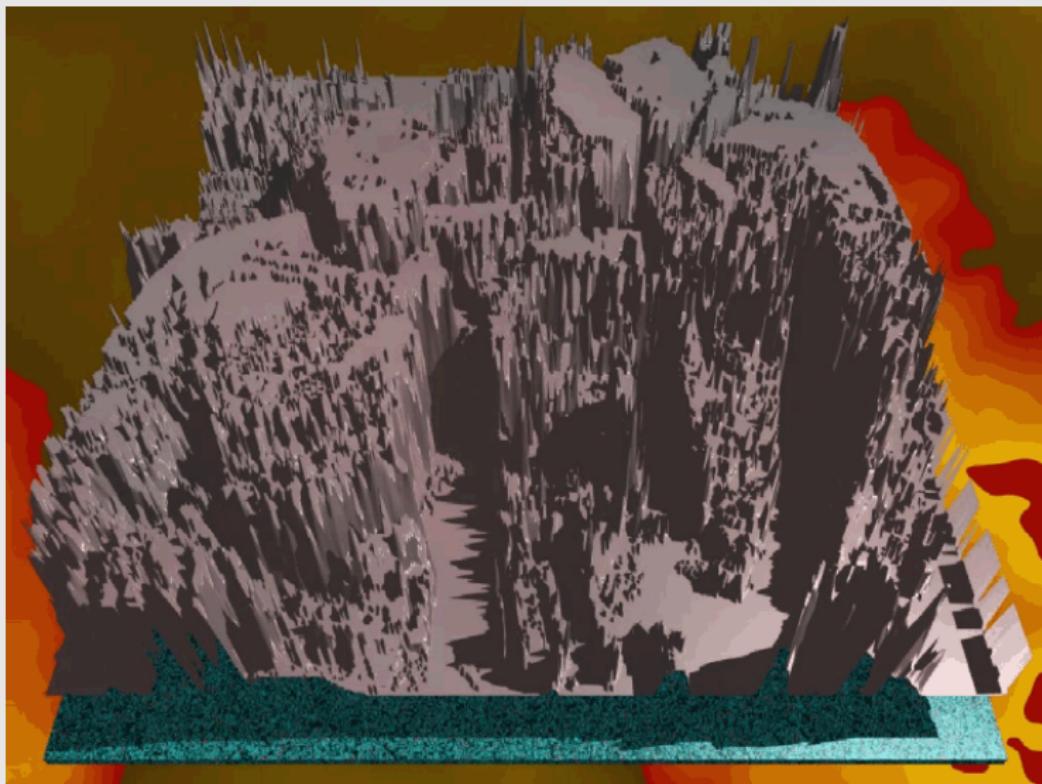
qrstuvwxyz

! . & ()

Vergleich von Bild und Grauwertgebirge



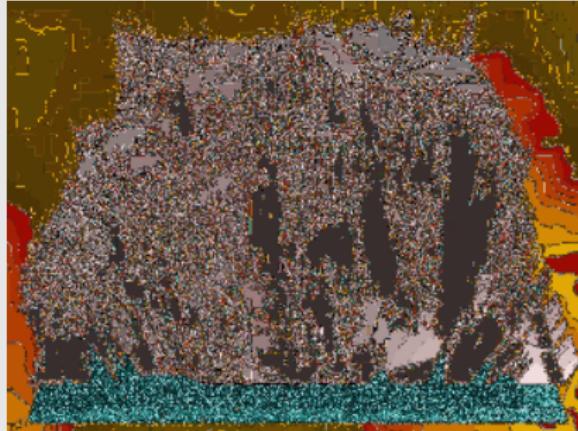
Die Bildfunktion eines komplexen Grautonbildes



Das Bild zum Grauwertgebirge



Vergleich von Bild und Grauwertgebirge



Multispektralbilder

Definition (**Multispektralbild**)

Die Bildfunktion eines Multispektralbildes bildet die Punkte der Bildebene auf einen Vektor von spektralen Messungen in mehreren spektralen Kanälen des elektromagnetischen Spektrums ab.

$$g(x, y) : X \times Y \Rightarrow \Re^C$$

Eigenschaften von Multispektralbildern

- ▶ Die Kanäle werden meist so gewählt, daß eine typische **spektrale Signatur** gemessen werden kann.
- ▶ Neben dem Bereich des sichtbaren Lichtes werden häufig auch Kanäle im nahen und thermischen Infrarot verwendet.
- ▶ Multispektralbilder werden insbesondere in der Fernerkundung eingesetzt.

Ein Beispiel für ein Multispektralbild

Flugzeug-Multispektralscanner, San Pablo Bay, CA

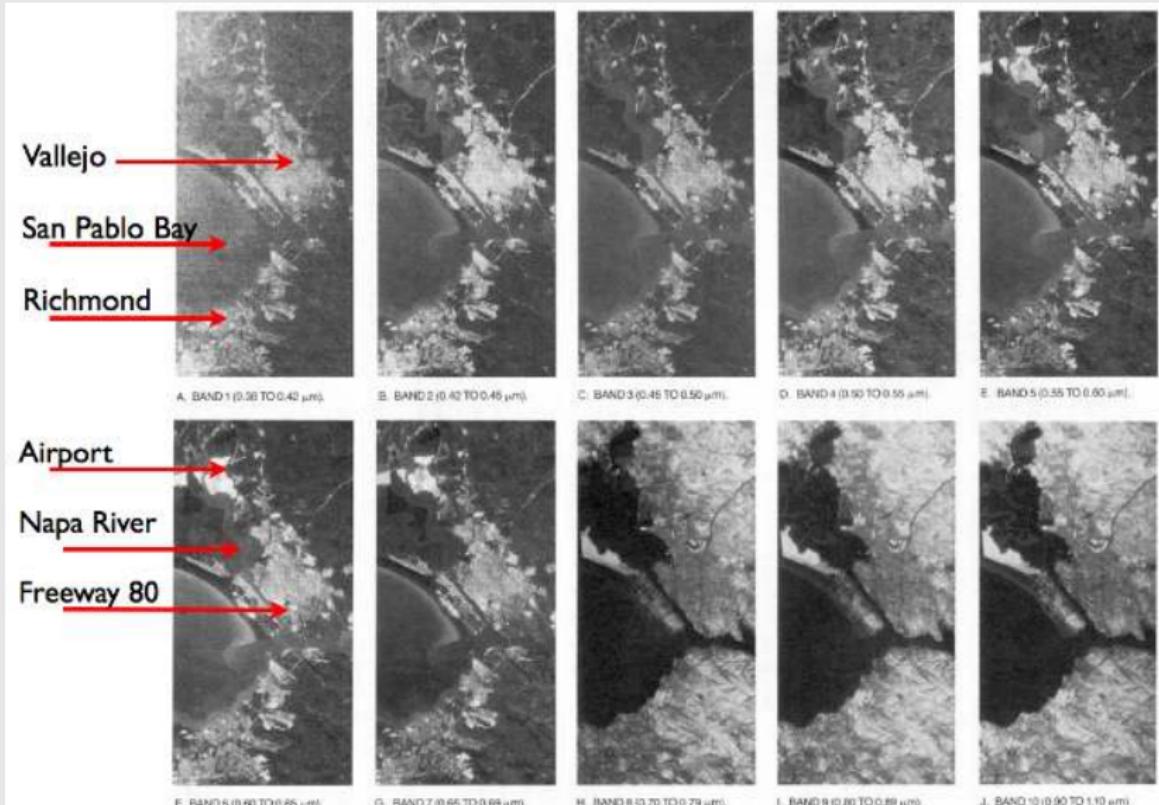
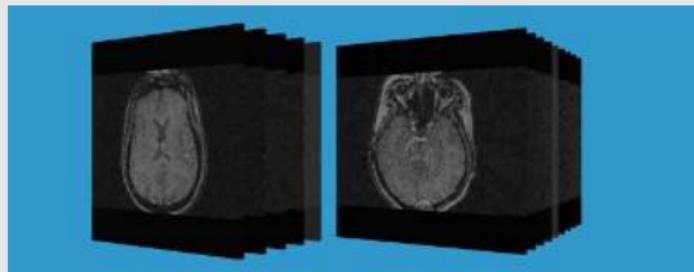


FIGURE 2.30 Aircraft multispectral scanner images of San Pablo Bay area, California, acquired March 29, 1990. Each image covers an area of 15 by 30 km. Courtesy NASA Ames Research Center.

Volumenbilder



Ein MRT

Definition (**Volumenbilder**)

- ▶ Der Definitionsbereich eines Volumenbildes ist keine Bildebene sondern ein dreidimensionales Bildvolumen.
- ▶ Der Wertebereich sind typischerweise skalare Meßwerte, die für den Anwendungsbereich charakteristisch sind.
- ▶ Volumenbilder werden häufig in der medizinischen Diagnostik eingesetzt (MRT, PET).

Beispiel: MRT-Volumenbild des Gehirns

Movie: MRT-Volumenbild des Gehirns

Die Schichten: MRT-Volumenbild des Gehirns

Blättern durch die Schichten

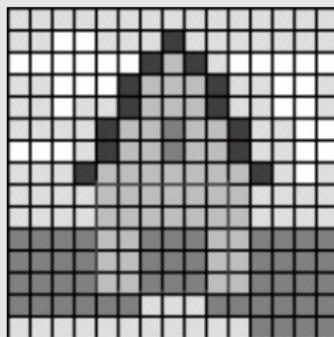
Der Ortsraum

Definition (Ortsraum)

Die Bildebene $X \times Y$ einer zweidimensionalen Bildes oder das Volumen $X \times Y \times Z$ eines Volumenbildes heißen der **Ortsraum**.

- ▶ Die Bildfunktionen $g(x, y)$ und $g(x, y, z)$ sind im Ortsraum definiert: Sie beschreiben den räumlichen Aufbau eines Bildes aus Punkten unterschiedlicher photometrischer Eigenschaften.
- ▶ Im Gegensatz dazu steht der **Frequenzraum**: Die Bildfunktionen werden durch ihr Fourier-Spektrum beschrieben.

Digitale Bildfunktionen



27

Analoge Bildfunktionen

28

Digitale Bildfunktionen

- Diskretisierung
- Rasterung
- Quantelung

29

Repräsentation von Bildern

Diskretisierung

Zur Verarbeitung im Digitalrechner müssen analoge Bilder diskretisiert werden:

Rasterung: Diskretisieren im **Definitionsbereich**.

Quantelung: Diskretisieren im **Wertebereich**.

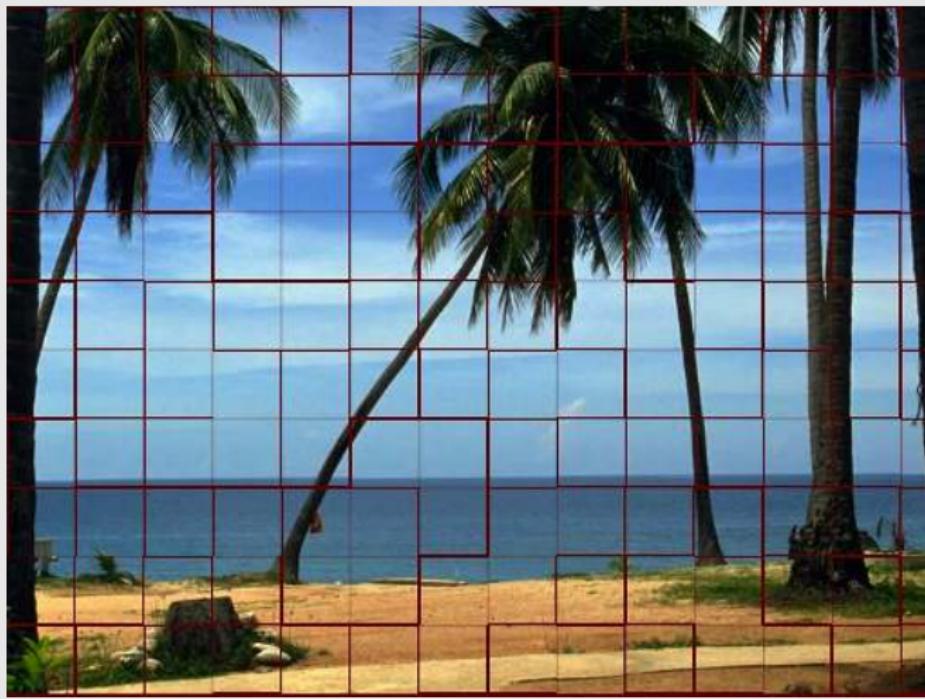
Terminologie: Diskretisierung

Rasterung: Auswahl von Rasterstellen, für die Funktionswerte definiert werden sollen.

Abtasten: Bestimmen der (analogen) Bildfunktionswerte an den Abtaststellen.

Quantelung: Bestimmen von diskreten Repräsentanten der analogen Abtastwerte.

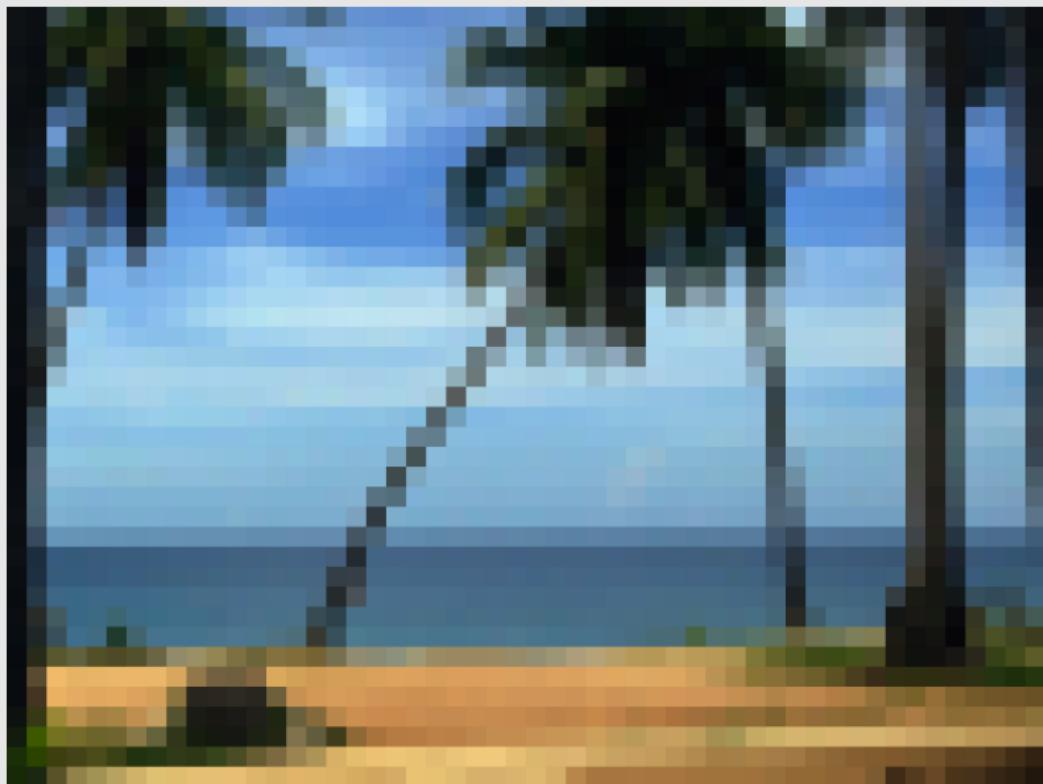
Beispiel: Rasterung eines Bildes



Grobes Raster und quantisierte Farbwerte



Feines Raster und quantisierte Farbwerte



Echtzeit-Rasterung (pixellate):

in Quartz-Composer

Movie: Mt.Tongariro, in unterschiedlicher Rasterung

Echtzeit-Rasterung (pointilize):

in Quartz-Composer

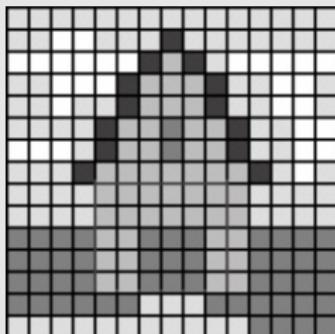
Mt. Tongariro

Ein Mosaik aus Ravenna



Der wundersame Fischzug,
Sant' Apollinare Nuovo, 6. Jahrhundert nach Chr.

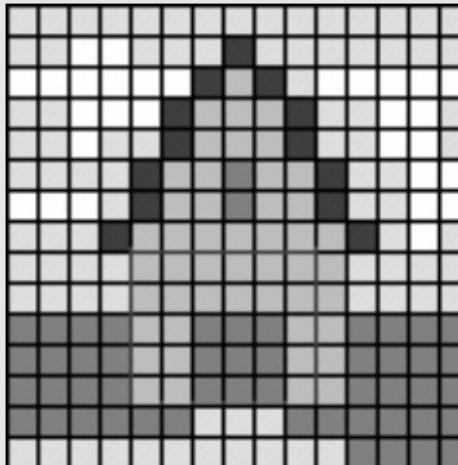
Matrix-Rasterung



$$g_{ij} : \begin{aligned} i &= 0 \dots I - 1 \\ j &= 0 \dots J - 1 \\ I &\quad \text{Zeilen} \\ J &\quad \text{Spalten} \end{aligned}$$

Das Raster muß nicht notwendigerweise **quadratisch** sein, s. TV-Bilder.

Pixel



Definition (**Pixel:**)

Die Elemente der Bildmatrix heißen **Pixel:** (picture elements).

Hexagonal-Rasterung



Hexagonal-Rasterung:

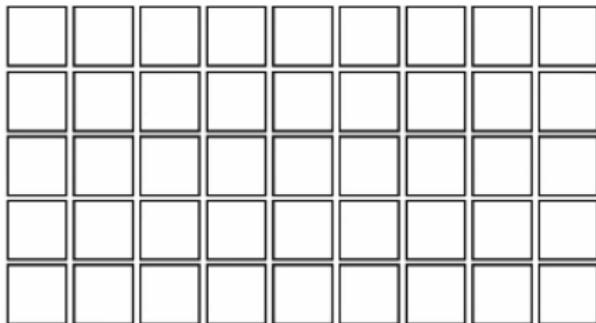
Die Pixel sind (gleichseitige) Sechsecke.

Vorteile:

- ☞ Die Pixelfläche ist kreisähnlicher.
- ☞ Keine Zusammenhangsprobleme (siehe nächste Folien).

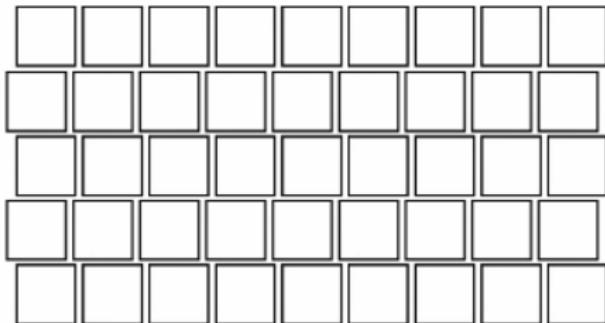
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



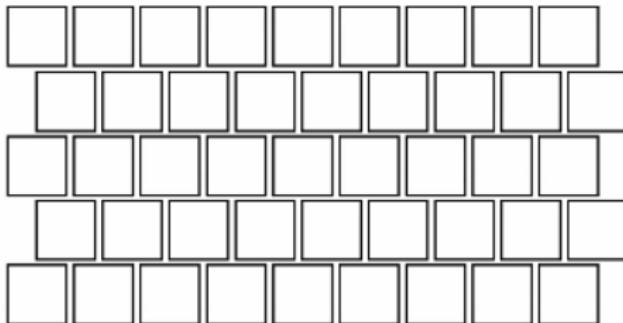
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



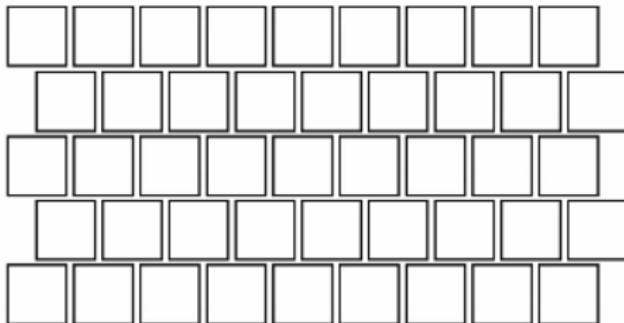
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



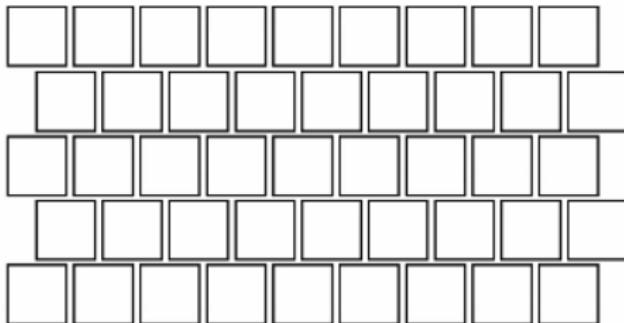
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



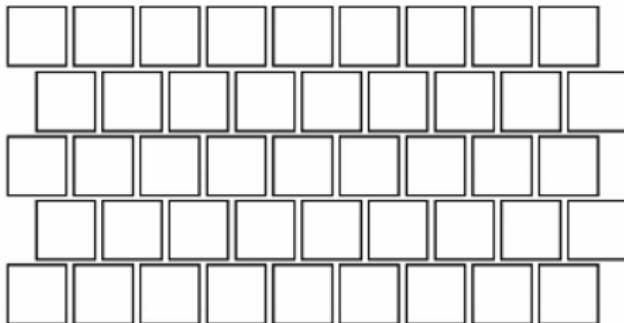
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



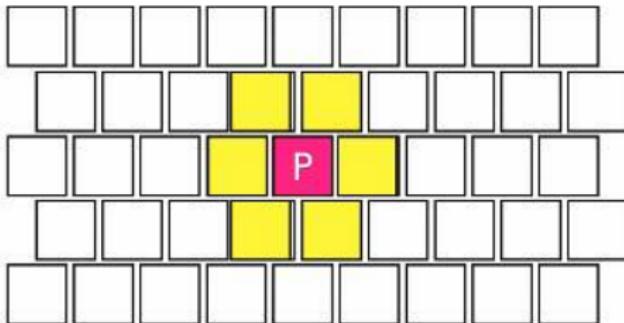
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



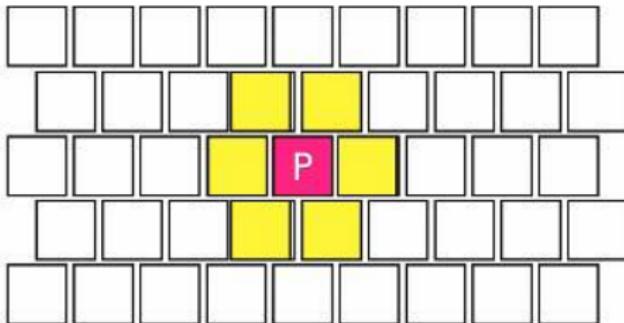
Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.



Vom Rechteckraster zum Hexagonalraster

- Jede zweite Zeile wird um ein halbes Pixel verschoben.

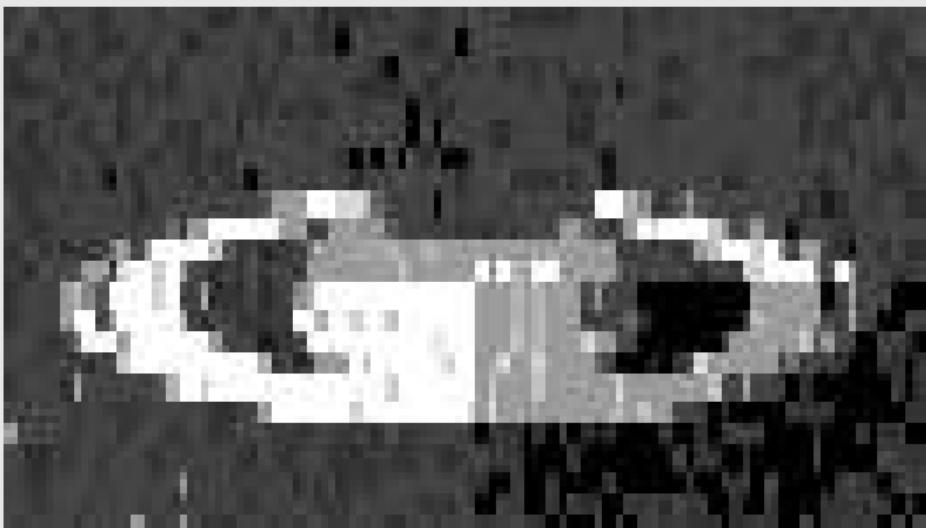


Die Pixel eines Grauwertbildes in Matrix-Rasterung

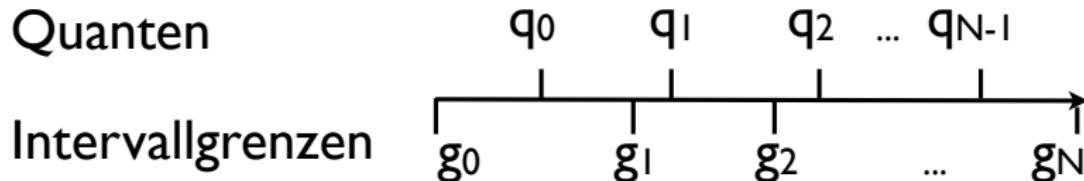
Jedes Druckzeichen steht für einen bestimmten Grauwert.

```
msvjirsndssssksqsjffssesssoihghpmosrihspssssgmfdlfsissgsjksissssrmsdoorjsljsnqps  
sscsjhjnsejkslssddsmssnssfsem .ssssseegs .ssssssssssssssssssssssssdfdssessjghsu  
jdsssislsssccsissssssgsdsssssssf .ssssfsdssemkjscssssssssssssqskgijsssjcsscsfg  
gckgrsrsrhssesdlgsssssjgssscsssdgdfschsssssskscfds#lsslsrsssssssekjssghhesj  
nsssdjossssgnsssshsemhsjhscssshsss .sffsssssssssshdssssscdssssdssssihsfkcsscss  
olcskskhjssdsssdscssssssssss .s .ssssfcsssscdssssgsssssssssssssfjhhssssss  
iselsssssesssogjrihsgissssssssc .s .s .scs ..ssssspsssshbssssssssssssssssssdsgscgs  
melhkscsf .shsssskgses .ssssssssesrsxssssscssvhgfhrsssdesssssdfssssdfdn  
scIssosssssshessssssswHHHVVTUNAsoss .slssssssssssssVVVKFqlyHvssssssssdaesshssde  
ssgkssssssnssCcHHVVVVVHssHHHEHypseseikdnsnssHHAsssHVSVVVRHsxHwsasssigssssss  
gssksenjsDHsHVVVVVPxjscgksEHHHHNHLHLMHHHHHHHHJHHHHHlqssjsHQVVSVHsHHsc#ssisfe  
kffuousHHnTVSVHtsfsuissppHHHHHHHHRHRRHHHHVHVMHVVHHHHHzsssscssnRSVVVVHsVHssshgs  
gssssHPExVVVUVsslHechsedsGKVVVVVVVVVVVHMHMLJHsHsss .....HHHHHH .sHssr ...  
hssspHVsHVVVVOnisskfisIVVLVVRVVQVVVVVVHHHLHOHHHsssszd .....sMHHHHHssHss .sp  
sssmgqCVVHVVVVVVHsssaftsWVVVVVVVSVVVHHHQOHHHSHHst .....gHHHHHHQsHss .pog.  
rsqospksRVVVVVVVVLfschHHHVQVVVVVUVHHTHHSHHHHBs .c .hsUHHHHHHsCssli .je .g  
jssssgHssssspvNVSVVVVVVVVSVVVVVVHQHHHPHHHHHHHHTHHHHHHHPssHssh .c .....i .c  
sqsspsqsnmjhssrrshessHVVVVVVSVVVVVVUHTHHHHHHHHHQPHHsro .f .....idid .....gg  
HoSSsqjsssisusflssgnsssesgisskokguhjnrssh .lih .h.q.d.c ....j ..c .Inlm ....j ..h .ck ..  
sqedskssmssslgsIsjssgfsscsssciemkmjgsmelssikj ....j .....s .gd ..e ..n ..smgh  
ssdmssstqpssrnssHospdsnssdlscofmsqssssdd .m ..i ..i ..e ..hdk .s ..jsie .kns  
ssllsssnsosssnsprpsqrnhssppsshsjssqsssiqfsq ..nl ..k .ned ..kilcsgm ..f ..f ..gngmi .n ..  
sjsssshssssHspshsjlgssscsjtsilsosfdrskrssjioslf .f ..s .p .nsipief .edmh ..s .immkr .s .s
```

Das Bild in Grauwertdarstellung



Quantelung: Diskretisieren der Grauwerte



Uniforme Quantelung: Am häufigsten verwendet; vorzugsweise Quantenzahl $N = 2^K$, z.B. $K = 8, N = 256$.

Optimale Quantelung: Statistisches Modell der Grauwertverteilung ist bekannt.

Modellbasiert: Die Kennlinie des Bildwandlers ist bekannt.

Optimale Quantisierung

Annahme: Die Grenzen g_0, g_N , die Wahrscheinlichkeitsdichte $p(g)$ und Zahl der Intervalle N seien bekannt.

Ziel: Minimiere den mittleren quadratischen **Quantisierungsfehler** δ_Q durch geeignete Wahl der Entscheidungsgrenzen g_1 bis g_{N-1} und der Repräsentanten q_0 bis q_{N-1} .

$$\delta_Q^2 = \sum_{n=0}^{N-1} \int_{g_n}^{g_{n+1}} (g - q_n)^2 p(g) dg$$

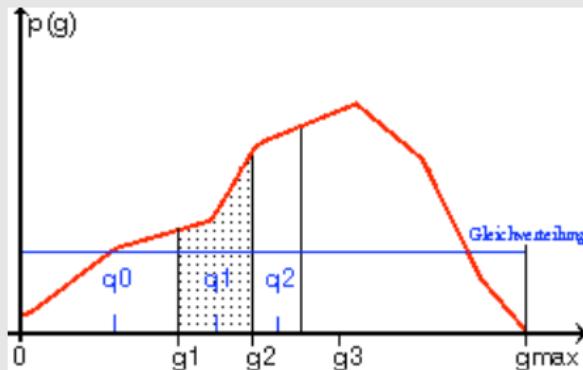
Lösung: Partielle Ableitungen nach den Unbekannten g_n und q_n bestimmen, Nullstellen berechnen:

Partielle Ableitungen nach g_n und q_n , Nullstellen:

$$\frac{\partial}{\partial g_n} \delta_Q^2 = (g_n - q_{n-1})^2 p(g_n) - (g_n - q_n)^2 p(g_n) = 0$$
$$n = 1 \dots N - 1$$

$$\frac{\partial}{\partial q_n} \delta_Q^2 = -2 \int_{g_n}^{g_{n+1}} (g - q_n) p(g) dg = 0 \quad n = 0 \dots N - 1$$

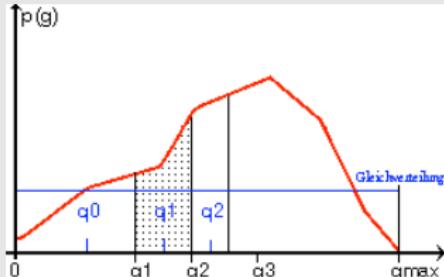
Man beachte, daß nicht nach g_0 und g_N differenziert wird, da die Grenzen des Wertebereichs von g als Randwerte festliegen. q_0 und q_{N-1} dagegen sind nicht a priori festgelegt und müssen aus den angrenzenden Intervallgrenzen ermittelt werden.



$$\Rightarrow g_n = \frac{1}{2}(q_{n-1} + q_n), \quad p(g_n) > 0, n = 1 \dots N-1 \quad (2)$$

$$\Rightarrow q_n = \frac{\int_{g_n}^{g_{n+1}} g p(g) dg}{\int_{g_n}^{g_{n+1}} p(g) dg} \quad n = 0 \dots N-1 \quad (3)$$

Interpretation der Lösung



Anschaulich bedeuten die Gleichungen folgendes:

- 1 Jede **Intervallgrenze** muß genau in der Mitte zwischen den **Quantenwerten** liegen.
- 2 Jeder **Quantenwert** ist genau die **Flächenschwerpunktskoordinate**.

Lösung

für $p(g) = \text{const}$: *Gleichmäßige Quantelung.*

Numerische Lösungsprozedur

- 1 Wählle q_0 , $n = 0$
- 2 $n := n + 1$, berechne g_n mit (2)
- 3 Berechne q_n mit (1)
- 4 Falls $q_n > g_N$ weiter bei 7
- 5 Falls $n < N - 1$ fahre fort bei 2
- 6 Falls q_{N-1} (2) befriedigt, ENDE
- 7 Wähle neues q_0 , $n = 0$ und fange bei 2 von vorne an.

Anwendungen: Umkodierung, Requantisierung

- ☞ Häufigkeitsverteilung von $g \Rightarrow p(g)$,
z.B. zur Kontrastverbesserung.
- ☞ Reduktion von K' Bits / Pixel auf $K < K'$.
- ☞ Berücksichtigung von Kennlinien der Bildwandler.
- ☞ Farbkorrektur durch Neuskalierung der Farbwerte
zwischen Schwarzpunkt und Weißpunkt.

Kontrastveränderung



Original



Veränderte Kennlinie

Farbkorrektur



Korallenriff, 4m Wassertiefe



nach Weißabgleich

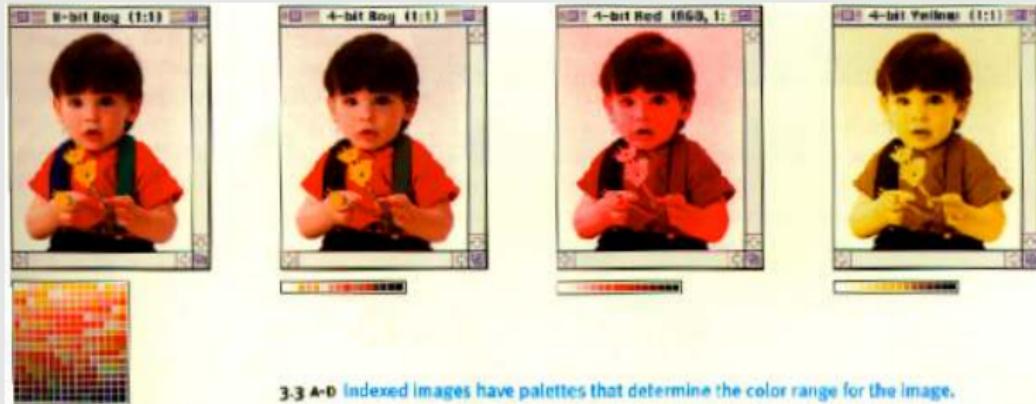
Speicherbedarf der Bildmatrix

- ▶ Bildmatrizen haben einen hohen Speicherbedarf, aber meist sehr viel **Redundanz**.
- ▶ Daher sind Kompressionstechniken sehr wichtig
 - ▶ für die langfristige Datenhaltung
 - ▶ und für die Datenübertragung.
- ▶ Für ein unkomprimiertes Bild ist der Speicherbedarf proportional zur
 - ▶ Anzahl der Pixel
 - ▶ und zum Speicherbedarf für den Farbwert eines Pixels.
- ▶ Für Animationen ist der Speicherbedarf zusätzlich proportional zur Länge der Bildfolge, typischerweise mindestens 25 Bilder pro Sekunde.

Speicherbedarf in Byte für quadratische Bildmatrizen

Seitenlänge	128	512	1024	8192
Binärbild	512	32 768	131 072	8 388 608
Grautonbild	4 096	262 144	1 048 576	67 108 864
256 Graust.				
RGB-Farbbild	12 288	786 432	3 145 728	201 326 592
256 ³ Farben				
Farben				
Multispektral,				
10 Kanäle				
je 256 Graust.	40 960	2 621 440	10 485 760	671 088 640

Quantelung von Farben: Farbpaletten

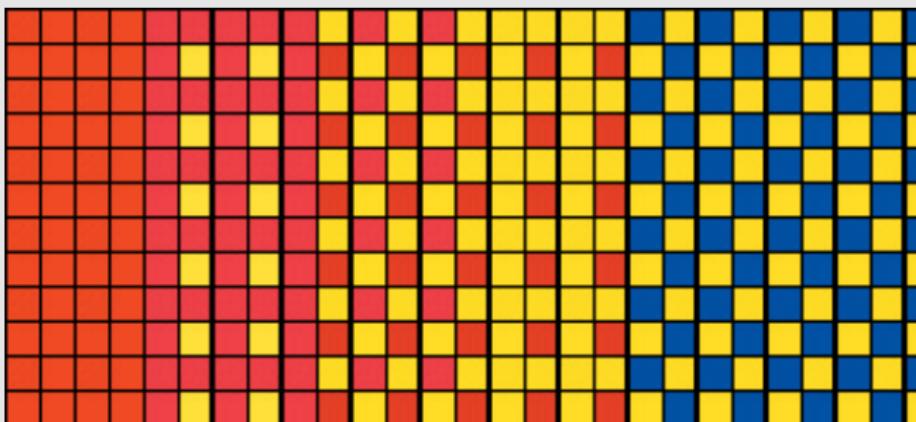


Die Farbwerte der Pixel eines Farbbildes können entweder quantisierte Farbvektoren oder **Indizes** einer Farbe in einer endlichen Palette von Farben sein (Pseudofarbbilder).

Dithering

Dithering

ist ein Verfahren, um Farbwerte, die nicht in einer Palette vorkommen, als Mischfarben von benachbarten Pixeln wiederzugeben.



Externe Bildformate

Externe Bildformate dienen

- ▶ zur Archivierung
- ▶ oder Datenübertragung.

Kompressionsverfahren: Zur Datenreduktion werden

- ▶ **reversible** und
- ▶ **nicht-reversible** Kompressionsverfahren verwendet.

Externe Bildformate

TIFF-Format: *Tag image file format*, für wissenschaftliche Bilddaten; verlustfreie Kompression.

GIF-Format: **Graphics interchange format**, weitverbreiteter Graphik-Standard, verlustfreie Kompression, besonders für Liniengrafik geeignet.

png-Format: *Portable Network Graphics Image*, verlustfreie Kompression.

JPEG-Format: *Joint photographic experts group* nicht-reversible, aber sehr effiziente Kompression durch Reihenentwicklung, besonders für natürliche Bilder (Photos) geeignet.

XPM-Format: *X pixel maps*, ASCII-Format für icons.

Teil XI

Repräsentation im Frequenzraum

Der Frequenzraum



30

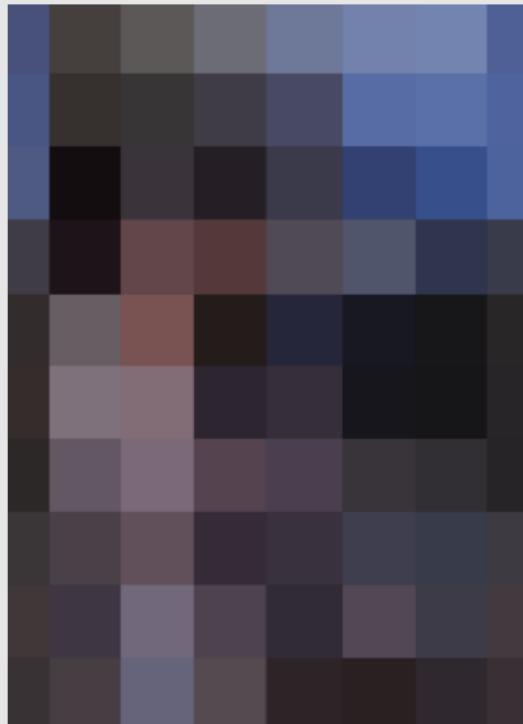
Auflösung eines Bildes

- Fourier-Reihenentwicklung
- Fourier-Transformation
(1-dimensional)

31

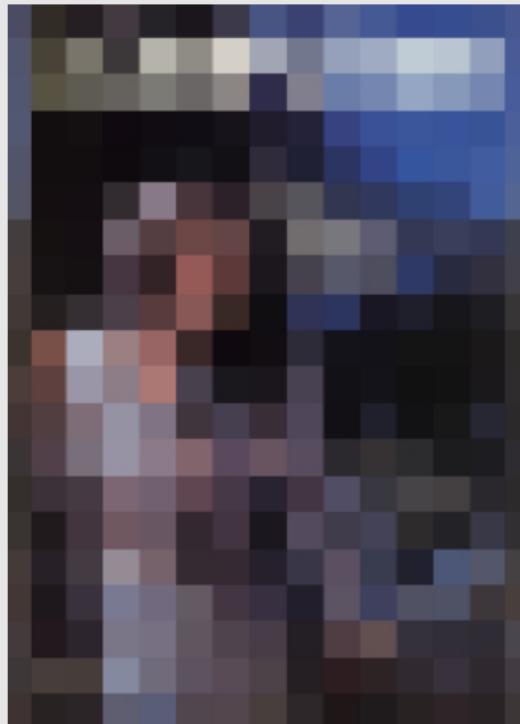
Abtasttheorem (sampling theorem)

Auflösung eines Bildes



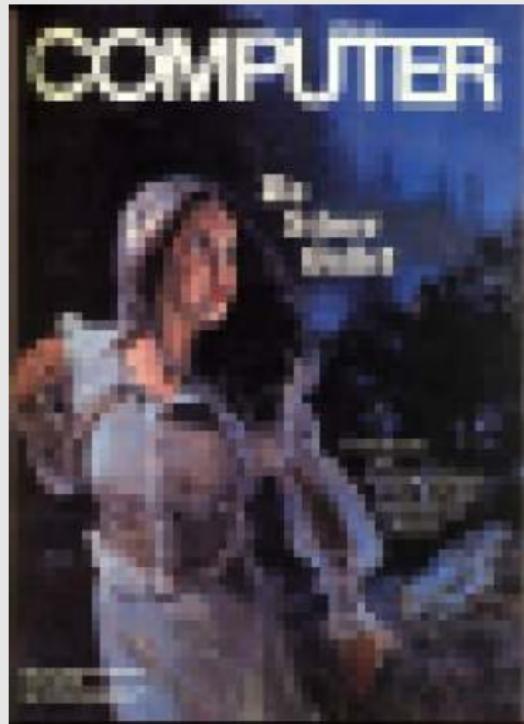
1 Pixel $\sim 128 \times 128$
Originalpixel

Auflösung eines Bildes



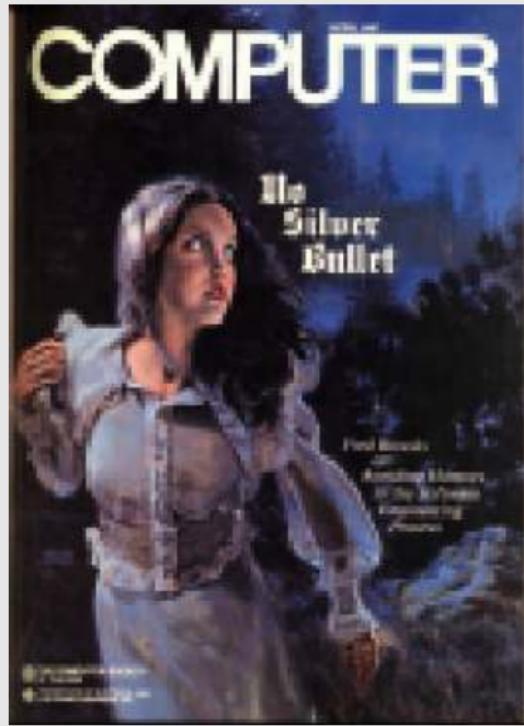
1 Pixel $\sim 64 \times 64$
Originalpixel

Auflösung eines Bildes



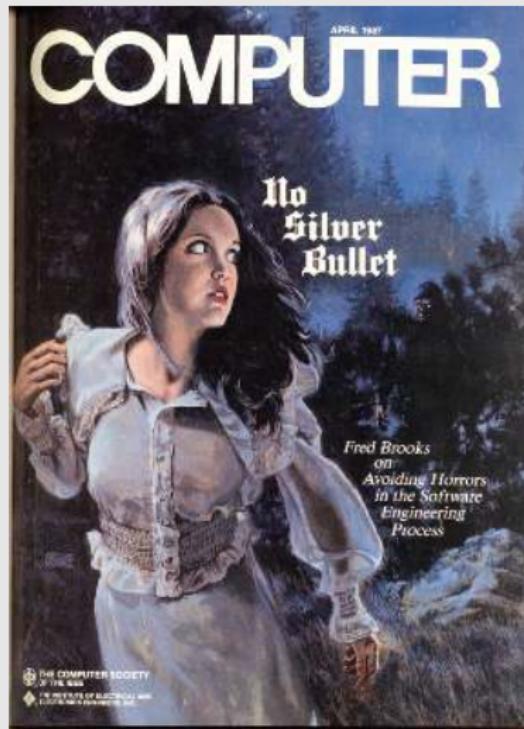
1 Pixel $\sim 16 \times 16$
Originalpixel

Auflösung eines Bildes



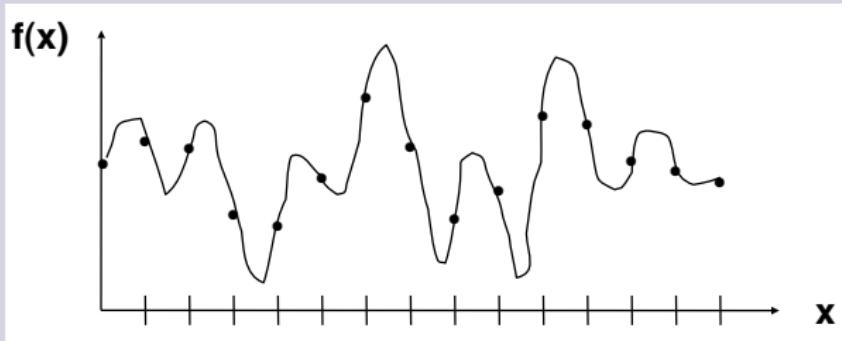
1 Pixel $\sim 8 \times 8$ Originalpixel

Auflösung eines Bildes



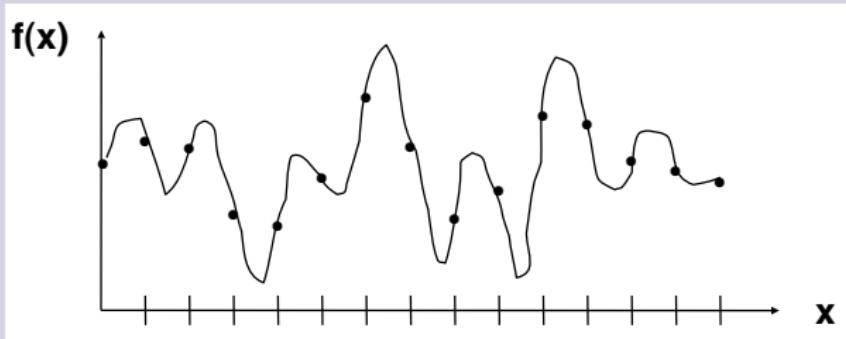
Originalbild

Problem (Optimale Rasterung:)



Gegeben sei eine Bildfunktion $g(x, y)$; wieviele Abtaststellen sind für eine optimale Abtastung nötig?

Problem (Optimale Rasterung:)

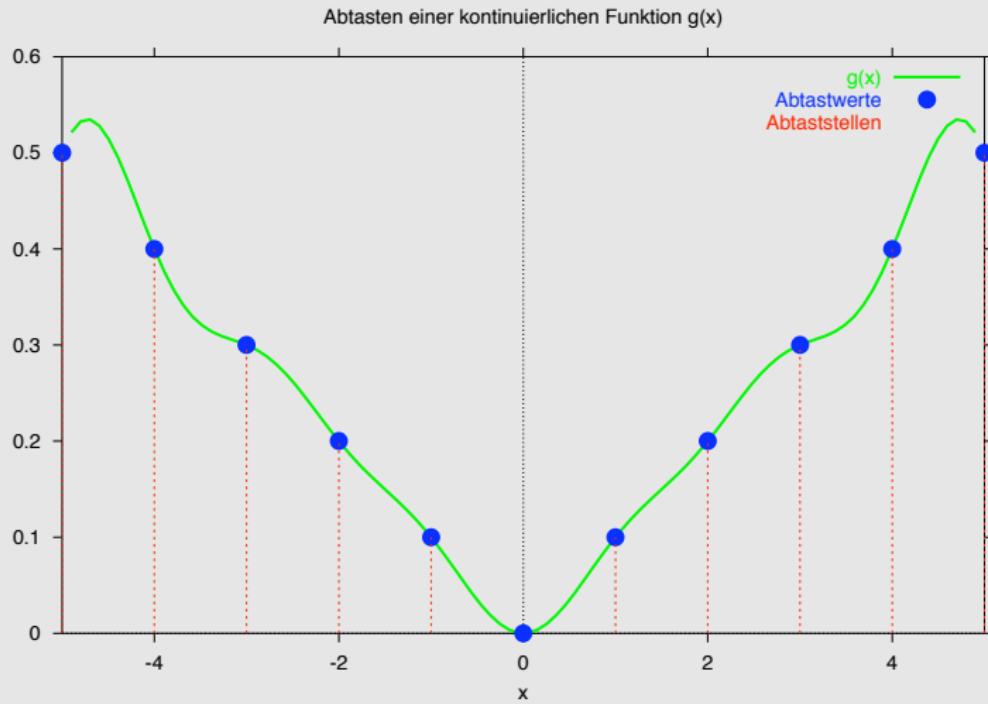


Gegeben sei eine Bildfunktion $g(x, y)$; wieviele Abtaststellen sind für eine optimale Abtastung nötig?

Plausible Lösung:

Je stärker $g(x, y)$ von einem Ort zum nächsten variiert, desto dichter muß abgetastet werden, wenn man keine Information verlieren will.

Beispiel: Eine aus den Abtaststellen rekonstruierte Kurve



Abtasttheorem:

Wieviele Abtaststellen sind für eine optimale Abtastung nötig?

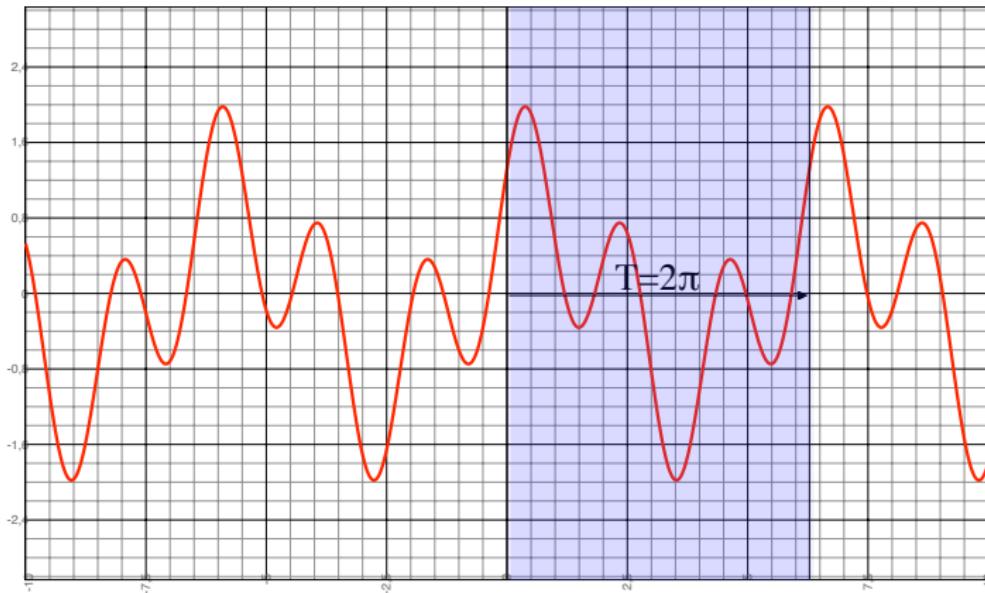
- ▶ Die präzise Antwort gibt das **Abtasttheorem** (sampling theorem).
- ▶ Es basiert auf einer Zerlegung von $g(x, y)$ in Sinus- und Cosinus-Bestandteile.

Herleitung: Zunächst *eindimensional*.

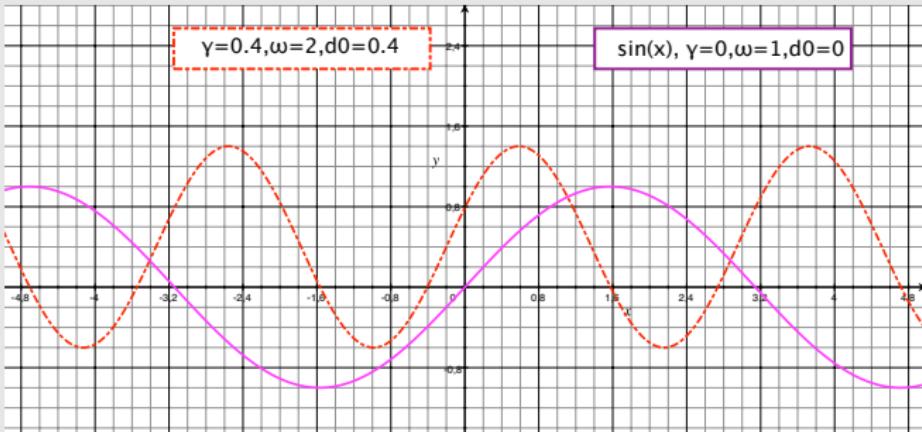
Fourier-Reihenentwicklung

Sei $g(t)$ eine **periodische** Funktion mit der Periode T ,

☞ d.h. $g(t) = g(t + T)$.



Die Parameter der Sinusfunktion



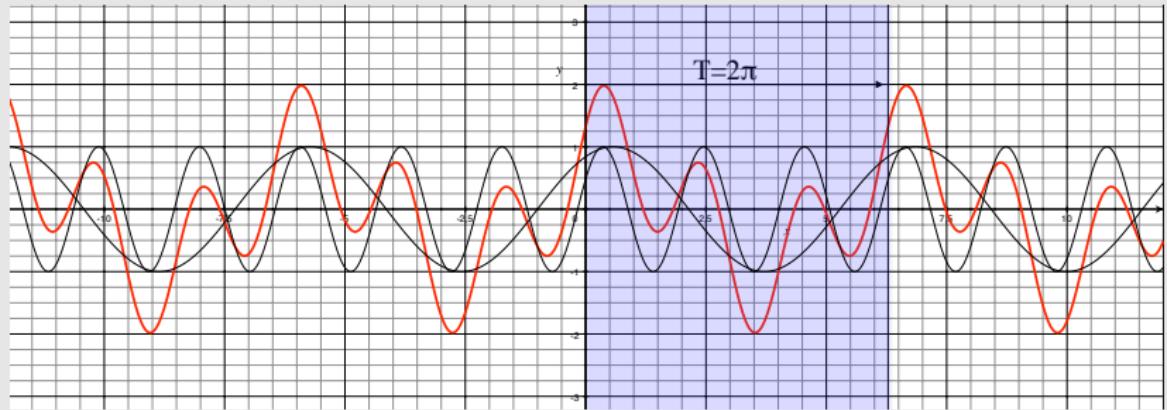
$$f(t) = d \sin(\gamma + \omega t) + d_0$$

γ : Phasenlage

ω : Kreisfrequenz

d : Amplitude

Repräsentation als Reihe von sinusförmigen Bestandteilen



$$g(t) = \sum_{k=1}^{\infty} d_k \sin(\gamma_k + \omega_k t) + d_0$$

Die Koeffizienten der Fourier-Reihe

Wegen der Periodizität von g muß ω_k ein Vielfaches der Grundfrequenz $\frac{2\pi}{T}$ sein: $\omega_k = \frac{2\pi k}{T}$.

$$g(t) = \sum_{k=1}^{\infty} d_k \sin \left(\gamma_k + \frac{2\pi k}{T} t \right) + d_0$$

mit Winkelsummensatz

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta$$

$$g(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[a_k \cos \left(\frac{2\pi k}{T} t \right) + b_k \sin \left(\frac{2\pi k}{T} t \right) \right]$$

Fourierreihe einer kontinuierlichen Funktion

$$g(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} \left[a_k \cos\left(\frac{2\pi k}{T}t\right) + b_k \sin\left(\frac{2\pi k}{T}t\right) \right]$$

$$e^{ix} = \cos x + i \sin x, \quad i = \sqrt{-1} \quad \textcolor{red}{Euler - Formel}$$

$$\cos x = \frac{1}{2}(e^{ix} + e^{-ix})$$

$$\sin x = \frac{1}{2i}(e^{ix} - e^{-ix}) \Rightarrow$$

$$\begin{aligned} g(t) &= \frac{a_0}{2} + \frac{1}{2} \sum_{k=1}^{\infty} [(a_k - ib_k)e^{(i2\pi kt/T)} \\ &\quad + (a_k + ib_k)e^{(-i2\pi kt/T)}] \end{aligned}$$

$$g(t) = \frac{a_0}{2} + \frac{1}{2} \sum_{k=1}^{\infty} [(a_k - ib_k)e^{(i2\pi kt/T)} + (a_k + ib_k)e^{(-i2\pi kt/T)}]$$

$$c_k = \frac{1}{2}(a_k - ib_k) \quad \text{komplexer Koeffizient, } k > 0$$

$$c_{-k} = c_k^* = \frac{1}{2}(a_k + ib_k) \quad \text{konjugiert kompl. K., } k > 0$$

$$c_0 = \frac{a_0}{2}$$

$$g(t) = \sum_{k=-\infty}^{\infty} c_k e^{i2\pi kt/T}$$

Die Fourier-Reihe in komplexer Darstellung

$$g(t) = \sum_{k=-\infty}^{\infty} c_k e^{i2\pi kt/T}$$

Satz

Für jede Funktion mit einer Periode T und der Eigenschaft, daß

$$\int_0^T |g(t)| dt < \infty$$

gibt es diese Reihenentwicklung.

Bildfunktionen erfüllen diese Bedingung, da sie beschränkt sind.

Die Koeffizienten c_k der Fourier-Reihe

$$g(t) = \sum_{k=-\infty}^{\infty} c_k e^{i2\pi kt/T}$$

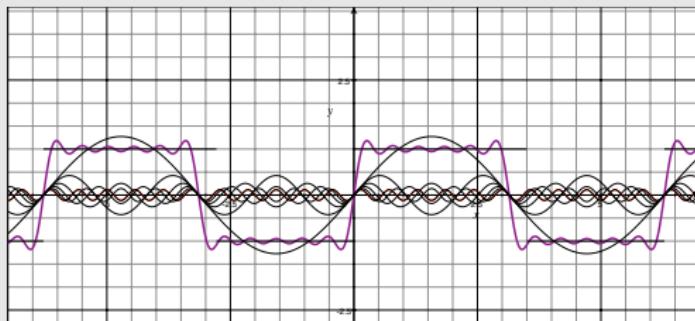
Die Koeffizienten sind

$$c_k = \frac{1}{T} \int_0^T g(t) \cdot e^{-\frac{i2\pi kt}{T}} dt, \quad -\infty \leq k \leq \infty$$

$$a_k = \frac{2}{T} \int_0^T g(t) \cdot \cos\left(\frac{2\pi kt}{T}\right) dt, \quad 0 \leq k \leq \infty$$

$$b_k = \frac{2}{T} \int_0^T g(t) \cdot \sin\left(\frac{2\pi kt}{T}\right) dt, \quad 0 \leq k \leq \infty$$

Beispiel: Rechteckfunktion



Rechteckfunktion: Amplitude A , Periode 2π ,

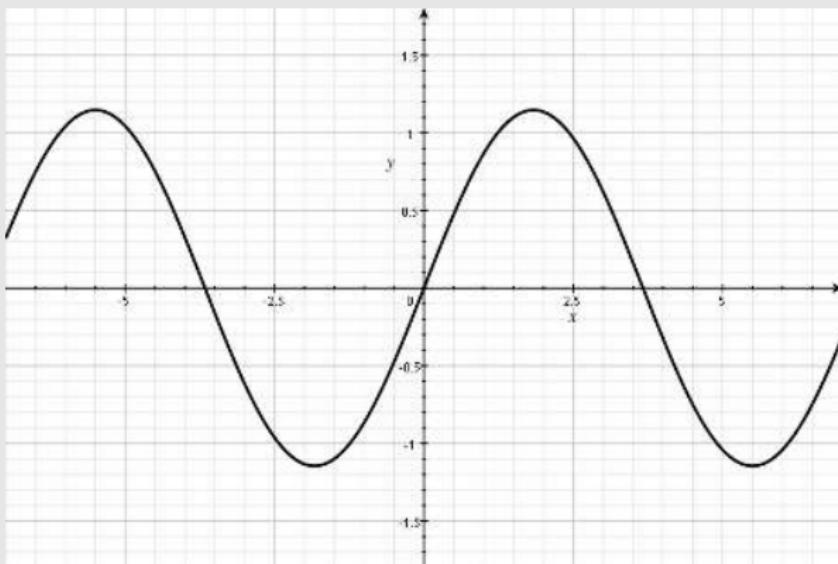
$$a_k = 0$$

$$b_k = \frac{1}{1 + 2k}$$

$$g(x) = \frac{4A}{\pi} \sum_{k=0}^{\infty} \frac{1}{1 + 2k} \sin((2k + 1)x)$$

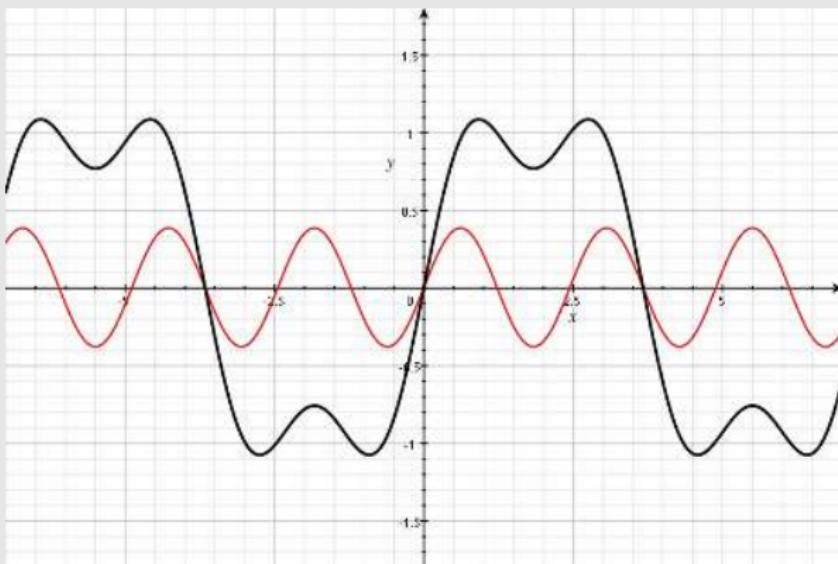
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



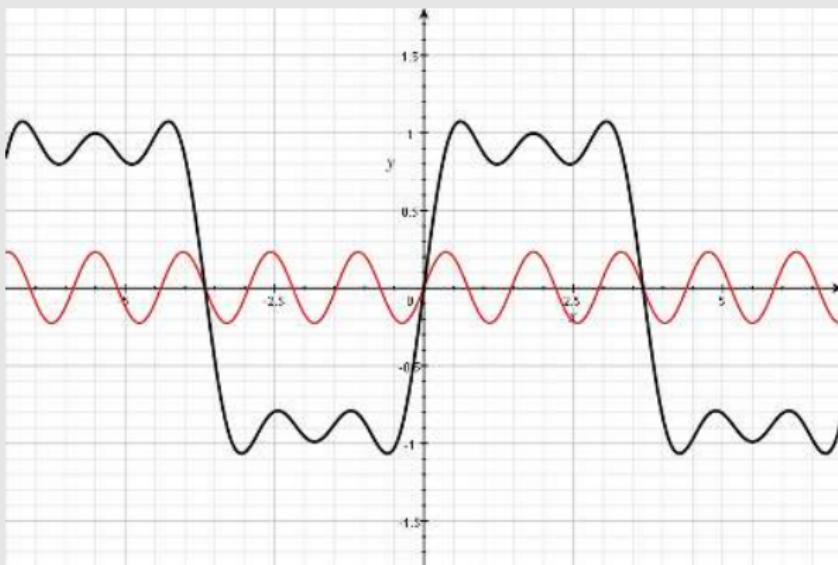
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



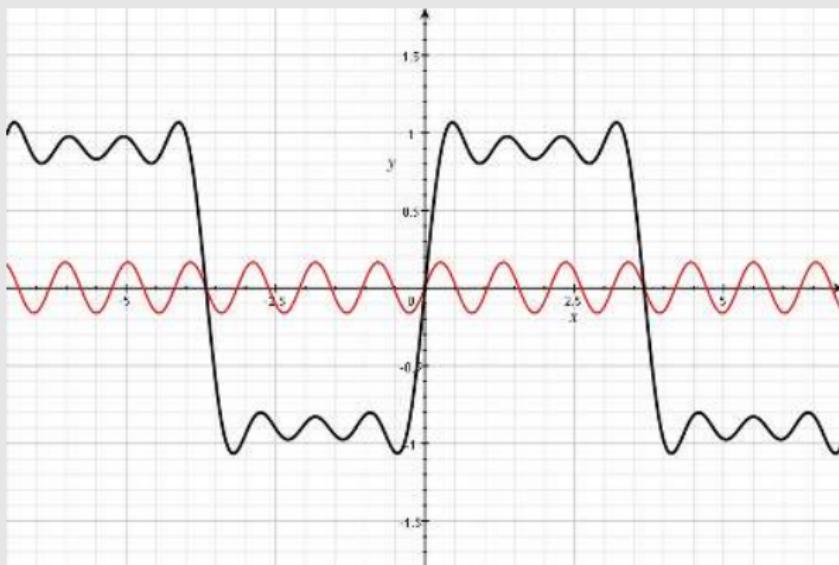
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



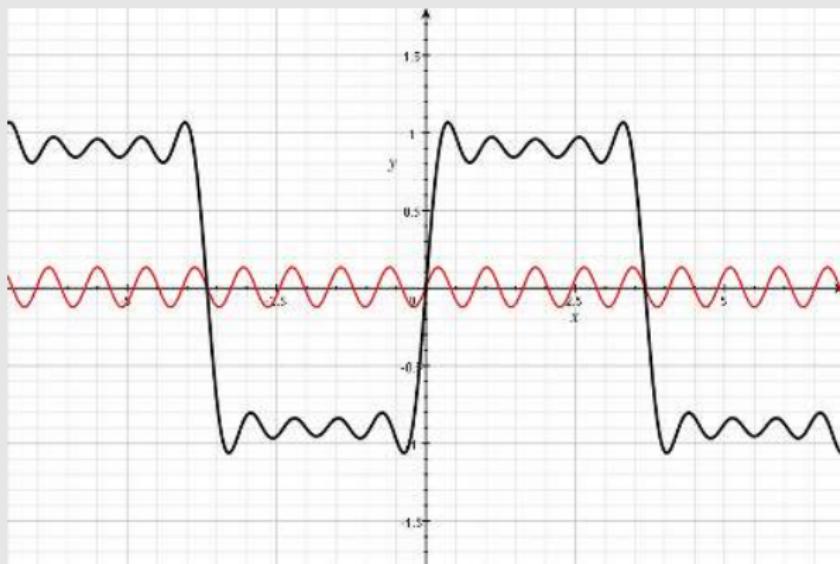
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



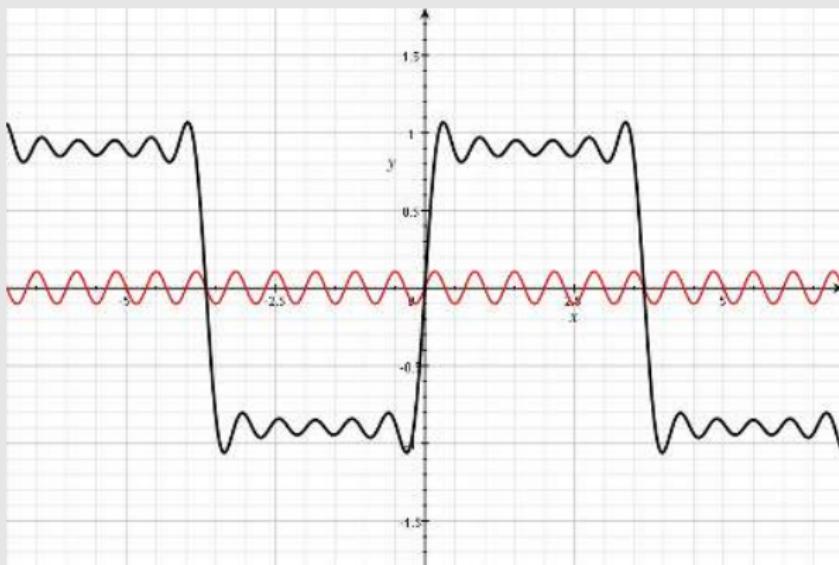
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



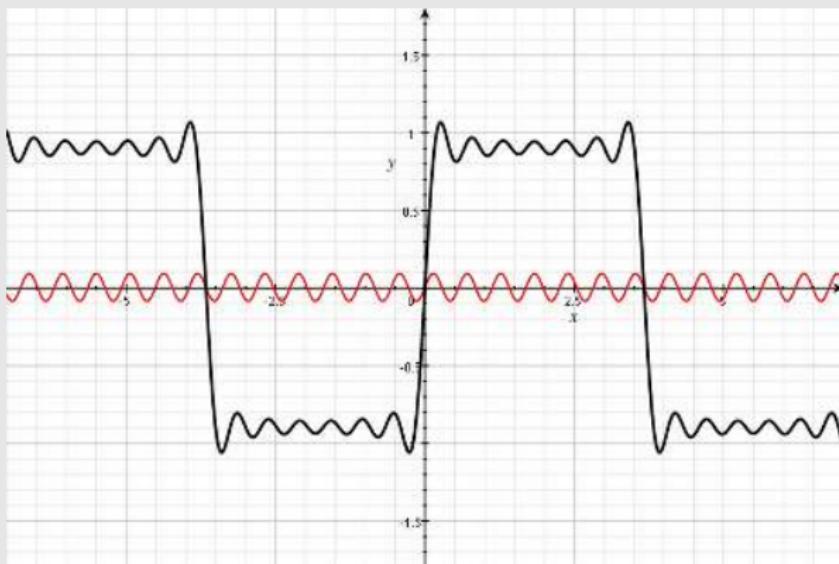
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



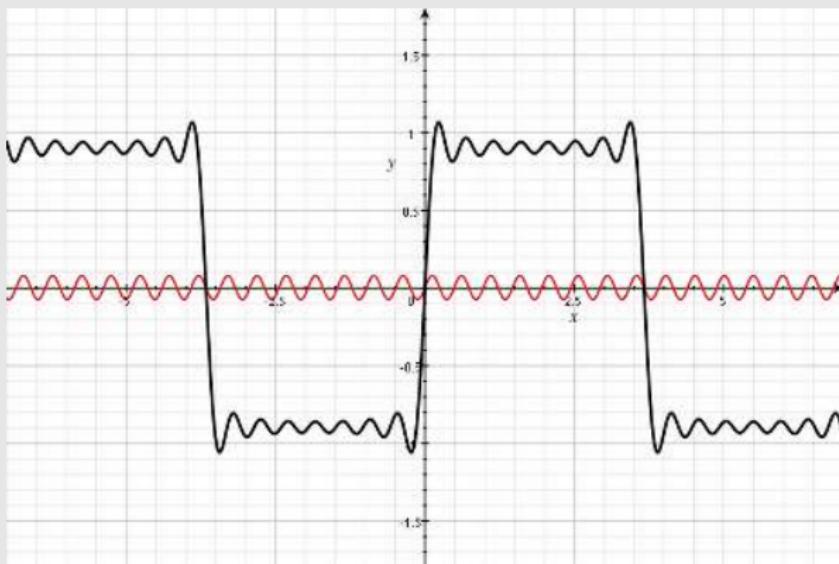
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



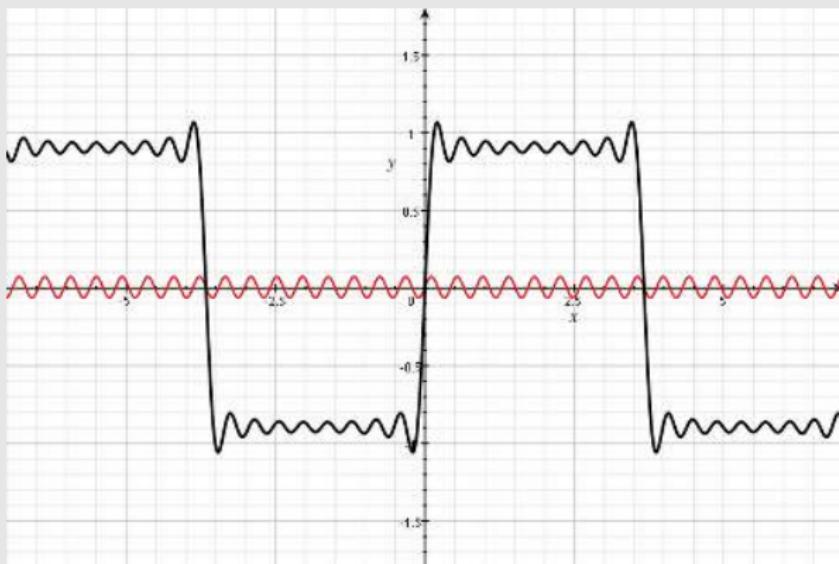
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



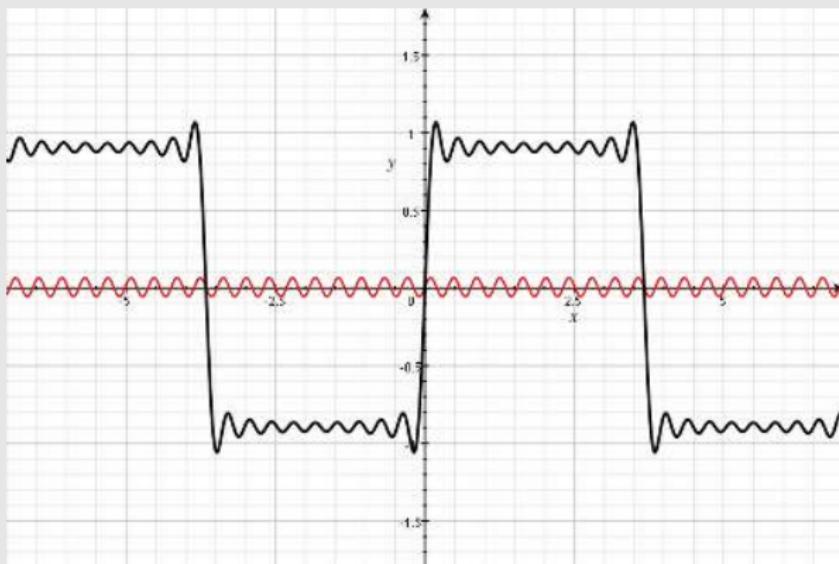
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



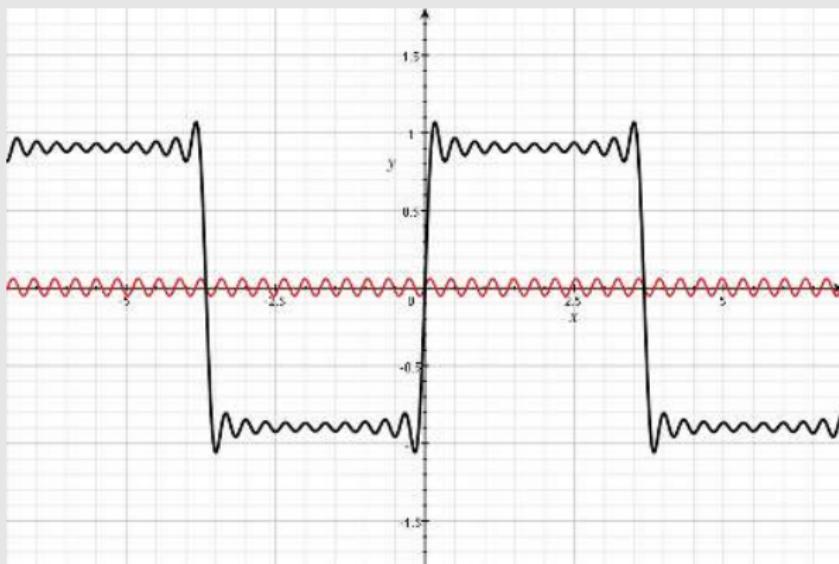
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



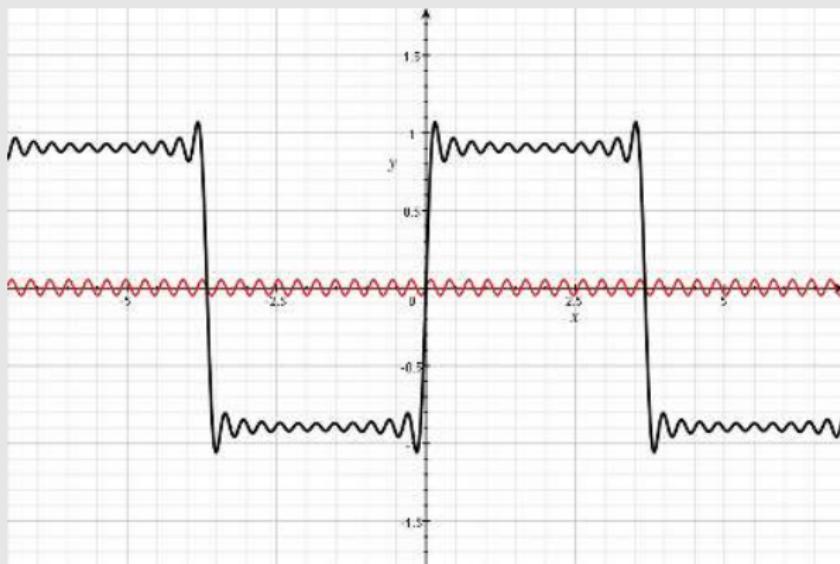
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



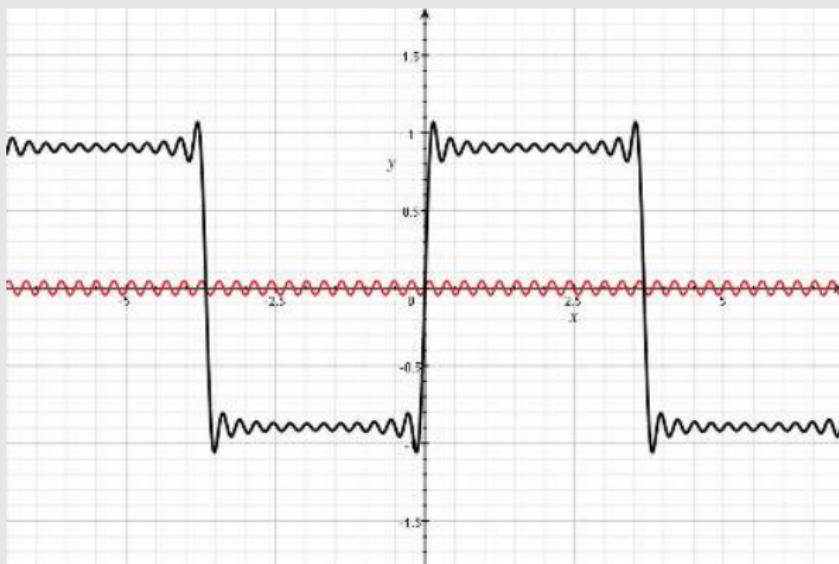
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



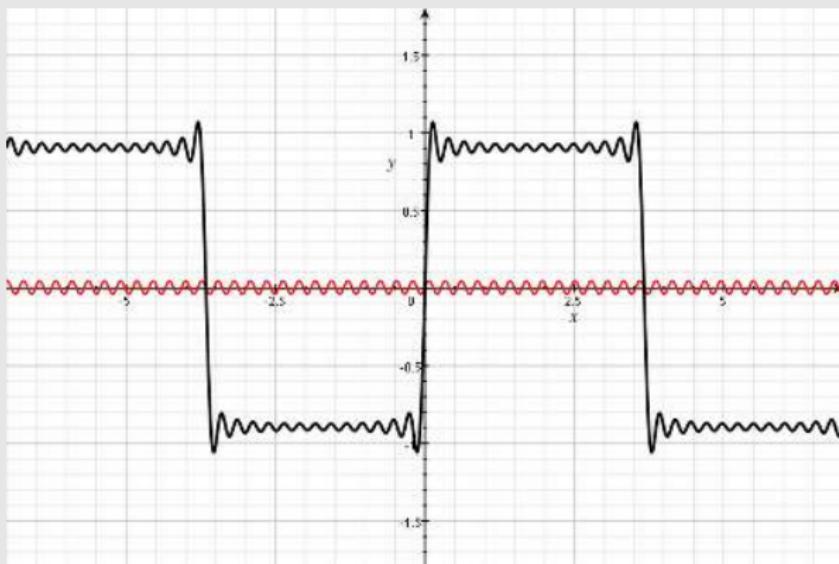
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



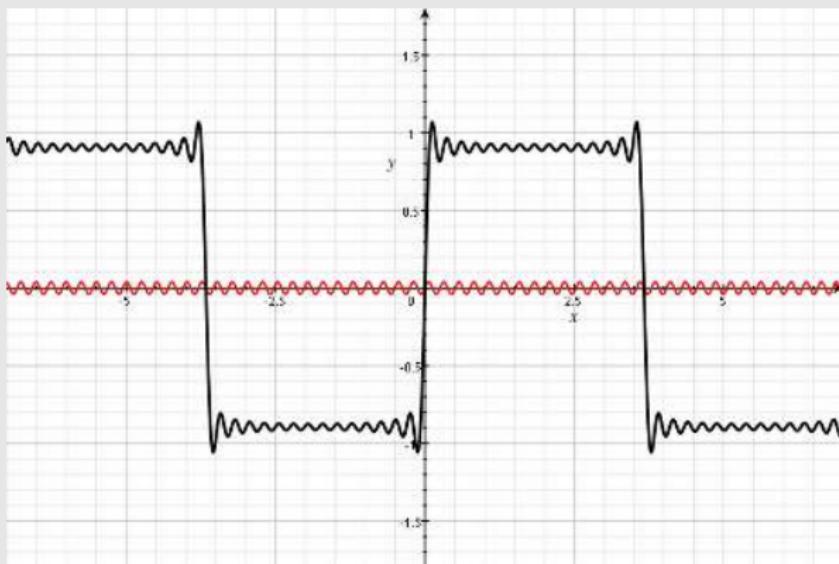
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



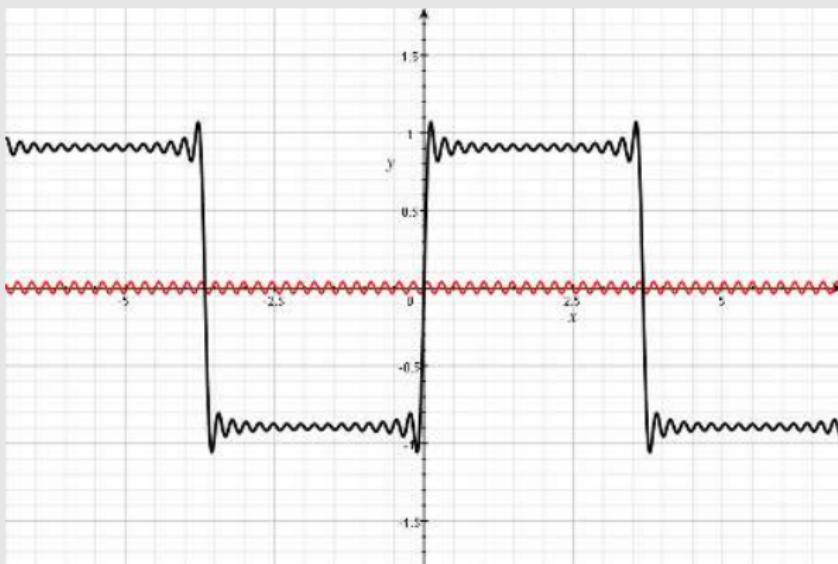
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



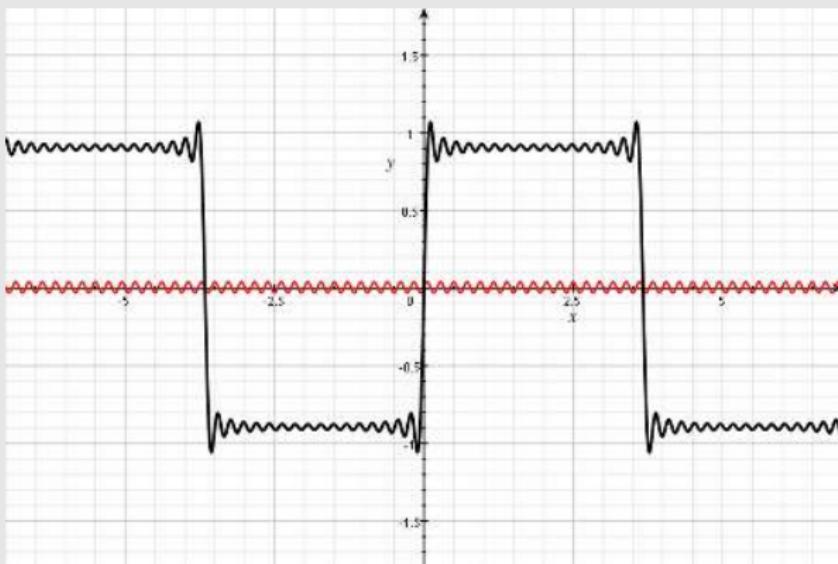
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



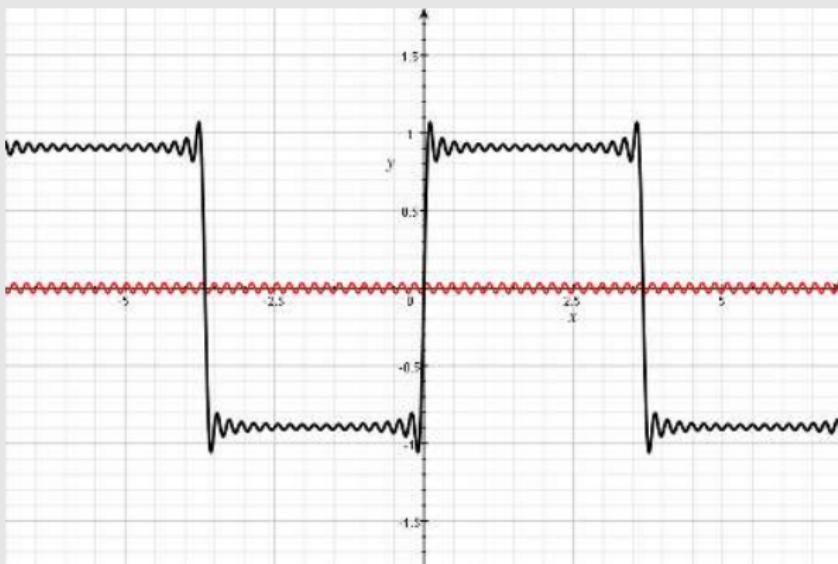
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



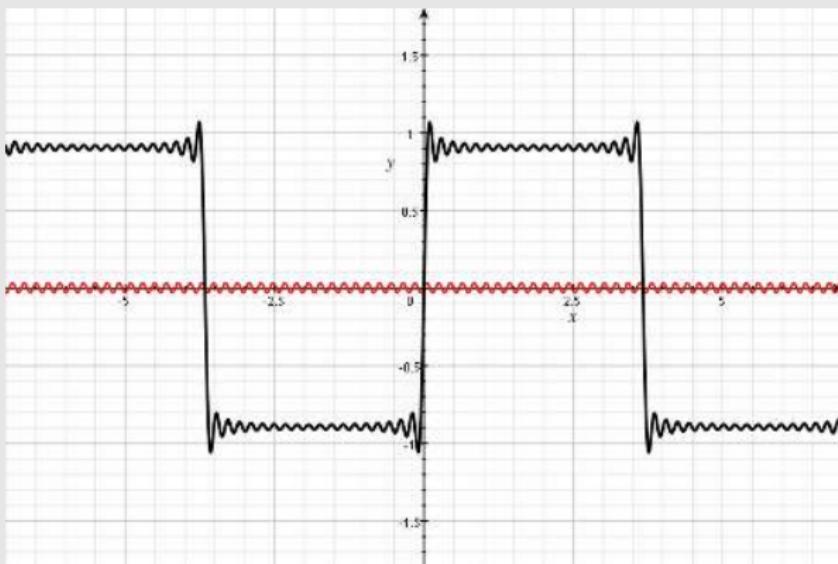
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



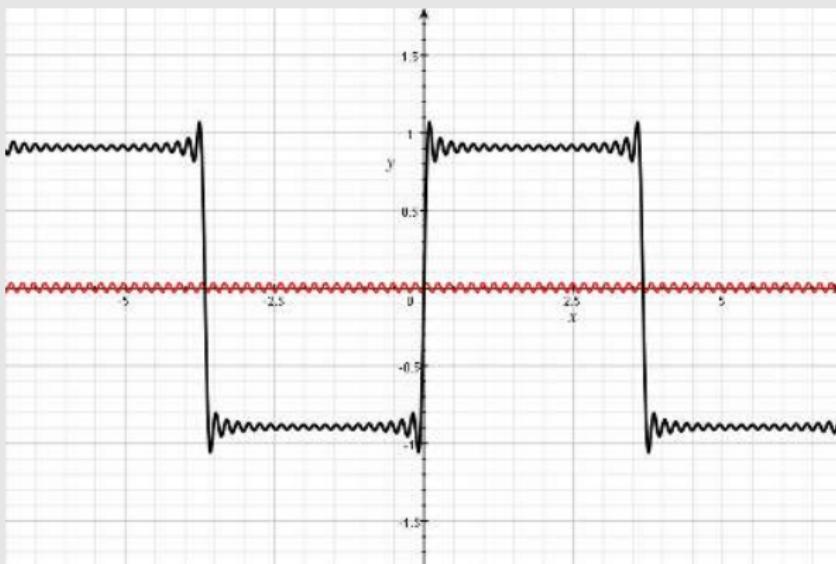
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



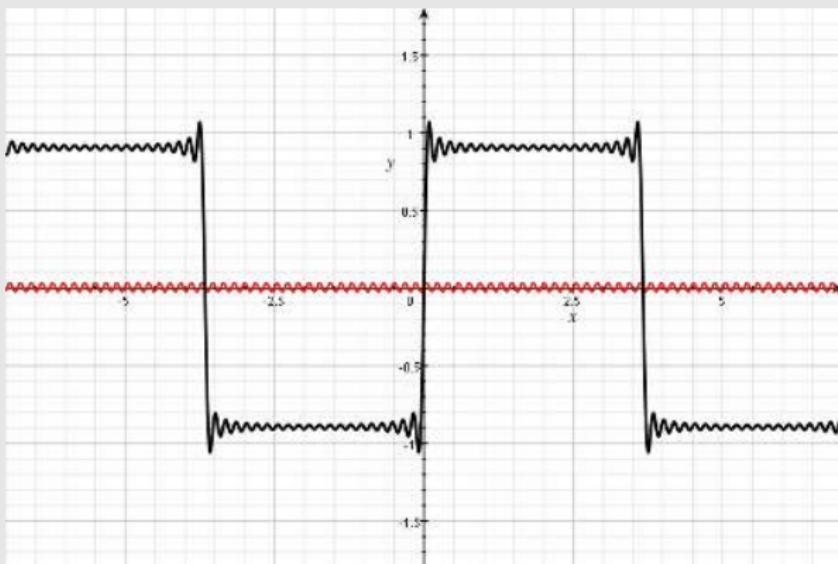
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



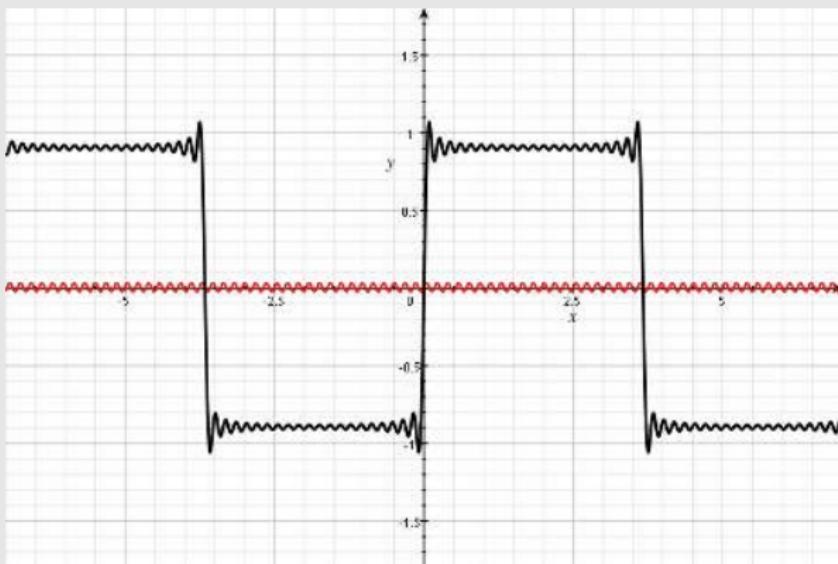
Beispiel: Rechteckfunktion

20 Terme der Fourierreihe

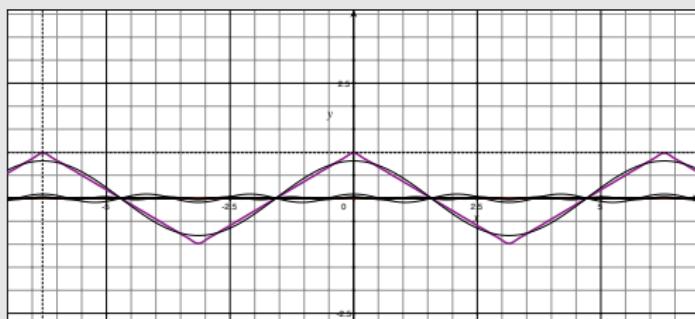


Beispiel: Rechteckfunktion

20 Terme der Fourierreihe



Beispiel: Dreieck-Funktion



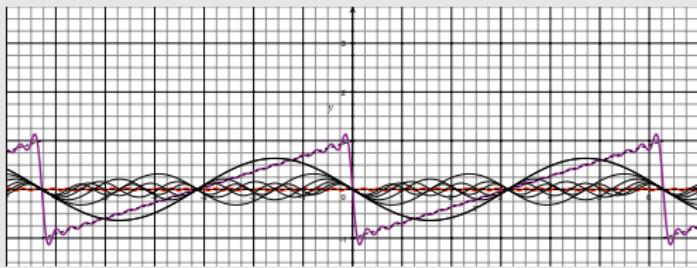
Dreieck-Funktion: Amplitude A , Periode 2π ,

$$a_k = \frac{1}{(1 + 2k)^2}$$

$$b_k = 0$$

$$g(x) = \frac{8A}{\pi\pi} \sum_{k=0}^{\infty} \frac{1}{(1 + 2k)^2} \cos((2k + 1)x)$$

Beispiel: Sägezahn-Funktion



Sägezahn-Funktion: Amplitude A , Periode 2π , die ersten 5 Terme

$$a_k = 0$$

$$b_k = \frac{1}{1+k}$$

$$g(x) = \frac{-2A}{\pi} \sum_{k=0}^{\infty} \frac{1}{1+k} \sin((k+1)x)$$

Eindimensionale Fourier-Transformation

Ähnlich wie Fourier-Reihenentwicklung, $T \rightarrow \infty$

$$\begin{aligned} G(f) &= \mathcal{F}\{g(t)\} \\ &= \int_{-\infty}^{\infty} g(t) \cdot e^{-2\pi i f t} dt \quad F\text{-Transf.} \end{aligned}$$

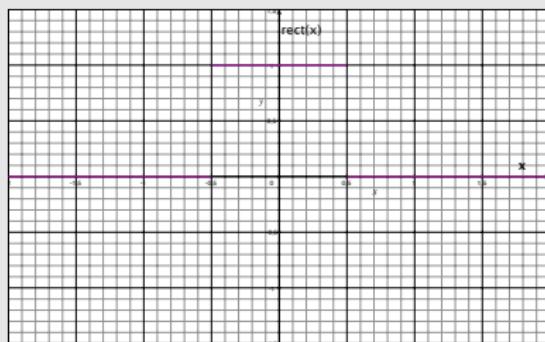
$$\begin{aligned} g(t) &= \mathcal{F}^{-1}\{G(f)\} \\ &= \int_{-\infty}^{\infty} G(f) \cdot e^{+2\pi i f t} df \quad \text{Inverse } F. \end{aligned}$$

$G(f)$ entspricht den c_k und gibt an, mit welchem Gewicht sinusförmige Bestandteile der Frequenz f in $g(t)$ enthalten sind.

- ▶ $G(f)$ ist eine andere, gleichwertige Repräsentation der Bildfunktion.
- ▶ Aus $G(f)$ kann $g(x)$ eindeutig rekonstruiert werden.
- ▶ **Achtung!** $G(f)$ ist i.a. **komplex**!
- ▶ $|G(F)|$ heißt **Amplitudenspektrum** von $g(t)$, meist kurz **Spektrum** genannt

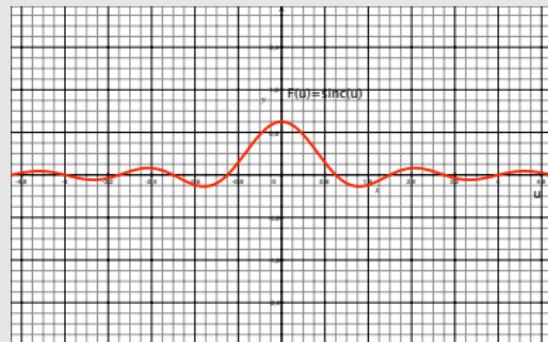
Die Fourier-Transformierte $\text{sinc}(u)$ der Rechteck-Funktion $\text{rect}(x)$

$f(x)$
Rechteck-Funktion



$$\text{rect}(x) = \begin{cases} 1 & -0.5 \leq x \leq 0.5 \\ 0 & \text{sonst} \end{cases}$$

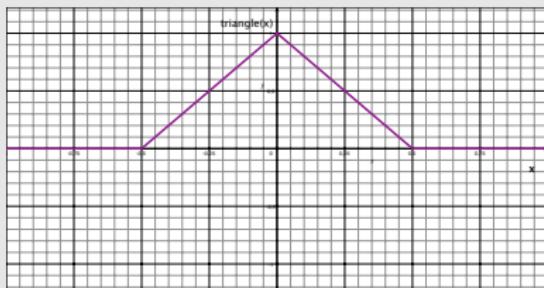
$F(u)$
sinc-Funktion



$$\text{sinc}(u) = \frac{\sin(\pi u)}{\pi u}$$

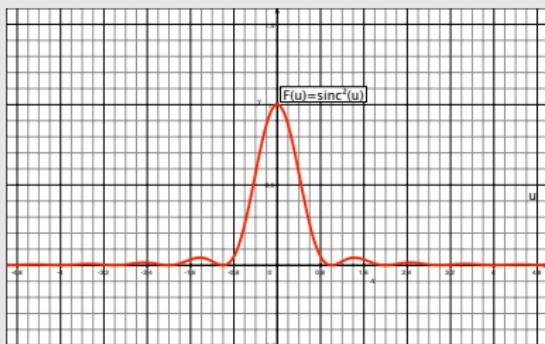
Die Fourier-Transformierte $T(u)$ der Dreieck-Funktion $\text{triangle}(x)$

$\text{triangle}(x)$
Dreieck-Funktion



$$\text{tria}(x) = \begin{cases} 1 - 2|x| & |x| \leq 0.5 \\ 0 & \text{sonst} \end{cases}$$

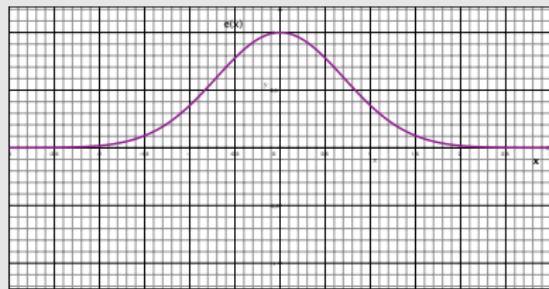
$T(u)$
 sinc^2 -Funktion



$$\text{sinc}^2(u) = \frac{\sin^2(\pi u)}{\pi^2 u^2}$$

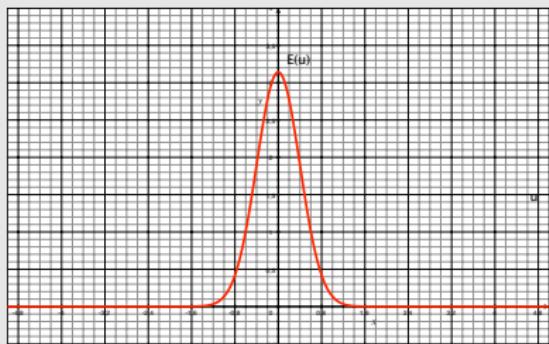
Die Fourier-Transformierte $E(u)$ der Gauß-Funktion $e(x)$

$e(x)$
Gauß-Funktion



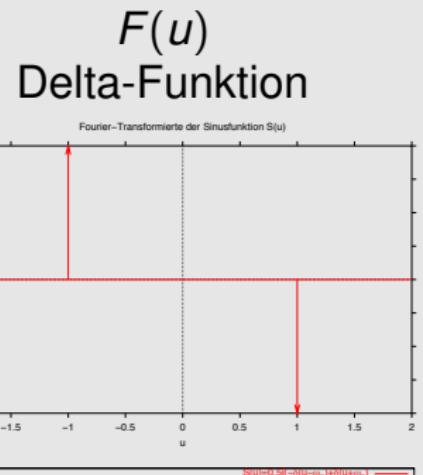
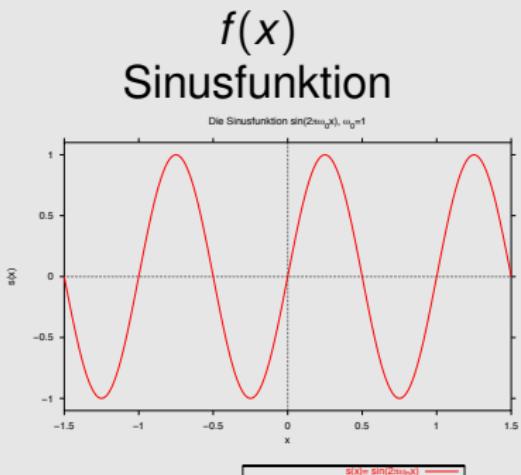
$$e(x) = e^{-\alpha x^2}$$

$E(u)$



$$E(u) = \frac{\pi}{\alpha} e^{-\frac{\pi u^2}{\alpha}}$$

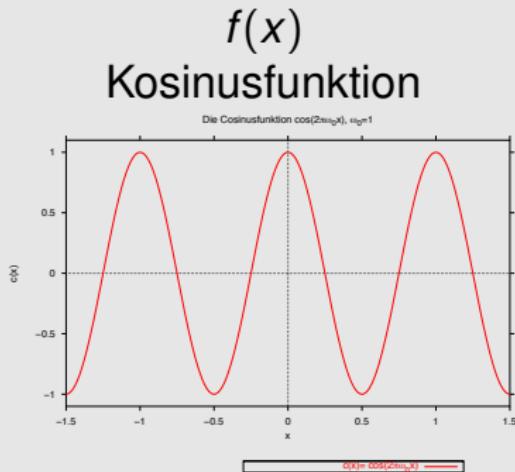
Die Fourier-Transformierte $F(u)$ der Sinus-Funktion $\sin(2\pi\omega_0 x)$



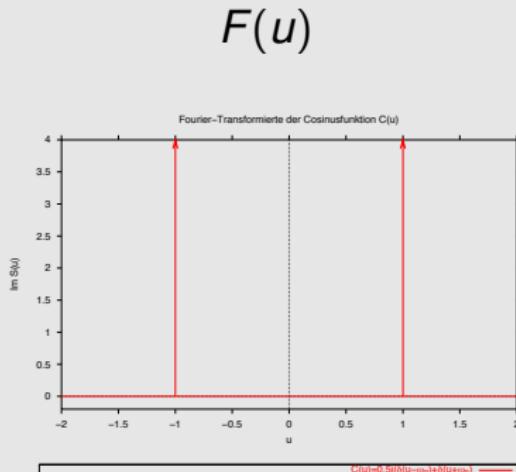
$$\sin(2\pi\omega_0 x)$$

$$\frac{1}{2}i[-\delta(u - \omega_0) + \delta(u + \omega_0)]$$

Die Fourier-Transformierte $F(u)$ der Kosinus-Funktion $\cos(2\pi\omega_0 x)$

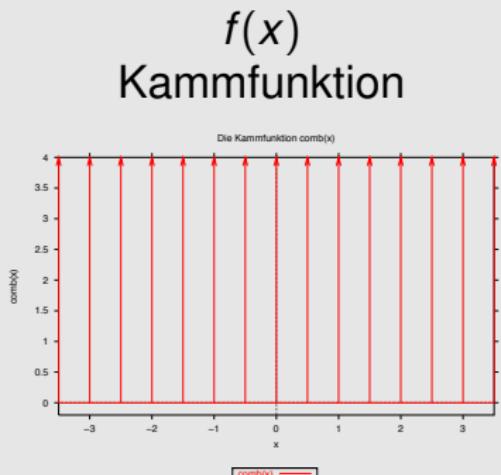


$$\cos(2\pi\omega_0 x)$$

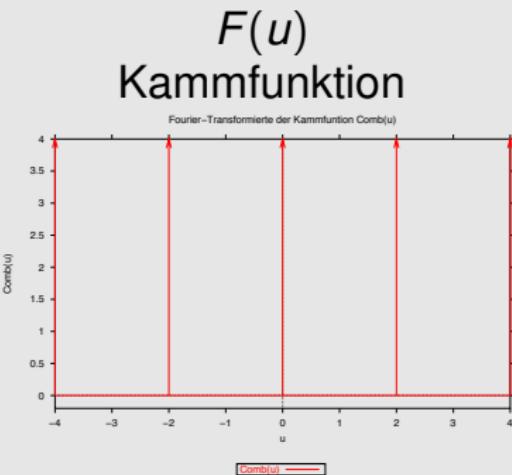


$$\frac{1}{2}[\delta(u - \omega_0) + \delta(u + \omega_0)]$$

Die Fourier-Transformierte $\text{Comb}(u)$ der Kamm-Funktion $\text{comb}(x)$



$$\sum_{n=-\infty}^{\infty} \delta(x - x_0)$$



$$\sum_{n=-\infty}^{\infty} \delta(u - \frac{n}{x_0})$$

Faltung und Faltungstheorem

Definition (Faltung)

Die Faltung zweier Funktionen f und g ist eine Funktion h einer Verschiebung y , mit dem Faltungsoperator $*$:

$$h(y) = f * g = \int_{-\infty}^{\infty} f(x)g(y - x)dx$$

Satz (Faltungstheorem)

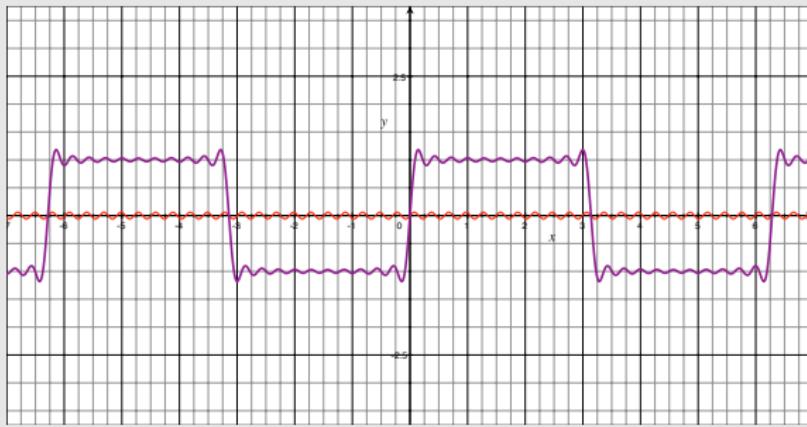
Die Fourier-Transformierte der Faltung zweier Funktionen f und g ist gleich dem Produkt $\mathcal{F}(f) \cdot \mathcal{F}(g)$ der Fourier-Transformierten der beiden Funktionen.

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g) = F \cdot G$$

Bandbreite und Grenzfrequenz

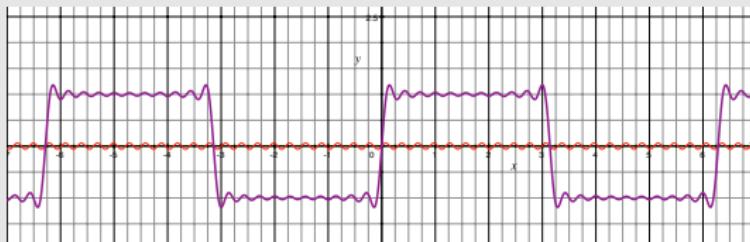
Definition (Grenzfrequenz)

$g(t)$ ist bandbeschränkt mit Grenzfrequenz W ,
wenn $G(f) = 0$ für $|f| > W$.



Eine Funktion und ihre Grenzfrequenz

Abtasttheorem (sampling theorem)



Satz (Abtasttheorem)

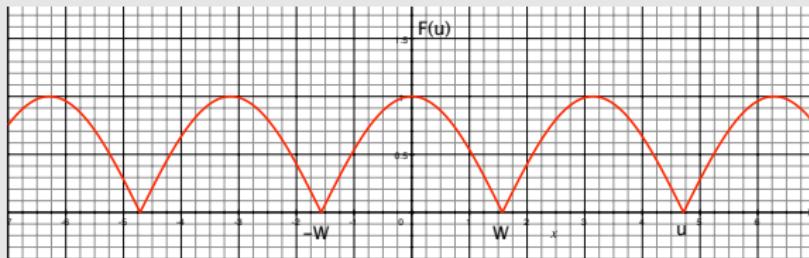
- ☞ Eine Funktion mit Bandbreite W kann exakt aus Abtastwerten zurückgewonnen werden, wenn diese höchstens $\frac{1}{2W}$ auseinanderliegen. (Nyquist?Shannon sampling theorem)

Der Beweis

Beweis.

☞ $g(t)$ sei bandbeschränkt mit Grenzfrequenz W .
 $\Rightarrow G(f) = 0$ für $|f| \geq W$

Setzt man $G(f)$ periodisch fort, kann für die Fourier-Transformierte eine Fourier-Reihenentwicklung angewendet werden. □



$$G(\textcolor{blue}{f}) = \sum_{k=-\infty}^{\infty} c_k \cdot e^{-i2\pi kf/(2W)} \quad G(f) \text{ hat die Periode } 2W$$

mit $c_k = \frac{1}{2W} \int_{-W}^W G(\textcolor{blue}{f}) e^{i2\pi kf/(2W)} df$

wegen $G(|\textcolor{blue}{f}| > W) = 0 \Rightarrow$

$$= \frac{1}{2W} \int_{-\infty}^{+\infty} G(\textcolor{blue}{f}) \cdot e^{i2\pi \textcolor{red}{fk}/(2W)} df$$

wegen Definition von $\mathcal{F}^{-1} \Rightarrow$

$$= \frac{1}{2W} \cdot g\left(\frac{k}{2W}\right)$$

$$g\left(\frac{k}{2W}\right) = \text{diskrete Abtastwerte im Abstand } \frac{1}{2W}$$

q.e.d.

Rekonstruktion aus Abtastwerten

$$\begin{aligned} g(t) &= \mathcal{F}^{-1}\{G(f)\} = \mathcal{F}^{-1} \left\{ \frac{1}{2W} \sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right) \cdot e^{-i2\pi f k/(2W)} \right\} \\ &= \int_{-W}^{+W} \frac{1}{2W} \sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right) e^{-i2\pi f k/(2W)} \cdot e^{i2\pi f t} df \\ &= \sum_{k=-\infty}^{\infty} \frac{1}{2W} \cdot g\left(\frac{k}{2W}\right) \underbrace{\int_{-W}^{+W} e^{i2\pi f(t - \frac{k}{2W})} df}_{S^*} \\ S^* &= \frac{1}{i2\pi(t - \frac{k}{2W})} \cdot e^{i2\pi f(t - \frac{k}{2W})} \Big|_{-W}^{+W} \quad \text{mit der Euler-Formel:} \end{aligned}$$

Rekonstruktion als Fourier-Reihe

Sei $\alpha = 2\pi \left(t - \frac{k}{2W} \right)$

$$S^* = \frac{1}{\alpha} \left\{ -i \cos(f\alpha) + \sin(f\alpha) \right\} \Big|_{-W}^{+W}$$

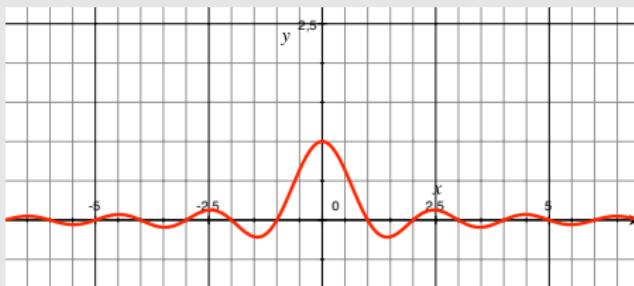
cos -Term verschwindet wegen $\cos(W) = \cos(-W)$

$$g(t) = \sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right) \frac{\sin[2\pi W(t - \frac{k}{2W})]}{2\pi W(t - \frac{k}{2W})}, \text{sinc}(u) = \frac{\sin(\pi u)}{\pi u}$$



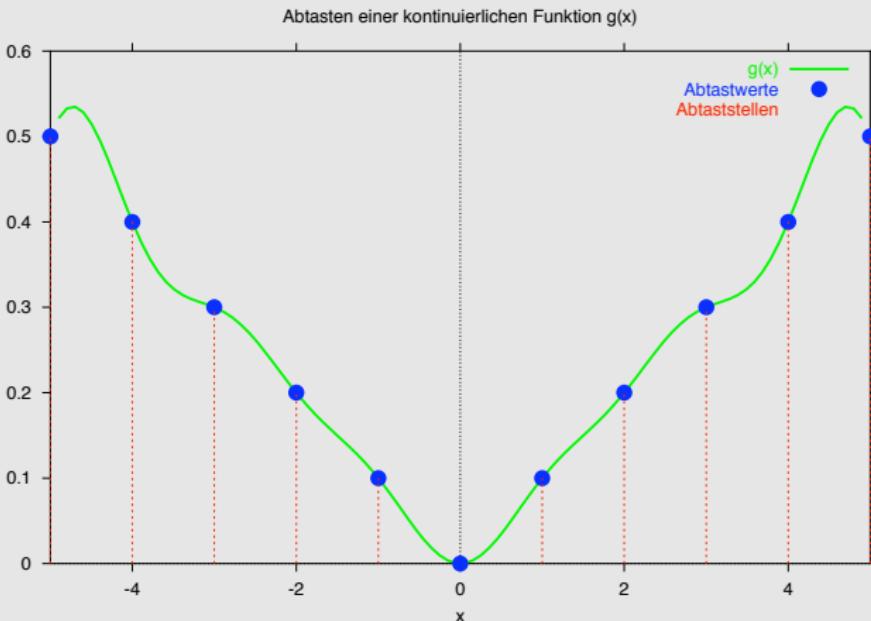
$$g(t) = \sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right) \text{sinc} \left[2W \left(t - \frac{k}{2W} \right) \right]$$

Rekonstruktion des Bildsignals

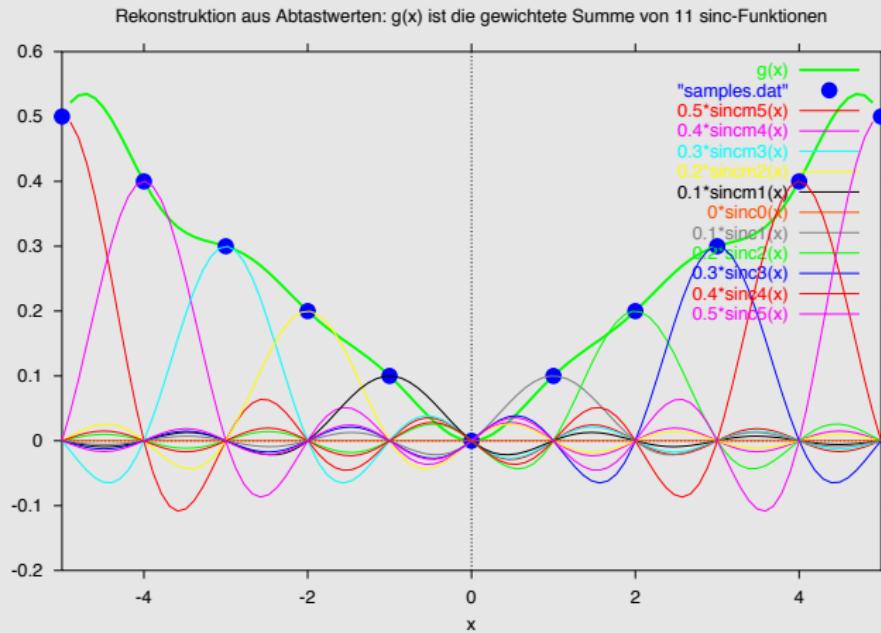


- ☞ Zur Rekonstruktion von $g(t)$ aus Abtastwerten
 - ▶ nehme man sinc-Funktionen,
 - ▶ zentriere diese um einen Abtastwert,
 - ▶ skaliere sie mit diesem
 - ▶ und addiere alle sinc-Funktionen auf.

Beispiel: Eine Kurve und die Abtaststellen



Die Rekonstruktion aus sinc-funktionen



Scheinfrequenzen: Aliasing

Wird das Abtasttheorem beim Abtasten eines Bildsignals oder beim Unterabtasten eines Digitalbildes nicht berücksichtigt und die Abtastfrequenz zu klein gewählt, dann entstehen Bilder, die Frequenzen enthalten, die im Ursprungsbild nicht vorkommen (Scheinfrequenzen).

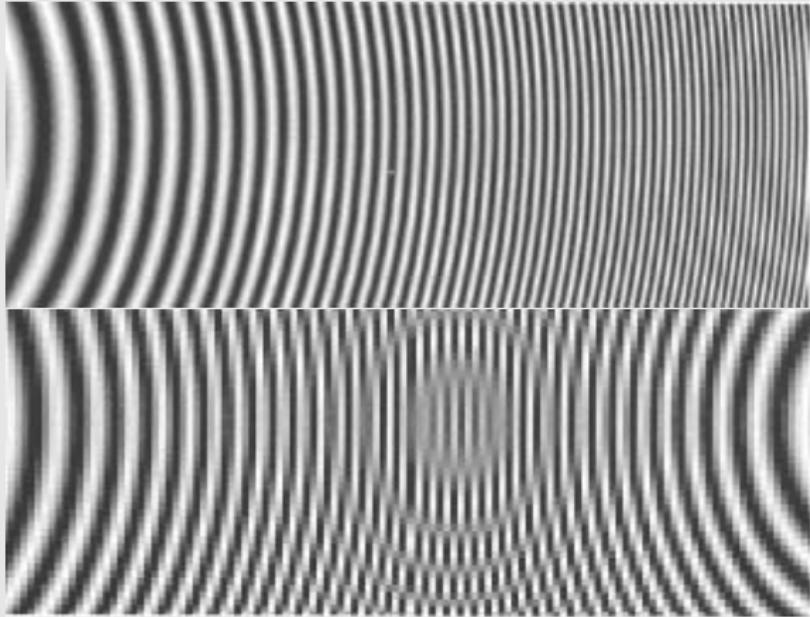
- ▶ Dieser Effekt heißt **aliasing** und tritt nicht nur beim Abtasten echter Signale auf sondern auch bei virtuellen (synthetischen) Signalen, z.B. beim **ray tracing**.
- ▶ Aliasing muß unbedingt vermieden werden, denn solche Bilddaten sind wegen der zufälligen Scheinfrequenzen **physikalisch sinnlos**, wenn sie auch graphisch gelegentlich sehr ansprechend sind.

Aliasing: Ein Beispiel

Ein 16×16 -Bild und vier unterabgetastete Bilder in den Größen 12×12 , 8×8 , 5×5 und 4×4



Scheinfrequenzen



Movie: Scheinfrequenzen

Movie: Rekonstruktionsfehler

Zur Entstehung von Scheinfrequenzen:

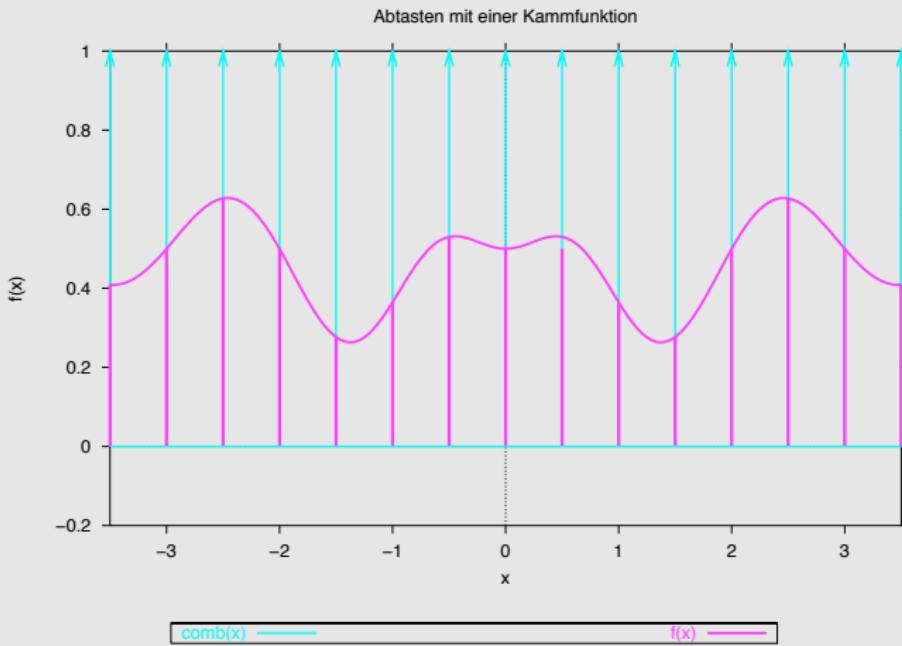
Wir können das Abtasten als ein **Produkt** der Bildfunktion f mit einer **Kammfunktion** modellieren:

- ▶ Die Nadelpulse der Kammfunktion geben die Abtaststellen an (Abstand x_0).
- ▶ Die Bildfunktion moduliert die Höhe der Nadelpulse.

Die diskrete Bildfunktion f_s ist dann:

$$f_s(x) = f(x) \sum_n \delta(x \cdot nx_0)$$

Die Kammfunktion



Abtasten als Faltung mit einer Kammfunktion

$$f_s = f(x) \sum_n \delta(x - nx_0)$$

$$f_s = \sum_n f(nx_0) \delta(x - nx_0)$$

$$F_s(u) = F(u) * \frac{1}{x_0} \sum_n \delta(u - \frac{n}{x_0})$$

wegen $\int_{-\infty}^{\infty} f(x) \delta(x - a) dx = f(a)$

$$F(u) * \delta(u - \frac{n}{x_0}) = F(u - \frac{n}{x_0})$$

$$F_s(u) = \frac{1}{x_0} \sum_n F(u - \frac{n}{x_0})$$

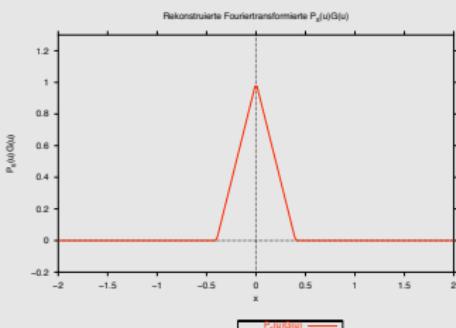
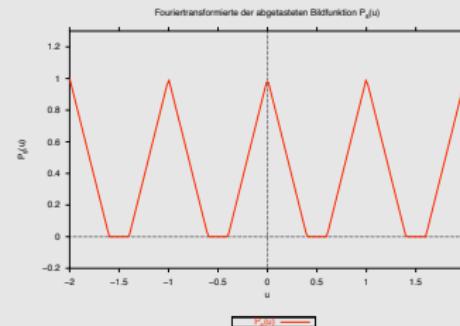
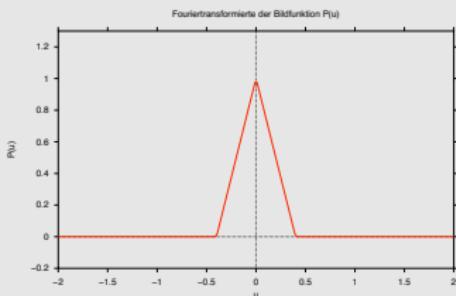
Fensterung der Fourier-Transformierten

- ☞ Abtasten mit einer Schrittweite von x_0 im Bild bedeutet, daß wir im Frequenzraum die Fouriertransformierte F im Abstand $\frac{1}{x_0}$ wiederholen.
- ☞ Wenn f bandbegrenzt ist mit Grenzfrequenz $\frac{1}{2x_0}$, dann überlappen sich diese wiederholten Teile in F_s nicht.
- ☞ F (und damit f) kann rekonstruiert werden, indem F_s mit einer geeigneten Fensterfunktion G multipliziert wird.[Ballard and Brown, 1982, S. 40]

$$G(u) = \begin{cases} 1 & |u| < \frac{1}{2x_0} \\ 0 & \text{sonst} \end{cases}$$

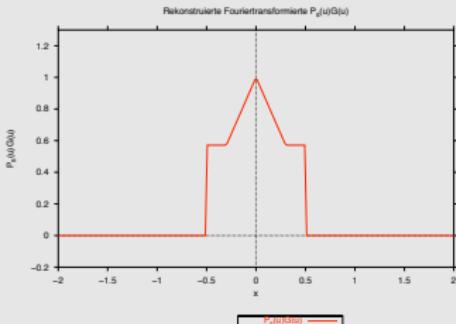
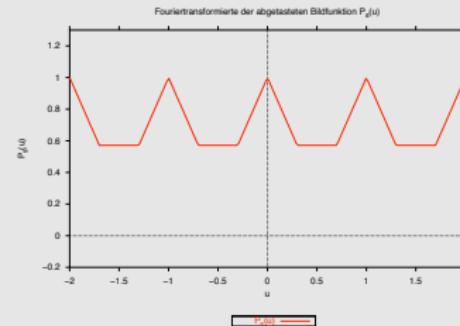
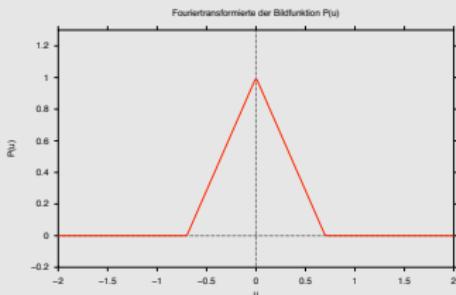
$$f(x) = \mathcal{F}^{-1}[F_s(u)G(u)]$$

Grenzfrequenz richtig



links oben: $P(u)$
recht oben: $P_s(u)$
links unten: rekonst. $P(u)$

Grenzfrequenz falsch



links oben: $P(u)$
recht oben: $P_s(u)$
links unten: rekonst. $P(u)$

Wie stark kann ein Signal variieren?

Eine bandbegrenzte Funktion kann nicht beliebig schnell variieren.

Bernsteins Ungleichung:

Ist $g(t)$ bandbeschränkt auf W , und $|g(t)| < M, \forall t$, so gilt:

$$\Rightarrow |g'(t)| \leq 2\pi WM$$

d.h. die erste Ableitung des Signals ist ebenfalls beschränkt .

Variation eines TV-Signals

Beispiel

Typisches TV-Signal 5 MHz

Bandbreite W : 5 MHz

Abstand zweier Abtastwerte: $\frac{1}{2W} = \frac{1}{2 \cdot 5 \text{MHz}} = 100 \text{ ns}$

Helligkeitsbereich:

$$g(t) = -128 \dots 128$$

$$|g'(t)| \leq 2\pi \cdot 5 \cdot 5 \cdot \text{MHz} \cdot 128 = 4021.24 \text{MHz}$$

Differenz zwischen Abtastwerten Δg :

$$|\Delta g| \leq |g'(t)| \cdot 100 \text{ns} \approx 400$$

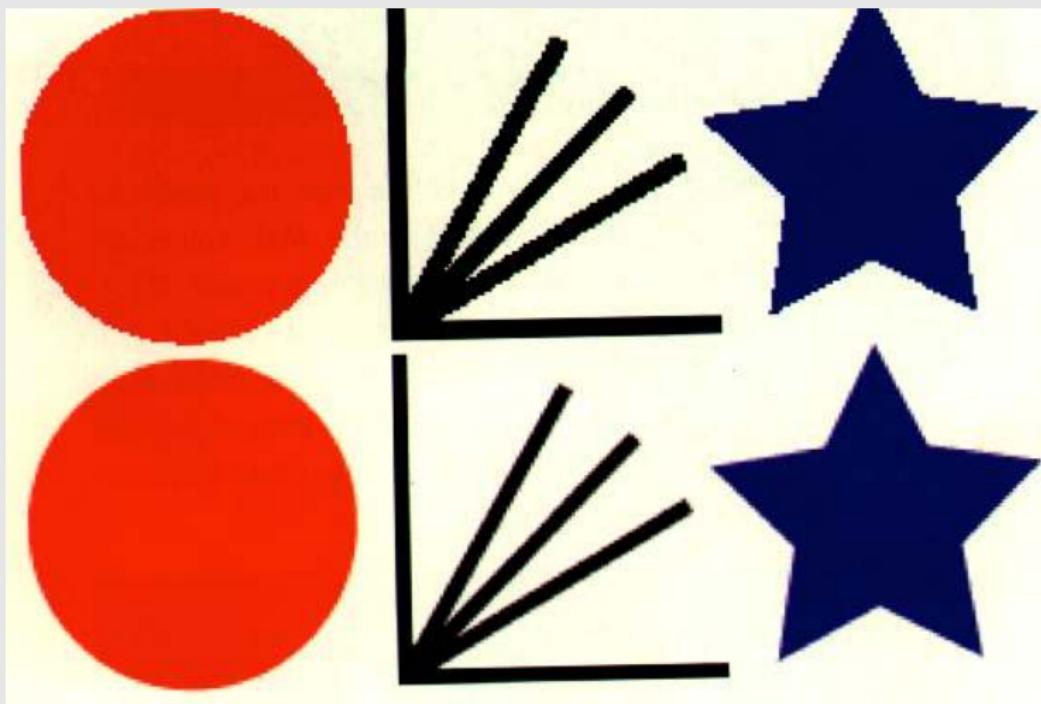
Antialiasing bei synthetischen Bildern

Synthetische Bilder verletzen oft das Abtasttheorem, indem sie steile Helligkeitssprünge enthalten, die bei natürlichen, bandbegrenzten Signalen nicht möglich wären. Dadurch erscheinen die Konturen von grafischen Objekten "zackig" und unnatürlich.

Antialiasing mildert Stufen "jaggies" bei gerasterten Konturen durch **Tiefpaßfilterung**:

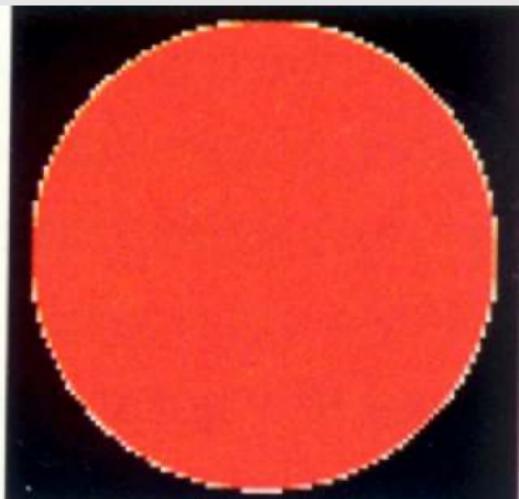
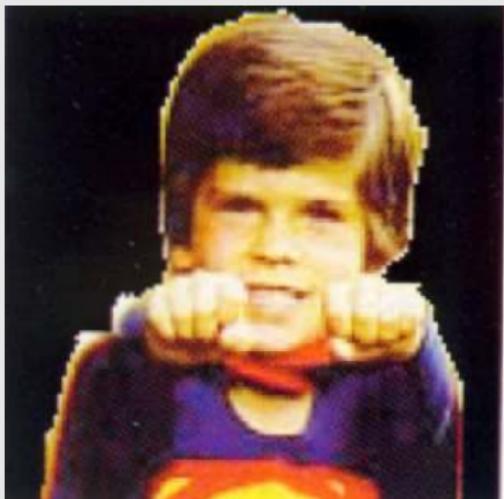
- ▶ Tiefpaßfilterung verringert die Grenzfrequenz und damit die Steilheit der Kante.
- ▶ Tiefpaßfilterung führt zu Zwischentönen; die Zacken im Rand einer synthetischen Kontur erscheinen glatter.

Beispiel: „Jaggies“ und Antialiasing

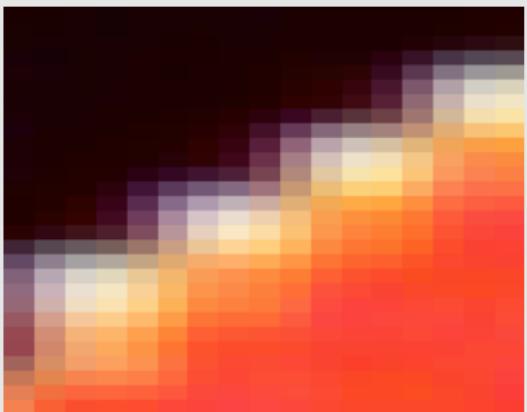
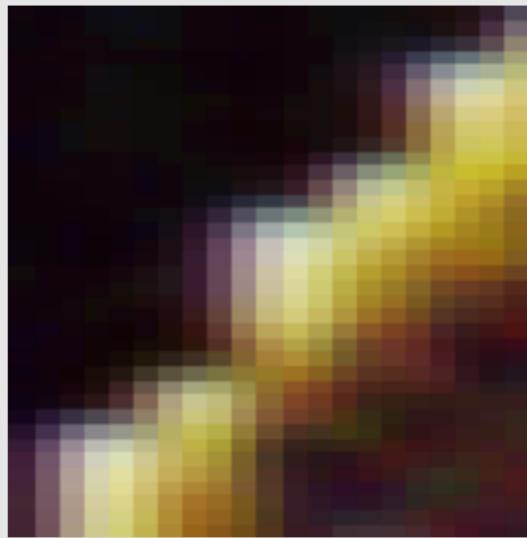


Halos: Falsches Antialiasing

Wenn nach dem Antialiasing der Hintergrund gewechselt wird, entstehen häßliche Halos:



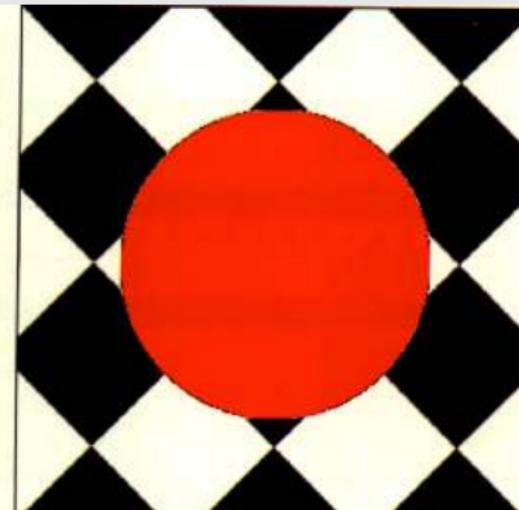
Ausschnitte aus den Halos



Antialiasing bei strukturiertem Hintergrund



Korrektes Antialiasing



falsches Antialiasing

Aliasing bei Blue Screen Matting

Korrekte Antialiasing bei Blue Screen Matting

Abtasten in der Zeit: Animationen, Filme



Definition (Bildfolgen)

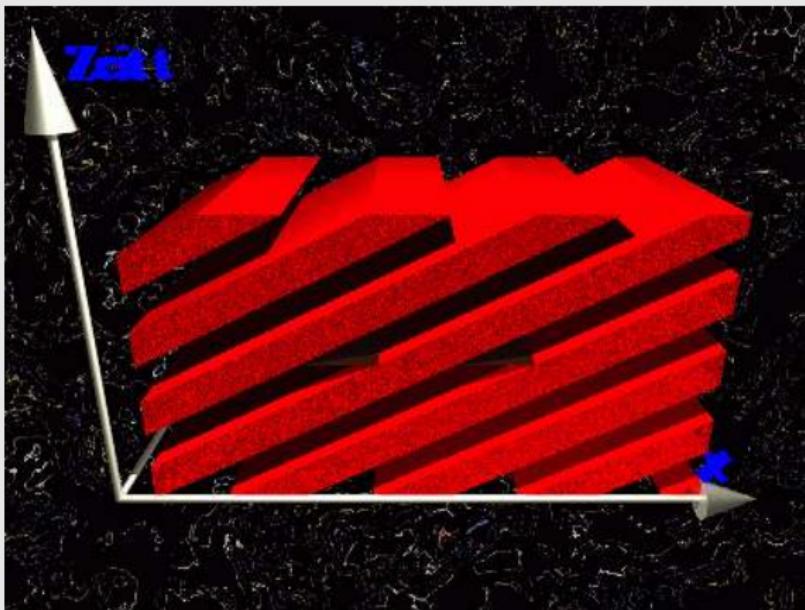
sind Folgen von Bildern (Filme), die durch eine zeitliche Abtastung einer Szene oder Synthetisieren einer Szene (Animationen) zu mehreren Zeitpunkten entstehen.

- ☞ Auch beim Abtasten in der Zeit ist das Abtasttheorem unbedingt zu beachten.
Sonst kann aliasing zu **Scheinbewegungen** führen.
- ☞ Bei zu geringer Abtastfrequenz können sich Bewegungen umkehren oder Objekte scheinbar stillstehen.

Verschiebung eines periodischen Musters

Movie: Verschiebung eines periodischen Musters

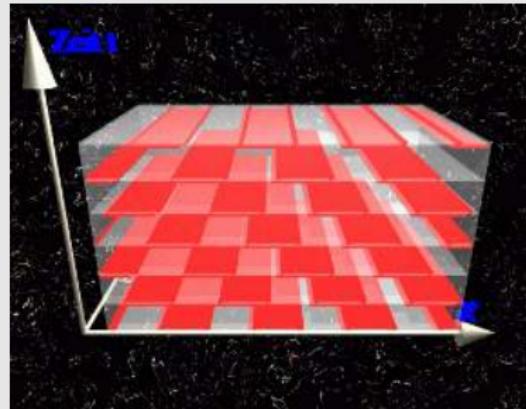
Ein zeitveränderliches analoges Bildsignal



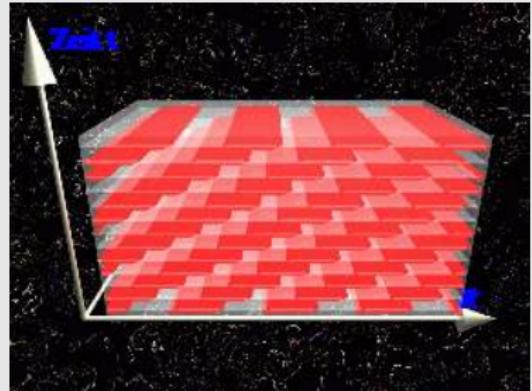
Bildfolge als 3D-Bild

- ▶ Eine Bildfolge kann als dreidimensionales Bild betrachtet werden, das neben den Zeilen- und Spaltenkoordinaten noch eine zusätzliche Zeitkoordinate hat.
- ▶ Ein bewegtes Streifenmuster bildet im dreidimensionalen Orts-Zeitraum eine periodische Folge von gescherten Quadern.
Die Scherung ist umso stärker, je schneller sich das Muster in der Bildebene verschiebt.
- ▶ Die Bewegungsgeschwindigkeit bestimmt, mit welcher Frequenz in der Zeit abgetastet werden muß.

Falsche und korrekte Abtastung

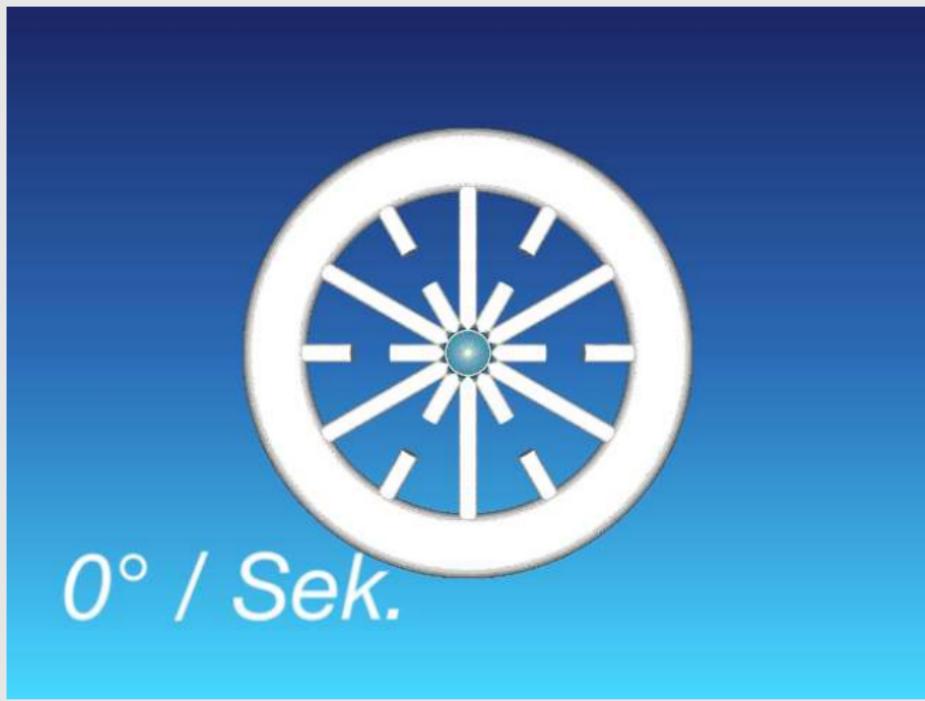


Falsche Abtastung



korrekte Abtastung

Die „broken wheel illusion“



Die „broken wheel illusion“

Movie: broken wheel illusion

2D-Fourier-Transformation

x, y = Orts **koordinaten**

f_x, f_y = Orts **frequenzen**

$\lambda = \frac{c}{f}$ = Orts **wellenlängen**

$$\begin{aligned} G(f_x, f_y) &= \mathcal{F}\{g(x, y)\} \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y) e^{-2\pi i(f_x x + f_y y)} dx dy \end{aligned}$$

$$\begin{aligned} g(x, y) &= \mathcal{F}^{-1}\{G(f_x, f_y)\} \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} G(f_x, f_y) e^{2\pi i(f_x x + f_y y)} df_x df_y \end{aligned}$$

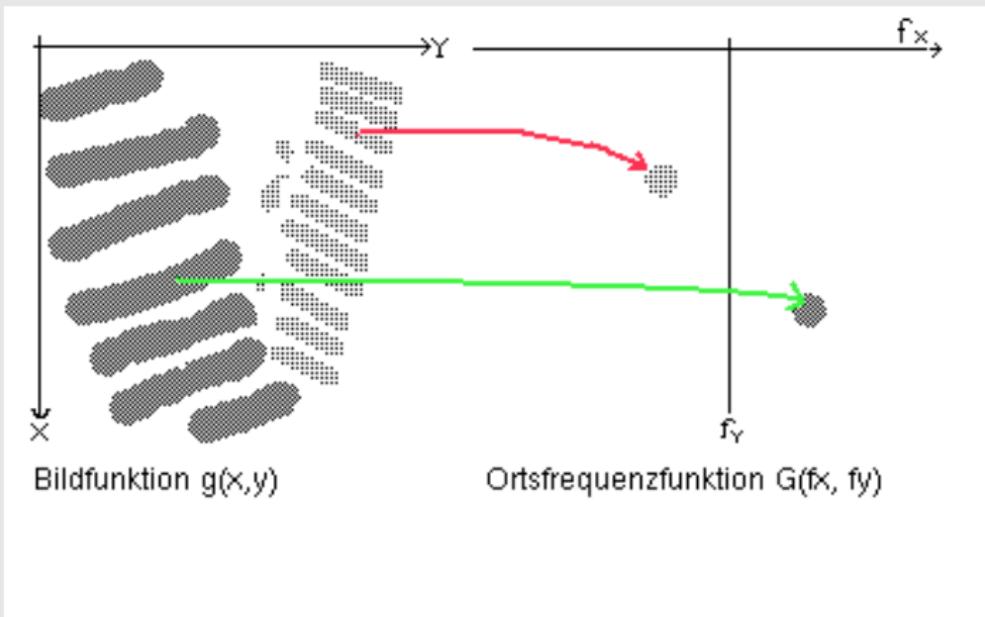
Interpretation des Spektrums

Bildfunktion: $g(x, y)$

Ortsfrequenzfunktion: $G(f_x, f_y)$

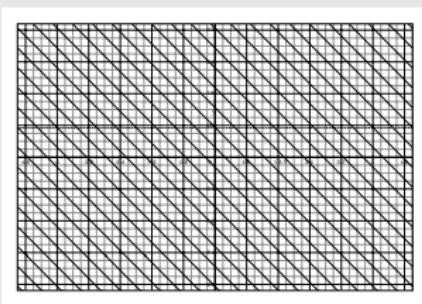
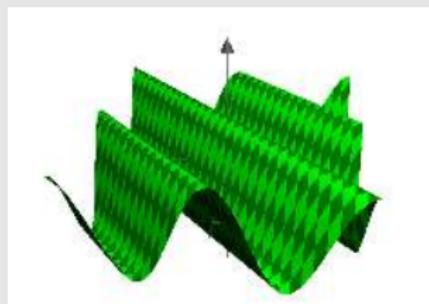
Interpretation: Der Funktionswert an der Stelle (f_x, f_y) gibt
Phase und **Betrag** der sinusförmigen
Grauwertkomponente in $g(x, y)$ mit Frequenz
in x-Richtung f_x und Frequenz in y-Richtung f_y
an.

Ein schematisches Beispiel



Die Ortsfrequenzen:

Als Beispiel wollen wir die Funktion $\cos(2\pi(u_x + v_y))$ näher betrachten:



- ▶ Die Nullstellen dieser Funktion sind parallele Geraden mit dem Abstand $1/\sqrt{u^2 + v^2}$.
- ▶ Die Geraden schneiden die y-Achse in Abständen von $1/v$ und die X-Achse in Abständen von $1/u$. Wegen dieser zyklischen Veränderungen in X- und Y-Richtung werden u und v **Ortsfrequenzen** genannt.

Eigenschaften der Fourier-Transformation

Linearität:

$$\begin{aligned}\mathcal{F}\{a \cdot g_1(x, y) + b \cdot g_2(x, y)\} \\ = a \cdot \mathcal{F}\{g_1(x, y)\} + b \cdot \mathcal{F}\{g_2(x, y)\}\end{aligned}$$

Skalierung:

$$\mathcal{F}\{g(cx, dy)\} = \frac{1}{|cd|} \cdot G\left(\frac{f_x}{c}, \frac{f_y}{d}\right)$$

Verschiebung:

$$\mathcal{F}\{g(x - x_0, y - y_0)\} = G(f_x, f_y) \cdot \underbrace{e^{-2\pi i(f_x x_0 + f_y y_0)}}_{\text{Betrag} = 1}$$

Das Faltungstheorem

Satz (Faltungstheorem)

$$\begin{aligned}\mathcal{F} \left\{ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g_1(\xi, \eta) \cdot g_2(\textcolor{red}{x} - \xi, \textcolor{blue}{y} - \eta) d\xi d\eta \right\} = \\ \mathcal{F}\{g_1(\textcolor{red}{x}, \textcolor{blue}{y})\} \cdot \mathcal{F}\{g_2(\textcolor{red}{x}, \textcolor{blue}{y})\} = G_1(f_x, f_y) \cdot G_2(f_x, f_y)\end{aligned}$$

Diese Eigenschaften sind eine wichtige Grundlage für schnelle Bildverarbeitungsalgorithmen.

Gängige Bezeichnungen

Frequenzspektrum von $g(x, y)$: $G(f_x, f_y)$,
die 2D-Fourier-Transformierte von $g(x, y)$.

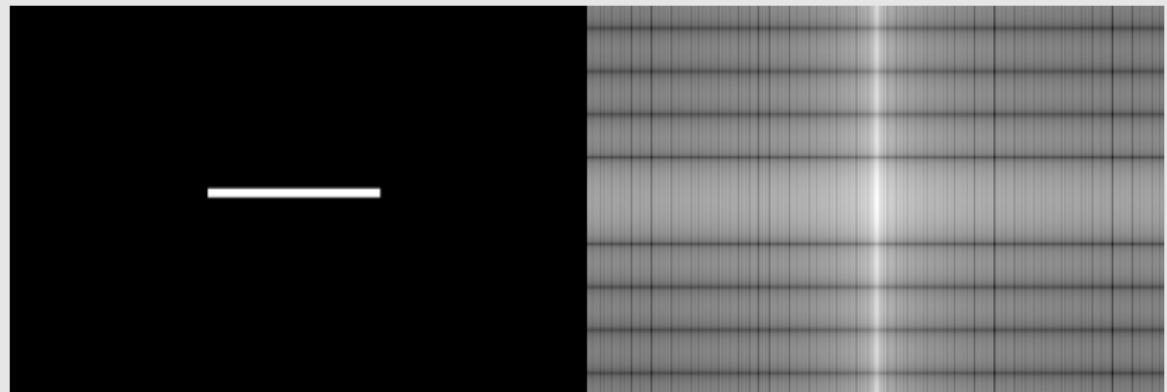
Betragsspektrum bzw. Amplitudenspektrum) von $g(x, y)$:

$$|G(f_x, f_y)| = \sqrt{\operatorname{Re}[G(f_x, f_y)]^2 + \operatorname{Im}[G(f_x, f_y)]^2}$$

Energiespektrum bzw. Powerspektrum von $g(x, y)$:

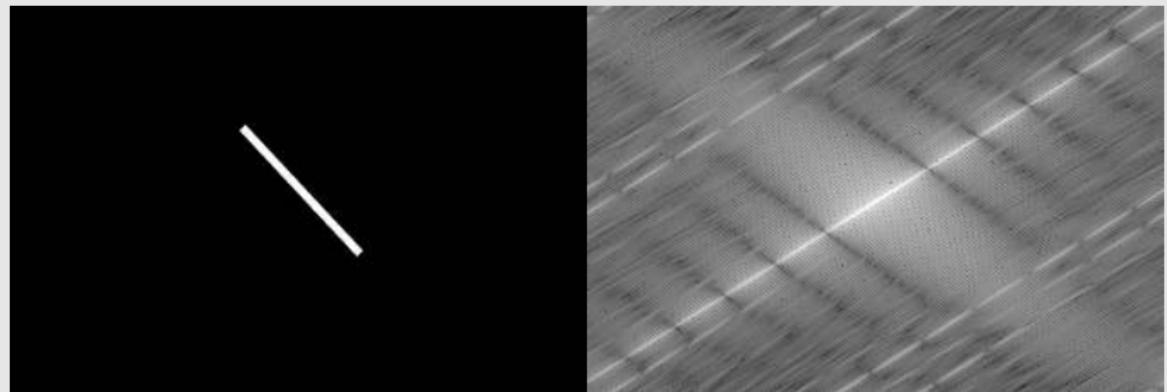
$$|G(f_x, f_y)|^2 = G(f_x, f_y) \cdot G^*(f_x, f_y)$$

Rotationseigenschaften der Fouriertransformation I



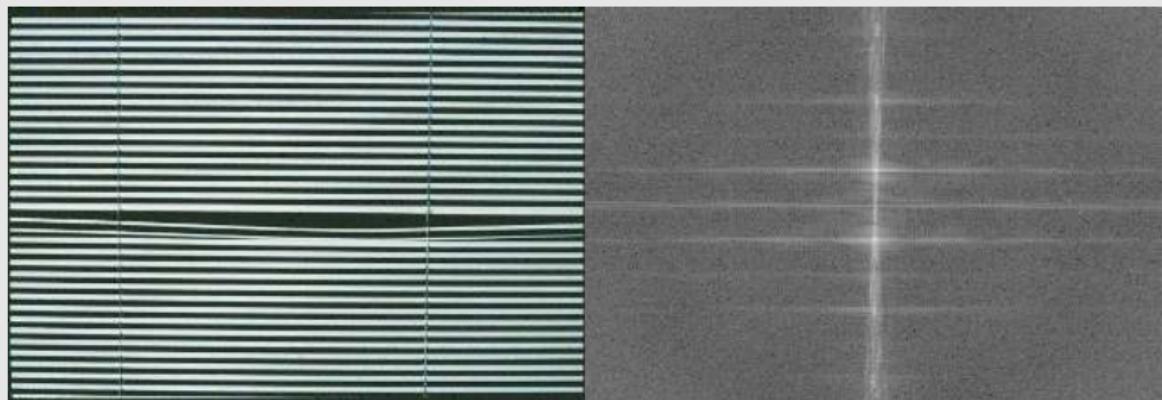
Ein einfaches Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Rotationseigenschaften der Fouriertransformation II



Das rotierte Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Weitere Beispiele: Jalousie



Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Weitere Beispiele: Zaun



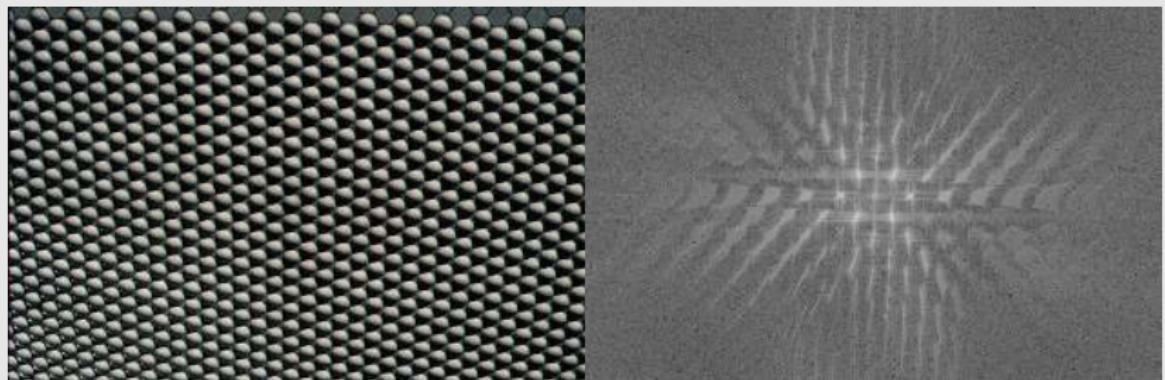
Grauwertbild und das dazugehörige Powerspektrum
(logarithmiert).

Weitere Beispiele: Treppe



Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Weitere Beispiele: Gitter

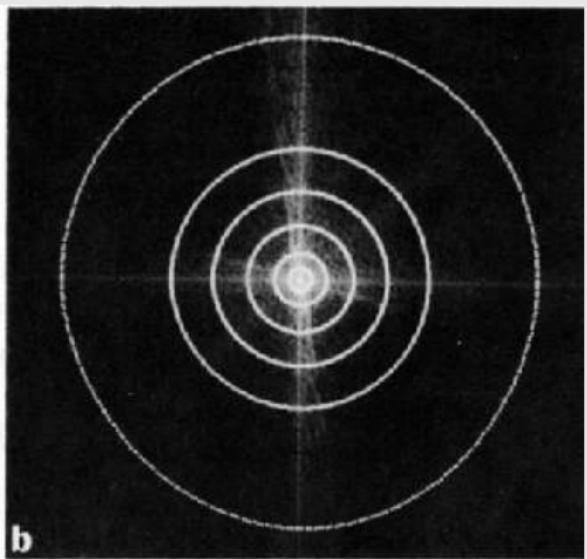


Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

256×256 Grauwertbild und Fourierspektrum

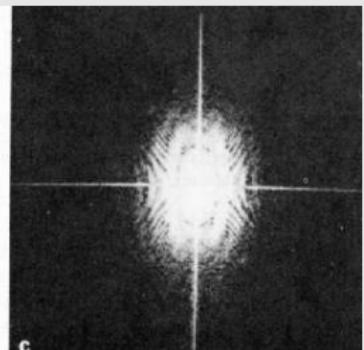
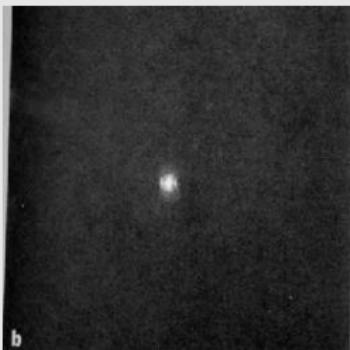
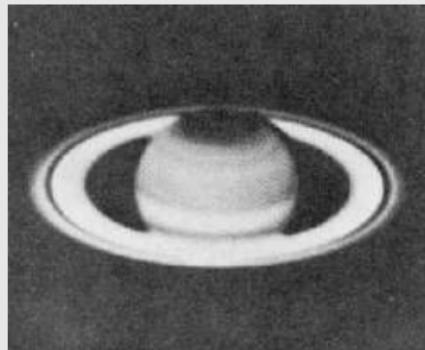


a



Die Kreise (Radien 5, 11, 22, 36, 53 und 98) umfassen jeweils 90,95, 98, 99, 99.5 und 99,9 % der Bildenergie.

Beispiel II: Saturn



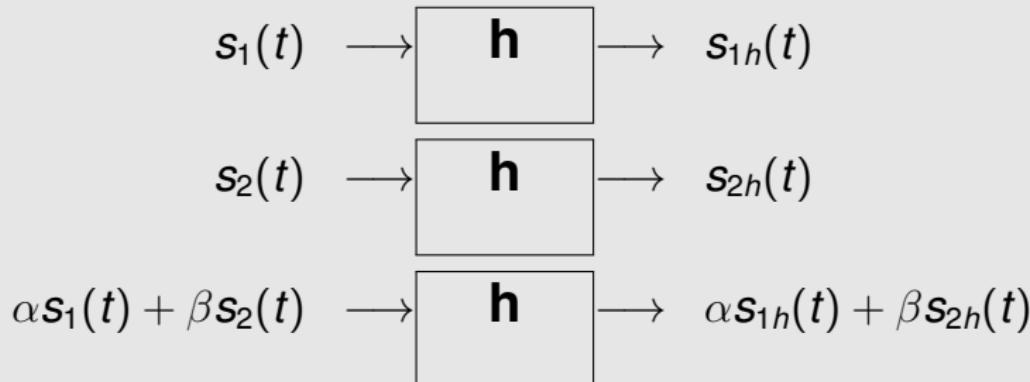
- a) Ein Bild des Planeten Saturn,
- b) Das Fourierspektrum: $|F(u, v)|$
- c) Das Fourierspektrum kontrastentzerrt:
 $\log(1 + |F(u, v)|)$

Lineares Übertragungsverhalten von Filtern

Aus der **Nachrichtentechnik**:

Viele Bausteine zur Signalübertragung zeigen (in Grenzen) **lineares** Übertragungsverhalten:

- ▶ Sei h ein Filter und seien $s_1(t), s_2(t)$ Signale am Eingang des Filters zum Zeitpunkt t ,
- ▶ $s_{1h}(t), s_{2h}(t)$: Ausgang des Filters zum Zeitpunkt t :



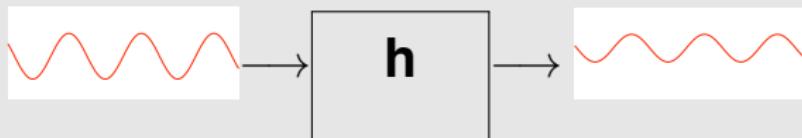
Übertragungsfunktion H(F)

Definition (Übertragungsfunktion H(F))

Ein Filter wird durch seine Übertragungsfunktion H(F) gekennzeichnet.

Diese gibt an, was aus **sinusförmigen** Einzelsignalen wird im Hinblick auf

- ▶ Phasenverschiebung und
- ▶ Dämpfung.



Allgemeine Signale, Frequenzgemische

Wegen der Linearität gilt für Frequenzgemische:
im Frequenzbereich:

$$S_h(f) = H(f) \cdot S(f)$$

im Zeitbereich:

$$s_h(t) = \underbrace{\int_{-\infty}^{\infty} h(t - \tau) \cdot s(\tau) d\tau}_{\text{Faltungsintegral, "convolution"}}$$

$$= h(t) * s(t)$$

Filter werden charakterisiert durch:

- ☞ Übertragungsfunktion $H(f)$, „Filterfunktion“
- ☞ Impulsantwort $h(t)$

$$\int_{-\infty}^{\infty} h(t - \tau) \cdot \delta(\tau) d\tau = h(t)$$

- ☞ δ = Impuls = Delta-Funktion = Dirac-Stoß

$$\delta(t) = \begin{cases} \infty & \text{wenn } t = 0 \\ 0 & \text{andernfalls} \end{cases}$$

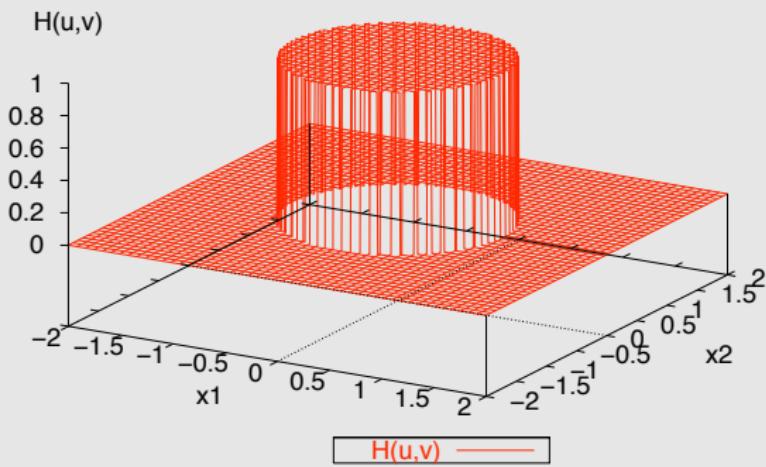
Bildverarbeitung im Frequenzraum

Filter werden in der Bildverarbeitung eingesetzt zur

- ▶ gezielten Modifikation des Spektrums
 - ▶ Hochpaß,
 - ▶ Tiefpaß,
 - ▶ Bandpaßfilterung
- ▶ modellbasierte Transformation (Wiener Filterung)
- ▶ Effizienten Berechnung von Transformationen

Ein optimaler Tiefpaß

Optimaler Tiefpaß, Grenzfrequenz 1



Tiefpaßfilter

Tiefpaßfilter:

sperren oder dämpfen hohe Frequenzen.

- ▶ Sie werden verwendet, um Bilder zu glätten
(Rauschunterdrückung, Weichzeichnung)
- ▶ oder um eine bestimmte Grenzfrequenz
sicherzustellen, z.B. vor einer Unterabtastung.
- ▶ Die vorherige Folie zeigt ein **optimales Tiefpaßfilter**:
Alle Frequenzen jenseits der Grenzfrequenz werden
vollständig gesperrt, alle anderen werden vollständig
durchgelassen.

Beispiel: Optimales Tiefpaßfilter I



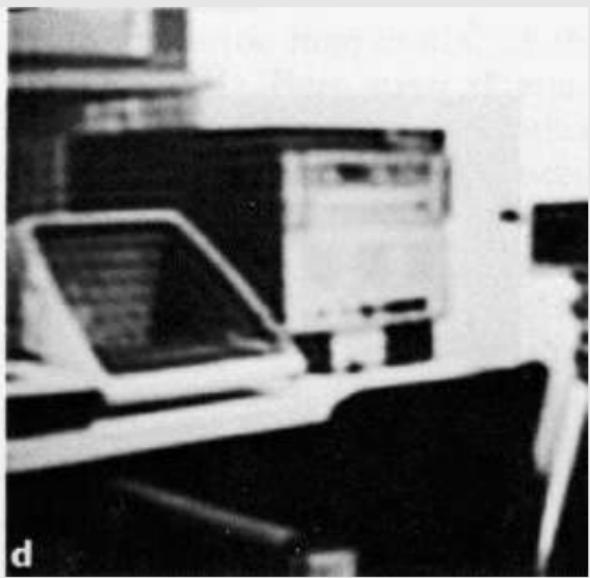
e



f

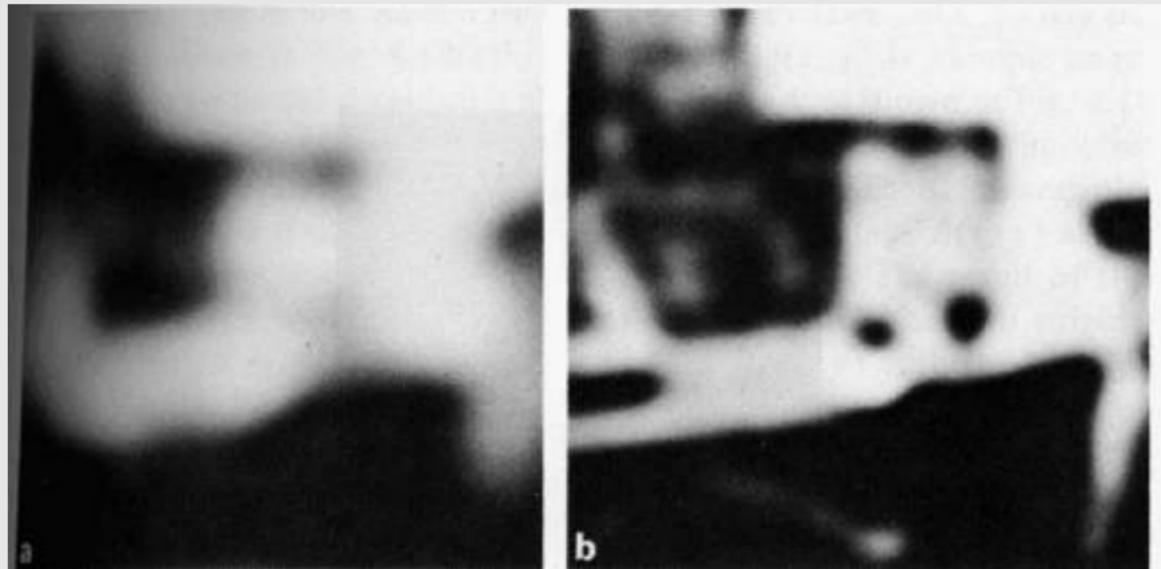
- links) Hohe Grenzfrequenz, leichte Weichzeichnung
rechts) Originalbild

Beispiel: Optimales Tiefpaßfilter II



Filterung bei mittleren Grenzfrequenzen, merklicher Weichzeichnereffekt, deutliches **ringing**.

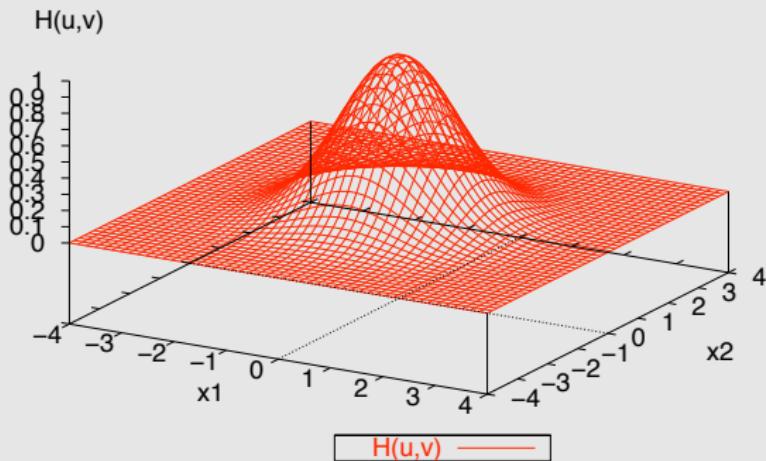
Beispiel: Optimales Tiefpaßfilter III



Sehr niedrige Grenzfrequenz, übertriebene Unschärfe

Exponential-Tiefpaß

Exponentaltiefpaß



Filter mit glattem Verlauf mildern das **ringing**,
z.B. Exponentialfilter, Butterworth-Filter.

Beispiel: Exponential-Tiefpaßfilter



Filterung bei mittlerer Grenzfrequenz, das **ringing** wird vermieden.

Beispiel: Exponential-Tiefpaßfilter II



Filterung bei sehr niedriger Grenzfrequenz, deutliche
Unschärfe.

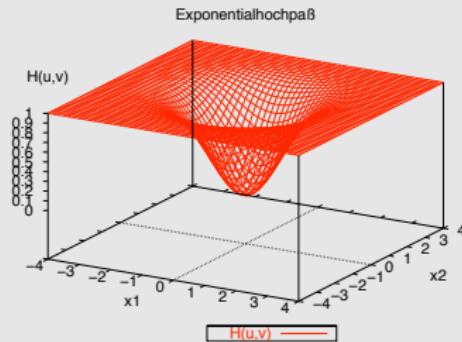
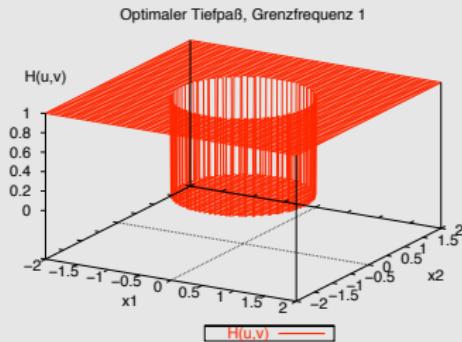
Echtzeitanimation: Gauß-Filter

Movie: Gauß-Filter

Echtzeitanimation: Gauß-Filter

Movie: Gauß-Filter

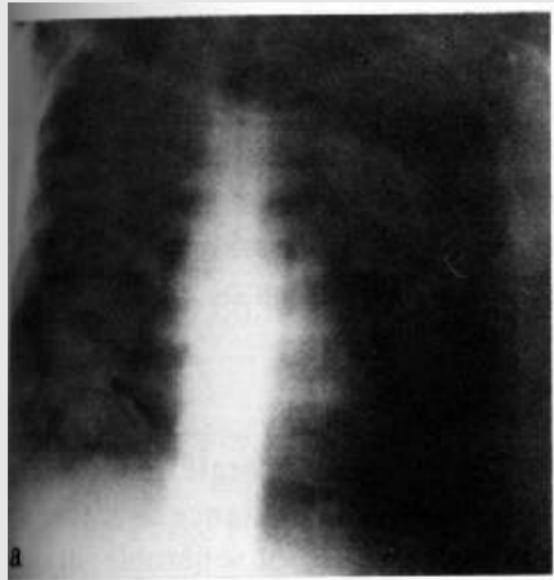
Hochpaßfilter



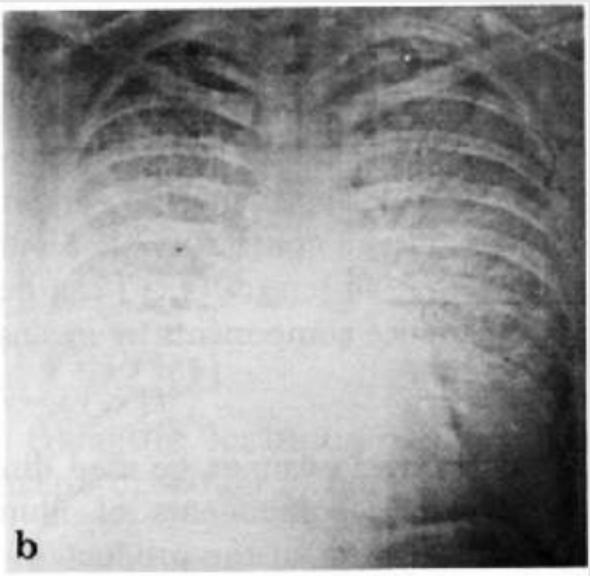
Hochpaßfilter

- ▶ unterdrücken oder dämpfen niedrige Frequenzen.
- ▶ Sie lassen Bilder schärfer erscheinen.
- ▶ Grautöne in glatten Bildbereichen erscheinen ausgewaschen.

Beispiel: Hochpaßfilterung I

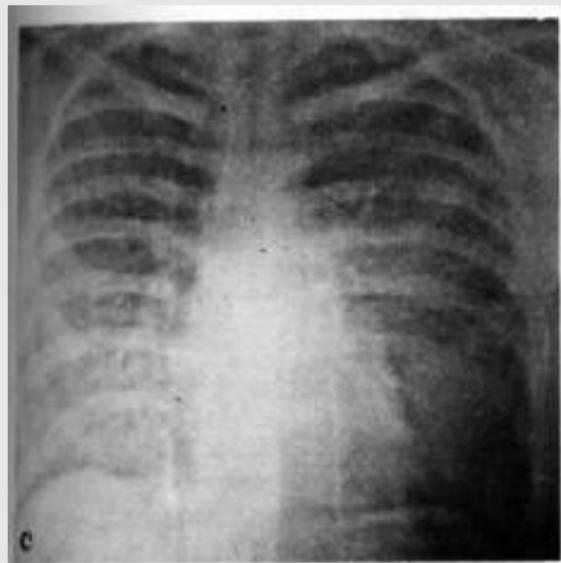


a) Originalbild

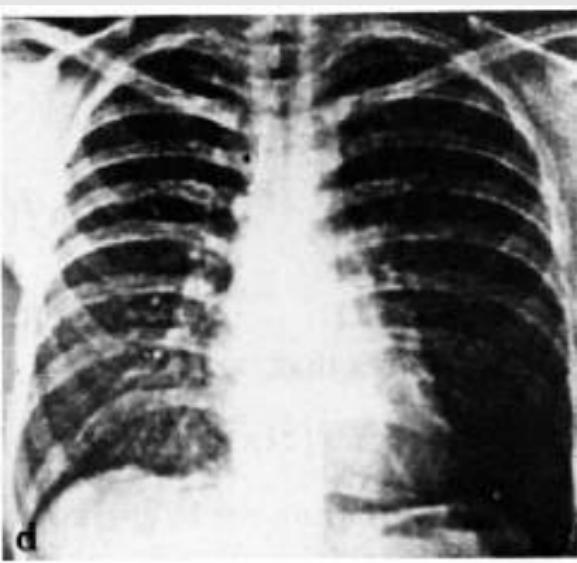


b) Hochpaß-Exponentialfilter

Hochpaßfilterung II



c) High frequency emphasis filter
kontrastentzerrt

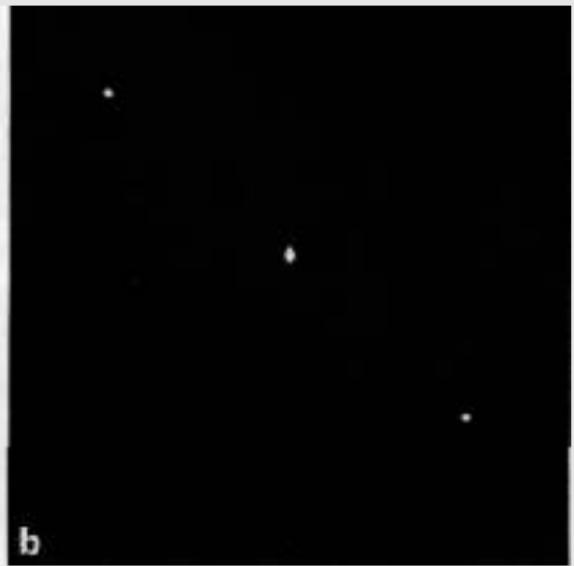
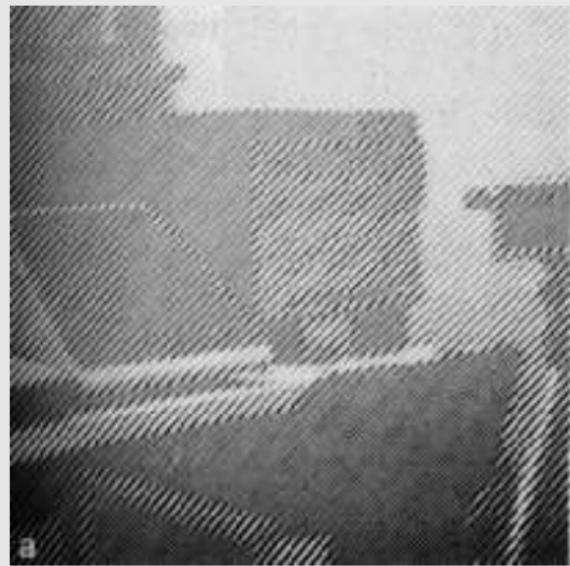


d) wie c), aber

Bandsperre zur Bildrestauration

- ▶ Beseitigung von additiven Störsignalen (Interferenzen): Hierzu eignen sich **Bandsperren** (Bandstoppfilter), das sind Filter, die ausgewählte Frequenzbereiche unterdrücken oder dämpfen.
- ▶ Rücktransformation von systematischen Bildveränderungen (Wiener Filterung)

Beispiel: Sinusförmige Interferenz und das Powerspektrum



Bildrestauration durch Bandsperre



Eine Bandsperre unterdrückt die Störfrequenz.

Wiener Filterung: Bewegungsunschärfe

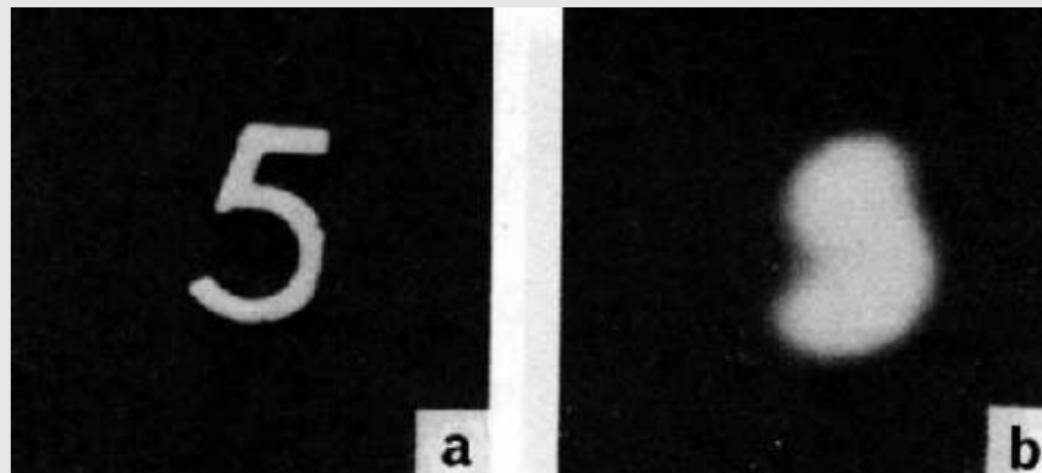


a



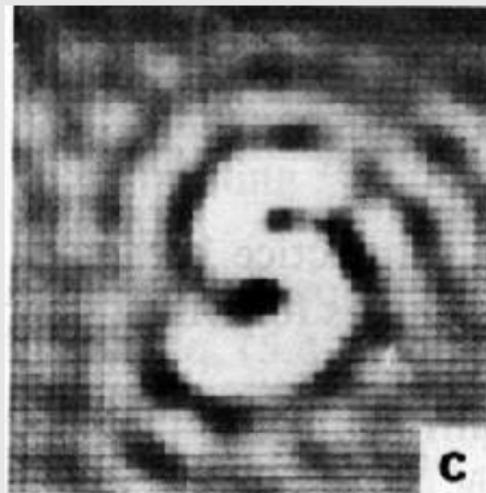
Originalbild und mit inverser Wiener Filterung restaurierte Version

Wiener Filterung: Fokussierungsunschärfe I

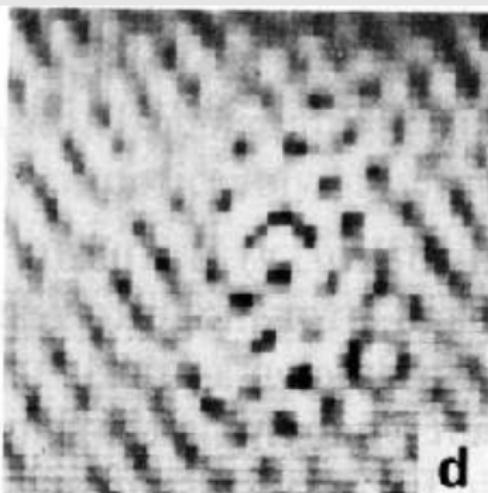


Originalbild und unscharfe Version

Fokussierungsunschärfe II



c



d

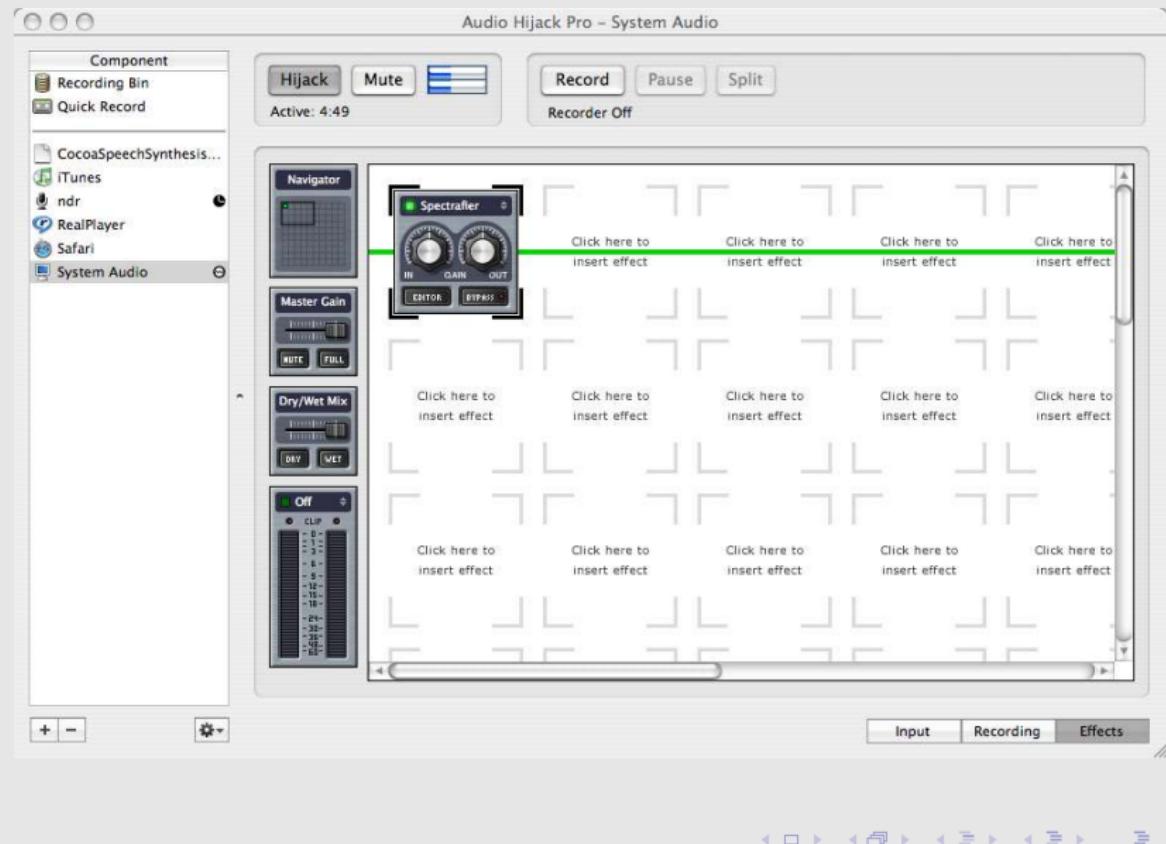
Restaurierte Bilder mit unterschiedlichen Modellen der Unschärfe

Audio-Filter

Auch für Audio-Signale können Filter eingesetzt werden:

- ▶ Tiefpaßfilter zur Betonung der Bässe.
- ▶ Hochpaßfilter zur Verstärkung der Höhen.
- ▶ Bandsperren zur Unterdrückung von Störgeräuschen.

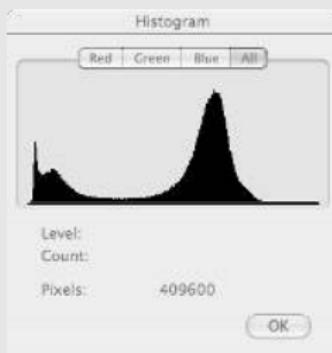
Beispiel: Filterung in Audio-Hijack



Teil XIII

Ikonische Algorithmen

Ikonische Bildverarbeitung



34

Globale Bildeigenschaften

- Statistische Merkmale

- Empirischer Mittelwert und Varianz
- Rauschen
- Einfache Rauschfilter
- Grauwerthistogramme

- Profile und Projektionen

35

Geometrische Operationen auf Bildern

36

Lokale Bildeigenschaften

Ikonische Algorithmen

Ikonische Algorithmen messen oder verändern **globale** Bildeigenschaften.

Photometrische Operationen: Kontrastverbesserung,
Schärfen, Glätten, Falschfarben

Geometrische Operationen: Vergrößern, Verkleinern,
Rotieren, Entzerren

Zwei Klassen von Operationen:

- ① im Ortsraum
- ② im Frequenzraum

Ikonische Bildverarbeitung (image processing)

- ▶ Die ikonische Bildverarbeitung ist weitgehend standardisiert (Din-Norm).
- ▶ BV-Geräte sind kommerziell erhältlich und werden industriell und genutzt.
- ▶ BV-Software ist für Hobby-Fotografen erhältlich.
- ▶ Ikonische Bildverarbeitungsalgorithmen werden oft als Hardware-Lösung implementiert.

Globale Bildeigenschaften

- ▶ Globale Bildeigenschaften beschreiben ein Bild als Ganzes, nicht die einzelnen Komponenten eines Bildes.
- ▶ Globale Bildeigenschaften werden häufig von Algorithmen zur Bildaufbereitung eingesetzt, die eine weitergehende Analyse vorbereiten.
 - ▶ Empirischer Mittelwert und Varianz
 - ▶ Histogramme
 - ▶ Projektionen
 - ▶ Profile
 - ▶ Frequenzspektrum

Empirischer Mittelwert

Definition (Empirischer Mittelwert)

Der empirische Mittelwert \bar{g} eines Bildes mit $M \times N$ Pixeln ist der Mittelwert der Grauwerte oder Farbwerte g_{mn} der Pixel eines Bildes.

$$\bar{g} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g_{mn}$$

Vereinfachte Notation: Die K Pixel eines Bildes werden linear aufgezählt:

$$\bar{g} = \frac{1}{K} \sum_{k=0}^{K-1} g_k$$

Empirische Varianz

Definition (Empirische Varianz)

Die empirische Varianz σ^2 eines Bildes mit K Pixeln ist der Mittelwert der quadrierten Abweichung der Grauwerte oder Farbwerte g_k der Pixel eines Bildes vom empirischen Mittelwert \bar{g} .

In vereinfachter Notation:

$$\sigma^2 = \frac{1}{K} \sum_{k=0}^{K-1} (g_k - \bar{g})^2$$

Inkrementelle Berechnung

Empirischer Mittelwert \bar{g}

$$\bar{g}_0 = 0$$

$$\bar{g}_k = \frac{\bar{g}_{k-1}(k-1) + g_k}{k}, k = 2 \dots K$$

Inkrementelle Berechnung

Empirischer Mittelwert \bar{g}

$$\bar{g}_0 = 0$$

$$\bar{g}_k = \frac{\bar{g}_{k-1}(k-1) + g_k}{k}, k = 2 \dots K$$

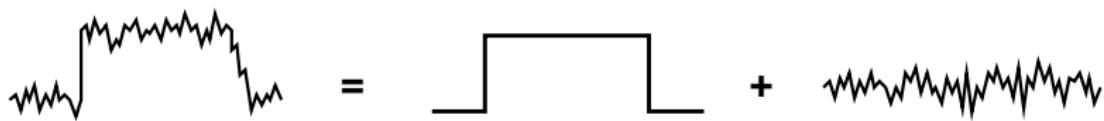
Empirische Varianz σ^2

$$\sigma_0^2 = 0$$

$$\begin{aligned}\sigma_k^2 &= \frac{(\sigma_{k-1}^2 + \bar{g}_{k-1}^2)(k-1) + g_k^2}{k} \\ &\quad - \left(\frac{\bar{g}_{k-1}(k-1) + g_k}{k} \right)^2, k = 2 \dots K\end{aligned}$$

Rauschen

- ▶ Ein Bildsignal, das mit einem Sensor aufgenommen wurde, zeigt stets zufällige Schwankungen - **das Rauschen**.
- ▶ Rauschen entsteht bei der Aufnahme des Signals im Verstärker oder bei der Signalübertragung.
- ▶ Das Rauschen kann oft als additives „weißes“ Rauschen modelliert werden - das Bild wird dann als Summe aus dem eigentlichen, idealen Bildsignal und einer überlagerten Rauschkomponente betrachtet.



Typische Eigenschaften weißen Rauschens

Weißes Rauschen ist

Normalverteilt: mit Mittelwert $\mu = 0$ und Varianz $\sigma^2 > 0$.

Wahrscheinlichkeitsdichte: $p(r)$

$$p(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{r^2}{2\sigma^2}}$$

Räumlich unkorreliert:

$$E[r_{ij}r_{mn}] = 0, \text{ für } ij \neq mn$$

Zeitlich unkorreliert:

$$E[r_{ij,t_1}r_{ij,t_2}] = 0, \text{ für } t_1 \neq t_2$$

Signal/Rauschverhältnis

Rauschen ist insbesondere bei schlechten Lichtverhältnissen problematisch, weil dann das Signal/Rauschverhältnis ungünstig ist, beispielsweise

- ▶ im Schatten,
- ▶ bei Nacht,
- ▶ in schlecht beleuchteten Gebäuden.

Wasserloch bei Nacht: Stark verrauscht

Stark vergrößerter Ausschnitt

Beispiel



Rauschbeseitigung durch Mitteln

Weißes Rauschen kann durch Mitteln beseitigt werden:

- ☞ **Idee:** Wenn der Mittelwert des Rauschens gegen Null geht, geht der Mittelwert der gemessenen Grauwerte gegen den wahren Grauwert.

$$\hat{r}_k = \frac{1}{k} \sum_{i=1}^k r_i \rightarrow 0$$

Es gibt zwei Möglichkeiten, Rauschen durch Mitteln zu glätten:

Räumliches Mitteln: Mittele die Grauwerte g_{mn} in der räumlichen Nachbarschaft von g_{ij} , sofern $g_{ij} \approx g_{mn}$

Zeitliches Mitteln: Mittele mehrere Stichproben des Pixels $g_{ij,t}$ zu verschiedenen Zeitpunkten $t = 1 \dots T$.

Unschärfe durch Mitteln

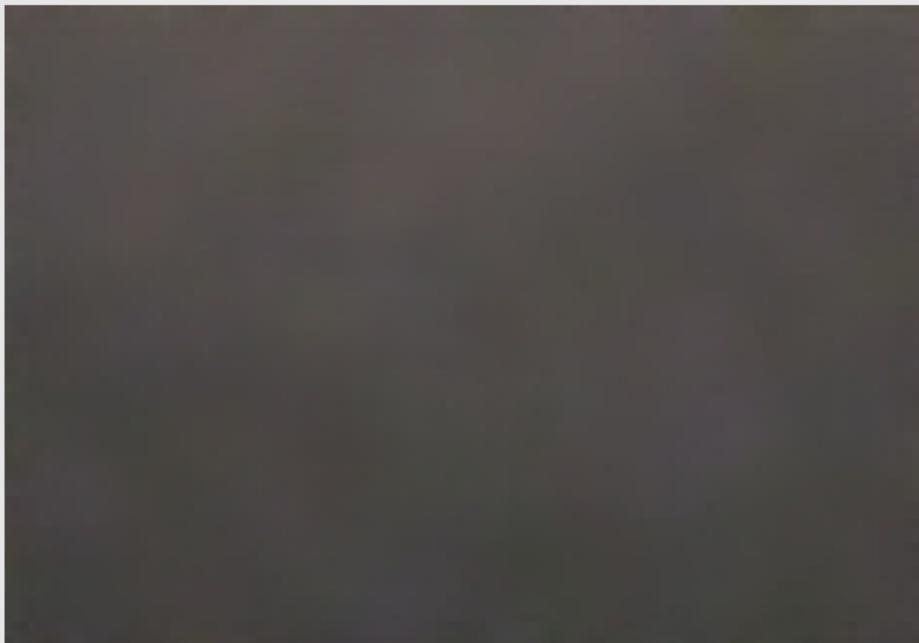
Beachte: Durch Mitteln entsteht Unschärfe:

Mitteln wirkt wie ein Tiefpaßfilter:

- ▶ Räumliches Mitteln verwischt Kanten.
- ▶ Zeitliches Mitteln führt bei bewegten Objekten zu Bewegungsunschärfe.

Gemittelter Ausschnitt

Der Ausschnitt von Beispiel 88 stark gemittelt (10×10).



Wie stark muß gemittelt werden?

Der geschätzte Rauschwert \hat{r}_k nach k Mittelungsschritten ist eine Zufallsvariable, deren Mittelwert und Varianz von k abhängt.

$$\hat{r}_k = \frac{1}{k} \sum_{i=1}^k r_i$$

$$E[\hat{r}_k] = \frac{1}{k} \sum_{i=1}^k E[r_i] = 0 \text{ Mittelwert}$$

$$\begin{aligned} E[(\hat{r}_k - E[\hat{r}_k])^2] &= E[\hat{r}_k^2] = E\left[\frac{1}{k^2}\left(\sum_{i=1}^k r_i\right)^2\right] \text{ Varianz} \\ &= \frac{1}{k^2} \sum_{i=1}^k E[r_k^2] = \frac{\sigma^2}{k} \end{aligned}$$

Beispiel

Um die Standardabweichung des Rauschens durch Mitteln zu halbieren, müssen vier Grauwerte gemittelt werden.

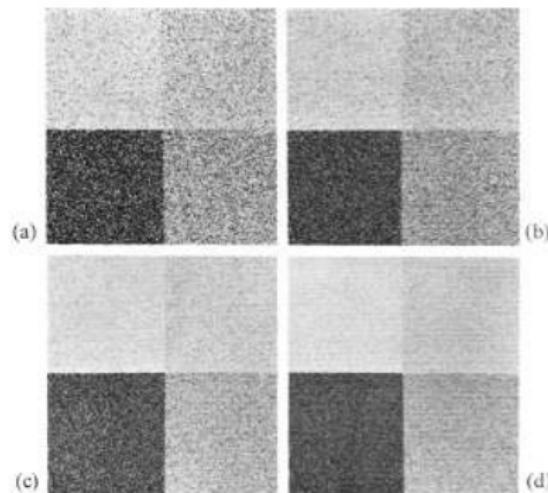


Figure 6.9. Smoothing by superposition of multiple copies. The numbers of copies in parts (a-d) are 2, 4, 8, and 16, respectively. [The pictures that were superimposed are versions of Figure 6.7(a).]

Beispiel für Mittelung

Example of Averaging



intensity averaging with
5 x 5 mask

$$\frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Simple Smoothing Operations

1. Averaging

$$\hat{g}_{ij} = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} \quad D \text{ is region around } g_{ij}$$

Example of
3-by-3 region D



2. Removal of outliers

$$\hat{g}_{ij} = \begin{cases} \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} & \text{if } \left| g_{ij} - \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} \right| \geq S \\ g_{ij} & \text{otherwise} \end{cases} \quad S \text{ is threshold}$$

Example of weights
in 3-by-3 region

1	2	1
2	3	2
1	2	1

3. Weighted average

$$\hat{g}_{ij} = \frac{1}{\sum w_k} \sum_{g_k \in D} w_k g_k \quad w_k = \text{weights in } D$$

Note that these operations are heuristics and not well founded!

Bimodal Averaging

To avoid averaging across edges, assume bimodal greyvalue distribution and select average value of modality with largest population.

1. Determine $\bar{g}_D = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$

2. $A = \{g_k \text{ with } g_k \geq \bar{g}_D\}$ $B = \{g_k \text{ with } g_k < \bar{g}_D\}$

3. $g_D' = \begin{cases} \frac{1}{|A|} \sum_{g_k \in A} g_k & \text{if } |A| \geq |B| \\ \frac{1}{|B|} \sum_{g_k \in B} g_k & \text{otherwise} \end{cases}$

Example:

11	14	15
13	12	25
15	19	26

→ A, B → $g_D' = 13$

Bimodal Averaging

To avoid averaging across edges, assume bimodal greyvalue distribution and select average value of modality with largest population.

1. Determine $\bar{g}_D = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$

2. $A = \{g_k \text{ with } g_k \geq \bar{g}_D\}$ $B = \{g_k \text{ with } g_k < \bar{g}_D\}$

3. $g_D' = \begin{cases} \frac{1}{|A|} \sum_{g_k \in A} g_k & \text{if } |A| \geq |B| \\ \frac{1}{|B|} \sum_{g_k \in B} g_k & \text{otherwise} \end{cases}$

Example:

B	11	14	15
	13	12	25
A	15	19	26

→ A, B → $g_D' = 13$

Bimodal Averaging

To avoid averaging across edges, assume bimodal greyvalue distribution and select average value of modality with largest population.

1. Determine $\bar{g}_D = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$

2. $A = \{g_k \text{ with } g_k \geq \bar{g}_D\}$ $B = \{g_k \text{ with } g_k < \bar{g}_D\}$

3. $g_D' = \begin{cases} \frac{1}{|A|} \sum_{g_k \in A} g_k & \text{if } |A| \geq |B| \\ \frac{1}{|B|} \sum_{g_k \in B} g_k & \text{otherwise} \end{cases}$

Example:

B	11	14	15	$\bar{g}_D = 16,7 \rightarrow$	A, B	$\rightarrow g_D' = 13$
	13	12	25			
	15	19	26	A		

Glätten mit rotierender Maske

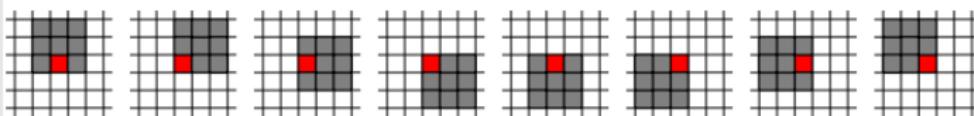
Averaging with Rotating Mask

Replace center pixel by average over pixels from the most homogeneous subset taken from the neighbourhood of center pixel.

Measure for (lack of) homogeneity is dispersion σ^2 (= empirical variance) of the greyvalues of a region D:

$$\bar{g}_{ij} = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} \quad \sigma_{ij}^2 = \frac{1}{|D|} \sum_{g_{mn} \in D} (g_{mn} - \bar{g}_{ij})^2$$

Possible rotated masks in 5×5 neighbourhood of center pixel:



Algorithm:

1. Consider each pixel g_{ij}
2. Calculate dispersion in mask for all rotated positions of mask
3. Choose mask with minimum dispersion
4. Assign average greyvalue of chosen mask to g_{ij}

Median Filter

Median of a distribution $P(x)$: x_m such that $P(x < x_m) = 1/2$

Median Filter:

$$\hat{g}_{ij} = \max a \text{ with } g_k \in D \text{ and } |\{g_k < a\}| < \frac{|D|}{2}$$

1. Sort pixels in D according to greyvalue
2. Choose greyvalue in middle position

Example:

11	14	15
13	12	25
15	19	26



11 11
12
13
14
15
15
19
25
26

greyvalue of center pixel
of region is set to 15

Median Filter reduces influence of outliers in either direction!

Rangordnungsfilter

Medianfilter sind ein Spezialfall der Rangordnungsfilter:

- ▶ Das Resultat der Filterung ist der Grauwert des Pixels mit mittlerem Rang in der Sortierfolge der Pixel.

Andere Variationen:

Expansion der dunklen Bereiche: Wähle das Pixel mit dem kleinsten Rang.

Expansion der hellen Bereiche: Wähle das Pixel mit dem höchsten Rang.

Gaussian Blur

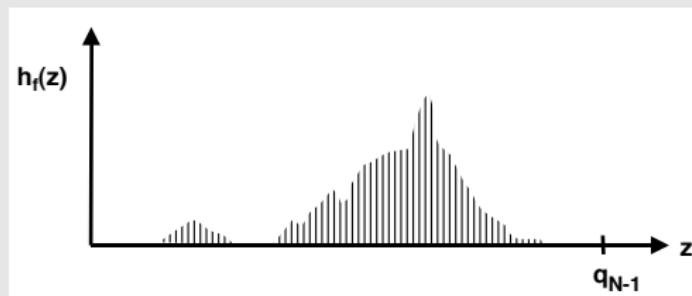
Gaussian Blur

Definition (Grauerthistogramm)

Ein Grauerthistogramm $h_f(z)$ eines Bildes f beschreibt die Häufigkeit der Grauwerte z im Bild.

Sei f ein Bild mit N Quantisierungsstufen:

- ☞ das Histogramm ist dann ein Vektor (array) mit N Elementen.

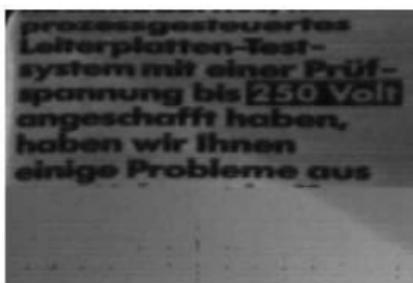


- ☞ Ein **Histogramm** beschreibt **diskrete** Werte,
- ☞ eine **Verteilung kontinuierliche**.

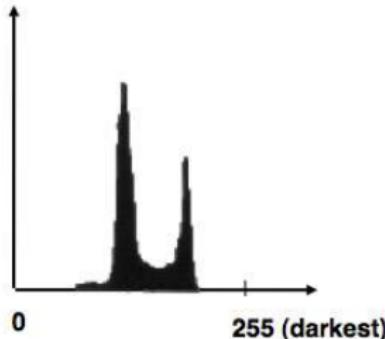
Beispiel

Example of Greyvalue Histogram

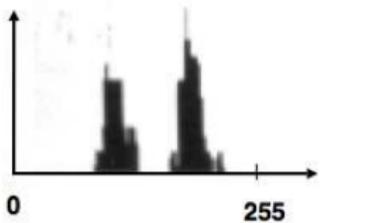
image



histogram



A histogram can be "sharpened" by
discounting pixels at edges
(more about edges later):



Maskierte Histogramme

Definition (Maskierte Histogramme)

Maskierte Histogramme sind Histogramme über Bildpunkte, die ein Prädikat Q befriedigen:

$$h_a = |g_{ij} = a \wedge Q(g_{ij})|$$

Beispiel („Histogramm schärfen“:)

Es werden nur Pixel berücksichtigt, an denen der Grauwert wenig variiert.

$$Q(g_{ij}) = |g_{ij} - g_{i-1,j}| + |g_{ij} - g_{i,j-1}| < \delta$$

Histogramme abgeleiteter Größen:

Beispiel

Gradientenhistogramm: Zähle die Häufigkeit der Gradientenwerte:

$$h(|g_{ij} - g_{i-1,j}| + |g_{ij} - g_{i,j-1}|)$$

Beispiel

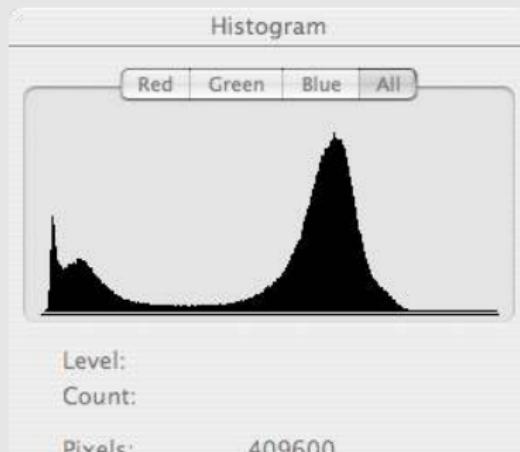
Original



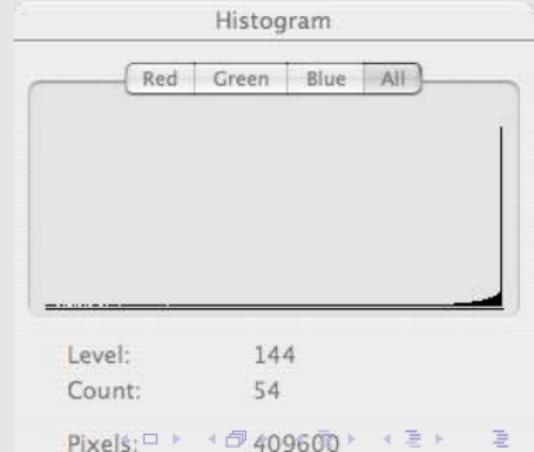
Kantenbild



Histogram des Originalbilds



Histogram des Kantenbilds



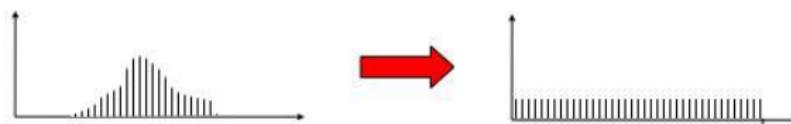
Histogram-Modifikation

Histogram Modification

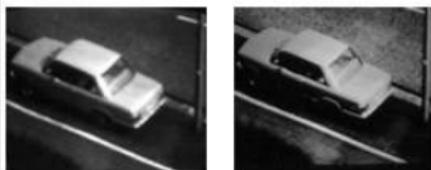
Greyvalues may be remapped into new greyvalues to

- facilitate image analysis
- improve subjective image quality

Example: Histogram equalization



1. Cut histogram into N stripes of equal area ($N = \text{new number of greyvalues}$)
2. Assign new greyvalues to consecutive stripes



Examples show improved resolution
of image parts with most frequent
greyvalues (road surface)

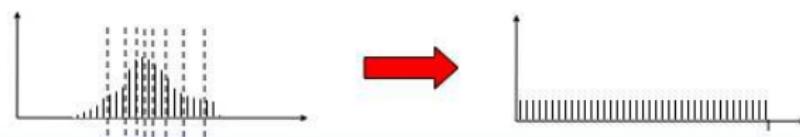
Histogram-Modifikation

Histogram Modification

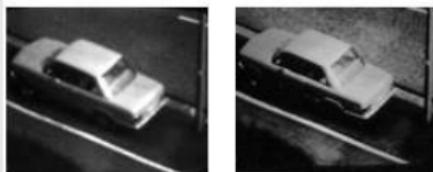
Greyvalues may be remapped into new greyvalues to

- facilitate image analysis
- improve subjective image quality

Example: Histogram equalization



1. Cut histogram into N stripes of equal area ($N = \text{new number of greyvalues}$)
2. Assign new greyvalues to consecutive stripes



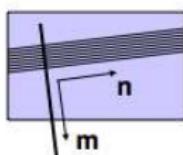
Examples show improved resolution
of image parts with most frequent
greyvalues (road surface)

Anwendung von Histogrammen

- ▶ Schwellenwertbestimmung
- ▶ Bereichszerlegung
- ▶ Quantisierung
- ▶ Bildwiedergabe

Projections

A projection of greyvalues in an image is the sum of all greyvalues orthogonal to a base line:

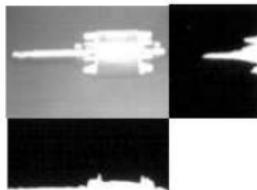
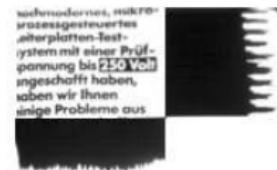


$$p_m = \sum_n g_{mn}$$

Often used:

"row profile" = row vector of all (normalized) column sums

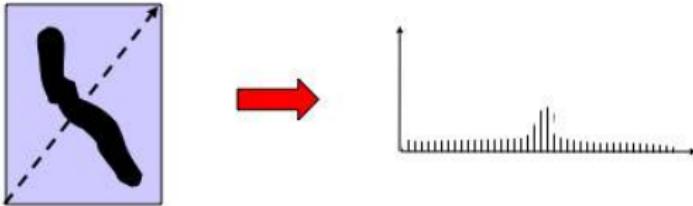
"column profile" = column vector of all (normalized) row sums



Cross-sections

A cross-section of a greyscale image is a vector of all pixels along a straight line through the image.

- fast test for localizing objects
- commonly taken along a row or column



Diskrete Abstandsmaße

Ein Abstandsmaß $d(x, y)$ für zwei Punkte x und y sollte eine **Metrik** sein:

Positiv:

$$\begin{aligned} d(x, y) &\geq 0 \\ d(x, y) = 0 &\rightarrow x = y \end{aligned}$$

Symmetrisch:

$$d(x, y) = d(y, x)$$

Dreiecksungleichung:

$$d(x, y) + d(y, z) \geq d(x, z)$$

Distance in Digital Images

Intuitive concepts of continuous images do not always carry over to digital images.

Several methods for measuring distance between pixels:

Eucledian distance

$$D_E((i, j), (h, k)) = \sqrt{(i - h)^2 + (j - k)^2}$$

costly computation of square root,
can be avoided for distance comparisons

City block distance

$$D_4((i, j), (h, k)) = |i - h| + |j - k|$$

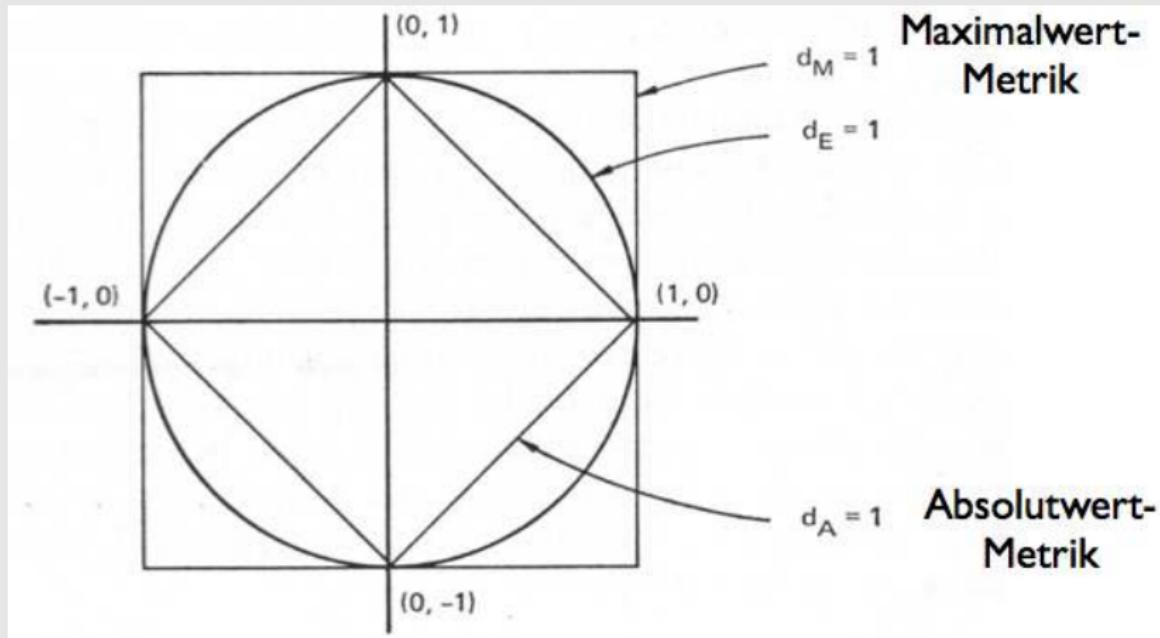
number of horizontal and vertical steps in a rectangular grid

Chessboard distance

$$D_8((i, j), (h, k)) = \max \{ |i - h|, |j - k| \}$$

number of steps in a rectangular grid if diagonal steps are allowed (number of moves of a king on a chessboard)

Einheitskreise diskreter Metriken



Einheitskreise einer Metrik und Zusammenhang

Absolutwertmetrik: $d_A = d_4$

- ▶ Der Einheitskreis der Absolutwertmetrik (City block Metrik) umfaßt die vier durch Kanten verbundenen Nachbarpixel.
- ▶ Der Einheitskreis von d_A definiert den 4-zusammenhang.

Schachbrettmetrik: $d_A = d_8$

- ▶ Der Einheitskreis der Maximalwertmetrik (Schachbrettmetrik) umfaßt die vier durch Kanten oder über Eck verbundenen Nachbarpixel.
- ▶ Der Einheitskreis von d_A definiert den 8-zusammenhang.

Keine Metriken

Satz (Diskrete Modifikationen des euklidischen Abstands:)

Vorsicht ☹! Einige diskrete Modifikationen des euklidischen Abstands sind keine Metriken:

- ▶ Der gerundete euklidische Abstand,
- ▶ Der quadrierte euklidische Abstand,
- ▶ Der abgeschnittene euklidische Abstand sind keine Metriken.

Satz (Gerundeter Euklidischer Abstand:)

Der gerundete Euklidische Abstand $\text{round}(d_e(x, y))$ ist keine Metrik.

Beweis.

Gegenbeispiel:

Sei $x = (0, 0)$, $y = (1, 1)$, $z = (2, 2)$

$$d(x, y) = \text{round}(\sqrt{2}) = \text{round}(1.41421 \dots) = 1$$

$$d(y, z) = \text{round}(\sqrt{2}) = \text{round}(1.41421 \dots) = 1$$

$$d(x, z) = \text{round}(2 \cdot \sqrt{2}) = \text{round}(2.82 \dots) = 3$$

$$d(x, y) + d(y, z) < d(x, z)$$

Die Dreiecksungleichung ist nicht erfüllt. □

Satz (Quadrierter Euklidischer Abstand:)

Der quadrierte Euklidische Abstand $d_e(x, y)^2$ ist keine Metrik.

Beweis.

Gegenbeispiel:

Sei $x = (1, 0)$, $y = (2, 0)$, $z = (3, 0)$

$$d(x, y) = 1$$

$$d(y, z) = 1$$

$$d(x, z) = 2 \cdot 2 = 4$$

$$d(x, y) + d(y, z) = 2 < d(x, z)$$

Die Dreiecksungleichung ist nicht erfüllt. □

Außerdem: Die Abstände sind nicht notwendigerweise ganze Zahlen.

Satz (Abgeschnittener Euklidischer Abstand):

Der abgeschnittene Euklidische Abstand $\lfloor d_e(x, y) \rfloor$ ist keine Metrik.

Beweis.

Gegenbeispiel:

Sei $x = (0, 0)$, $y = (1, 1)$, $z = (3, 4)$

$$d(x, y) = \lfloor \sqrt{2} \rfloor = \lfloor 1.41 \dots \rfloor = 1$$

$$d(y, z) = \lfloor \sqrt{13} \rfloor = \lfloor 3.6 \dots \rfloor = 3$$

$$d(x, z) = \lfloor \sqrt{25} \rfloor = 5$$

$$d(x, y) + d(y, z) = 4 < d(x, z)$$

Die Dreiecksungleichung ist nicht erfüllt. □

Geometric Transformations

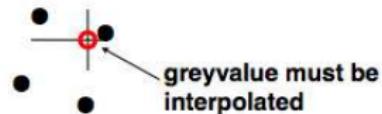
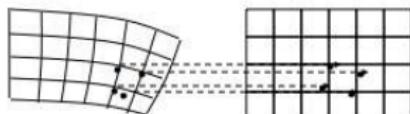
Various applications:

- change of view point
- elimination of geometric distortions from image capturing
- registration of corresponding images
- artificial distortions, Computer Graphics applications

Step 1: Determine mapping $T(x, y)$ from old to new coordinate system

Step 2: Compute new coordinates (x', y') for (x, y)

Step 3: Interpolate greyvalues at grid positions from greyvalues at transformed positions



Polynomiale Koordinatentransformation

- assume polynomial mapping between (x, y) and (x', y') of degree m
- determine corresponding points
- a) solve linear equations for a_{rk}, b_{rk} ($r, k = 1 \dots m$)
- b) minimize mean square error (MSE) for point correspondences

General format of transformation:

$$x' = \sum_{r=0}^m \sum_{k=0}^{m-r} a_{rk} x^r y^k$$

$$y' = \sum_{r=0}^m \sum_{k=0}^{m-r} b_{rk} x^r y^k$$

Approximation by biquadratic transformation:

$$x' = a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 + a_{02}y^2 \quad \text{at least 6 corresponding pairs needed}$$

$$y' = b_{00} + b_{10}x + b_{01}y + b_{11}xy + b_{20}x^2 + b_{02}y^2$$

Approximation by affine transformation:

$$x' = a_{00} + a_{10}x + a_{01}y$$

at least 3 corresponding pairs needed

$$y' = b_{00} + b_{10}x + b_{01}y$$

Zur Erinnerung: Affine 2D-Transformationen

Translation, Rotation, Scaling, Skewing

Translation by vector t :

$$\underline{v}' = \underline{v} + \underline{t} \quad \text{with} \quad \underline{v}' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad \underline{v} = \begin{bmatrix} x \\ y \end{bmatrix} \quad \underline{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Rotation of image coordinates by angle α :

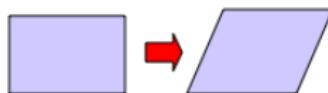
$$\underline{v}' = R \underline{v} \quad \text{with} \quad R = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

Scaling by factor a in x-direction and factor b in y-direction:

$$\underline{v}' = S \underline{v} \quad \text{with} \quad S = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$$

Skewing by angle β :

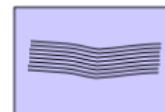
$$\underline{v}' = W \underline{v} \quad \text{with} \quad W = \begin{bmatrix} 1 & \tan \beta \\ 0 & 1 \end{bmatrix}$$



Example of Geometry Correction by Scaling

Distortions of electron-tube cameras may be

1 - 2 % => more than 5 lines for TV images



ideal image

actual image

Correction procedure may be based on

- fiducial marks engraved into optical system
- a test image with regularly spaced marks

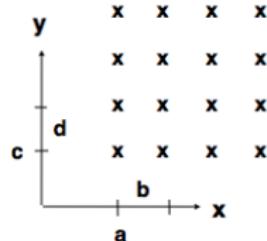
Ideal mark positions:

$$x_{mn} = a + mb, \quad y_{mn} = c + nd \quad m = 0 \dots M-1$$

$$\text{Actual mark positions:} \quad n = 0 \dots N-1$$

$$x'_{mn}, y'_{mn}$$

Determine a, b, c, d such that MSE (mean square error) of deviations is minimized



Minimierung des Anpassungsfehlers

Minimizing the MSE

$$\begin{aligned}\text{Minimize } E &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x_{mn} - x'_{mn})^2 + (y_{mn} - y'_{mn})^2 \\ &= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (a + mb - x'_{mn})^2 + (c + nd - y'_{mn})^2\end{aligned}$$

From $\delta E / \delta a = \delta E / \delta b = \delta E / \delta c = \delta E / \delta d = 0$ we get

$$a = \frac{2}{MN(M+1)} \sum_m \sum_n (2M-1-3m) x'_{mn}$$

$$b = \frac{6}{MN(M^2-1)} \sum_m \sum_n (2m-M+1)x'_{mn}$$

$$c = \frac{2}{MN(N+1)} \sum_m \sum_n (2N-1-3n) y'_{mn}$$

$$d = \frac{6}{MN(N^2-1)} \sum_m \sum_n (2n-N+1)y'_{mn}$$

Special case M=N=2:

$$a = 1/2 (x'_{00} + x'_{01})$$

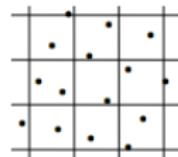
$$b = 1/2 (x'_{10} - x'_{00} + x'_{11} - x'_{01})$$

$$c = 1/2 (y'_{00} + y'_{01})$$

$$d = 1/2 (y'_{01} - y'_{00} + y'_{11} - y'_{10})$$

Principle of Greyvalue Interpolation

Greyvalue interpolation = computation of unknown greyvalues at locations $(u'v')$ from known greyvalues at locations $(x'y')$



Two ways of viewing interpolation in the context of geometric transformations:

- A Greyvalues at grid locations $(x y)$ in old image are placed at corresponding locations $(x'y')$ in new image: $g(x'y') = g(T(x y))$
=> interpolation in new image
- B Grid locations $(u'v')$ in new image are transformed into corresponding locations $(u v)$ in old image: $g(u v) = g(T^{-1}(u'v'))$
=> interpolation in old image

We will take view B:

Compute greyvalues between grid from greyvalues at grid locations.

Grauwertinterpolation: Nächster Nachbar

Nearest Neighbour Greyvalue Interpolation

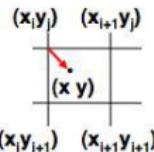
Assign to (x, y) greyvalue of nearest grid location

(x_i, y_j) (x_{i+1}, y_j) (x_i, y_{j+1}) (x_{i+1}, y_{j+1})

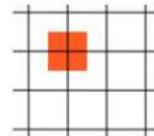
grid locations

(x, y)

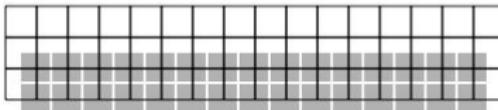
location between grid with
 $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$



Each grid location represents the greyvalues in a rectangle centered around this location:



Straight lines or edges may appear step-like after this transformation:



Grauwertinterpolation: Nächster Nachbar

Nearest Neighbour Greyvalue Interpolation

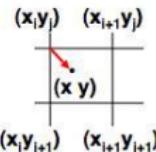
Assign to (x, y) greyvalue of nearest grid location

(x_i, y_j) (x_{i+1}, y_j) (x_i, y_{j+1}) (x_{i+1}, y_{j+1})

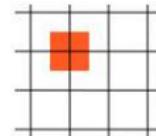
grid locations

(x, y)

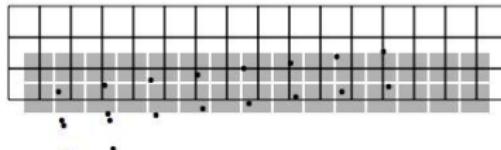
location between grid with
 $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$



Each grid location represents the greyvalues in a rectangle centered around this location:



Straight lines or edges may appear step-like after this transformation:



Grauwertinterpolation: Nächster Nachbar

Nearest Neighbour Greyvalue Interpolation

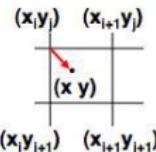
Assign to (x, y) greyvalue of nearest grid location

(x_i, y_j) (x_{i+1}, y_j) (x_i, y_{j+1}) (x_{i+1}, y_{j+1})

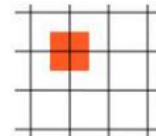
grid locations

(x, y)

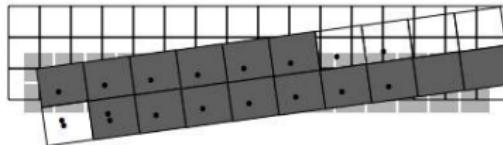
location between grid with
 $x_i \leq x \leq x_{i+1}$, $y_j \leq y \leq y_{j+1}$



Each grid location represents the greyvalues in a rectangle centered around this location:



Straight lines or edges may appear step-like after this transformation:



Bilinear Greyvalue Interpolation

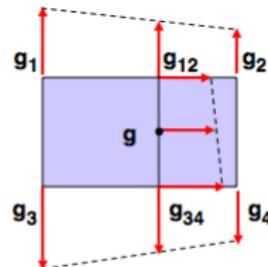
The greyvalue at location (x, y) between 4 grid points (x_i, y_j) , (x_{i+1}, y_j) , (x_i, y_{j+1}) , (x_{i+1}, y_{j+1})

+1) (x_{i+1}, y_{j+1}) is computed by linear interpolation in both directions:

$$g(x, y) = \frac{1}{(x_{i+1} - x_i)(y_{j+1} - y_j)} \left\{ (x_{i+1} - x)(y_{j+1} - y)g(x_i, y_j) + (x - x_i)(y_{j+1} - y)g(x_{i+1}, y_j) + (x_{i+1} - x)(y - y_j)g(x_i, y_{j+1}) + (x - x_i)(y - y_j)g(x_{i+1}, y_{j+1}) \right\}$$

Simple idea behind long formula:

1. Compute g_{12} = linear interpolation of g_1 and g_2
2. Compute g_{34} = linear interpolation of g_3 and g_4
3. Compute g = linear interpolation of g_{12} and g_{34}



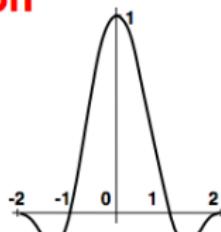
The step-like boundary effect is reduced.
But bilinear interpolation may blur sharp edges.

Bikubische Grauwertinterpolation

Bicubic Interpolation

Each greyvalue at a grid point is taken to represent the center value of a local bicubic interpolation surface with cross section h_3 .

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{for } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$



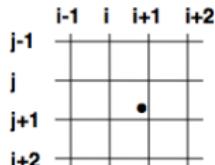
cross section of
interpolation kernel

The greyvalue at an arbitrary point $[u, v]$ (black dot in figure) can be computed by

- 4 horizontal interpolations to obtain greyvalues at points $[u, j-1] \dots [u, j+2]$ (red dots), followed by
- 1 vertical interpolation (between red dots) to obtain greyvalue at $[u, v]$.

Note:

For an image with constant geyvalues g_0 the interpolated greyvalues at all points between the grid lines are also g_0 .

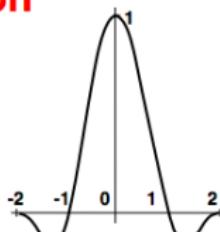


Bikubische Grauwertinterpolation

Bicubic Interpolation

Each greyvalue at a grid point is taken to represent the center value of a local bicubic interpolation surface with cross section h_3 .

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{for } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$



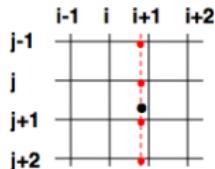
The greyvalue at an arbitrary point $[u, v]$ (black dot in figure) can be computed by

- 4 horizontal interpolations to obtain greyvalues at points $[u, j-1] \dots [u, j+2]$ (red dots), followed by
- 1 vertical interpolation (between red dots) to obtain greyvalue at $[u, v]$.

Note:

For an image with constant geyvalues g_0 the interpolated greyvalues at all points between the grid lines are also g_0 .

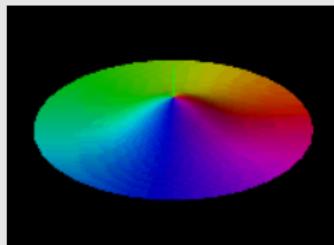
cross section of
interpolation kernel



Teil XIII

Ikonische Algorithmen

Lokale Bildeigenschaften



- 34 Globale Bildeigenschaften
- 35 Geometrische Operationen auf Bildern
- 36 Lokale Bildeigenschaften
 - Scharfzeichnungsfilter

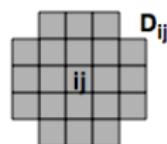
Local Neighbourhood Operations

Many useful image transformations may be defined as an instance of a local neighbourhood operation:

Generate a new image with pixels \hat{g}_{mn} by applying operator f to all pixels g_{ij} of an image

$$\hat{g}_{mn} = f(g_1, g_2, \dots, g_K) \quad g_1, g_2, \dots, g_K \in D_{ij}$$

example of
neighbour-
hood



Pixel indices i, j may be incremented by steps larger than 1 to obtain reduced new image.

Scharfzeichnungsfilter: Unscharfes Maskieren



intensity sharpening
with 3×3 mask

-1	-1	-1
-1	9	-1
-1	-1	-1



"unsharp masking" =
subtraction of blurred image

$$\hat{g}_{ij} = g_{ij} - \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$$

Erklärungsmodelle für die Wirkungsweise

- ① Effekt des Mittelns rückgängig machen
- ② Faltung mit dem Laplace-Operator: Verstärkt die Variation des Bildes
- ③ Kanten überhöhen
- ④ Diffusionsprozeß
- ⑤ Im Frequenzraum: Hochpaßfilter

Heuristik: Umkehr des Mittelns

Heuristischer Zugang:

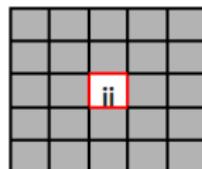
Effekt des Mittelns rückgängig machen

$$\text{Mitteln} : g'_{ij} = 1/M \sum_{g_k \in D_{ij}} g_k$$

$$\text{Schärfen} : g_{ij} = M \cdot g'_{ij} - \sum_{g_k \in \hat{D}_{ij}} g_k, \quad \hat{D}_{ij} = D_{ij} - \{g_{ij}\}$$

$$\text{geschätzt} : \approx M \cdot g'_{ij} - \sum_{g'_k \in \hat{D}_{ij}} g'_k$$

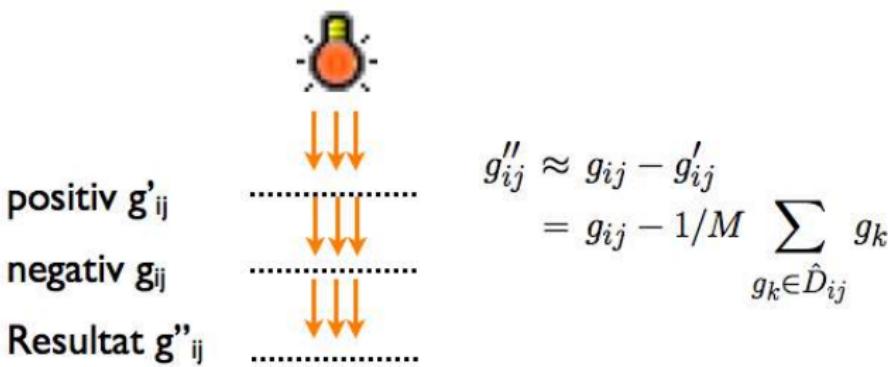
$\hat{D}_{ij}, M=25$



Fotografische Lösung

„Unscharfes Maskieren“:

Beim Herstellen von Positiv-Abzügen ein unscharfes Negativbild in den Lichtstrahl einbringen und damit das scharfe Negativ belichten:



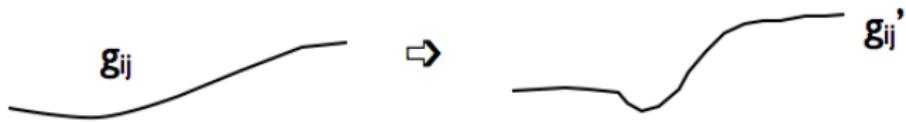
Scharfzeichnen: Kanten überhöhen

● Kanten überhöhen:

- einfacher Kantenoperator mißt Kantenhöhe:

$$\begin{aligned}dg &= g_{ij} - g_{i-1,j} + g_{ij} - g_{ij+1} \\g_{ij}' &= g_{ij} + \alpha dg\end{aligned}$$

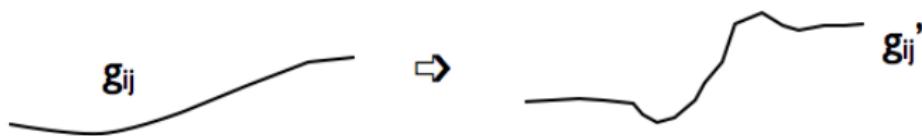
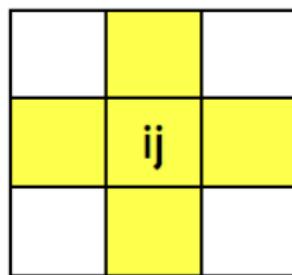
i-1,j	
i,j	i, j+1



Symmetrische Lösung: Laplace-Operator

- Symmetrische Lösung:

$$\begin{aligned}dg &= 4g_{ij} - g_{i-1j} - g_{ij-1} - g_{ij+1} \\g_{ij}' &= g_{ij} + \alpha dg\end{aligned}$$



Teil XIV

Binarisieren

Segmentierung von Bildern



37

Grundlagen der Segmentierung

- Ziele der Segmentierung
- Uniformitätsannahme

38

Binarisieren

Segmentierung

Definition (**Segmentierung**)

ist die Zerlegung des Bildes in Komponenten, die als sinnvolle Einheiten weiterverarbeitet werden können,

- ▶ z.B. Partitionierung in Bereiche, die potentiell Objekten oder Objektteilen entsprechen.

Beispiel



Bildelemente

Welche Bildelemente können Szenenkomponenten entsprechen?

Die Antwort hängt von der Art der Bilder und der Komplexität der Szene ab:

- ☞ Je **allgemeiner** die Anwendung, desto **konservativer** muß segmentiert werden.

Segmentierung in

Ganze Objekte: Beispielsweise zur

- ▶ Zeichenerkennung
- ▶ Industrieller Objekterkennung
- ▶ medizinischer Zellanalyse

Linien oder Kantenzüge: Luftbilder, Innenräume

Kantenelemente, Ecken, Gruppen: Natürliche Szenen

Sekundäre Ziele der Segmentierung

- ▶ Eine Hierarchie von unterschiedlichen Auflösungen für weiterverarbeitende Prozesse zur Verfügung stellen.
- ▶ Reduktion der Datenmengen

Auflösung

Weiterverarbeitende Prozesse brauchen unterschiedlich feine Auflösung, z.B.

Grobe Auflösung wird benötigt zur:

- ▶ Analyse des Bildaufbaus (Horizont, Vordergrund usw.),
- ▶ Steuerung der Aufmerksamkeit,
- ▶ Planung von Feinanalyse.

Feine Auflösung wird benötigt zur Messung von Positionen und Geschwindigkeiten:

- ▶ Detailbeschreibung,
- ▶ Stereoskopie,
- ▶ Bewegungsanalyse

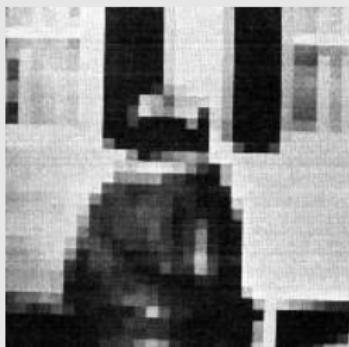
Datenreduktion

- ▶ Rohbilder sind auch wegen ihres Datenvolumens nicht als grundlegende Datenstruktur für die Bildanalyse geeignet.
- ▶ Segmentierung reduziert die Bilder auf eine symbolische Repräsentation, die wesentlich weniger Speicheraufwand erfordert.

Beispiel: Landhausszene des VISIONS-Systems



Landhausszene
(VISIONS System [Hanson and Riseman, 1978])



Oberkörper und Bein einer Person
(VISIONS System [Hanson and Riseman, 1978])

Probleme bei der Segmentierung

Schlechter Kontrast: Die Grauwerte der Vordergrundobjekte können von Hintergrund ununterscheidbar sein.

Beleuchtungsartefakte: Schatten, Glanzlichter, Verdeckung:

- ☞ Generell ist für eine erfolgreiche Segmentierung szenenspezifisches Kontextwissen erforderlich.

Uniformitätsannahme

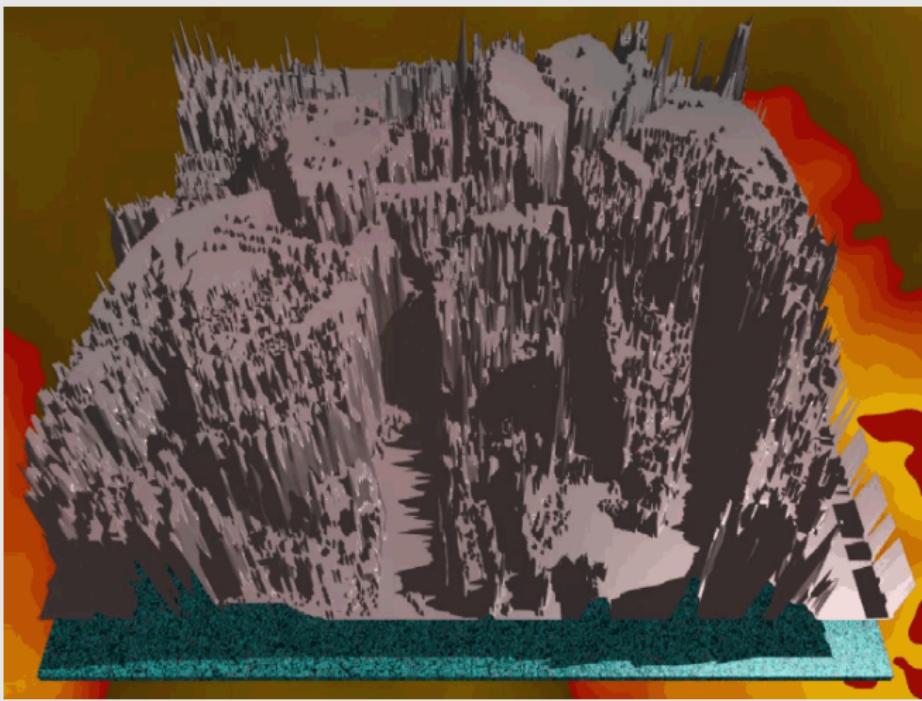


Uniformitätsannahme

Den meisten Segmentierungsverfahren liegt eine Uniformitätsannahme für die Bildfunktion zugrunde:

- ▶ Objekten entsprechen Bildbereiche, für die ein Uniformitätsprädikat gilt → **Bereichszerlegung**
- ▶ Duale Aussage: Objektgrenzen entsprechen Diskontinuitäten eines Uniformitätsprädikats → **Kantenfinder**

Beispiel: Grauwertgebirge



Typische Uniformitätsprädikate

- ▶ Grauwerte innerhalb fester Schranken (für Schwarz-Weiß-Bilder)
- ▶ Grauwertgradient klein: Grenzen an Stellen, wo der Grauwertgradient Extremwerte hat
- ▶ Uniforme statistische Eigenschaften: gleiche lokale Verteilung, gleiche Textur
- ▶ Glattheit in 3D: Grenzen an Entfernungsdiskontinuitäten

Binarisieren



- ▶ Segmentieren in kontrollierten Umgebungen
- ▶ mit wenigen Objekttypen

Binarisieren

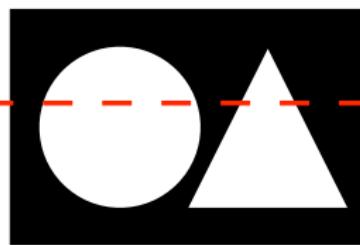
Definition (Binarisieren)

Binarisierung ist die Transformation einer Bildfunktion in ein **Binärbild**.

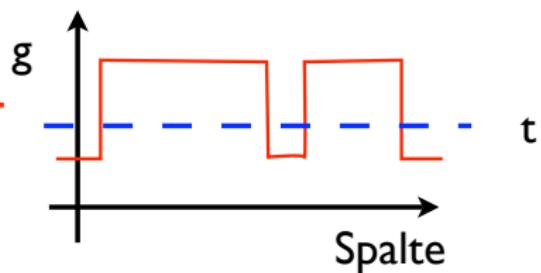
Für viele Anwendungen ist es ausreichend, nur zwischen **zwei** Grauwerten zu unterscheiden:

- ▶ „SCHWARZ“ und „WEISS“
- ▶ oder „1“ und „0“.

Vorgehen: Schwellenwertverfahren



Bild



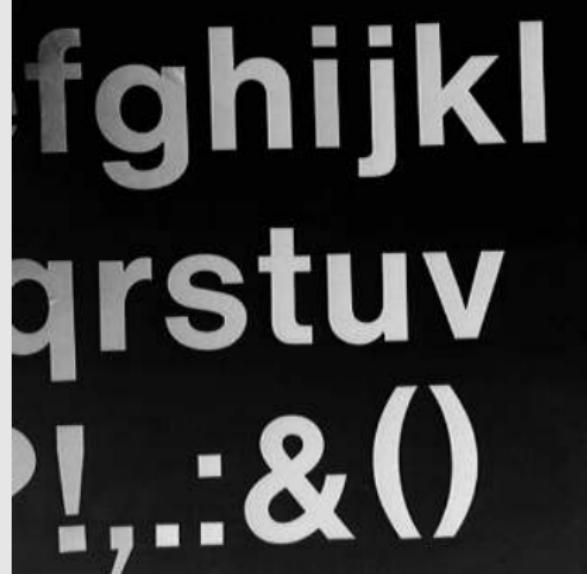
Zeilenprofil

Schwellenwertverfahren

Der Grauwert $g(x, y)$ wird auf „0“ abgebildet, wenn er kleiner als eine Schwellenwert t ist, sonst auf „1“.

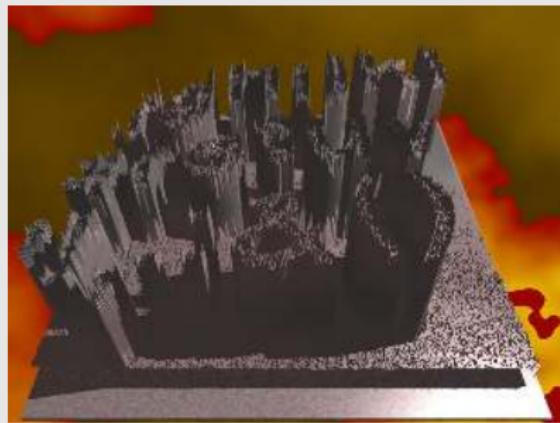
$$g^t(x, y) = \begin{cases} 0, & g(x, y) < t \\ 1, & g(x, y) \geq t \end{cases}$$

Beispiel: Segmentieren von Schriftzeichen



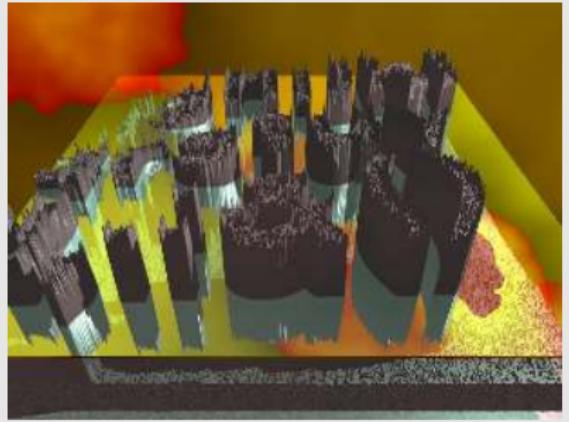
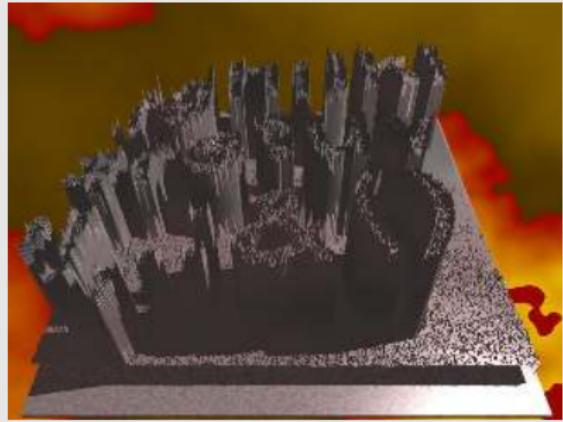
fghijkl
qrstuv
?!,:&()

Beispiel: Segmentieren von Schriftzeichen



Grauwertgebirge und Schwellenwert

Beispiel: Segmentieren von Schriftzeichen



Grauwertgebirge und Schwellenwert

Beispiel: Segmentieren von Schriftzeichen



t zu niedrig

(Negativ vom vorherigen Bild für bessere Druckbarkeit)

t optimal

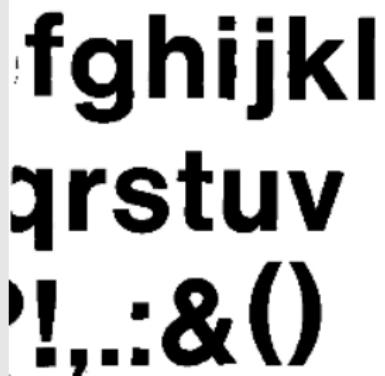
t zu hoch

Beispiel: Segmentieren von Schriftzeichen



t zu niedrig

(Negativ vom vorherigen Bild für bessere Druckbarkeit)



t optimal

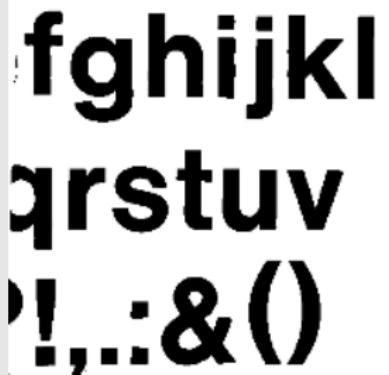
t zu hoch

Beispiel: Segmentieren von Schriftzeichen

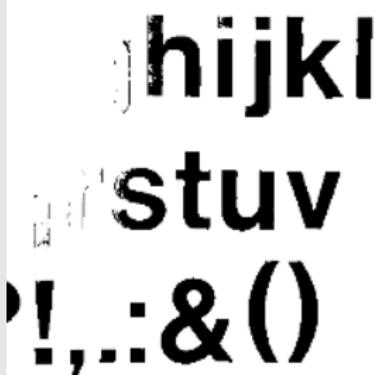


t zu niedrig

(Negativ vom vorherigen Bild für bessere Druckbarkeit)



t optimal



t zu hoch

Abwandlung: Konturfindung

Der Grauwert $g(x, y)$ wird auf „1“ abgebildet, wenn er in einem charakteristischen Grauwertintervall zwischen zwei Schwellen u und v liegt, sonst auf „0“.

$$g^{u,v}(x, y) = \begin{cases} 1, & u \geq g(x, y) \geq v \\ 0, & \text{sonst} \end{cases}$$

Beispiel: Konturen als Grauwertintervall



Originalbild

Konturen

Beispiel: Konturen als Grauwertintervall



Originalbild

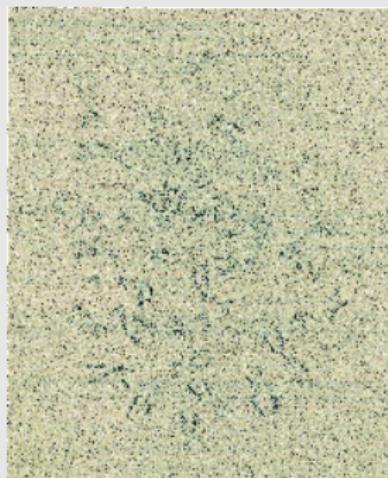


Konturen

Allgemein

- ☞ Schwellenwertverfahren sind anwendbar, um Objekt und Hintergrund zu trennen,
 - ▶ wenn die *Grauwerte* im Objekt einen charakteristischen Wert haben,
 - ▶ oder wenn eine *Funktion der Grauwerte* im Objekt einen charakteristischen Wert hat.

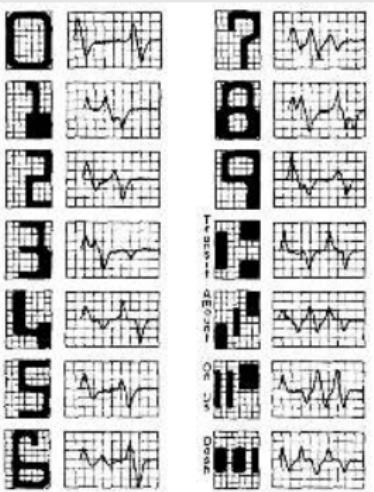
Beispiele für geeignete Anwendungsszenarien



Chromosomen

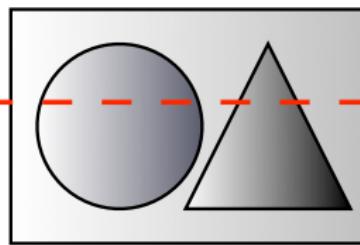


Fingerabdrücke

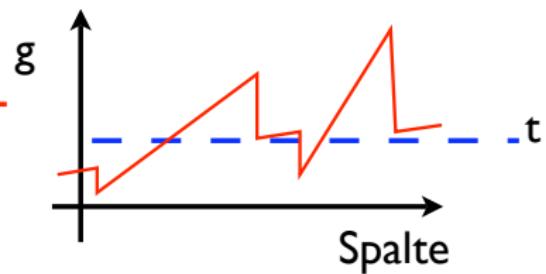


OCR-Schrift

Vorgehen: Charakteristische Funktion der Grauwerte



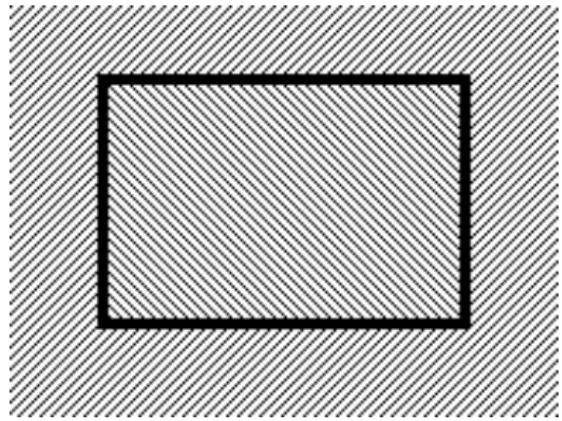
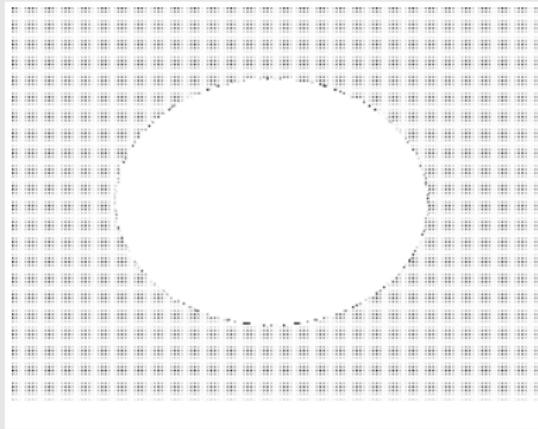
Bild



Zeilenprofil

In diesem Beispiel ist der Gradient der Grauwerte charakteristisch für Objekt und Hintergrund.

Segmentieren von texturierten Bildern



Mögliche Homogenitätskriterien:

- ▶ Mittlerer Grauwert
- ▶ Richtung der Linien

Verfahren zur Schwellenwertwahl

Manuell: systematisches Probieren, eventuell sehr zeitaufwendig

2-Klassen-Klassifikationsproblem: Klassifizierte die Pixel entsprechend der klassenbedingten Wahrscheinlichkeiten der Grauwerte als Objektpixel oder Hintergrundpixel.

Histogram: Das Histogram der Grauwerte sollte typischerweise bimodal sein - ein Maximum für die Objektgrauwerte, ein Maximum für die Hintergrundgrauwerte.

Vorwissen über die Szene: Kriteriumsfunktionen

Anwendbar, wenn aus dem Vorwissen über die Szene eine Kriteriumsfunktion für die Güte der Segmentierung abgeleitet werden kann: Beispiele für Kriterien:

- ▶ Die **Fläche** der gesuchten weißen Objekte ist bekannt.
- ▶ Die **Strichdicke** der resultierenden schwarzen Linien ist bekannt.
- ▶ Die **Anzahl** der weißen und schwarzen Komponenten ist bekannt.

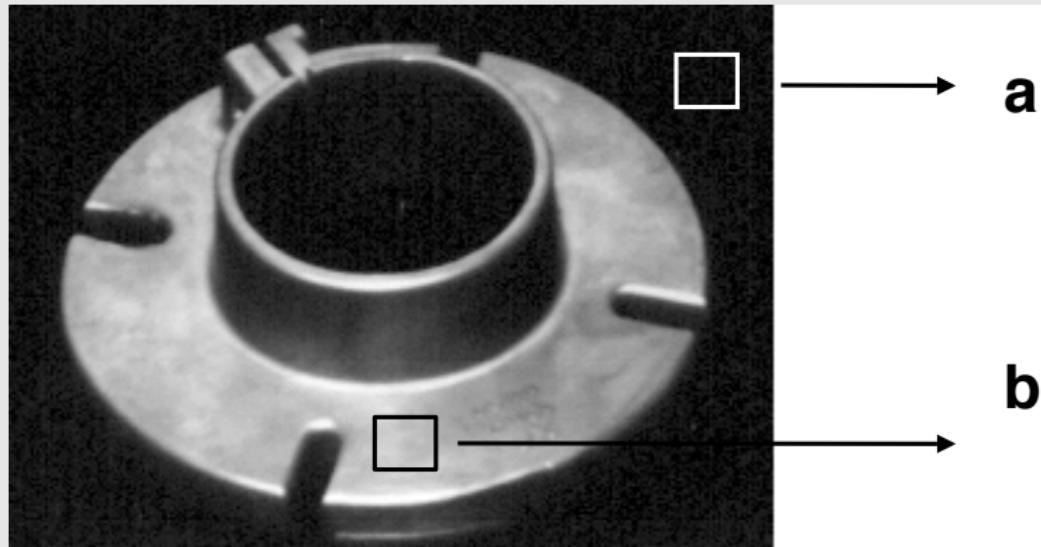
Vorgehen: Variiere systematisch den Schwellenwert (beispielsweise mit binärer Suche), bis das resultierende Binärbild die Kriteriumsfunktion erfüllt.

Schwellenwertwahl an Referenzposition

- ▶ Wenn die Position der Objekte ungefähr bekannt ist, kann die Schwelle anhand bekannter Objekt- und Hintergrundpixel geschätzt werden.
- ▶ Sei a ein Referenzgrauwert im Objekt und b ein Referenzgrauwert im Hintergrund, dann ist eine mögliche Schwelle t :

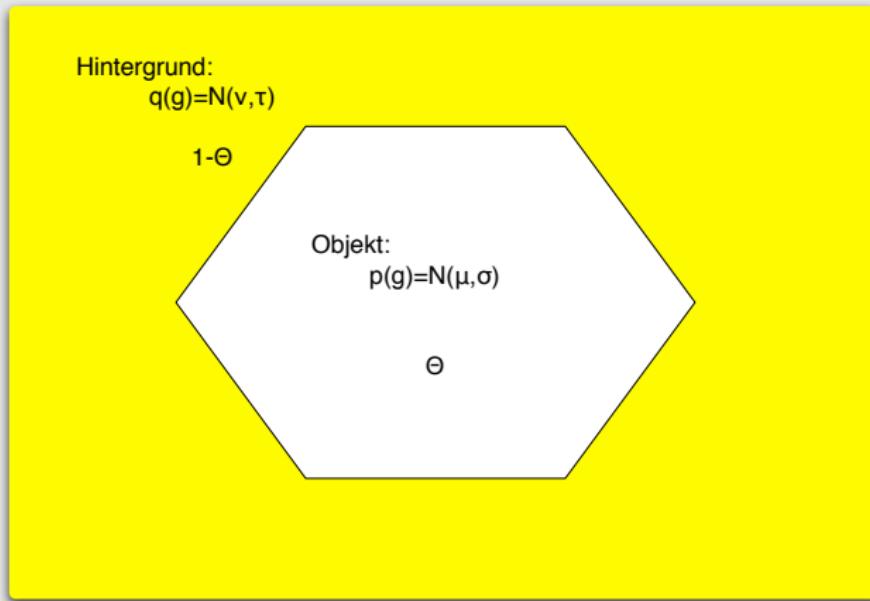
$$t = \frac{a + b}{2}$$

Beispiel: Teile auf dem Fließband in bekannter Position



Verteilungsbasierte Schwellenwertwahl

Annahme: die Dichtefunktionen der Grauwerte im Objekt und im Hintergrund seien bekannt, z.B. seien die Grauwerte normalverteilt.



Die klassenbedingten Wahrscheinlichkeiten

A-priori-Wahrscheinlichkeit der Objektpixel: θ

A-priori-Wahrscheinlichkeit der Hintergrundpixel: $1 - \theta$

Wahrscheinlichkeitsdichte der Objektgrauwerte:

Normalverteilt mit Mittelwert μ und Varianz σ .

$$p(g) = N(\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(g-\mu)^2}{2\sigma^2}}$$

Wahrscheinlichkeitsdichte der Hintergrundgrauwerte:

Normalverteilt mit Mittelwert ν und Varianz τ .

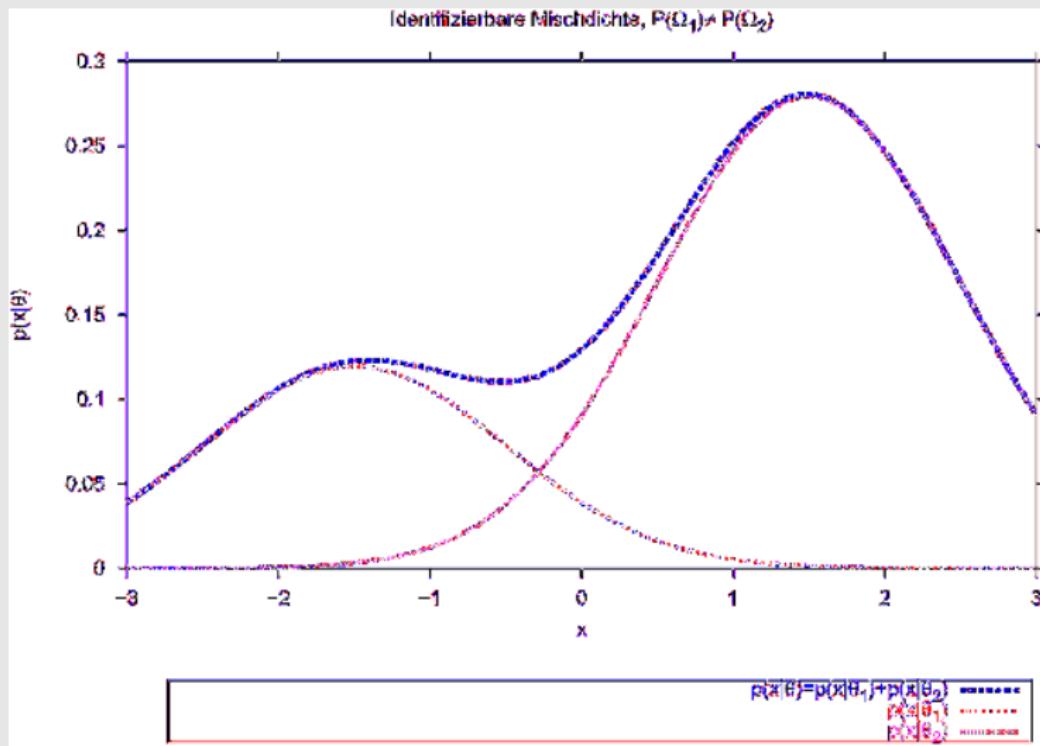
$$\mu < \nu$$

$$q(g) = N(\nu, \tau) = \frac{1}{\sqrt{2\pi}\tau} e^{\frac{-(g-\nu)^2}{2\tau^2}}$$

Mischdichte des Bildes:

$$b(g) = \theta \cdot p(g) + (1 - \theta) \cdot q(g)$$

Die Mischdichte



Fehler bei gegebener Schwelle t und $\mu < \nu$

Fehler: Hintergrund als Objekt:

$$Q(t) = \int_{-\infty}^t q(g)dg$$

Fehler: Objekt als Hintergrund:

$$1 - P(t) = \int_t^{\infty} p(g)dg$$

Fehlersumme: Gewichtet mit a-priori-Wahrscheinlichkeit:

$$\theta \cdot (1 - P(t)) + (1 - \theta) \cdot Q(t)$$

Minimum: Differenzieren nach t , Ableitung = 0

$$(1 - \theta) \cdot q(t) = \theta \cdot p(t)$$

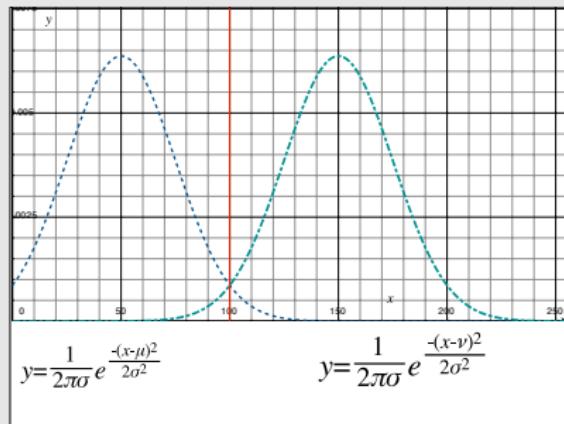
$$\begin{aligned}
 (1 - \theta) \cdot q(t) &= \theta \cdot p(t) \\
 p(t) &= \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \quad \text{Objekt} \\
 q(t) &= \frac{1}{\sqrt{2\pi}\tau} e^{\frac{-(t-\nu)^2}{2\tau^2}} \quad \text{Hintergrund}
 \end{aligned}$$

logarithmieren:

$$\begin{aligned}
 \ln \sigma + \ln(1 - \theta) - \frac{(t - \tau)^2}{2\tau^2} &= \ln \tau + \ln \theta - \frac{(t - \mu)^2}{2\sigma^2} \quad \text{oder} \\
 \tau^2(t - \mu)^2 - \sigma^2(t - \nu)^2 &= 2\sigma^2\tau^2 \ln \frac{\tau\theta}{\sigma(1 - \theta)}
 \end{aligned}$$

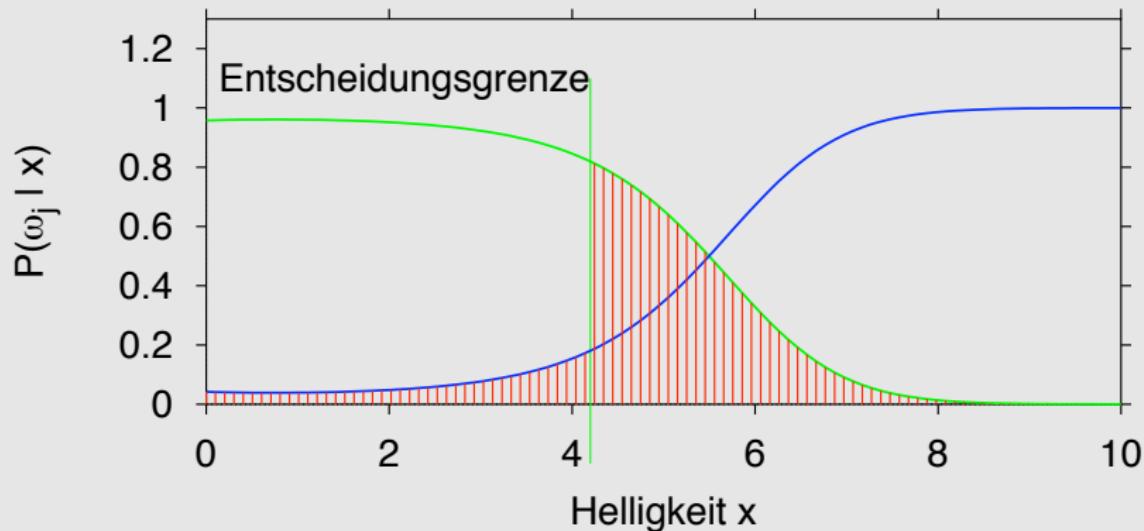
Spezialfall: $\theta = \frac{1}{2}, \tau = \sigma$

$$\begin{aligned}(t - \mu)^2 &= (t - \nu)^2 \\ t &= \frac{\mu + \nu}{2}\end{aligned}$$

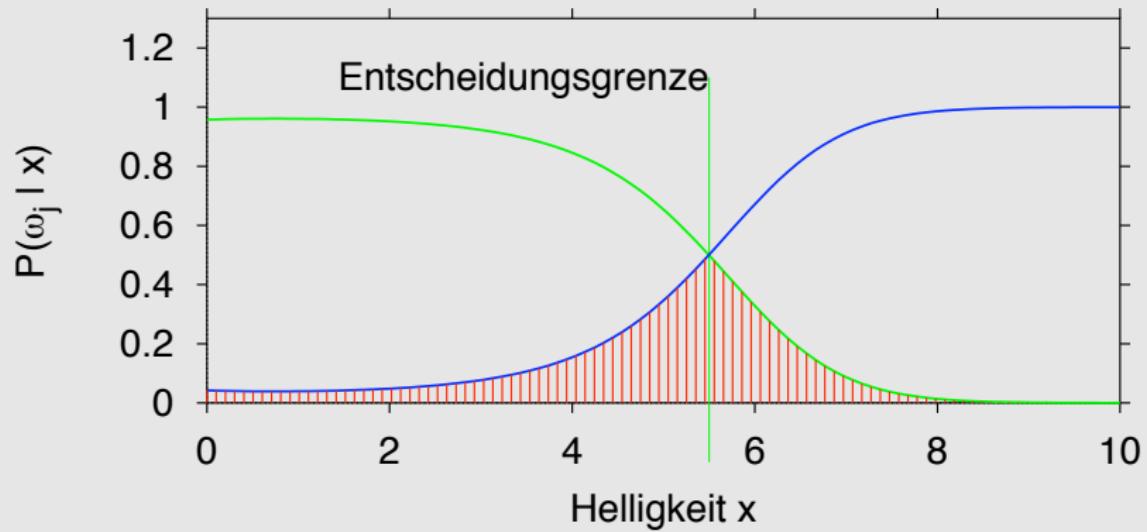


Beispiel: $\mu = 50, \nu = 150, \sigma = \tau = 20, t = 100$

Komponenten der Fehlerwahrscheinlichkeit

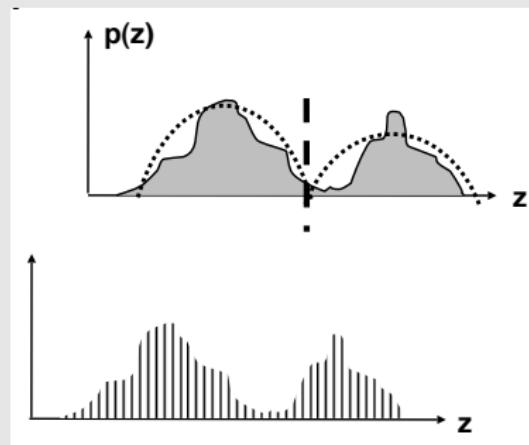
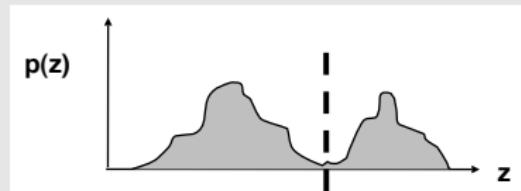


Optimale Schwelle im Schnittpunkt der Dichten



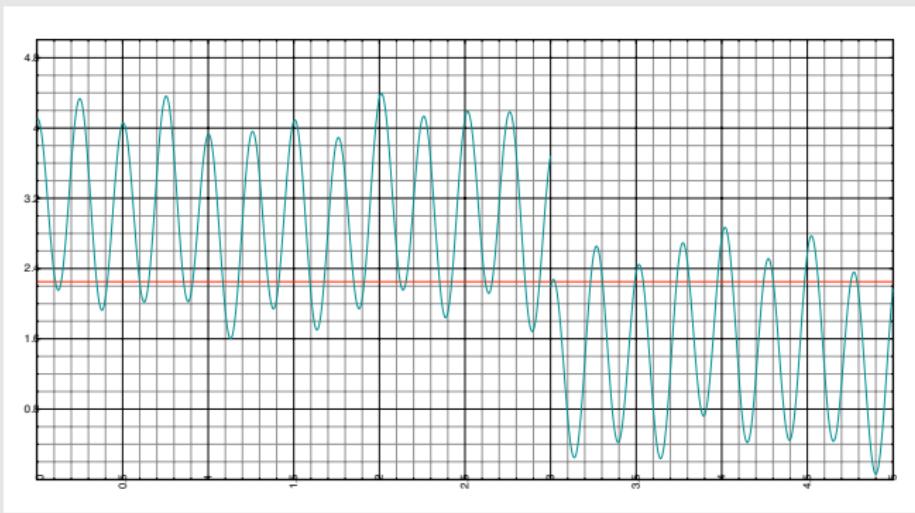
Schätzung der Dichten über Grauwerthistogramme

Heuristik: Suche das Tal zwischen den beiden Maxima der Grauwertdichte



Histogramme geben eine diskrete Approximation der Grauwertdichte.

Zeilenprofil eines stark verrauschten Bildes



Bei stark verrauschten Bildern

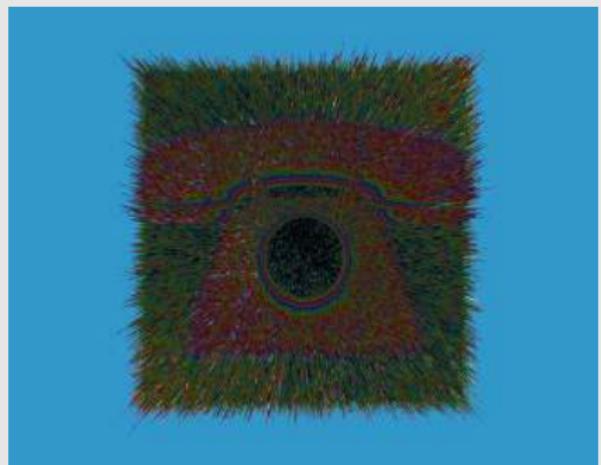
Bei starkem Rauschen ist eine Rauschunterdrückung notwendig, bevor ein Schwellenwertverfahren angewendet werden kann.

- ▶ Bilder glätten (Tiefpaßfilter, Medianfilter)
- ▶ Oder Kontext nutzen: benachbarte Pixel gehören oft zur selben Klasse.

Beispiel: ein stark verrauschtes Bild



Verrausches Bild



Grauwertgebirge

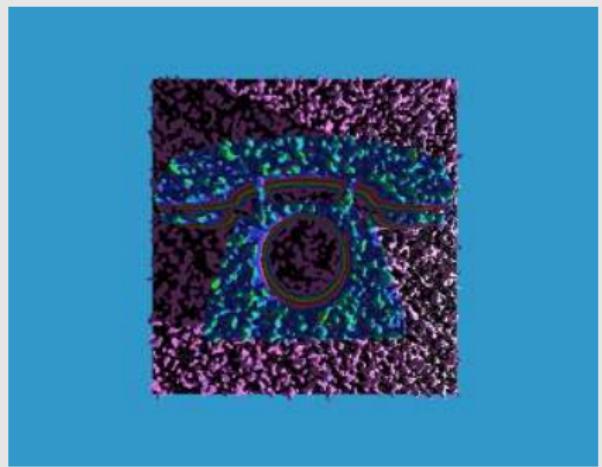
Grauwertgebirge und Schwellenwertebene

Telefonbild

Beispiel: Das Bild mit Medianfilter, Rang 5



Geglättetes Bild



Grauwertgebirge

Grauwertgebirge und Schwellenwertebene

Telefonbild

Kontrollierte Aufnahmebedingungen

Für industrielle Anwendungen ist es sehr nützlich, wohldefinierte Aufnahmebedingungen zu haben.

- ▶ Segmentierungsverfahren sind besonders einfach, wenn ein guter Kontrast sichergestellt werden kann.
- ▶ Besonders vorteilhaft ist es, wenn Teile auf einem transparenten Fließband bei Gegenlicht aufgenommen werden können.

Kontrollierte Aufnahmebedingungen

Image Capturing for Thresholding

If the image capturing process can be controlled, thresholding can be facilitated by a suitable choice of

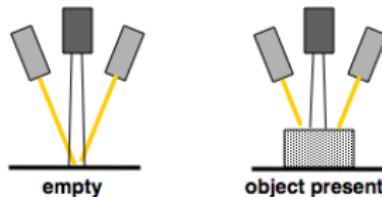
- illumination
- camera position
- object placement
- background greyvalue or colour
- preprocessing

Example: Backlighting

Illumination from the rear gives bright background and shadowed object

Example: Slit illumination

On a conveyor belt illuminated by a light slit at an angle, elevations give rise to displacements which can be recognized by a camera.



fghijkl
qrstuvwxyz
!,:&()

39

Zusammenhang und
Komponentenanalyse

40

Zusammenhang und
Komponentenanalyse

- Komponentenfärbung

41

Konturverfolgung:

Zusammenhang

Definition (Zusammenhang)

Der Zusammenhang ist eine Äquivalenzrelation Z zwischen Pixeln P mit den folgenden Eigenschaften:

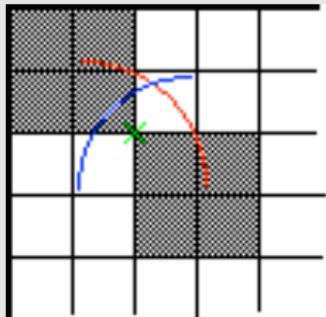
$$Z \subset P \times P$$

Reflexiv $\forall_{x \in P} Z(x, x)$

Symmetrisch $\forall_{x,y \in P} Z(x, y) \rightarrow Z(y, x)$

Transitiv $\forall_{x,y,z \in P} Z(x, y) \wedge Z(y, z) \rightarrow Z(x, z)$

4er- und 8er-Zusammenhang



4er-Zusammenhang: Die Pixel haben eine gemeinsame *Kante*.

8er-Zusammenhang: Die Pixel haben eine gemeinsame *Ecke* oder *Kante*.

Widerspruch:

4er-Zusammenhang: Sowohl Vordergrund als auch Hintergrund hängen nicht zusammen.

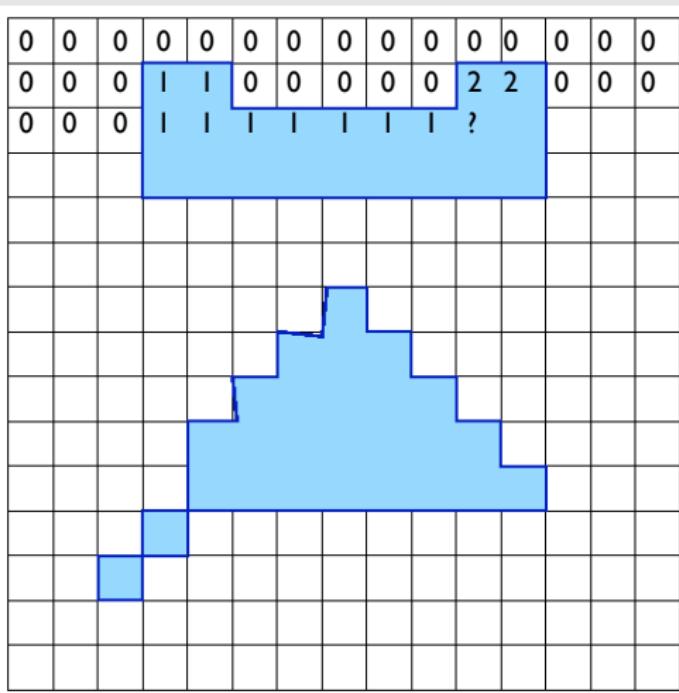
8er-Zusammenhang: Sowohl Vordergrund als auch Hintergrund hängen zusammen.

Bei der Hexagonal-Rasterung gibt es solche Probleme nicht.

Der Zusammenhang ist eine Äquivalenzrelation.

- Zusammenhängende Bildkomponenten zu finden bedeutet eine Unterteilung des Bildes in Äquivalenzklassen.
- Jedem Pixel wird eine eindeutige Komponentennummer (Klasse) zugeordnet.
- Zwei Ansätze:
 - 1 Nummerierung der Pixel
 - 2 oder Konturverfolgung

Komponentennummerierung (Blob-Coloring)



Komponentennumerierung (Blob-Coloring)

Jedem Pixel wird eine eindeutige Nummer zugeordnet, die seiner Äquivalenzklasse entsprechend der Zusammenhangsrelation entspricht.

- ▶ Numeriere die Pixel von links nach rechts, von oben nach unten im Hintergrund mit „0“, im Objekt mit einer laufenden Objektnummer.
- ▶ Wenn man aus dem Hintergrund kommend auf ein Objektpixel trifft, das einen Nachbarn in der oberen Zeile hat, numeriere es mit der Nummer des direkten Nachbarn.

Transitivität

- ▶ Wenn ein Pixel zwei direkte Nachbarn mit verschiedener Objektnummer hat, fahre mit der niedrigeren Nummer fort.
- ▶ Notiere daß die beiden Nummern zur selben Komponente gehören und mische die beiden Komponenten.
- ▶ Zum Mischen der Komponenten eignet sich eine Tabelle (array), in der für jede Klasse eingetragen wird, zu welcher Klasse sie äquivalent ist.

Berechnung von Merkmalen der Komponenten

Im selben Durchgang durch das Bild, in dem die Zusammenhangskomponenten ermittelt werden, können weitere statistische Merkmale über die Komponenten gesammelt werden:

Die Fläche: Zähle die Anzahl der Pixel in jeder Komponente.

Die Schwerpunktskoordinaten: Summiere die Zeilen- und Spaltenkoordinaten der Komponenten.

Die Trägheitsmomente: Summiere die Quadrate der Zeilen- und Spaltenkoordinaten der Komponenten.

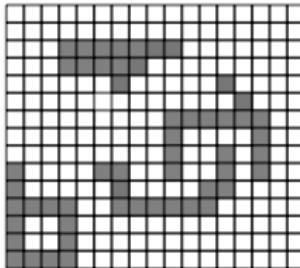
Rauschunterdrückung

- ▶ Komponenten mit sehr kleiner Fläche gehen vermutlich auf Rauschen zurück.
- ▶ Diese können dem Hintergrund zugeordnet werden.

Komponentenanalyse: Schema

Component Labelling

Determining connected regions in B/W images



Component 1
(2 3 9)(3 3 7)(4 6 6)

Component 2
(4 12 12)

Component 3
(5 13 13)(6 9 14)(7 9 9 14 14)(8 9 9 14 14)(9 9 9 14 14)

Component 4
(9 0 0)(10 0 0)(11 0 3)(12 0 0 3 3)(13 0 0 3 3)(14 0 0 3 3)

Component 5
(9 5 6 12 12)(10 6 6 11 12)(11 6 11)

(line fromColumn toColumn)

In this example:
component
descriptions using run-length coding

Component labelling of B/W images with 4-neighbourhood

Scan image left to right, top to bottom:

if pixel is white then continue

if pixel is black then

if left neighbour is white and upper neighbour is white then assign new label

if left neighbour is black and upper neighbour is white then assign left label

if left neighbour is white and upper neighbour is black then assign upper label

if left neighbour is black and upper neighbour is black then

assign left label, merge left label and upper label

Konturverfolgung: Contour Following Bug



Eine Wanze torkelt die Kontur entlang und sammelt dabei die Koordinaten der Kreuzungspunkte.

Konturverfolgung: Contour Following Bug

contour following

Konturverfolgung

Verfahren bei kontinuierlichen Bildfunktionen:

- ▶ Gehe nach rechts, bis die Kontur überschritten wird.
- ▶ In der Figur drehe entgegen dem Uhrzeigersinn.
- ▶ Außerhalb der Figur drehe im Uhrzeigersinn.
- ▶ Beim Überqueren der Kontur notiere den Kreuzungspunkt.

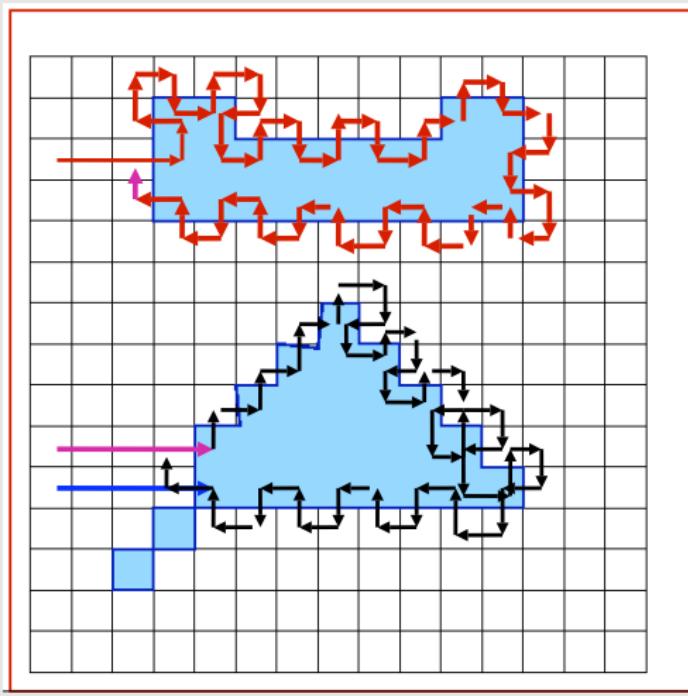


Konturverfolgung im Binärbild

Verfahren bei diskreten Bildfunktionen

- ▶ Gehe Pixel für Pixel nach rechts, bis die Kontur überschritten wird.
- ▶ In der Figur drehe 90° entgegen dem Uhrzeigersinn und gehe ein Pixel weiter.
- ▶ Außerhalb der Figur drehe 90° im Uhrzeigersinn und gehe ein Pixel weiter.
- ▶ Beim Überqueren der Kontur notiere den Kreuzungspunkt.

Konturverfolgung für zwei Komponenten



Mehrdeutigkeit bzgl. Zusammenhang

- ▶ **Warnung:** Es ist nicht garantiert, daß das Verfahren sich eindeutig an eine Zusammenhangsdefinition hält.
- ▶ Sogar **innerhalb** einer Komponente kann der Zusammenhang mal so, mal so ausgewertet werden.

Probleme

Stark gekrümmte Konturen: Wenn die Kontur stärker gekrümmmt ist als der Wendekreis der Wanze, wird sie falsch abgetastet.

Löcher im Objekt müssen separat untersucht werden.

Mehrdeutig bzgl. 4er- und 8er-Zusammenhang.

Teil XVI

Bereichsfinder

Homogene Bereiche



42

Bereichsverschmelzung (Region Growing)

- Histogramverfahren
- Teilen und Verschmelzen (Split-and-Merge)

43

Ballungsanalyse (Clustering)

Bereichsverschmelzer (Region Growing)

Kleine Saatbereichen können zu größeren Bereichen verschmolzen werden, wenn diese ein **Homogenitätskriterium** erfüllen.

Bereichsverschmelzer (Region Growing)

Kleine Saatbereichen können zu größeren Bereichen verschmolzen werden, wenn diese ein **Homogenitätskriterium** erfüllen.

Homogenitätskriterien

- 1 Verschmelze zwei benachbarte Bereiche, wenn der **verschmolzene Bereich** ein Homogenitätskriterium erfüllt, beispielsweise die Varianz der Grauwerte klein genug ist.

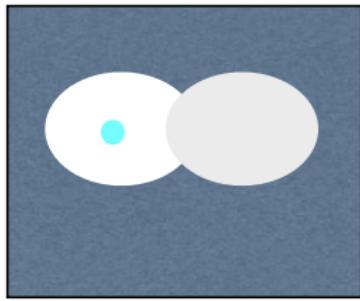
Bereichsverschmelzer (Region Growing)

Kleine Saatbereichen können zu größeren Bereichen verschmolzen werden, wenn diese ein **Homogenitätskriterium** erfüllen.

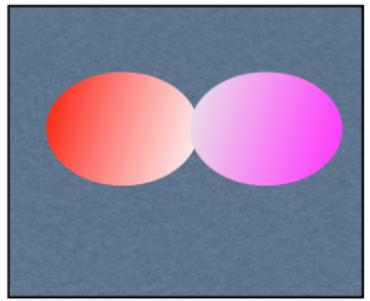
Homogenitätskriterien

- ① Verschmelze zwei benachbarte Bereiche, wenn der **verschmolzene Bereich** ein Homogenitätskriterium erfüllt, beispielsweise die Varianz der Grauwerte klein genug ist.
- ② Verschmelze zwei benachbarte Bereiche, wenn die **Grenze zwischen ihnen** ein Kriterium erfüllt, z.B. nur schwachen Kontrast hat.

Schwierigkeiten bei der Segmentierung

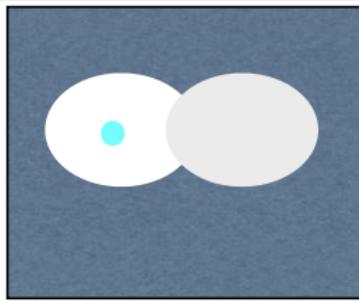


Ungleiche Größe

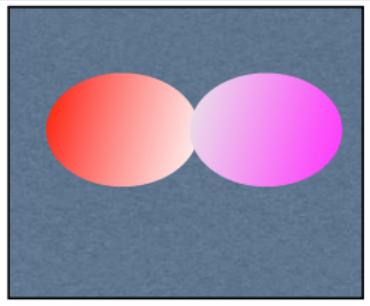


Schwacher Kontrast

Schwierigkeiten bei der Segmentierung



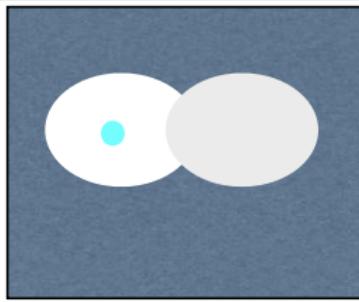
Ungleiche Größe



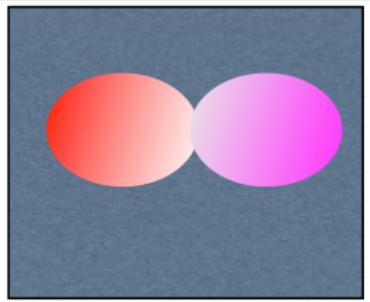
Schwacher Kontrast

- 1 Problem mit (1): Große Bereiche können kleine „verschlucken“, auch wenn sie deutlich verschieden sind.

Schwierigkeiten bei der Segmentierung



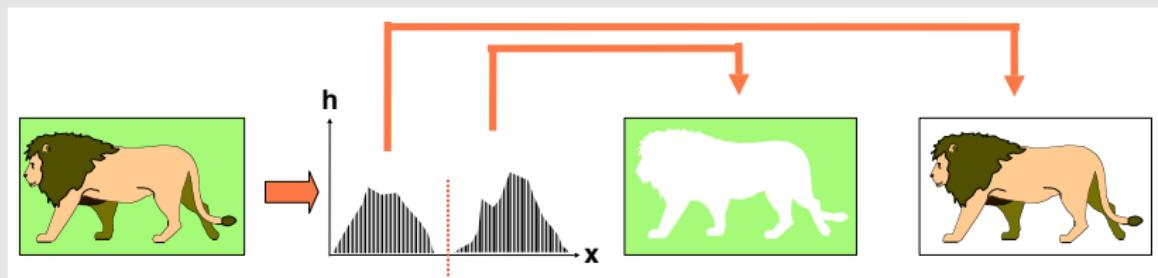
Ungleiche Größe



Schwacher Kontrast

- ① Problem mit (1): Große Bereiche können kleine „verschlucken“, auch wenn sie deutlich verschieden sind.
- ② Problem mit (2): Deutlich verschiedene Bereiche können verschmolzen werden, nur weil die Grenze zwischen ihnen schwach ist.

Rekursive Zerlegung mit Histogrammen



- ▶ Berechne Histogramme für verschiedene Pixelmerkmale (z. B: R, G, B, Intensitäts-Histogramme).
- ▶ Wähle das klarste Histogramm für eine Binarisierung.
- ▶ Wende das Verfahren rekursiv auf die Teilbereiche an.

Schwierigkeiten der Histogrammerzeugung

- ▶ Die Histogramme berücksichtigen nicht die Nachbarschaft der Pixel.

Schwierigkeiten der Histogrammerzeugung

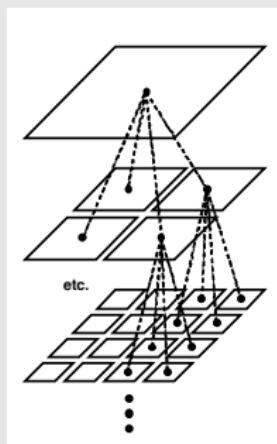
- ▶ Die Histogramme berücksichtigen nicht die Nachbarschaft der Pixel.
- ▶ Die Moden der Histogramme sind manchmal nicht klar zu erkennen.

Schwierigkeiten der Histogrammerzeugung

- ▶ Die Histogramme berücksichtigen nicht die Nachbarschaft der Pixel.
- ▶ Die Moden der Histogramme sind manchmal nicht klar zu erkennen.
- ▶ Frühe Fehlentscheidungen können später nicht korrigiert werden.

Split-and-Merge

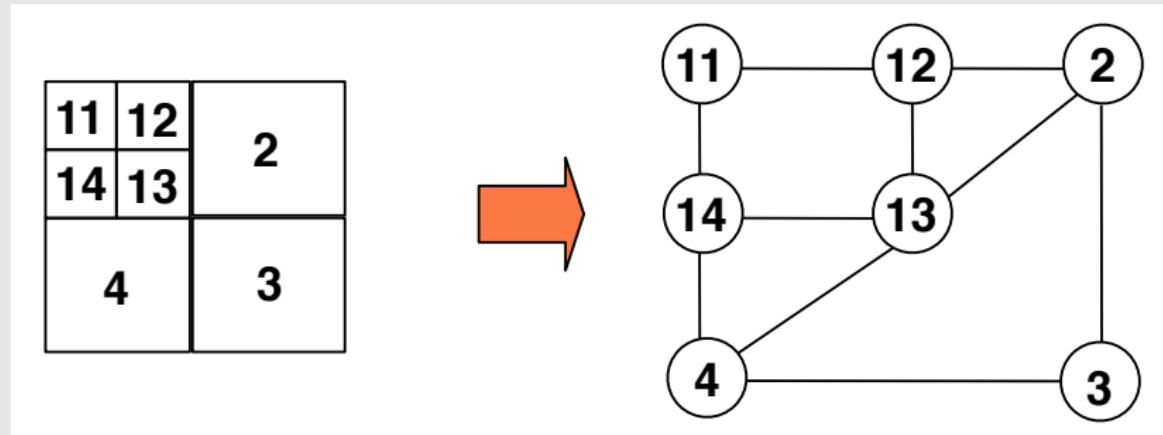
Bereichsgrenzen werden entlang von quadtree-Grenzen festgelegt.



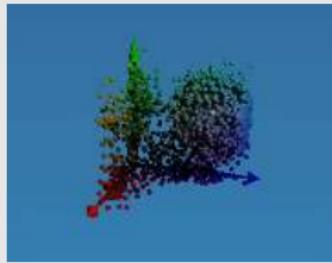
- ▶ Beginne mit einer willkürlichen Bereichszerlegung in einer quadtree-Ebene.
- ▶ Teile jeden Bereich, der ein Homogenitätskriterium verletzt, in vier Kindbereiche.
- ▶ Verschmelze rekursiv alle Bereiche, die zusammen ein Homogenitätskriterium erfüllen.[?]

Der Bereichs-Nachbarschafts-Graph

Als unterstützende Datenstruktur für die merge-Phase ist der Bereichs-Nachbarschafts-Graph (region adjacency graph) nützlich.



Ballungsanalyse (Clustering)



42

Bereichsverschmelzung (Region Growing)

43

Ballungsanalyse (Clustering)

- Segmentierung als Klassifikationsproblem
- Der Merkmalsraum

Segmentierung durch Klassifikation

Idee:

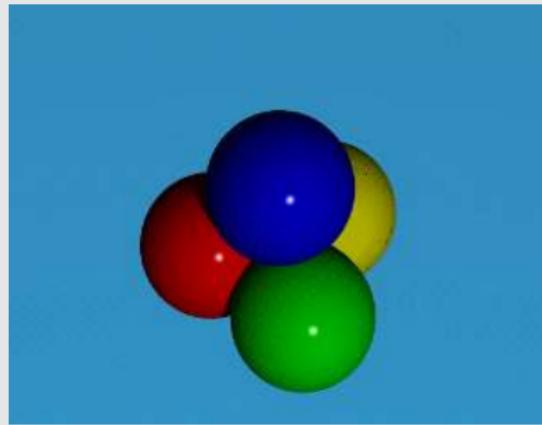
Die Pixel homogener Bereiche bilden kompakte Ballungen im Merkmalsraum, z.B. im Raum der RGB-Vektoren.

Ein Farbbild und die Pixel im RGB-Merkmalsraum

Segmentierung durch Klassifikation

Idee:

Die Pixel homogener Bereiche bilden kompakte Ballungen im Merkmalsraum, z.B. im Raum der RGB-Vektoren.

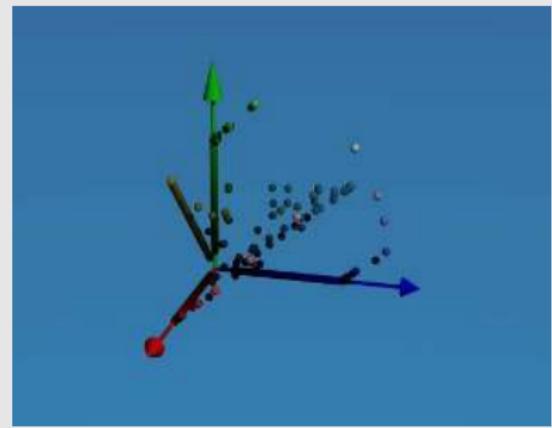
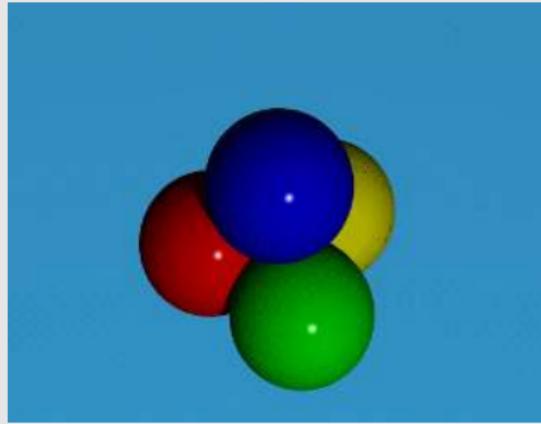


Ein Farbbild und die Pixel im RGB-Merkmalsraum

Segmentierung durch Klassifikation

Idee:

Die Pixel homogener Bereiche bilden kompakte Ballungen im Merkmalsraum, z.B. im Raum der RGB-Vektoren.

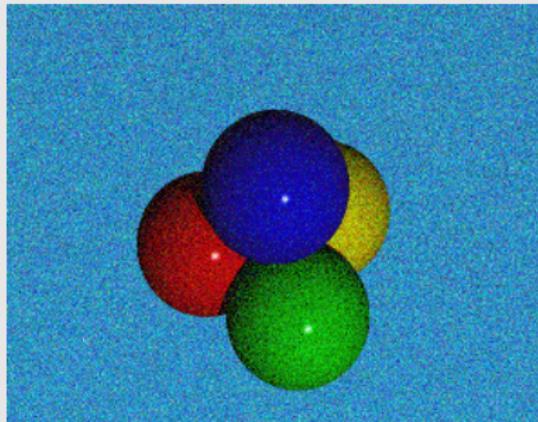


Ein Farbbild und die Pixel im RGB-Merkmalsraum

Der RGB-Merkmalsraum, synthetisches Bild

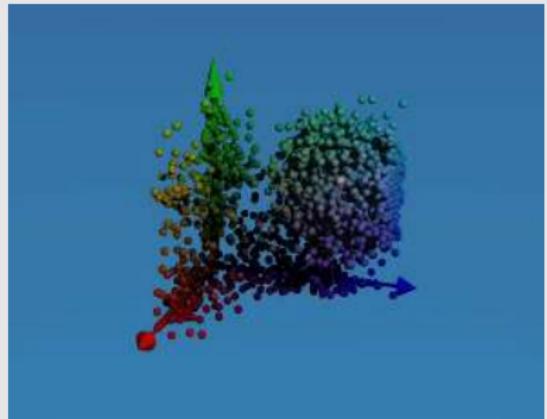
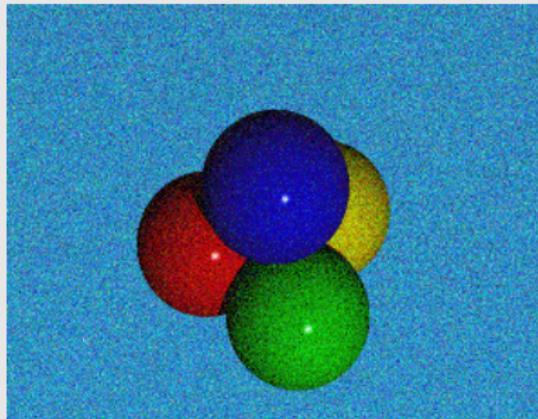
Bild unverrauscht

Rauschen vergrößert die Ballungen



Ein verauschtet Bild und die Pixel im RGB-Merkmalsraum

Rauschen vergrößert die Ballungen



Ein verauschttes Bild und die Pixel im RGB-Merkmalsraum

Der RGB-Merkmalsraum: Verrauscht Bild

Bild verrauscht

Testbild Porsche



Der RGB-Merkmalsraum: Verrauscht Bild

Bild verrauscht

Teil XVII

Kantenfinder

Kantenfinder



- 44 Objektgrenzen und Kanten
 - Entstehung von Kanten
- 45 Gradientenbasierte Verfahren
- 46 Laplace-basierte Operatoren

Heuristischer Ansatz:

An Objektgrenzen ändert sich das Oberflächenmaterial und damit das Reflexionsvermögen.

- ▶ Änderungen im Reflexionsvermögen führen zu Änderungen in der Abbildung des Objektes.
- ▶ Aus Diskontinuitäten der Abbildung kann daher auf Objektgrenzen geschlossen werden.

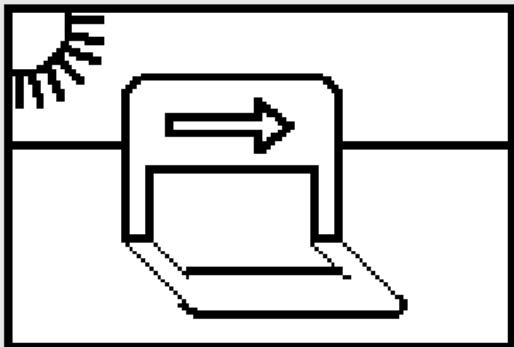
Vorsicht!

Nicht alle Diskontinuitäten im Bild gehen auf Materialgrenzen zurück!

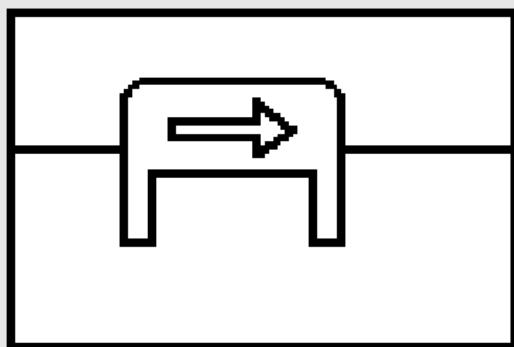
Weitere Ursachen für Grauwertdiskontinuitäten:

- ▶ Änderung der Oberflächeneigenschaften (Farbe, Material, Glattheit)
- ▶ Änderung der Oberflächenorientierung zum Betrachter
- ▶ Änderung der Beleuchtung (Schatten, Sekundärbeleuchtung)
- ▶ Rauschen, Quantisierung

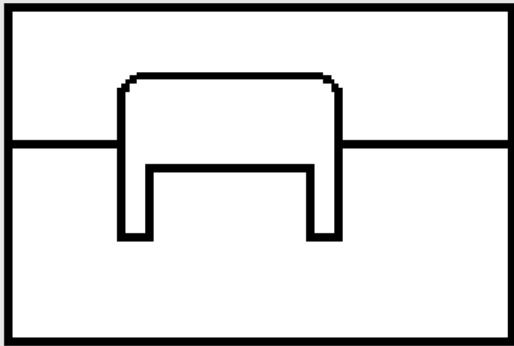
Entstehung von Kanten



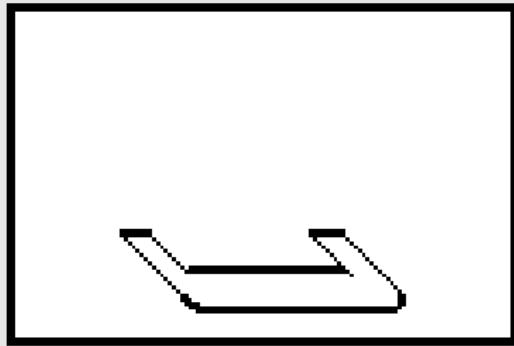
Ein Bild



Reflektanzkanten



Orientierungskanten



Beleuchtungskanten

Kontrastumkehr

Kanten in Realweltbildern



Der Kontrast entlang einer Kante kann abhängig von der Beleuchtung stark variieren und sich gelegentlich auch umkehren: siehe die senkrechte Kante in Bildmitte.

©Wolfgang
Förstner

Ansätze zur kantenbasierten Segmentierung

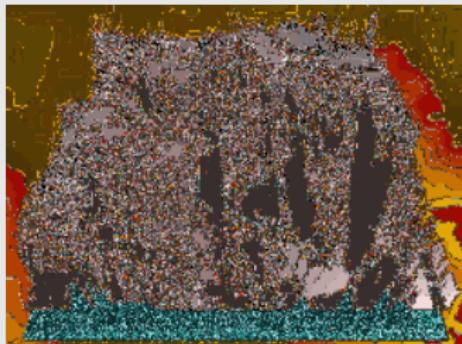
Definition (**Kante**)

Eine starke lokale Variation der Helligkeit (oder Farbe) im Bild heißt Kante.

Es gibt hunderte von Ansätzen zum Kantenfinden; im Wesentlichen drei Gruppen von Vorgehensweisen:

- ▶ Approximation des Gradienten oder des Laplace-Operators (Roberts, Sobel, Marr, Canny,...)
- ▶ Diskrete Näherung, Reihenentwicklung,
- ▶ Schablonentest (Kirsch), Anpassung von parametrisierbaren Kantenmodellen (Hummel)
- ▶ Hypothesentest (Yakimovski)

Gradientenbasierte Kantenfinder



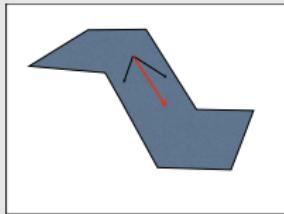
- ▶ Ableitungen der Bildfunktion
Digitale Gradienten,
Kantenoperatoren
- ▶ Ableitungen 2. Ordnung
Laplace,
- ▶ Robuste Kantenfinder: Marr
und Hildreth, Canny

Vorgehensweise: Ableitungsorientiert

Pixel mit starker Variation der Bildfunktion erkennen wir an hohen Werten der **ersten Ableitung** der Bildfunktion.

Vorgehensweise zur Kantendetektion:

- ▶ Berechne den Gradienten der Bildfunktion.
- ▶ Suche lokale Maxima des Gradientenbetrages.
- ▶ Die so gefundenen Bildpunkte sind Kandidaten für Kantenelemente (**edgel**).



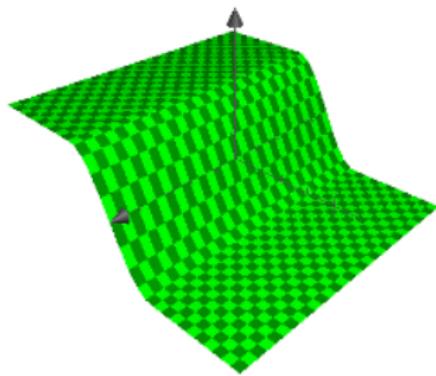
Gegeben: Eine Bildfunktion $g(x, y)$.

Sei $\vec{\nabla}_{(x,y)} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix}$ der Gradientenoperator.

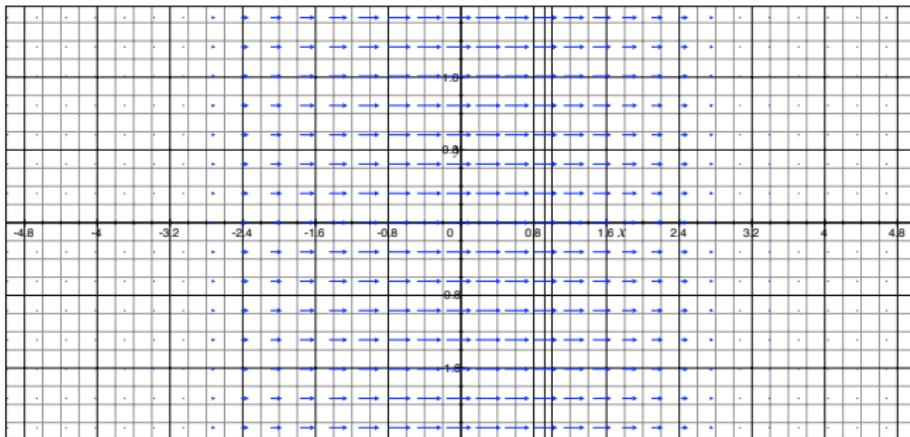
$\|\vec{\nabla}_{(x,y)} g(x, y)\|$ = ein Maß für die Variation von $g(x, y)$.

$\vec{\nabla}_{(x,y)} g(x, y)$ zeigt in die Richtung des stärksten Grauwertgefälles.

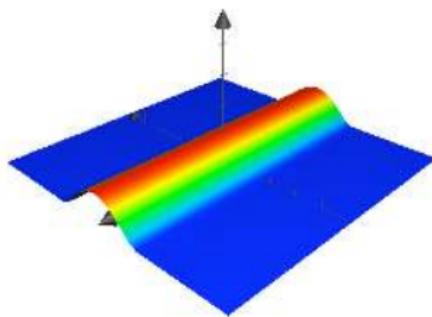
Beispiel: Stufenkante



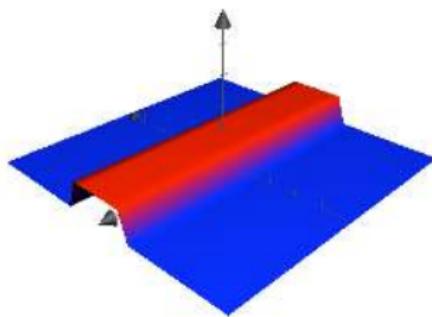
Gradientenfeld der Stufenkante



Gradientenbetrag der Stufenkante

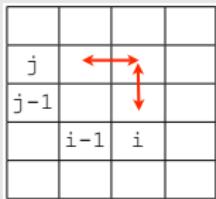


Gradientenbetrag mit Schwellenwert



Gradienten im Digitalbild

Diskrete Ableitungen in x , y und in Richtung θ :
Differenzen statt Differentiale



$$\Delta_x g(i, j) = g(i, j) - g(i - 1, j)$$

$$\Delta_y g(i, j) = g(i, j) - g(i, j - 1)$$

$$\Delta_\theta g(i, j) = \Delta_x g(i, j) \cdot \cos \theta + \Delta_y g(i, j) \cdot \sin \theta$$

Bildaus-
schnitt

Digitaler Gradient: (Betrag)

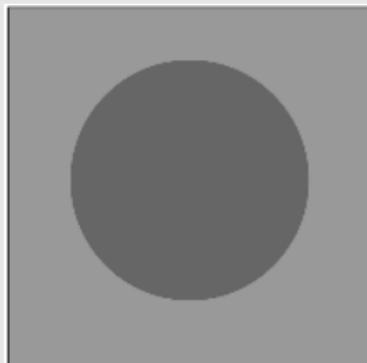
$$||\vec{\nabla}_{(x, y)} g(i, j)|| = \sqrt{(\Delta_x g(i, j))^2 + (\Delta_y g(i, j))^2}$$

oft vereinfacht zu $\approx |\Delta_x g(i, j)| + |\Delta_y g(i, j)|$

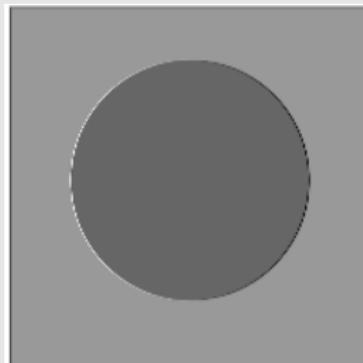
oder $\approx \max(|\Delta_x g(i, j)|, |\Delta_y g(i, j)|)$

Genaugigkeit der Näherung

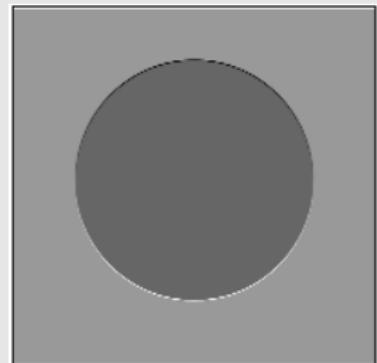
- ▶ Die Näherung über die Summe der Absolutwerte ist nur für horizontale und vertikale Geraden exakt.
- ▶ In den diagonalen Richtungen ist der Wert um den Faktor $\sqrt{2}$ falsch.



Original



$\Delta_x g(i, j)$

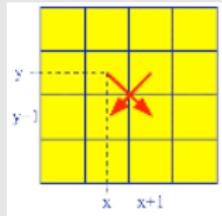


$\Delta_y g(i, j)$
dem Original überlagert.

Roberts-Cross-Operator

Ein symmetrischer digitaler Gradient:

Roberts cross operator $\mathcal{R}(x, y)$:



$$\Delta_1 = g(x, y) - g(x + 1, y + 1)$$

$$\Delta_2 = g(x, y + 1) - g(x + 1, y)$$

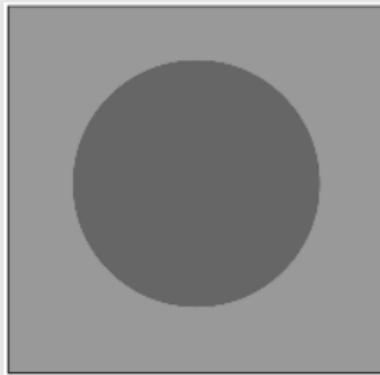
$$||\vec{\nabla}(x, y)g(x, y)|| \approx \mathcal{R}(x, y)$$

Näherungen

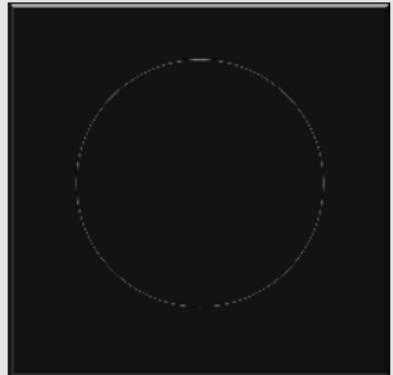
$$\mathcal{R}(x, y) = \sqrt{\Delta_1^2 + \Delta_2^2}$$

$$\mathcal{R}(x, y) \approx \mathcal{F}(x, y) = |\Delta_1| + |\Delta_2|$$

Beispiel: Roberts-Cross-Operator

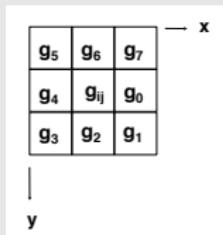


Original



Roberts-Cross-Operator-Maxima

Sobel-Operator



- ▶ Berechnet die Gradientenkomponenten Δ_x und Δ_y anhand einer 3×3 -Umgebung.
- ▶ Die Bildfunktion wird gleichzeitig durch mittenbetontes Mitteln geglättet.

$$\Delta g_x = (g_1 + 2g_0 + g_7) - (g_3 + 2g_4 + g_5)$$

$$\Delta g_y = (g_1 + 2g_2 + g_3) - (g_7 + 2g_6 + g_5)$$

$$||\vec{\nabla}_{(x,y)} g_{ij}|| = \sqrt{\Delta g_x^2 + \Delta g_y^2}$$

$$\tan \gamma = \frac{\Delta g_y}{\Delta g_x}$$

Der Sobel-Operator in Maskendarstellung

-1	0	1
-2	0	2
-1	0	1

Δ_x

-1	-2	-1
0	0	0
1	2	1

Δ_y

Vorteil:

Der Sobel-Operator unterdrückt wegen der größeren Operator-Maske das Rauschen besser als der Roberts-Cross-Operator.

Beispiel: Sobel

1	1	1
0	1	1
0	0	1

Pixel des Bildes

$$\begin{aligned}\Delta g_x &= -1 - 0 - 0 + 1 + 2 + 1 = 3 \\ \Delta g_y &= -1 - 2 - 1 + 0 + 0 + 1 = -3 \\ \tan \gamma &= \frac{\Delta g_y}{\Delta g_x} = -1 \\ \gamma &= -45^\circ\end{aligned}$$

Beispiel 2: Sobel

0	1	2
0	1	2
0	1	2

Pixel des Bildes

$$\Delta g_x = -0 - 0 - 0 + 2 + 4 + 2 = 8$$

$$\Delta g_y = -0 - 2 - 2 + 0 + 2 + 2 = 0$$

$$\tan \gamma = \frac{\Delta g_y}{\Delta g_x} = 0$$

$$\gamma = 0^\circ$$

Beispiel: Sobel-Operator 3



$g(x, y)$
greyvalue image

0 = black
255 = white

Δg_x
x-component of
greyvalue gradient

0 = greyvalue 128

Δg_y
y-component of
greyvalue gradient

0 = greyvalue 128

Objektgrenzen und Kanten
Gradientenbasierte Verfahren
Laplace-basierte Operatoren

Der Laplace-Operator
Der Mexican-Hat Operator
Der Canny-Operator

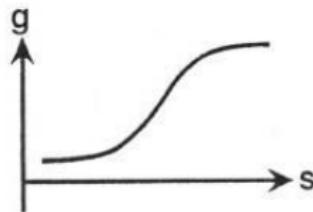
Teil XVII

Kantenfinder

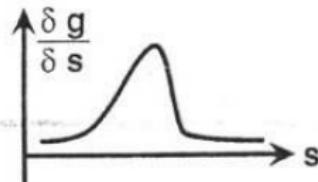
Der Laplace-Operator

- ▶ Nulldurchgänge der 2. Ableitung der Bildfunktion weisen auf Extremwerte der 1. Ableitung hin.
- ▶ Daher können steile Gradienten der Bildfunktion mittels des Laplace-Operators gefunden werden.
- ▶ Der Laplace-Operator ∇^2 , angewendet auf eine Bildfunktion g , liefert ein richtungsunabhängiges Maß für die 2. Ableitung von g .

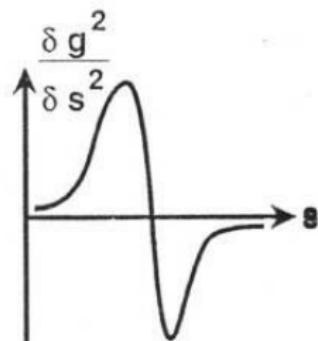
$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$



Grauwertverlauf senkrecht zu einer Kante ("Kantenmodell")



Verlauf des Gradienten-Betrages



Verlauf der 2. Ableitung

Diskrete Approximation des Laplace-Operators

Zur diskreten Approximation der partiellen 2. Ableitungen werden Differenzen von Differenzen berechnet.

$g_{i-1,j-1}$	$g_{i,j-1}$	$g_{i+1,j-1}$
$g_{i-1,j}$	g_{ij}	$g_{i+1,j}$
$g_{i-1,j+1}$	$g_{i,j+1}$	$g_{i+1,j+1}$

$$\begin{aligned}\nabla^2 g_{ij} &= (g_{i+1,j} - \textcolor{red}{g_{ij}}) - (\textcolor{red}{g_{ij}} - g_{i-1,j}) \\ &\quad + (g_{ij+1} - \textcolor{red}{g_{ij}}) - (\textcolor{red}{g_{ij}} - g_{ij-1}) \\ &= g_{i+1,j} + g_{i-1,j} + g_{ij+1} + g_{ij-1} - 4\textcolor{red}{g_{ij}}\end{aligned}$$

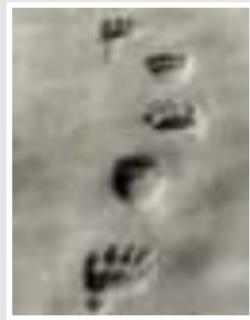
Pixel um g_{ij}

Die diskrete Approximation des Laplace-Operators berechnet den Unterschied zwischen dem Grauwert eines Pixels und dem Mittelwert der Grauwerte der Umgebungspixel.

Beispiel: Laplace-Operator



Originalbild



Laplace-Operator

- ☞ Der Laplace-Operator ist extrem rauschempfindlich, da die 2. Ableitung das Rauschen betont.
- ☞ Er sollte nur auf gefilterten oder rauscharmen Bildern angewendet werden.

Beispiele

1 1 1

1 **1** 1

$$\nabla^2 f = 4 - 4 = 0$$

1 1 1

1 2 3

1 **2** 3

$$\nabla^2 f = 2 + 2 + 1 + 3 - 8 = 0$$

1 2 3

1 2 1

1 **2** 1

$$\nabla^2 f = 6 - 8 = -2$$

1 2 1

1 1 3

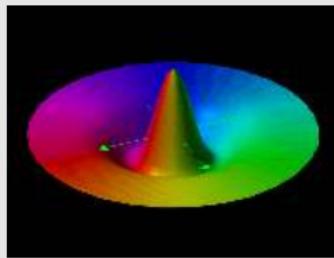
1 **1** 3

$$\nabla^2 f = 6 - 4 = 2$$

1 1 3

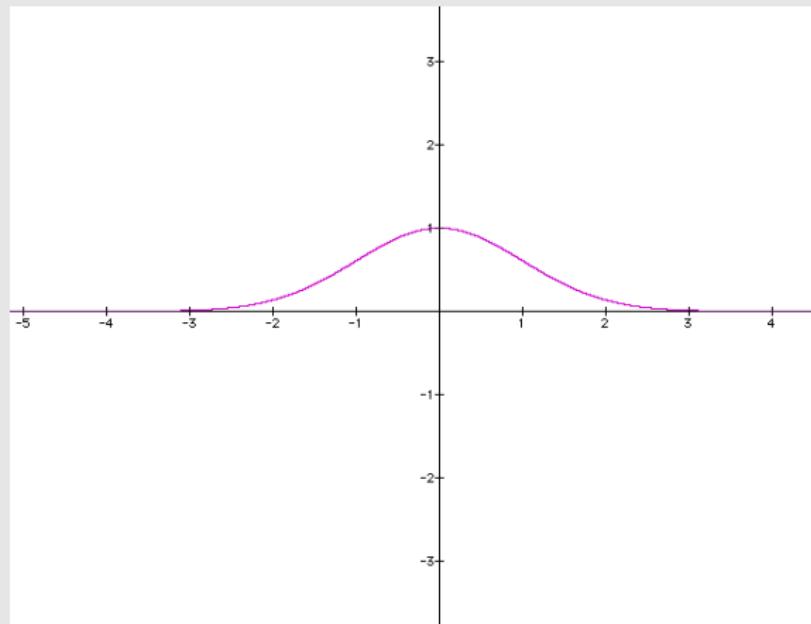
Der Mexican-Hat Operator

Marr und Hildreth 1980



- ▶ Glätten des Bildes mit einem Gauß-Filter
- ▶ Lokalisation der Extremwerte des Gradienten durch die Nulldurchgänge der 2. Ableitung

Querschnitt eines Gauß-Filters, $\sigma = 1$

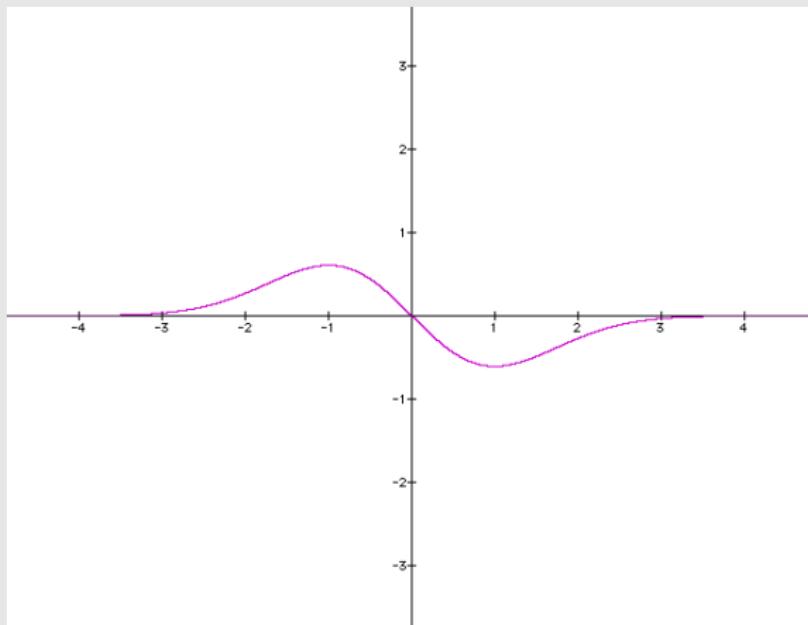


$$Z = e^{-\frac{r^2}{2}}$$

2D-Gauß-Filter, $\sigma = 1$

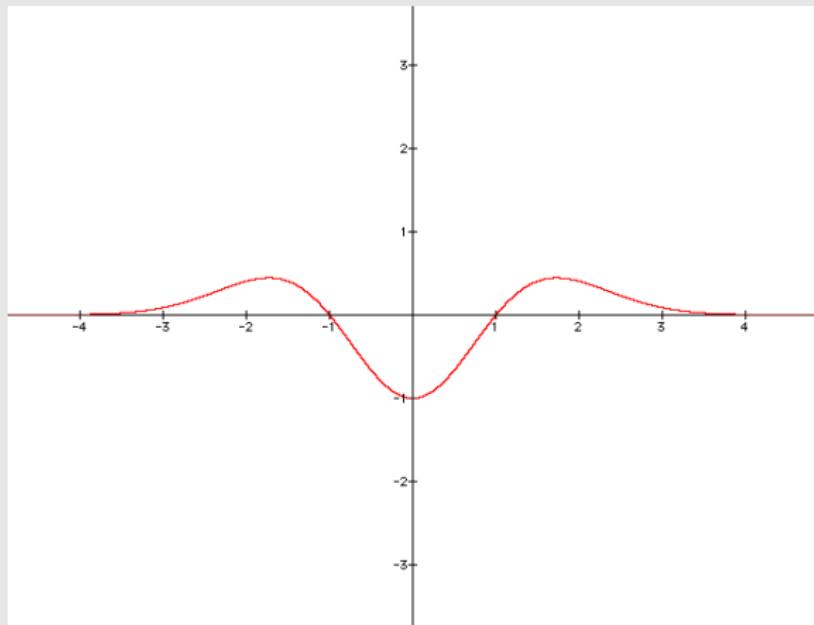
Glockenkurve

Erste Ableitung der Gauß-Funktion, $\sigma = 1$



$$z = -r \cdot e^{-\frac{r^2}{2}}$$

Zweite Ableitung der Gauß-Funktion, $\sigma = 1$

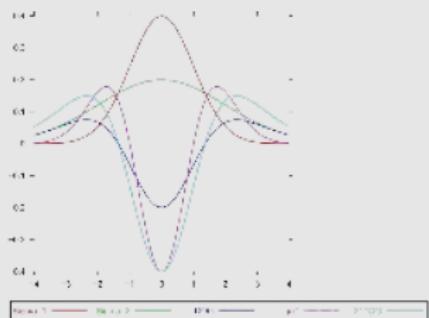


$$z = (r^2 - 1) \cdot e^{-\frac{r^2}{2}}$$

Zweite Ableitung der Gauß-Funktion, $\sigma = 1$

Der Mexican-Hat Operator: Ein umgekehrter Sombrero

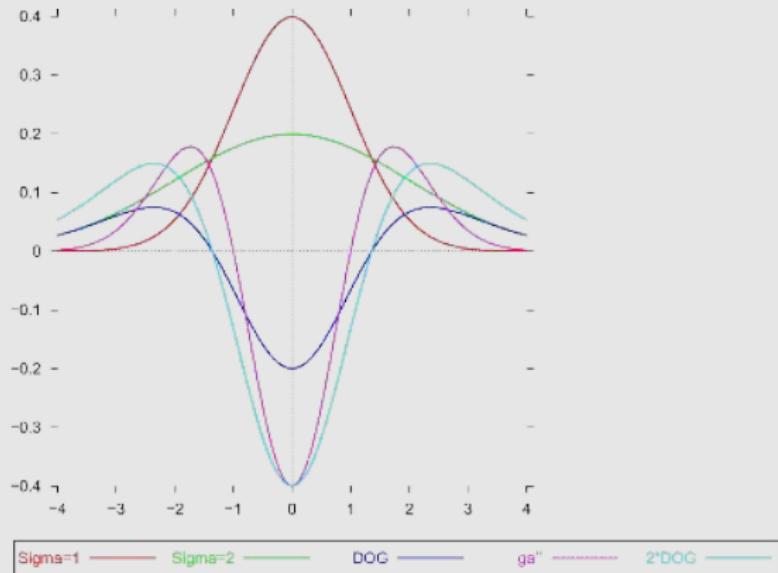
Eine andere Interpretation: DoG-Operator



Difference of Gaussians (DOG):

- ▶ Die 2. Ableitung einer Gauß-Funktion kann als Differenz zweier Gauß-Funktionen mit unterschiedlicher Varianz approximiert werden.
- ▶ So lassen sich rezeptive Felder der Netzhaut modellieren
(On/Off-Zellen)[Marr, 1976]

Differenz zweier Gauß-Funktionen mit unterschiedlicher Varianz



Der DoG-Operator: Ein umgekehrter Sombrero

Marr-Hildreth-Operator: Robuster Operator

Suche Kanten als Nulldurchgänge der 2. Ableitung eines geglätteten Bildes: [Marr and Hildreth, 1980]

Laplacian of Gaussian (LoG):

$$\nabla^2[f(x, y, \sigma) \cdot g(x, y)]$$

Gauß-Filter:

$$f(x, y) = e^{\frac{x^2+y^2}{2\sigma^2}}$$

Umkehren von Differenzieren und Falten:

$$\nabla^2[f(x, y, \sigma)] \cdot g(x, y) = h(x, y) \cdot g(x, y)$$

$$h(x, y) = c \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{\frac{x^2+y^2}{2\sigma^2}}$$

Diskrete Approximation von $h(x, y)$

$$\begin{vmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{vmatrix} \approx h(x, y) = c \left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{\frac{x^2+y^2}{2\sigma^2}}$$

Difference of Gaussians (DoG)

Der Mexican-Hat-Operator kann als DoG-Operator approximiert werden.

$$\begin{aligned}f(x, y, \sigma) &= e^{\frac{x^2+y^2}{2\sigma^2}} \\h(x, y) &\approx f(x, y, \sigma_1) - f(x, y, \sigma_2)\end{aligned}$$

Die beste Approximation ergibt sich für

$$\sigma_2 \approx 1.6\sigma_1.$$

Beispiel: DoG



Originalbild

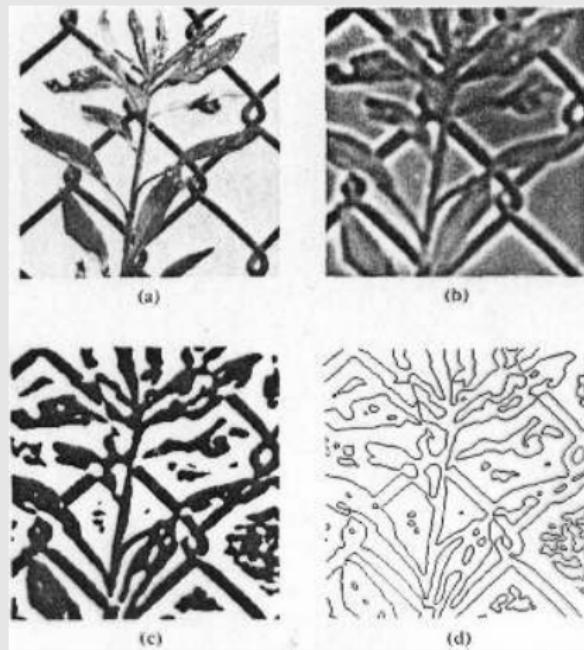


DoG-Filterung mit $\sigma_1 = 1, \sigma_2 = 1.6$

Kantensuche mit dem Mexican-Hat-Operator

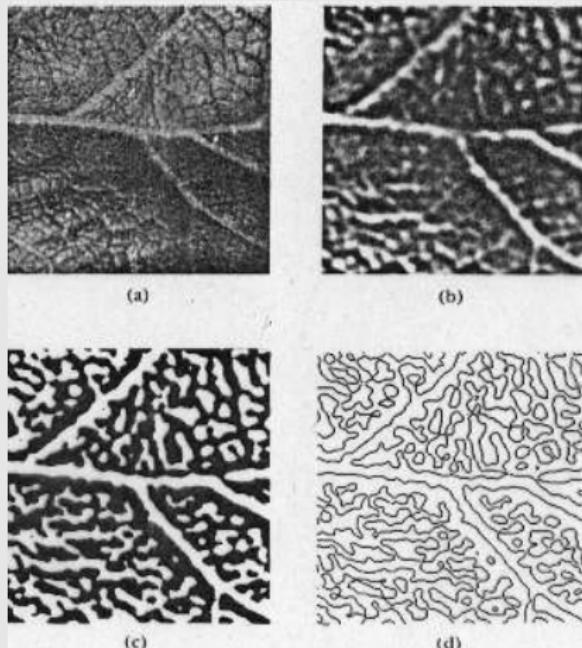
- ▶ Die Nulldurchgänge des mit dem Mexican-Hat-Operator gefilterten Bildes weisen auf Extremwerte des Gradienten hin.
- ▶ Kantenzüge werden gefunden, indem Nulldurchgangslinien verfolgt werden.
- ▶ Das Verfahren liefert also nicht isolierte edgel, sondern geschlossene Kantenzüge.

Beispiel: Maschendrahtzaun



Original, Mexican-Hat-Resultat, positive und negative
Mexican-Hat-Resultate, Nulldurchgänge

Beispiel: Blattstrukturen



Original, Mexican-Hat-Resultat, positive und negative
Mexican-Hat-Resultate, Nulldurchgänge

Beispiel

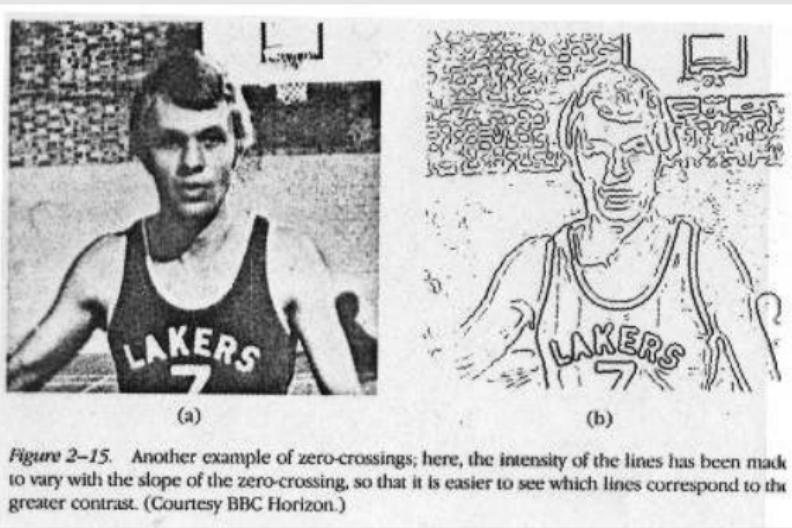
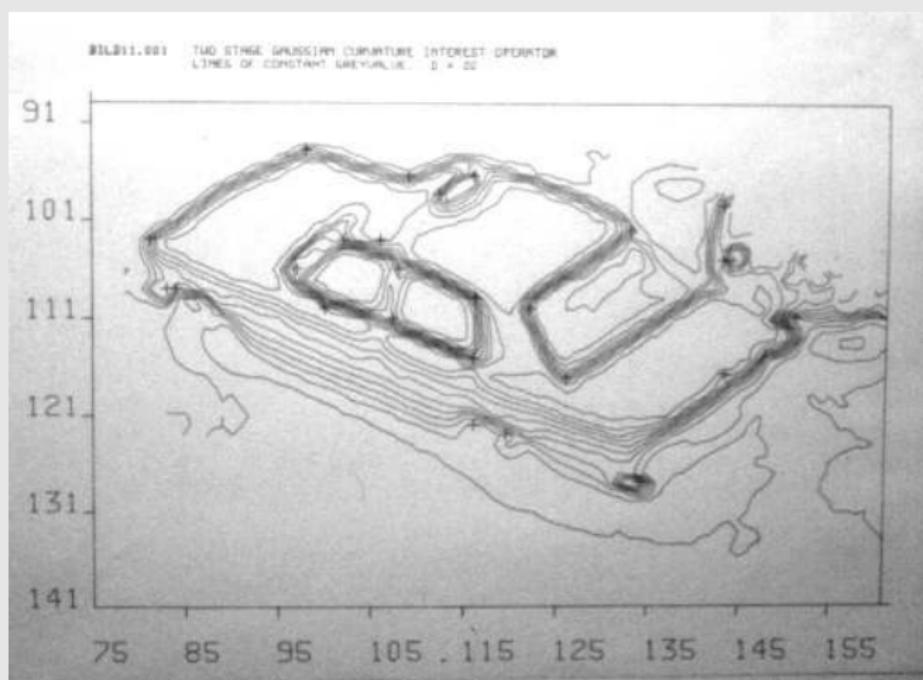


Figure 2-15. Another example of zero-crossings; here, the intensity of the lines has been made to vary with the slope of the zero-crossing, so that it is easier to see which lines correspond to the greater contrast. (Courtesy BBC Horizon.)

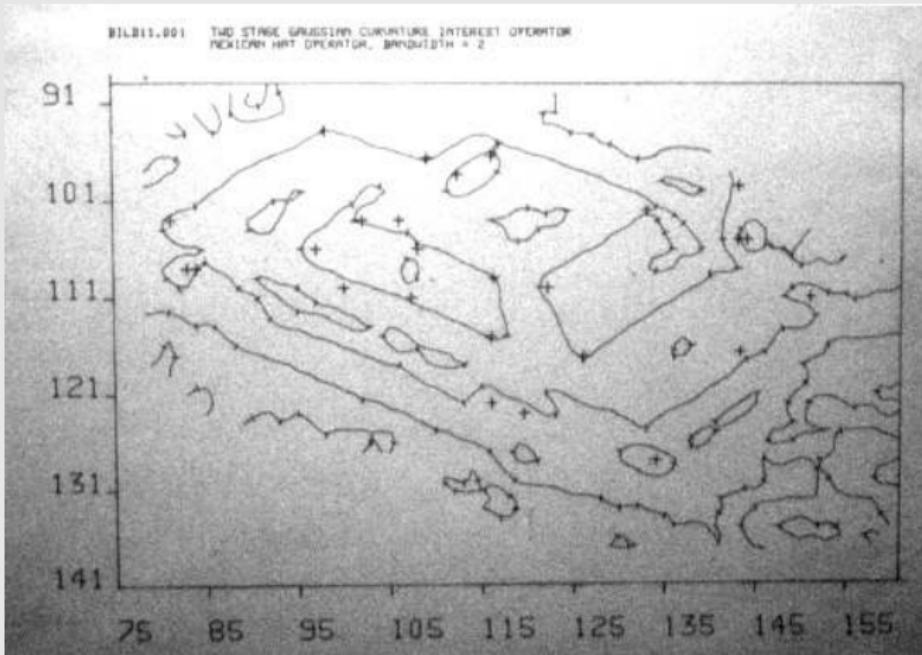
Beispiel: Taxi-Szene, Originalbild



Beispiel: Taxi-Szene: Höhenlinien des Grauwertgebirges



Beispiel: Taxi-Szene



Nulldurchgangslinien des Mexican-Hat-Operators,
Durchmesser 5

Edge-Work: Das Vorzeichen der 2. Ableitung

Edge-Work-Filter

Gradientenbasiert vs. Laplace-basiert

Schwelle an den Gradienten:

- ⊕ Sichere Detektion von starken Kanten, wenn der Operator auch glättet.
- ⊖ Schlechte Lokalisation, Mehrfachdetektion
- ⊖ Nur edgel, keine Kantenzüge

Nulldurchgänge des Laplace-Operators:

- ⊕ Durchgehende Kantenzüge
- ⊕ Gute Lokalisation
- ⊖ Rauschempfindlichkeit
- ⊖ unterscheidet nicht zwischen schwachen und starken Kanten.

Der Canny-Operator

Ein optimaler Kantendetektor für Stufenkanten, die durch weißes Rauschen gestört sind. [Canny, 1986]

Optimalitätskriterien:

Lokalisierung: Die Kante soll da gefunden werden, wo der Gradient am steilsten ist.

Eindeutigkeit: Jede Kante soll nur einmal gefunden werden.

Robustheit: Jede Kante soll im passenden Frequenzbereich gefunden werden.

Lösungsansatz

Idee:

Kombination der guten Lokalisation des Mexican-Hat-Operators mit der Robustheit eines Gradientenoperators.

- ▶ Berechne die Richtung und den Betrag des Gradienten mit Ableitungsfiltern, die eine Gauß-Glättung beinhalten (1. Ableitung der Gaußfunktion).
- ▶ Lokalisiere die Kante bei den Nulldurchgängen der 2. Ableitung.
- ▶ Unterdrücke schwache Kanten durch eine Hysteresis-Schwelle an den Gradientenbetrag.

Canny-Detektionsschema

Normalenvektor \vec{n} zur Kantenrichtung: f Gauß-Filter,
 g Grauwertbild:

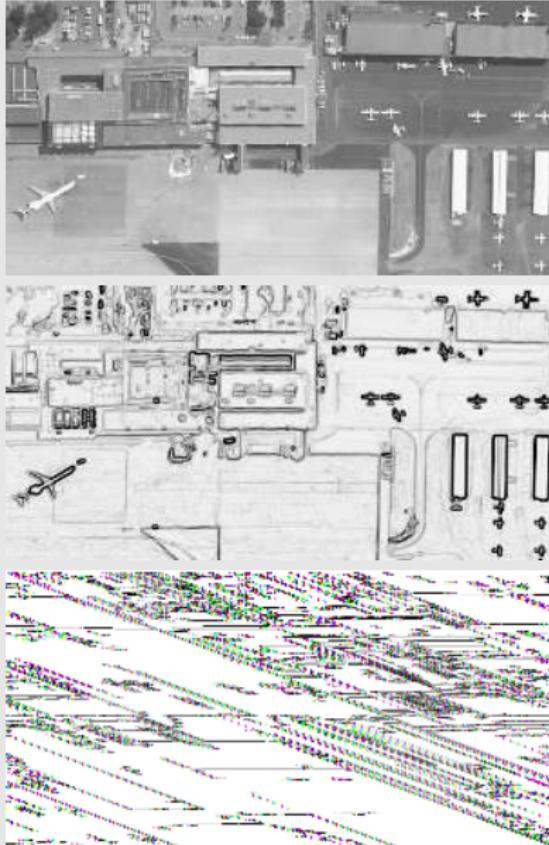
$$\vec{n} = \frac{\nabla(f \cdot g)}{|\nabla(f \cdot g)|}$$

Lokalisation: Die Kante liegt beim lokalen Maximum von $|\nabla(f \cdot g)|$ in Gradientenrichtung.

$$\frac{\partial^2}{\partial} f \cdot g = 0$$

Kantenstärke: $|\nabla(f \cdot g)|$

Beispiel: Canny-Operator



Beispiel (Flughafen Nürnberg)

Originalbild, Betrag des Gradienten und mit dem Canny-Operator lokalisierte Kanten

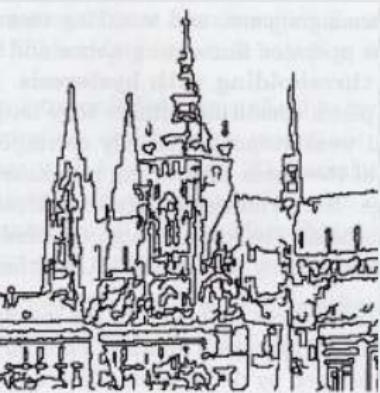
Multi-Skalen-Ansatz

- ▶ Bei komplexen Szenen können die Kanten unterschiedliche Gauß-Filter erfordern.
 - ▶ Scharfe Kanten erfordern kleine Werte für σ ,
 - ▶ weiche Kanten hohe Werte für σ .
- ▶ Das Canny-Verfahren analysiert das Bild bei verschiedenen Werten von σ und aggregiert die gefundenen Kanten über einen speziellen *feature synthesis*-Algorithmus.

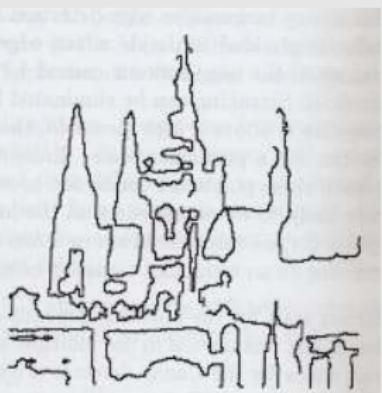
Skalenvergleich



Original



Canny Operator $\sigma = 1$



Canny Operator $\sigma = 2.8$

Hysteresis-Schwellenwert



Alle edgel

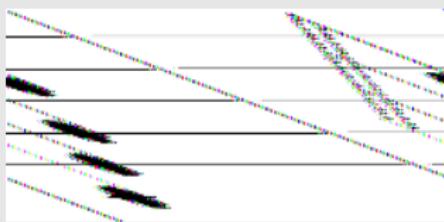


gefiltert

Teil XVIII

Gruppierung von Kanten

Hough-Transformation



47

Hough-Transformation

- Beispiele für Parzen-Fenster-Schätzung
- Senkrechte Geraden

Geradensuche

Für viele Bildverarbeitungsaufgaben ist es nötig, Geraden oder andere regelmäßige Linien erkennen zu können, beispielsweise zur

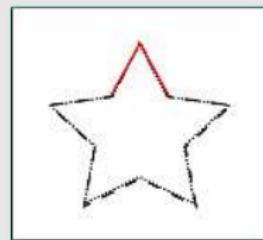
Geraden sind wichtig zur

- ▶ Erkennung von Gebäuden und Straßen,
- ▶ Analyse von Dokumenten,
- ▶ Auswertung von Fluchtpunkten und Perspektive,
- ▶ Erkennung natürliche Objekte, wie Baumstämme.

Edgel \Rightarrow Geraden: Hough-Transformation



Bild



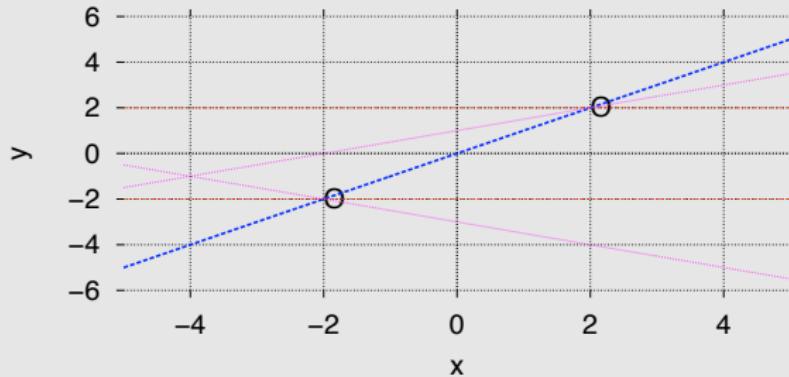
edgel und einige Geraden

Problem:

Welche Teilmengen der edgel können Geradenstücken zugeordnet werden?

- ☞ Ein edgel repräsentiert nur einen **Punkt** auf der möglichen Geraden und steht für ein ganzes **Geradenbüschel** von unendlich vielen Geraden, die durch diesen Punkt gehen könnten.

Geradenbüschel im Bildraum



Zwei edgel und
einige Geraden
ihrer Geraden-
büschel

$y = x$	$y = 0.5x + 1$
$y = 2$	$y = -0.5x - 1$
$y = -2$	

Gesucht:

Die Gerade, die im Durchschnitt aller Geradenbüschel der edgel liegt.

Gruppieren von edgeln: Hough-Transformation

Wir überlagern dem Bildraster ein $x - y$ -System. Dann können wir Geraden im Bild durch Gleichungen der Form

$$y = \underbrace{m}_{\text{Steigung}} x + \underbrace{c}_{\text{Achsenabschnitt}}$$

Sei (x_0, y_0) ein Bildpunkt, an dem eine Kante gefunden wurde — ein **edgel**. Alle Geraden durch diesen Punkt erfüllen die Gleichung:

$$\begin{aligned} y_0 &= mx_0 + c \\ c &= \underbrace{-x_0}_{\text{Steigung}} \cdot m + \underbrace{y_0}_{\text{Achsenabschnitt}} \end{aligned}$$

[Hough, 1962]

Parameterraum

$$c = \underbrace{-x_0}_{\text{Steigung}} \cdot m + \underbrace{y_0}_{\text{Achsenabschnitt}}$$

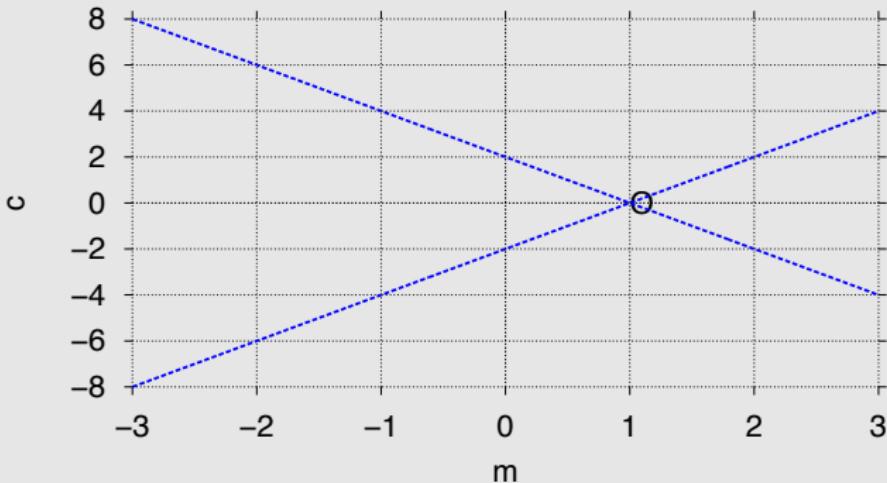
Für die Geraden eines Geradenbüschels besteht ein linearer Zusammenhang zwischen den Parametern m und c .

Wenn wir im Parameterraum m gegen c auftragen, dann liegen die Parameter der Geraden eines Geradenbüschels wieder auf einer Geraden.

- ☞ Die Parameter einer Geraden, die zu mehreren Geradenbüscheln gehört, liegen im Schnittpunkt der Parametergeraden der Büschel.

Geradenbüschel im Parameterraum $M \times C$

Der Parameterraum: Parzen–Schätzung für Geraden



$$c = -2m + 2 \quad \text{-----}$$

$$c = 2m - 2 \quad \text{-----}$$

Evidenz für eine Gerade im Bildraum

Es gibt zwei Verfahren, die Evidenzen zu kombinieren:

- ① **Das Parzen-Fensterverfahren**, siehe
[Duda and Hart, 1973]: Das Parzen-Fensterverfahren ist eine numerische Schätzung für eine kontinuierliche Wahrscheinlichkeitsdichte $p(m, c)$.
- ② **Die Hough-Transformation**: Ein Histogram-Ansatz:
[Ballard and Brown, 1982] Der Parameterraum wird diskretisiert und in (m, c) -Zellen unterteilt. In jeder Zelle wird gezählt, wieviele Geraden g_i diese Zelle passieren.

Bei beiden Verfahren geben lokale Maxima Hinweise auf mögliche Geraden.

Geradensuche mit dem Parzen-Fenster-Verfahren:

- ▶ Wir suchen diejenigen Paare (m, c) , die aufgrund der beobachteten edgel am wahrscheinlichsten sind.
- ▶ Wir haben eine Stichprobe von n edgeln, die uns eine Menge von Geraden g_i im MC-Parameterraum definieren.
- ▶ Wir schätzen $p(m, c)$, indem wir für den Punkt (m, c) im MC-Parameterraum den Abstand $\Delta((m, c), g_i)$ zu jeder Geraden g_i bestimmen.

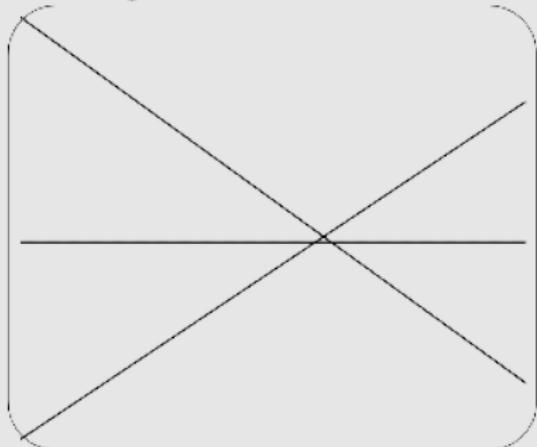
$$p(m, c) = \frac{1}{n} \sum_{i=1}^n \delta_n(\Delta((m, c), g_i))$$

Ein einfaches Beispiel

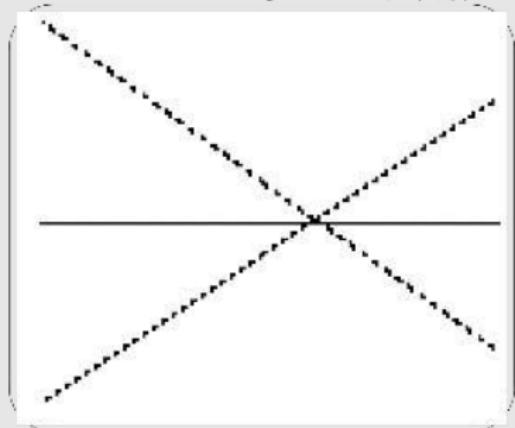
Beispiel

Ein einfaches Bild und die berechneten edgel

Anwendung: Suche alle Geraden in einem Bild



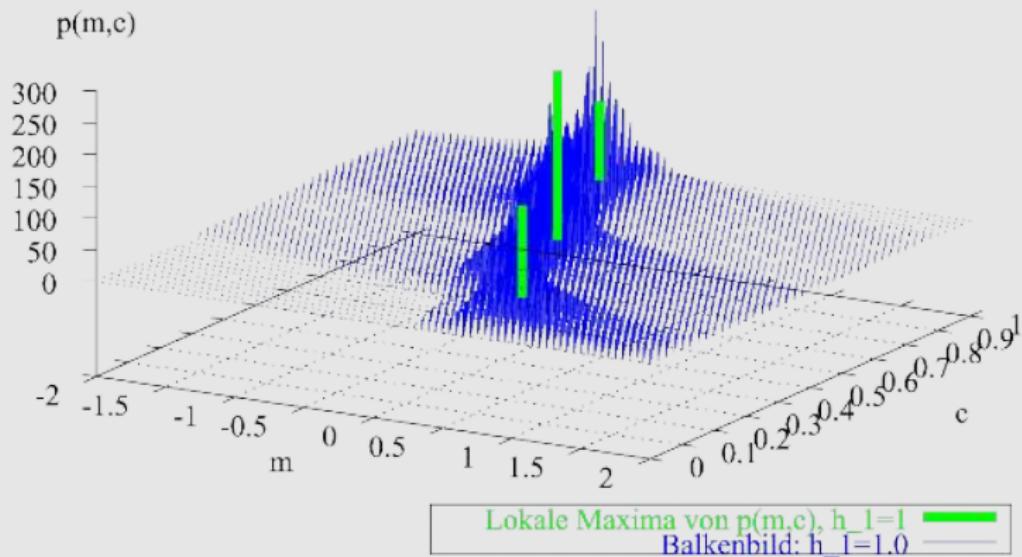
Pixel mit hohem Grauwertgradienten ($\mathcal{F}(x, y) > 10$)



Dichtefunktion, Fensterbreite $h = 1$

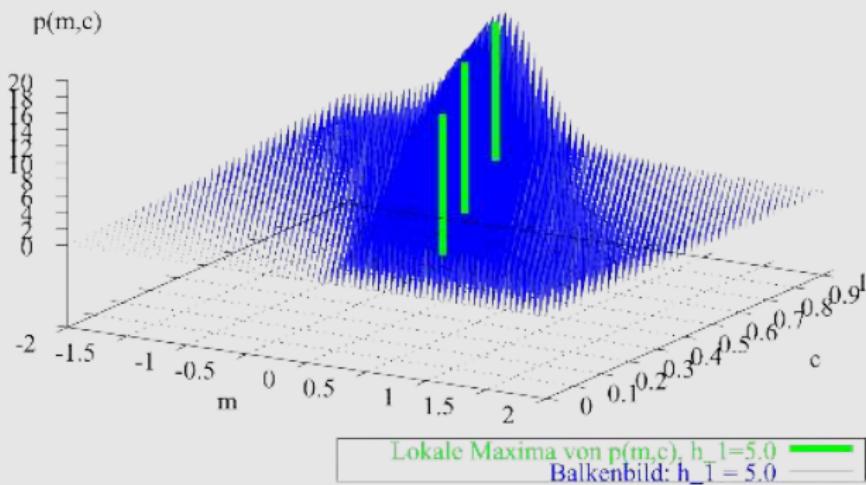
Der Parameterraum: $h_1 = 1.0$

Der Parameterraum: Parzen-Schätzung für Geraden



Dichtefunktion, Fensterbreite $h = 5$

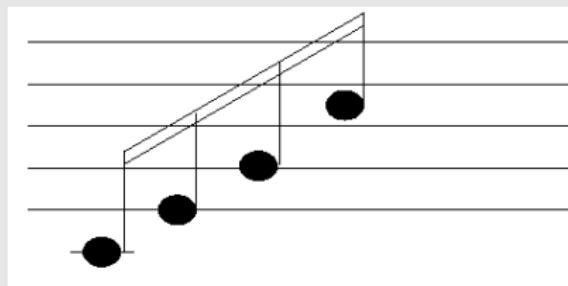
Der Parameterraum: $h_1 = 5.0$
Der Parameterraum: Parzen-Schätzung für Geraden



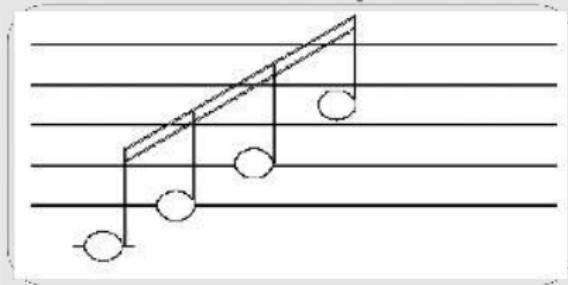
Ein weiteres Beispiel

Beispiel

Ein Notenbild und die berechneten edgel



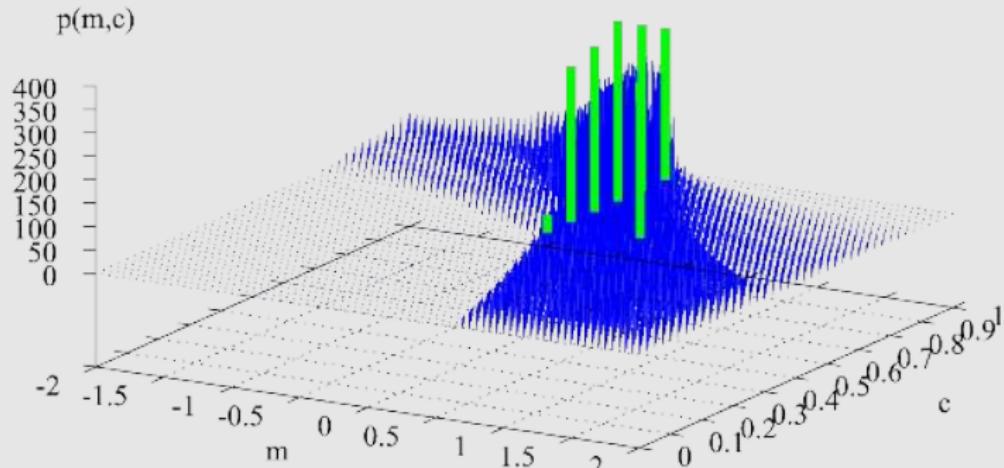
Pixel mit hohem Grauwertgradienten



Dichtefunktion, Fensterbreite $h = 1$

Der Parameterraum:

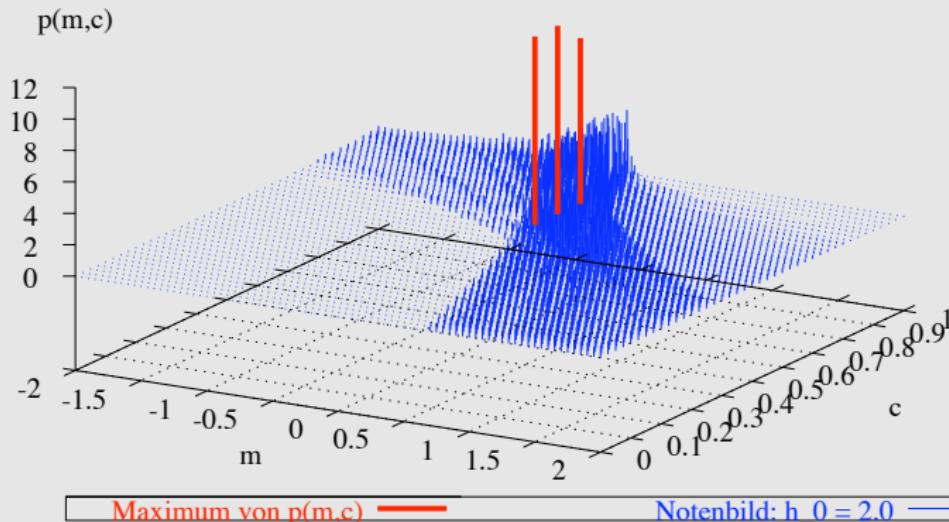
Der Parameterraum: Parzen-Schätzung für Geraden



Notenbild: Lokale Maxima von $p(m,c)$, $h=1$ ■
Notenbild: $h=1.0$ —

Dichtefunktion, Fensterbreite $h = 2$

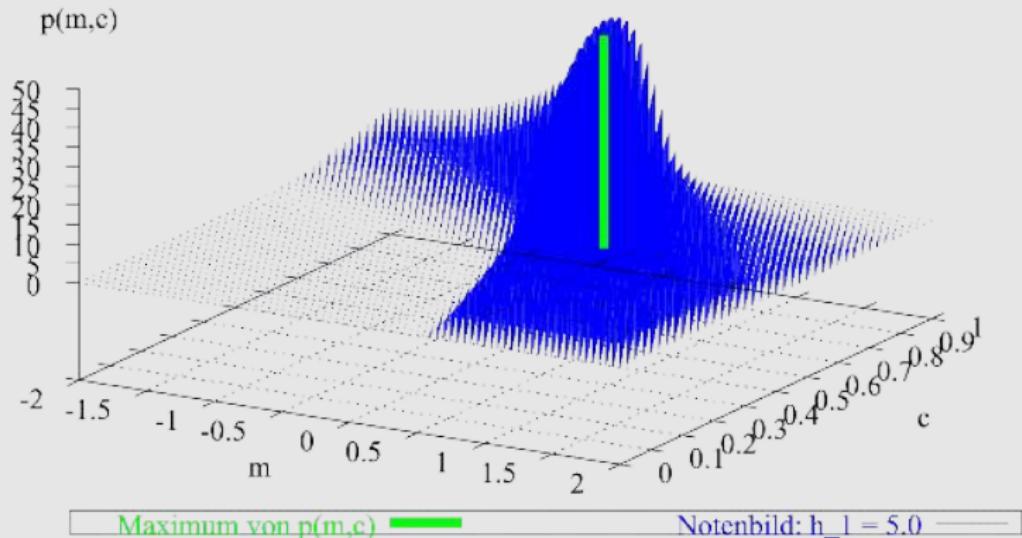
Der Parameterraum: Parzen-Schätzung für Geraden



Dichtefunktion, Fensterbreite $h = 5$

Der Parameterraum:

Der Parameterraum: Parzen-Schätzung für Geraden



Senkrechte Geraden

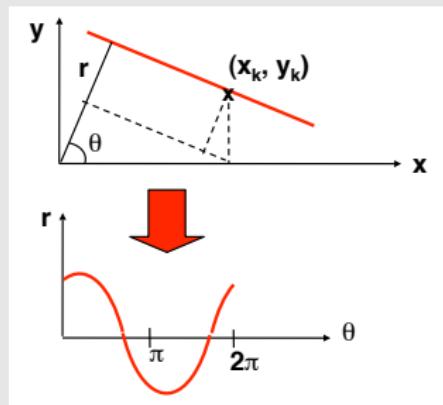
Darstellung der Geraden

Die Steigungs-Achsenabschnitts-Form der Geradengleichung ist ungünstig, da m nicht proportional zum Steigungswinkel wächst und steile Geraden wegen der Polstelle bei einer Steigung von 90° im (m, c) -Parameterraum nicht darstellbar sind.

Wie lassen sich senkrechte Geraden finden? Es gibt prinzipiell zwei Wege:

- 1 Am einfachsten: Das Bild um 90° drehen und das Verfahren ein zweites Mal anwenden, um die fehlenden Geraden zu finden.
- 2 Etwas aufwendiger: Eine anderen Geradendarstellung verwenden. In Polardarstellung bleiben alle Parameter endlich.

Polarform der Geradengleichung



r : Abstand vom Ursprung
 θ : Richtung der Normalen

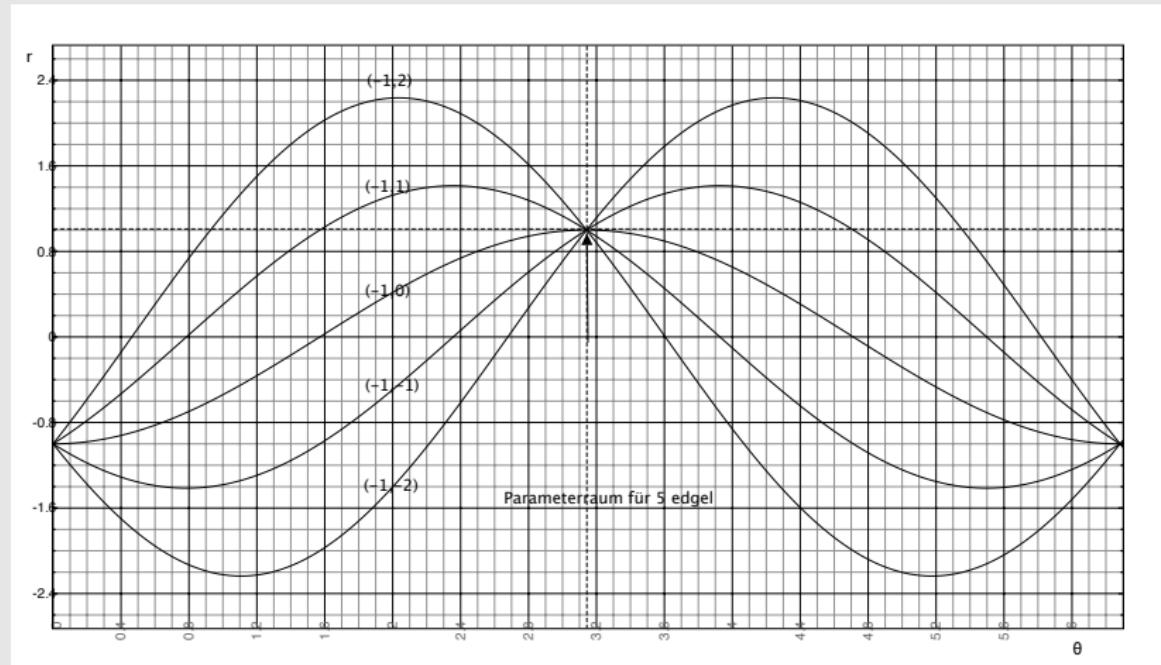
Im Parameterraum ergibt sich eine sinusähnliche Kurve.

Geradendarstellungen

Parameterraum in Polardarstellung

Kurven für die edgel

$(-1, -2), (-1, -1), (-1, 0), (-1, 1), (-1, 2)$



Berücksichtigung der Gradientenrichtung

Wenn der Kantenfinder auch die Gradientenrichtung schätzt, können die Einträge im Parameterraum auf diejenigen Parameterkombinationen beschränkt werden, die in einem Intervall δ um die Gradientenrichtung ϕ liegen.

$$(x_k, y_k, \phi) = \text{edgel mit Gradientenrichtung } \phi$$
$$x_k \cos \theta + y_k \sin \theta = r, \phi - \delta \leq \theta \leq \phi + \delta \text{ im Parameterraum}$$

Hough-Transformation für Kreise

Der Parameterraum:

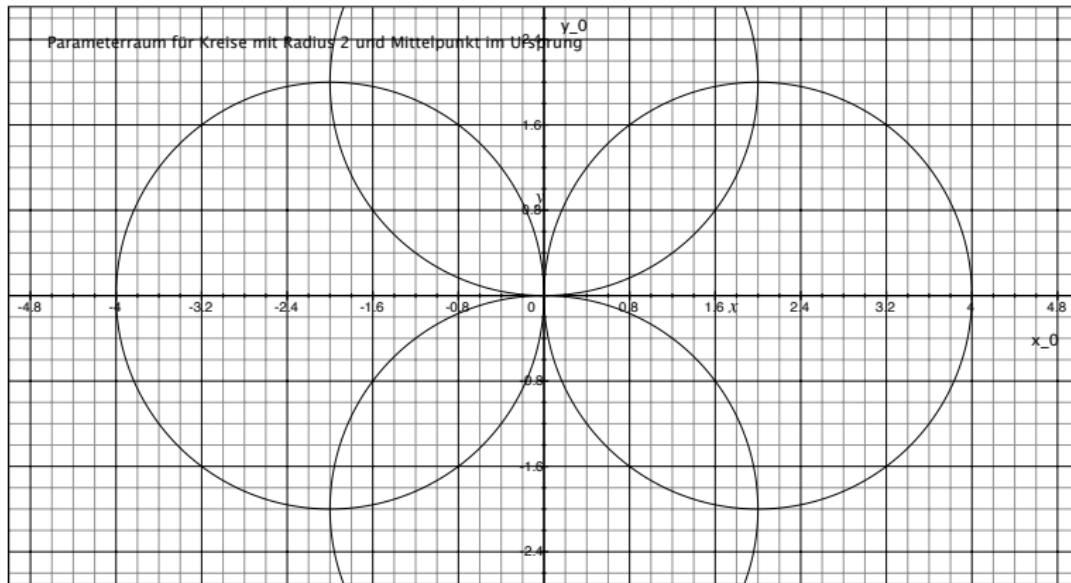
r : Radius

x_0, y_0 : Koordinaten des Mittelpunkts

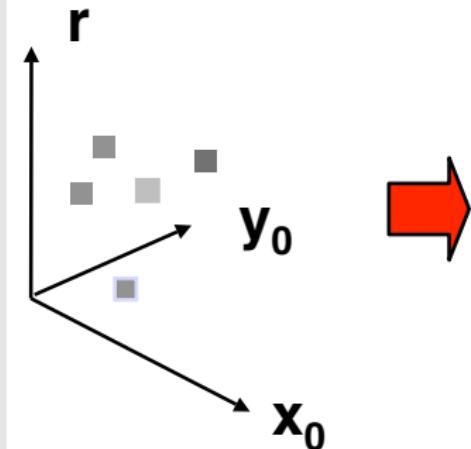
x_k, y_k : Koordinaten des edgels

$$(x_k - x_0)^2 + (y_k - y_0)^2 = r^2$$

Beispiel: Parameterraum für Kreis mit Radius 2 und Mittelpunkt im Ursprung



Anwendung: Münzen auf strukturiertem Untergrund



Teil XIX

Geometrische Szenenrekonstruktion

Szenenrekonstruktion



Alte und junge
Frau

48

Geometrische Szenenrekonstruktion

- Tiefen- und Formhinweise
 - Motorische Hinweise
 - Scheinbare Größe
 - Texturgradienten
- Shape from Contour

49

Photometrische Szenenrekonstruktion

50

Das Generalitätsprinzip

Mehrdeutigkeit bzgl. der Tiefe

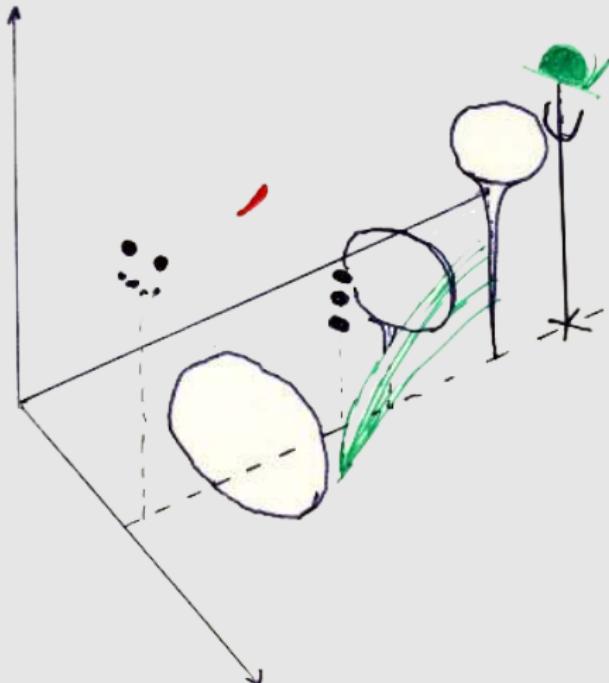


Ansicht 1

Mehrdeutigkeit bzgl. der Tiefe



Ansicht 1



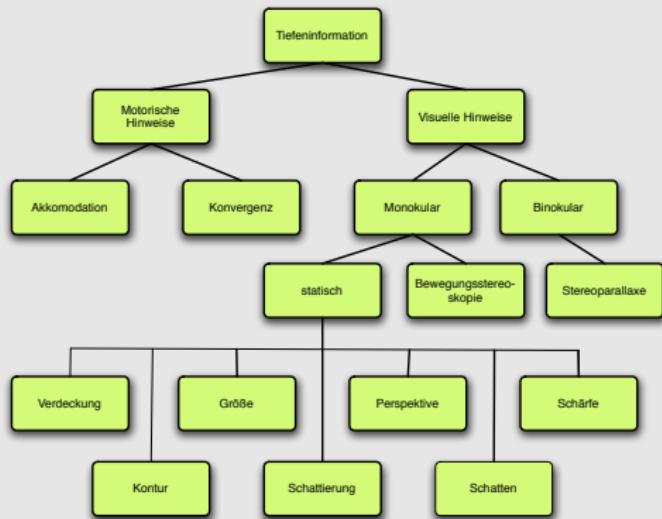
Ansicht 2

Das Problem

Die Projektion ist **keine bijektive** Abbildung: Unendlich viele Raumpunkte werden auf denselben Bildpunkt projiziert.

- ☞ Bilder sind daher mehrdeutig, aber nicht alle der möglichen Interpretationen sind physikalisch sinnvoll.
- ☞ Die Gesetze der Physik und Vorwissen über die abgebildete Szene schränken die Interpretationsmöglichkeiten ein.
- ☞ Aus solchen Tiefenhinweisen können wir entweder absolute Entfernung, relative Entfernung oder Orientierungen von Oberflächen ableiten.
- ☞ Wenn wir bestimmte Annahmen machen müssen, sprechen wir von **indirekten** Tiefenhinweisen, sonst von **direkten** Tiefenhinweisen.

Tiefenhinweise



Motorische Hinweise:



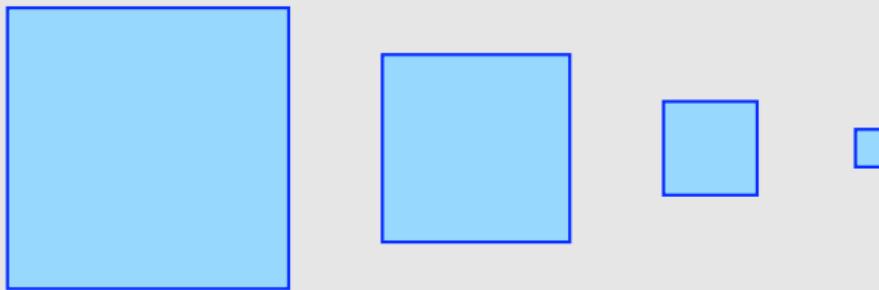
Das visuelle System des Menschen kann bei der Auswertung der visuellen Eindrücke berücksichtigen, wie die Augen ausgerichtet sind und wie stark die Linse gewölbt ist.

Beides sind **direkte** Tiefenhinweise, da sie unabhängig von der abgebildeten Szene anwendbar sind.

Akkomodation: Auf welche Entfernung fokussiert die Linse?

Vergenz: Unter welchem Winkel wird ein Objekt fixiert?

Große Objekte erscheinen näher als kleine.



Objektmodelle

Wenn wir wissen, welches Objekt abgebildet ist, können wir seine Entfernung ermitteln, indem wir seine **bekannte absolute Größe** mit der **beobachteten scheinbaren Größe** im Bild vergleichen.

- ▶ **Objektmodelle** liefern **indirekte** Tiefenhinweise, da wir nicht direkt die Entfernung sondern eine andere physikalische Größe messen.
- ▶ Ebenso möglich:

$$\frac{\text{scheinbare Geschwindigkeit}}{\text{absolute Geschwindigkeit}} \quad \text{oder} \quad \frac{\text{scheinbare Helligkeit}}{\text{absolute Helligkeit}}$$

Tiefenhinweis: Vertraute Größe



Tiefenhinweis: Vertraute Größe 2



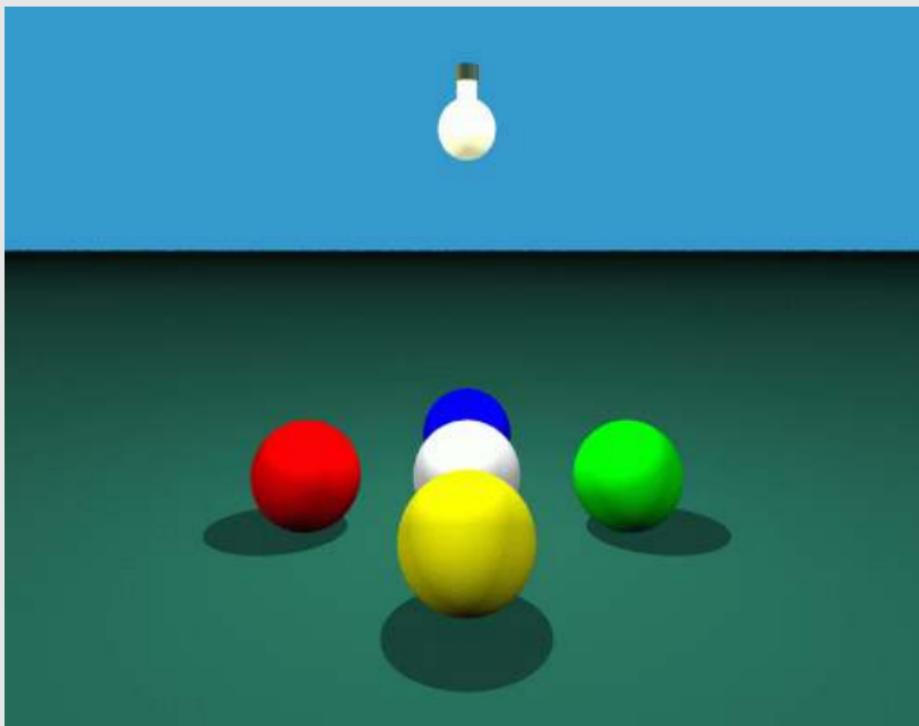
Tiefenhinweis: Vertraute Größe 2



Tiefenhinweis: Verdeckung



Tiefenhinweis: Schattierung und Schattenwurf



Beispiel: Totemaske

Maske

- ▶ Schattierung ist mehrdeutig bzgl. konvex-konkav.
- ▶ Im Zweifelsfall wählt unser Sehsinn die Interpretation, bei der das Licht von oben kommt.

Tiefenhinweis: Schattierung und Schattenwurf 2



Atmosphärische Trübung



- ▶ Das von der Atmosphäre gestreute Licht (Luftlicht) bildet einen Schleier, der das Objekt umso weißer erscheinen lässt, je weiter es entfernt ist.

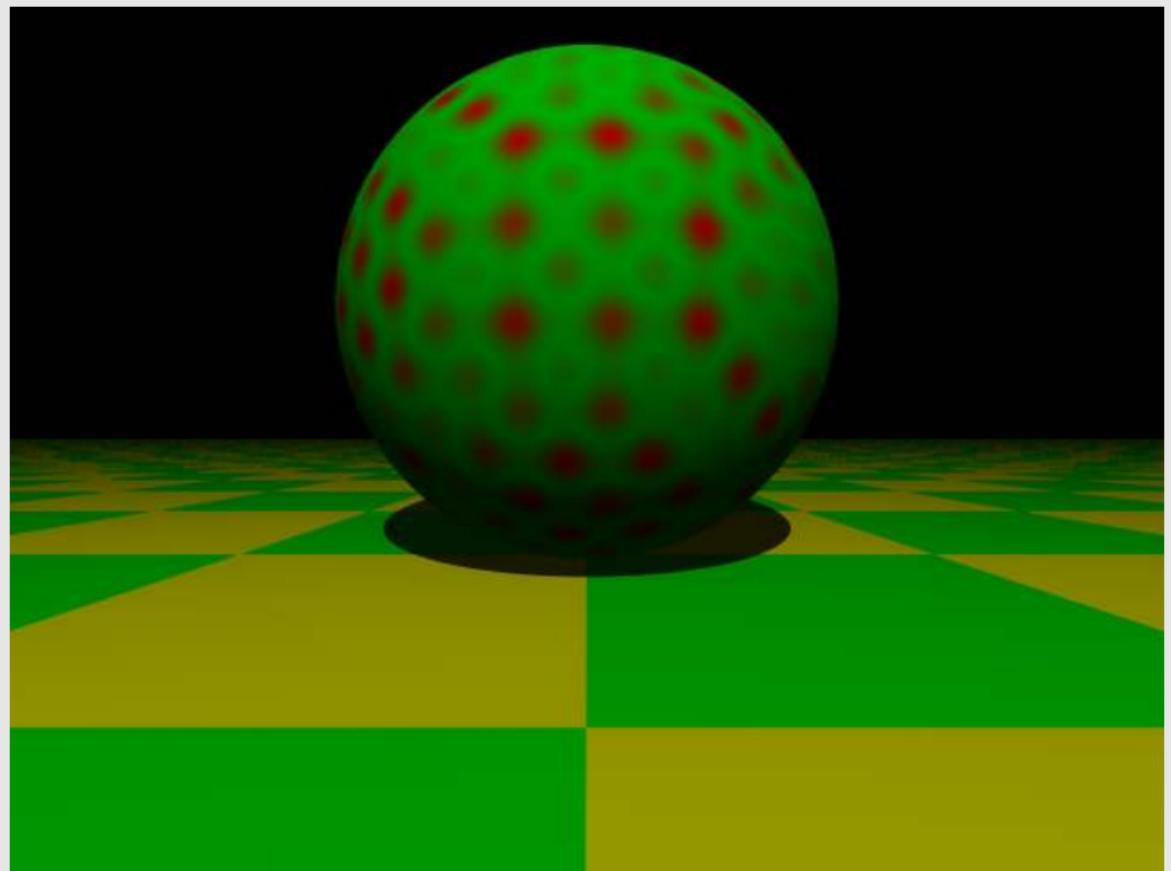
Atmosphärische Trübung



Atmosphärische Trübung



Tiefenhinweis: Texturgradient



Texturgradienten

Die **perspektivische Verkürzung** beeinflußt die Abbildung von texturierten Flächen.

Größe der Texel: Die Texel werden mit zunehmender Entfernung **kleiner**.

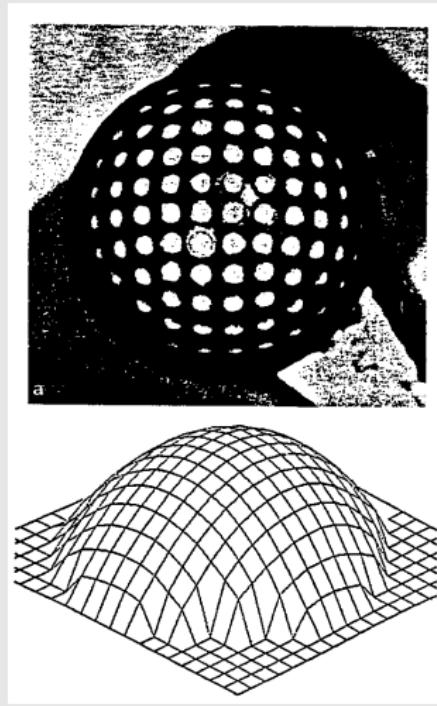
Form der Texel: Die Form der Texel hängt von der Blickrichtung ab — Kreise werden als Ellipsen abgebildet.

Dichte der Texel: Die Abstände zwischen den Texeln werden perspektivisch verkürzt, so daß sie mit zunehmender Entfernung dichter erscheinen.

Die Variation der Textur in unterschiedliche Richtungen wird durch **Texturgradienten** beschrieben.

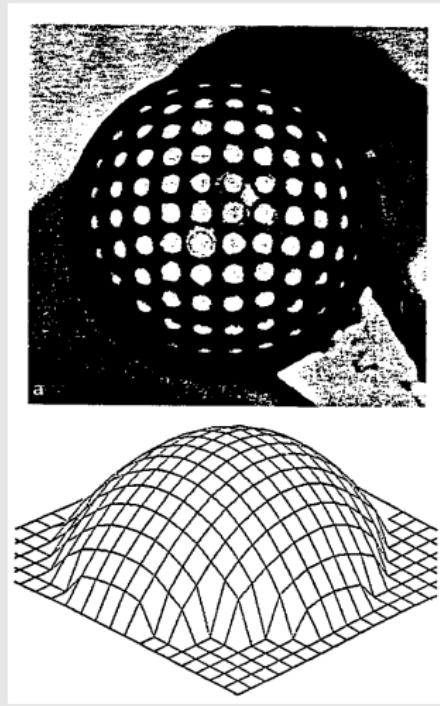
- ☞ Je nach Merkmal (Form, Größe, Dichte) gibt es unterschiedliche Texturgradienten.
- ☞ Aus den Texturgradienten kann die Orientierung der Fläche relativ zur Betrachtungsrichtung abgeleitet werden [Ballard and Brown, 1982].

Texturgradient: Perspektivische Verkürzung der Kreise



Rekonstruktion einer Kugeloberfläche

Texturgradient: Perspektivische Verkürzung der Kreise



Rekonstruktion einer Kugeloberfläche

Texturgradient



Perspektivische Verkürzung der Texel

Texturgradient:



Perspektivische Verkürzung der Blüten und der Abstände

Texturgradient:



Perspektivische Verkürzung der Blüten und der Abstände

Texturgradient:



Perspektivische Verkürzung der Blüten und der Abstände

Texturgradient:



Perspektivische Verkürzung der Blüten und der Abstände

Texturgradient:



Perspektivische Verkürzung der Blüten und der Abstände

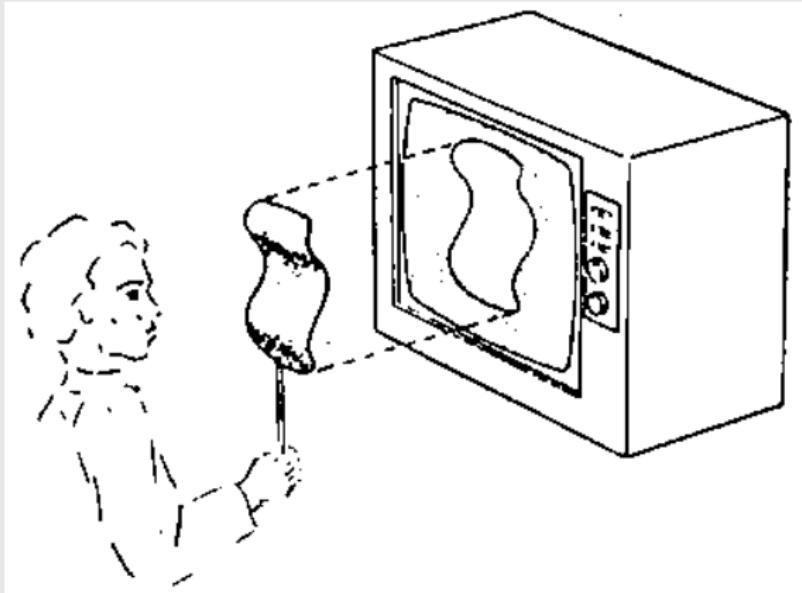
Shape from Contour

Geometrische Szenenrekonstruktion nach
[Barrow and Tenenbaum, 1981]:



- ▶ Rekonstruiere die sichtbaren Objektkonturen im Raum.
- ▶ Rekonstruiere die von den Konturen umschlossenen Flächen.

Konturen und Flächen



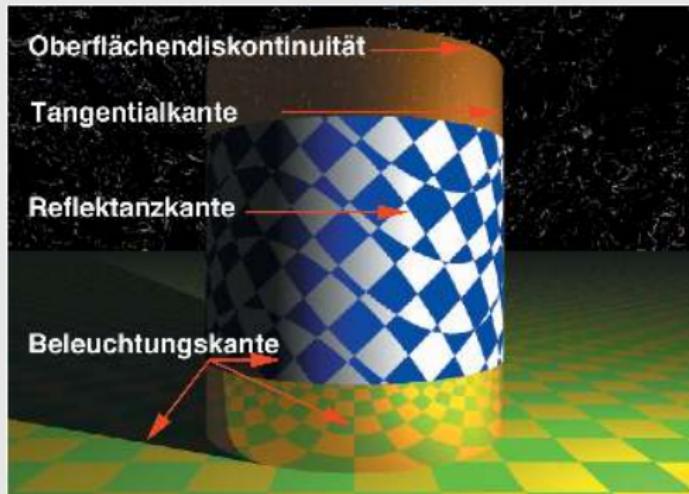
Unser Sehsystem füllt Linienzüge automatisch mit glatten

Objektkonturen

Konturen können Hinweise auf die Form einer Oberfläche geben.

- ▶ Linienzeichnungen sind Abstraktionen für diejenigen Stellen im Bild, an denen eine Bildfunktion stark variiert, sogenannte Kanten.
- ▶ Kanten sind Hinweise auf Diskontinuitäten
 - ▶ der Objektoberfläche (Orientierung, Pigmentierung)
 - ▶ oder der Beleuchtung.

Shape from Contour



Arten von Bildstrukturen

Oberflächendiskontinuitäten: (discontinuity boundaries)

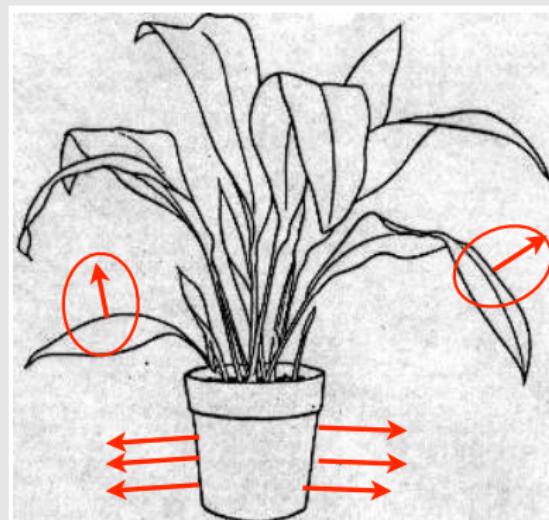
Die Oberflächennormale einer Fläche ändert sich sprunghaft.

Tangentialkanten (extremal boundaries) treten nur bei gekrümmten Flächen auf. Wir sehen sie dort, wo die Projektionsstrahlen in der Tangentialebene der Fläche liegen.

Reflektanzkanten: Auffällige Änderung in der Pigmentierung.

Beleuchtungskanten: Auffällige Änderung in der Beleuchtungsstärke, strukturiertes Lichtfeld.

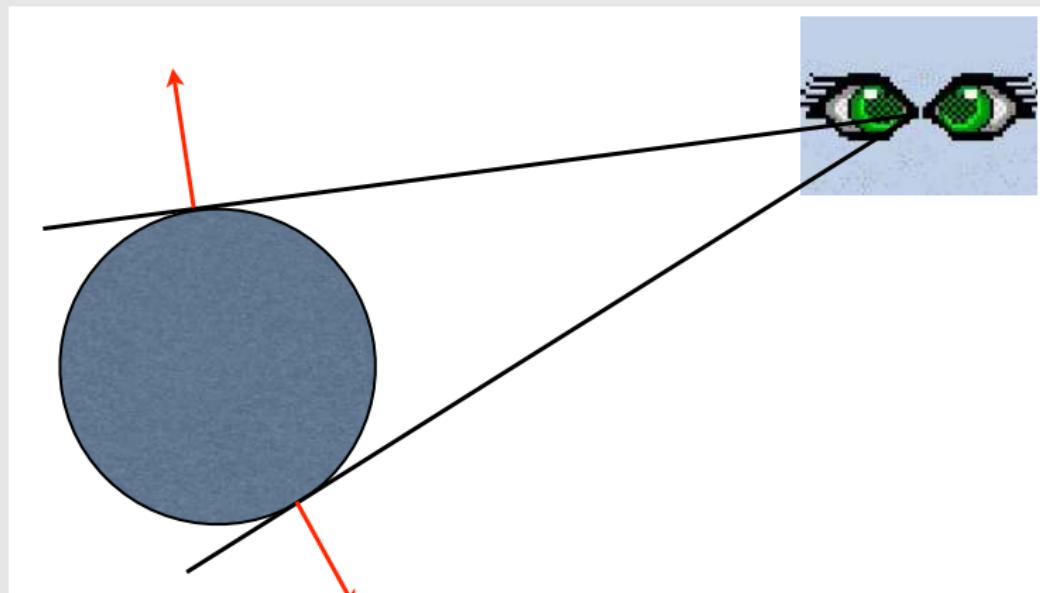
Oberflächendiskontinuitäten und Tangentialkanten



boundaries

- ▶ Tangentialkanten
(extremal boundaries)
- ▶ Oberflächendiskontinuitäten
(discontinuity boundaries)

Oberflächendiskontinuitäten und Tangentialkanten



Tangentalkanten

Aus Tangentialkanten (extremal boundaries) können wir eindeutig die Lage der Fläche im Raum ableiten:

Oberflächennormale

An einer Tangentialkante steht die Oberflächennormale der gekrümmten Fläche

- ☞ senkrecht auf der sichtbaren Kontur
- ☞ und senkrecht auf dem Projektionsstrahl, der ja in der Tangentialebene liegt.

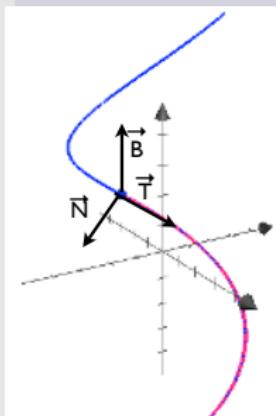
Mount Palomar: Beispiel für Tangentialkanten



Gekrümmten Raumkurve

Gegeben:

Eine differenzierbare, gekrümmte Raumkurve am Punkt der Bogenlänge s :



Tangentenvektor $\vec{T}(s)$: Ein Einheitsvektor in Richtung der ersten Ableitung einer parametrischen Raumkurve.

Hauptnormalenvektor $\vec{N}(s)$: Ein Vektor senkrecht zur Kurve in der Schmiegeebene der Kurve.

Binormalenvektor $\vec{B}(s)$: $\vec{T}(s) \times \vec{N}(s)$

Das begleitende Dreibein

Definition (Begleitendes Dreibein)

Die Vektoren $\vec{T}(s)$, $\vec{N}(s)$, $\vec{B}(s)$ sind Einheitsvektoren eines orthonormierten Rechtssystems und heißen **begleitendes Dreibein**.

Krümmung einer Kurve

Krümmung

- ▶ Der Krümmungsvektor ist der Vektor der zweiten Ableitung einer parametrischen Raumkurve.
- ▶ Der Krümmungsvektor ist zum Tangentenvektor orthogonal, ist aber kein Einheitsvektor.
- ▶ Der Betrag des Krümmungsvektors am Punkt der Bogenlänge s heißt Krümmung $\chi(s)$.

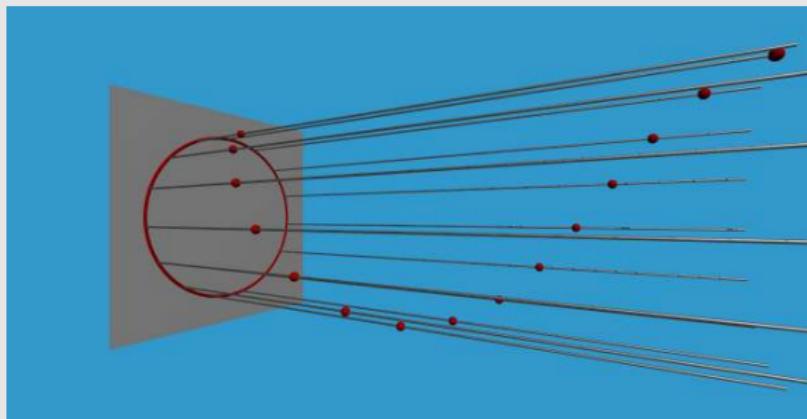
Die Krümmung beschreibt die Richtungsänderung des Tangentenvektors in der Schmiegeebene der Kurve.

Die Torsion (Windung) einer Raumkurve

Torsion

- ▶ Die Torsion $\tau(s)$ beschreibt die Richtungsänderung des Binormalenvektors in der Normalenebene der Kurve.
- ▶ Der Betrag der Torsion $\tau(s)$ ist der Betrag der ersten Ableitung der Binormalen.
- ▶ Das Vorzeichen der Torsion gibt die Richtung an, in die die Kurve nach oben oder unten aus der Schmiegeebene ausgelenkt wird.

Der Rekonstruktionsalgorithmus für Raumkurven



- ▶ Berechne für alle edgel einer zusammenhängenden Kontur die Projektionsstrahlen.
- ▶ Wähle für jedes edgel diejenige Entfernung vom Projektionszentrum, die die glattesten Raumkurve ergibt.

Glattheitsmaße für Raumkurven

Forderung 1:

$$\int |\chi'(s)| ds = \text{Minimum}$$

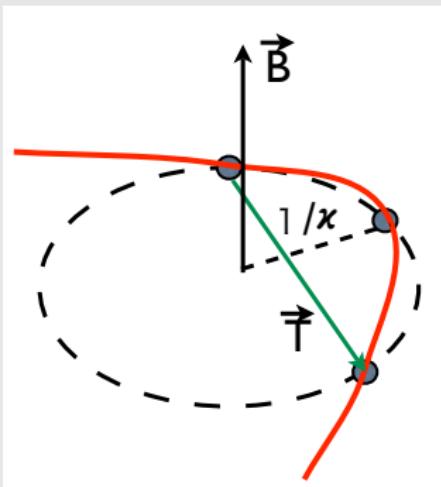
Forderung 2:

$$\int \tau(s) ds = \text{Minimum}$$

$$\int \left(\frac{d(\chi(s) \cdot \vec{B})}{ds} \right)^2 ds = \int (\chi'(s)^2 + \chi(s)^2 \tau^2) ds$$

Minimisierung dieses Ausdrucks bedeutet, diejenige Raumkurve zu finden, die einem ebenen Kreisbogen am ähnlichsten ist.

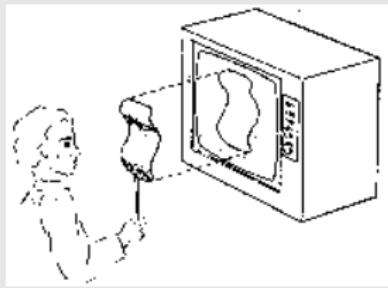
Lokale Optimierung



- ▶ Aus drei Punkten entlang der Kontur lassen sich lokal \vec{T} , \vec{B} und χ schätzen.
- ▶ Verschiebe lokal die Punkte entlang der Projektionsstrahlen, so daß das Integral lokal minimiert wird.

Dieses Verfahren rekonstruiert Ellipsen als gekippte Kreise.

Interpolation von Oberflächen



Problem (Füllen der Fläche:)

Gegeben: *Eine rekonstruierte Kontur,*

Gesucht: *Der Verlauf der eingeschlossenen Fläche*

Annahme über physikalische Oberflächen

- ▶ Orientierung und Tiefe sind stetig.
- ▶ Lokale Oberflächenmerkmale variieren so glatt wie möglich.
- ▶ Die totale Variation über die Oberfläche ist minimal.

Haupt- und Nebenkrümmung χ_1, χ_2

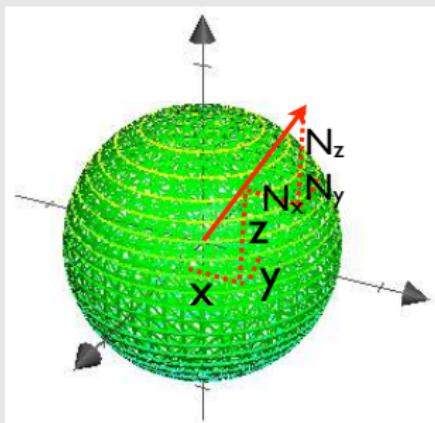
Elastische Membranen, Seifenfilme: $\chi_1 + \chi_2 = 0$

Gebogene Papierbögen: $\chi_1 \cdot \chi_2 = 0$

Ebenen, Zylinder, Kugeln: $\chi_1 = \text{const.}, \chi_2 = \text{const.}$

Gesucht: Ein Verfahren, daß exakte Ergebnisse für den Fall von Kugeloberflächen liefert und „vernünftige“ Näherungen in allen anderen Fällen.

Normalen einer Kugel mit Radius R



Kugeloberfläche

$$N_x = x/R$$

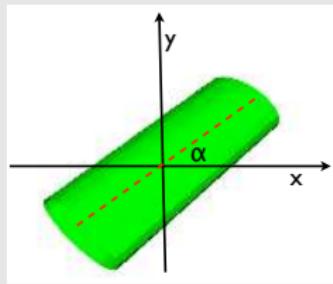
$$N_y = y/R$$

$$N_z = z/R$$

Normalenvektor \vec{N} :

Die Komponenten des Normalenvektors sind linear von den Bildkoordinaten x und y abhängig.

Normalen eines Zylinders mit Radius R



Zylinderoberfläche

$$N_x = ((y \cos \alpha - x \sin \alpha) \cdot \sin \alpha) / R$$

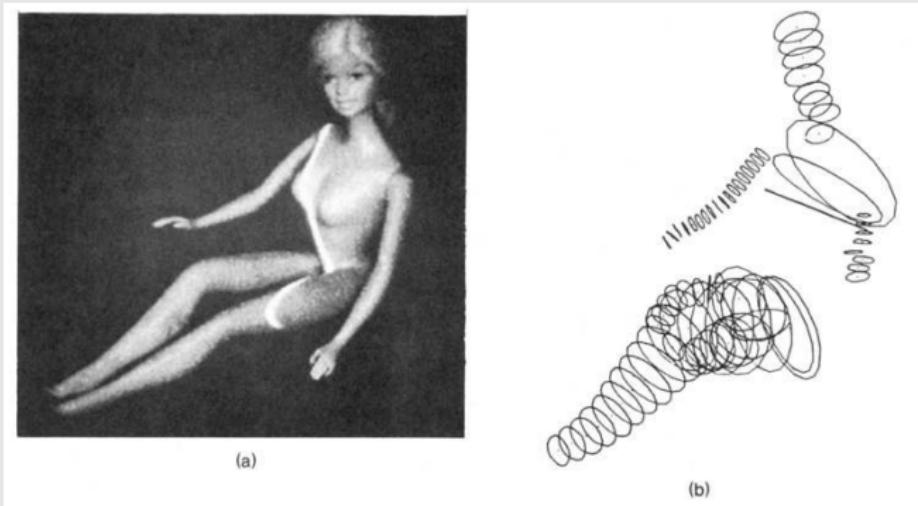
$$N_y = ((y \cos \alpha - x \sin \alpha) \cdot \cos \alpha) / R$$

$$N_z = 1 - N_x^2 - N_y^2$$

Implementation

- ▶ Auf Kugeloberflächen und Zylindern kann der Normalenvektor global interpoliert werden.
- ▶ Für andere Flächen wird lokal so interpoliert, daß N_x, N_y möglichst linear in x und y variieren.
 - ▶ Lineare Anpassung einer Ebene an N_x, N_y .
 - ▶ Iterative Verbesserung, bis die Fläche glatt ist.

Shape from Contour mit verallgemeinerten Zylindern



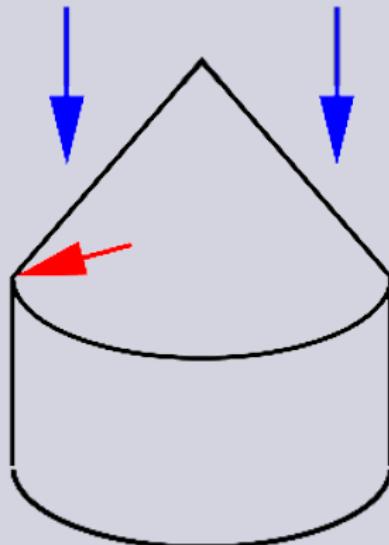
aus [Ballard and Brown, 1982] nach Agin-1972

Klassifikation der Konturen

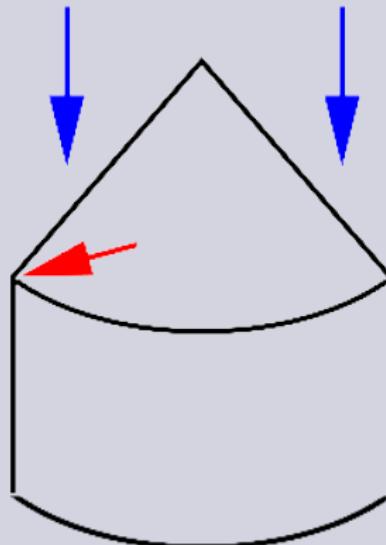
- ▶ Damit das Flächeninterpolationsverfahren angewendet werden kann, müssen Tangentialkanten erkannt werden.
- ▶ Indizien für Tangentialkanten sind beispielsweise
 - ▶ Symmetrie
 - ▶ Schattierung
 - ▶ Textur

Türmchen oder Torte?

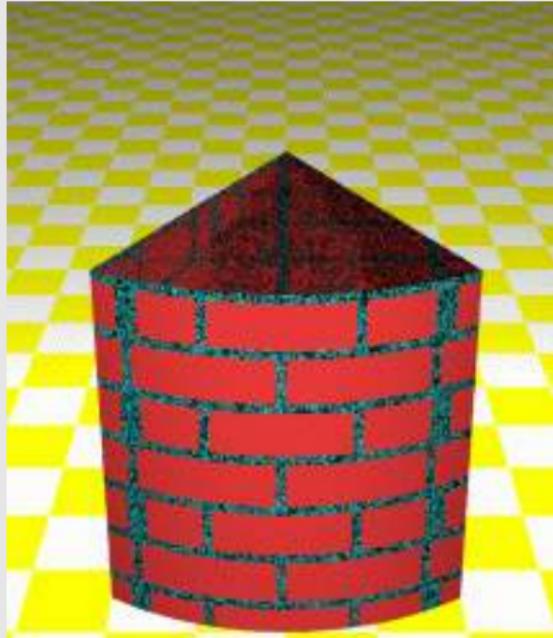
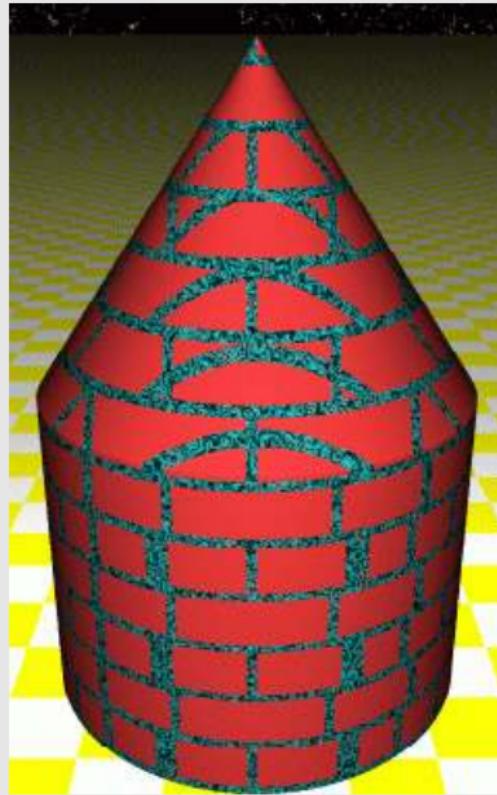
Tangentialkanten



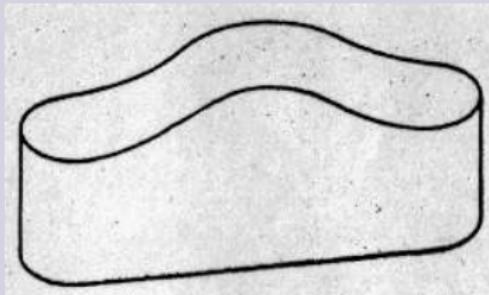
Oberflächendiskontinuitäten



Die von den Konturen begrenzten Flächen

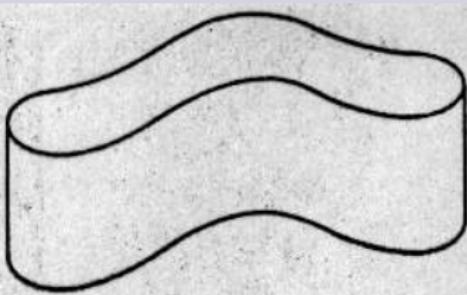


Der Kontext einer Kontur



(a)

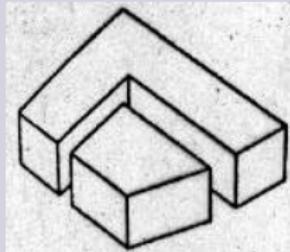
planare Seitenfläche



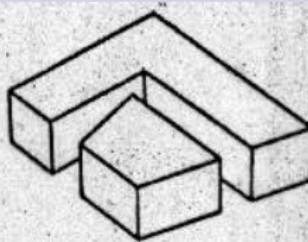
(b)

gekrümmte Seitenfläche

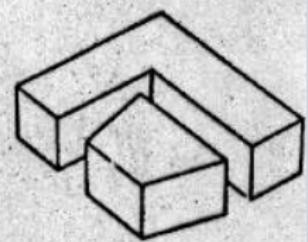
Im Kontext eines benachbarten Objektes



(a)



(b)



(c)

Abhangigkeit von der Interpretation

Die alte und die junge Frau



Teil XIX

Geometrische Szenenrekonstruktion

Photometrische Szenenrekonstruktion

48

Geometrische Szenenrekonstruktion



49

Photometrische Szenenrekonstruktion

- Photometrische Bildentstehung (image formation)
 - Messen von Strahlungswerten
 - Modellierung des Sensors
 - Modellierung der Oberfläche
 - Beispiele für Reflexionsfunktionen
 - Die Reflektanzkarte
- Photometrisches Stereo
- Shape from Shading

Literatur

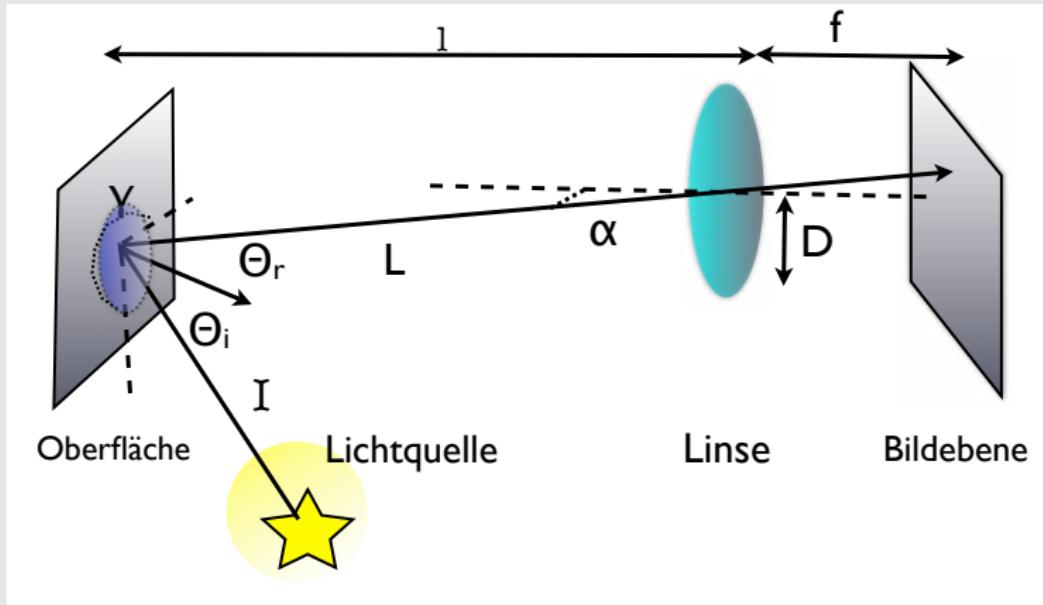
- ❖ Horn, B. K. P. (1986).
Robot Vision.
The MIT Press, Cambridge, Mass.McGraw-Hill, New York.
- ❖ Ikeuchi, K. and Horn, B. K. P. (1981).
Numerical shape from shading and occluding boundaries.
Artificial Intelligence, 17:141–184.
- ❖ Woodham, R. G. (1981).
Analysing images of curved surfaces.
Artificial Intelligence, 17:117–140.

Bildentstehung (image formation)

Zur Rekonstruktion der Objektoberflächen aus der Objektschattierung:

- ▶ Wie hängt die Bildintensität von der einfallenden Strahlung ab?
- ▶ Wie hängt die auf das Bild einfallende Strahlung von der durch das Objektiv abgebildeten Objektstrahlung ab?
- ▶ Wovon hängt die vom Objekt ausgesendete Strahlung ab?

Strahlungspfade bei der Aufnahme



Meßgrößen für Strahlung

Es gibt drei Systeme zur Messung von Strahlungswerten:
Die Systeme verwenden unterschiedliche Einheiten.

Das radiometrische System mißt die Meßgrößen integriert über alle Frequenzen des elektromagnetischen Spektrums.

Das spektrometrische System mißt die Meßgrößen spezifisch für jeweils eine Frequenz.

Das photometrische System mißt die Meßgrößen für den Bereich des sichtbaren Lichts.

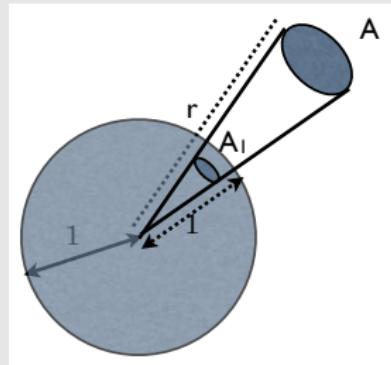
Allen Systeme verwenden skalare Größen und leider keine Vektoren, obwohl Strahlung ein gerichtetes Phänomen ist.

Das radiometrische System

Das radiometrische System mißt die Strahlungsenergie bezogen auf

- ▶ den Raumwinkel ω ,
- ▶ die Größe der bestrahlten oder emittierenden Fläche A
- ▶ und die Dauer t der Bestrahlung.

Der Raumwinkel



$$\omega = A_1 = \frac{A}{r^2}$$

Raumwinkel ω

Der **Raumwinkel ω** wird durch das Flächensegment A_1 der Einheitskugel angegeben, das von dem Winkel ω aus der Oberfläche der Kugel ausgeschnitten wird.

Die Grundgrößen

Strahlungsenergie Q :

Strahlungsleistung P (oder ϕ):

$$P = \frac{dQ}{dt}$$

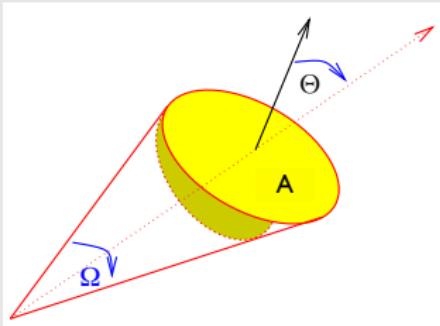
Strahlstärke I : Strahlungsleistung je Raumwinkelement
(radiant intensity)

$$I = \frac{dP}{d\omega}$$

Bestrahlungsstärke E : Leistung der pro Flächenelement
einfallenden Strahlung (irradiance).

$$E = \frac{dP}{dA}$$

Die Strahldichte L (radiance)



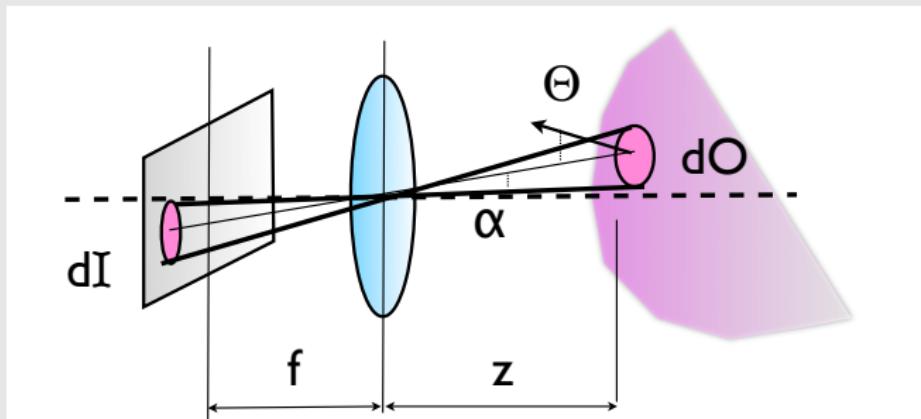
Raumwinkel ω

- ▶ Die **Strahldichte L** (radiance) berücksichtigt auch die Orientierung der strahlenden Fläche A zum Raumwinkel ω .
- ▶ Sie gibt die Leistung P pro Raumwinkelement ω . und pro Flächenelement A senkrecht zum Strahl an.

Strahldichte L (radiance)

$$L = \frac{d^2P}{d\omega dA}$$
$$\omega = \frac{A \cos \vartheta}{R^2}$$

Modellierung des Sensors: Bildintensität



- ▶ Die Helligkeit (oder Intensität) einer Fläche im Bild ist proportional zu der auf diese Fläche eingestrahlten Leistung (Bestrahlungsstärke E_i).

Intensität \sim Bestrahlungsstärke E_i

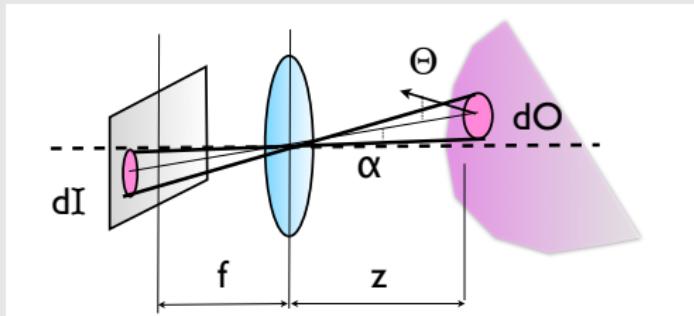
Berechnung der Strahlstärke E_i

Problem (Gesucht:)

Bestrahlstärke E_i der Bildfläche dI

Gegeben: Sei

- ▶ L : Strahldichte der Objektfläche dO .
 - ▶ $d\omega_i$: der Raumwinkel der Bildfläche dI im optischen Zentrum
 - ▶ $d\omega_o$: der Raumwinkel der Objektfläche dO im optischen Zentrum.
-
- ▶ Strahlen, die durch das optische Zentrum gehen, werden nicht gebrochen.
 - ▶ Daher ist $d\omega_i = d\omega_o$



$$d\omega_i = d\omega_o$$

$$d\omega_i = \frac{dl \cdot \cos \alpha}{(f / \cos \alpha)^2}$$

$$d\omega_o = \frac{dO \cdot \cos \vartheta}{(z / \cos \alpha)^2}$$

$$\frac{dO}{dl} = \frac{\cos \alpha}{\cos \vartheta} \left(\frac{z}{f}\right)^2$$

Die Strahlungsleistung dP von dO durch die Linse

Welche Strahlungsleistung dP geht von der Objektfläche dO aus durch die Linse mit Durchmesser d ?

- ▶ Sei ω der Raumwinkel, den die Linsenfläche $\frac{\pi}{4}d^2$ vom Objekt aus gesehen bildet:

$$\omega = \frac{\pi}{4} \cdot \frac{d^2 \cos \alpha}{(z / \cos \alpha)^2} = \frac{\pi}{4} \cdot \left(\frac{d}{z}\right)^2 \cdot \cos^3 \alpha$$

- ▶ Die Strahlungsleistung dP , die von der Objektfläche dO aus unter dem Winkel ϑ durch die Linse geht:

$$\begin{aligned} dP &= L \cdot \omega \cdot dO \cdot \cos \vartheta \\ &= L \cdot dO \cdot \frac{\pi}{4} \cdot \left(\frac{d}{z}\right)^2 \cdot \cos^3 \alpha \cos \vartheta \end{aligned}$$

Die Bestrahlstärke E_i des Bildelements dI

Die Linse richtet die Strahlungsleistung dP auf ein Bildelement mit der Fläche dI :

- Die Bestrahlstärke E_i der Bildfläche dI :

$$E_i = \frac{dP}{dI} = L \cdot \underbrace{\frac{dO}{dI}}_{\frac{\cos \alpha}{\cos \vartheta} \left(\frac{z}{f}\right)^2} \cdot \frac{\pi}{4} \cdot \left(\frac{d}{z}\right)^2 \cdot \cos^3 \alpha \cos \vartheta$$
$$E_i = L \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha$$

Ergebnis

Satz (Bestrahlstärke E_i eines Bildelements)

$$E_i = L \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha$$

Die Bestrahlstärke E_i eines Bildelements dI ist

- ☞ proportional zur Strahldichte L von der Objektfläche dO durch die Linse,
- ☞ unabhängig von der Entfernung z
- ☞ und proportional zum Verhältnis $(\frac{d}{f})^2$

Relative Öffnung eines Objektivs

Definition (Relative Öffnung:)

Der Quotient $\frac{f}{d}$ heißt effektive oder relative Öffnung eines Objektives.

- ☞ Je kleiner die relative Öffnung ist, desto lichtstärker ist das Objektiv.

Vignetierung

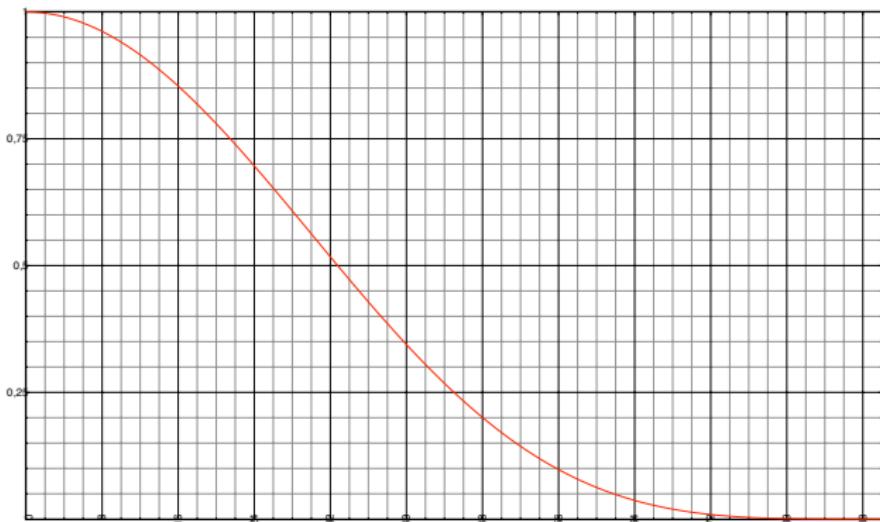
Der Term $\cos^4 \alpha$ bewirkt eine Verdunklung des Bildes zum Bildrand hin.

Er ist bei kleinen Werten von α vernachlässigbar, muß aber bei Meßkammern und Weitwinkelobjektiven korrigiert werden.

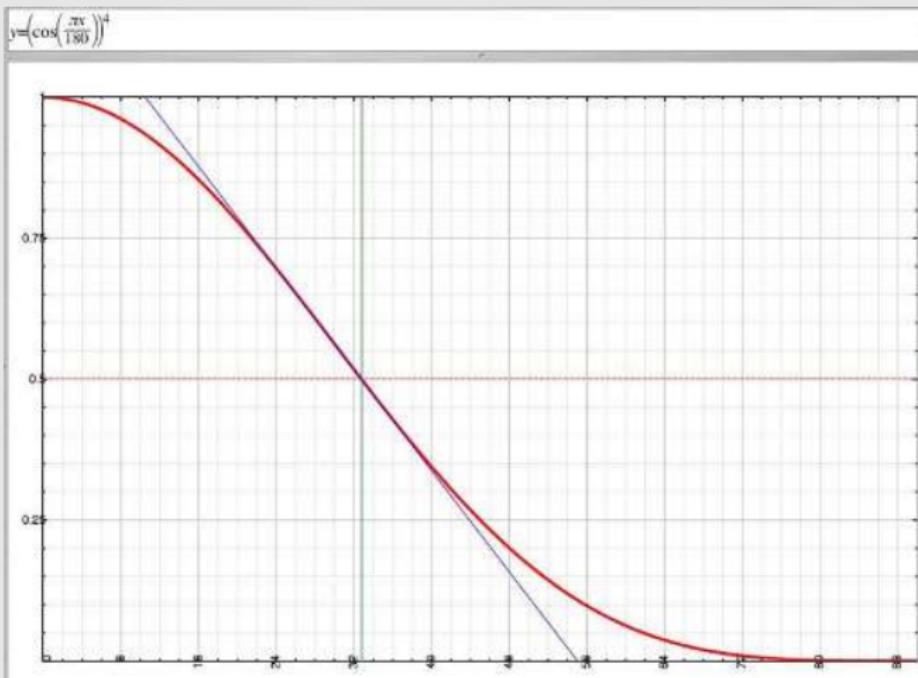
Definition (**Natürliche Vignetierung**)

Die Randverdunklung proportional zu $\cos^4 \alpha$ heißt **natürliche Vignetierung**.

Natürliche Vignetierung: $\cos^4 \alpha$



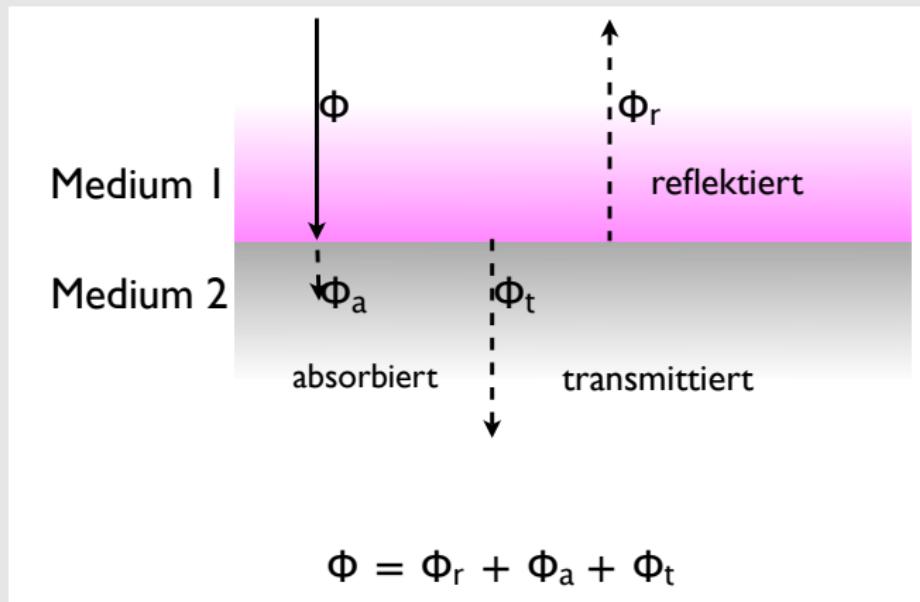
Natürliche Vignetierung: $\cos^4 \alpha$



Strahlungsfluß an der Grenzfläche zweier Medien

Der einfallende Strahlungsfluß wird entweder

- reflektiert, absorbiert oder transmittiert.

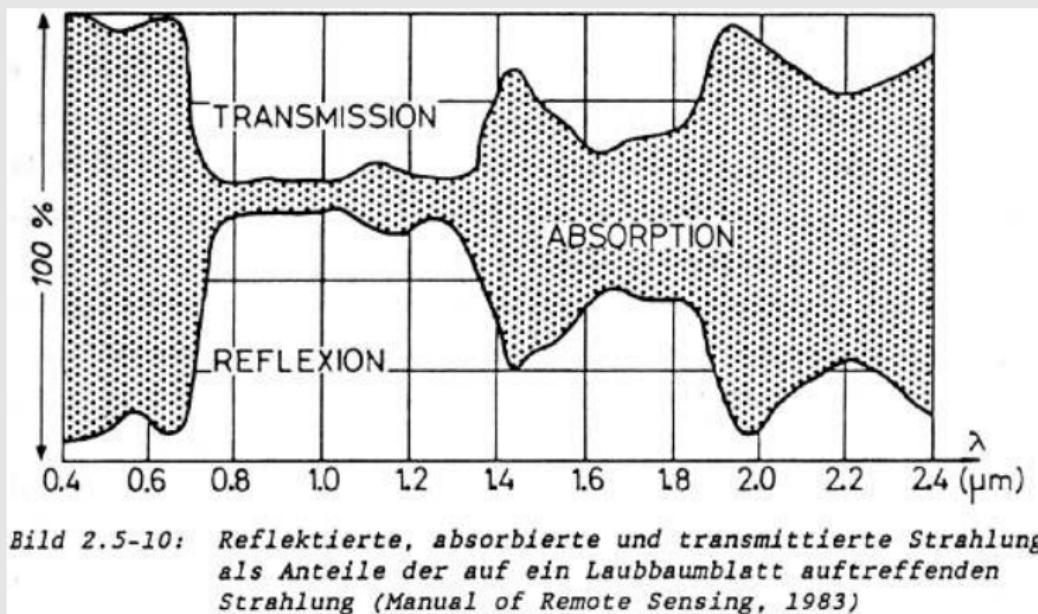


Beispiel: Laub

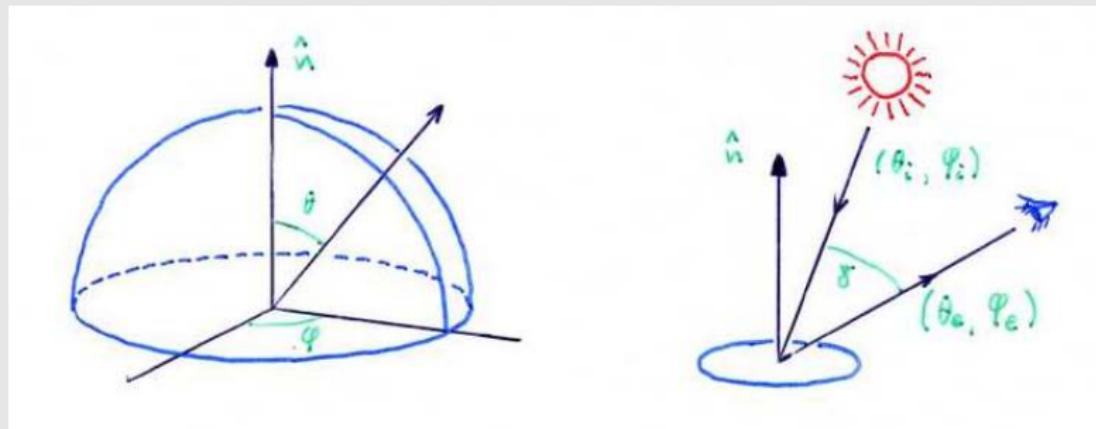


Abhangigkeit von der Wellenlange, Beispiel Laub

Laub reflektiert stark im Infrarotbereich und absorbiert stark im Rotkanal.



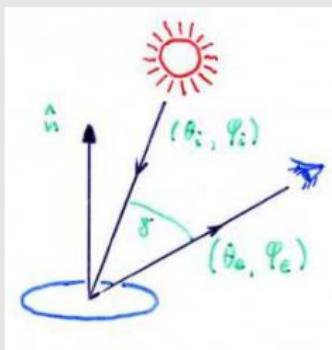
Die am Objekt reflektierte Strahldichte L



- Die vom Objekt reflektierte Strahldichte L hängt ab von:
- ▶ der Bestrahlstärke E ,
 - ▶ der Reflektivität des Oberflächenmaterials,
 - ▶ der Geometrie der Lichtreflexion.

BDRF: Reflexionsfunktion

Bestrahlstärke E_0 : Strahlungsleistung, die pro Flächenelement auf das Objekt einstrahlt.



Reflexionsgrad ρ des Oberflächenmaterials:
Anteil der Strahlungsleistung, die nicht absorbiert sondern reflektiert wird.

Geometrische Anordnung:
Die **Reflexionsfunktion**
(bidirectional reflectance distribution function) BDRF.

Die Reflexionsfunktion (BDRF)

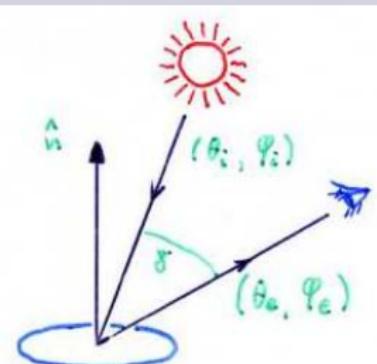
Definition (Reflexionsfunktion:)

Die Reflexionsfunktion ϕ (*bidirectional reflectance distribution function*) beschreibt das Reflexionsverhalten einer Oberfläche in Abhängigkeit von der Beleuchtungsrichtung (ϑ_e, φ_e) und der Beobachtungsrichtung (ϑ_r, φ_r) .

Die Reflexionsfunktion $\phi(\vartheta_e, \varphi_e, \vartheta_r, \varphi_r)$ erlaubt die Berechnung der StrahldichteVerteilung $L_r(\vartheta_r, \varphi_r)$ der reflektierten Strahlung bei gegebener StrahldichteVerteilung $L_e(\vartheta_e, \varphi_e)$.

Der Phasenwinkel

Phasenwinkel γ



- ☞ Für viele Materialien ändert sich die Strahldichte L_e nicht, wenn die Oberfläche um den Normalenvektor \vec{n} rotiert wird. Ausnahme: z.B. Vogelfedern
- ☞ Die Strahldichte L_e hängt dann nur von der Differenz $(\varphi_i - \varphi_e)$ oder dem **Phasenwinkel γ** ab.

Parameter der Bildentstehung

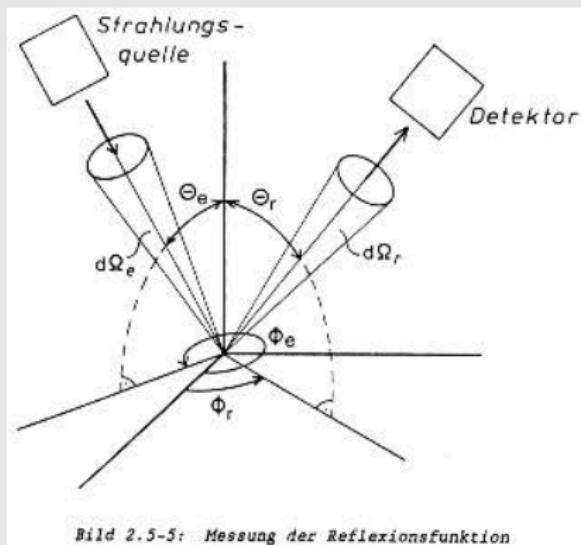
Die Intensität ist damit abhängig von

- ① Beleuchtungsorientierung,
- ② Sensororientierung,
- ③ Orientierung der Objektoberfläche.

Shape from Shading

Sofern 1. und 2. bekannt sind, kann die Oberflächenorientierung aus der Intensität berechnet werden:

Messung der BDRF



Die BDRF muß durch Messung der Strahldichte für alle Winkelkombinationen $(\vartheta_i, \vartheta_e, \varphi_i, \varphi_e)$ ermittelt werden.

Der ideale Spiegel

Der ideale Spiegel reflektiert nur,

- ▶ wenn der Einfallswinkel ϑ_i gleich dem Ausfallswinkel ϑ_e ist
- ▶ und wenn der Phasenwinkel $\gamma = \pi$ ist.

$$\vartheta_i = \vartheta_e$$

$$\gamma = \pi$$

$$\phi(\vartheta_i, \vartheta_e, \varphi_i - \varphi_e) = \delta(\vartheta_i - \vartheta_e) \cdot \delta(\pi - (\varphi_i - \varphi_e))$$

$$\text{Kronneckersymbol } \delta(x) = \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}$$

Lambert'scher Strahler

Die ideale matte Oberfläche

(Lambert'scher Strahler):

$$\phi(\vartheta_i, \vartheta_e, \gamma) = \cos(\vartheta_i)$$

- ▶ Eine ebene Lambert'sche Oberfläche, die mit parallelem Licht bestrahlt wird, sieht aus allen Richtungen gleich hell aus, da die Strahldichte L in diesem Fall unabhängig vom Ausfallwinkel ϑ_e ist.
- ▶ Beispiele: Angenähert Lambert'sche Strahler sind Papier, Schnee.

Die Reflexionsfunktion der Mondoberfläche



Reflexionsfunktion der Mondoberfläche ϕ :

$$\phi_{\mathfrak{D}}(\vartheta_i, \vartheta_e, \gamma) = \frac{\cos(\vartheta_i)}{\cos(\vartheta_e)}$$

- ▶ Eine ebene, mit parallelem Licht bestrahlte Scheibe wird bei wachsendem Ausfallwinkel immer heller.
- ▶ Bei Vollmond sieht der Mond wie eine flache Scheibe aus, da dann $\vartheta_i = \vartheta_e$ ist.

Typische Oberflächen

Natürliche Materialien haben oft matte und spiegelnde Anteile:

Reflektor für Einfallswinkel ϑ_i gleich Ausfallswinkel ϑ_e
und Phasenwinkel $\gamma = \pi$,

Lambert'scher Strahler sonst.

Reflexionsfunktionen in Pov-Ray

In Pov-Ray wird das Verhältnis von spiegelndem zu Lambert'schen Anteil in der finish-Spezifikation beschrieben:

Beispiel

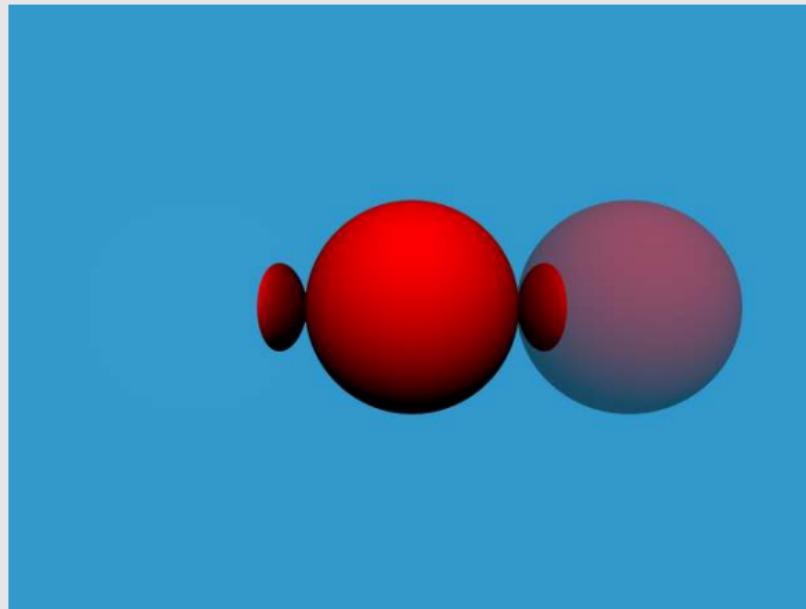
Ein perfekter Spiegel, ein Lambert'scher Strahler und ein gemischter Reflektor:

```
#declare Mirror =
    finish {ambient 0 diffuse 0 reflection 1}
#declare Lambert =
    finish {ambient 0 diffuse 1 reflection 0}
#declare Mixed =
    finish {ambient 0 diffuse 0.8 reflection 0.2}
```

Beispiel

Drei Kugeln mit den Oberflächen

finish { Mirror } , **finish { Lambert }** , **finish { Mixed }**



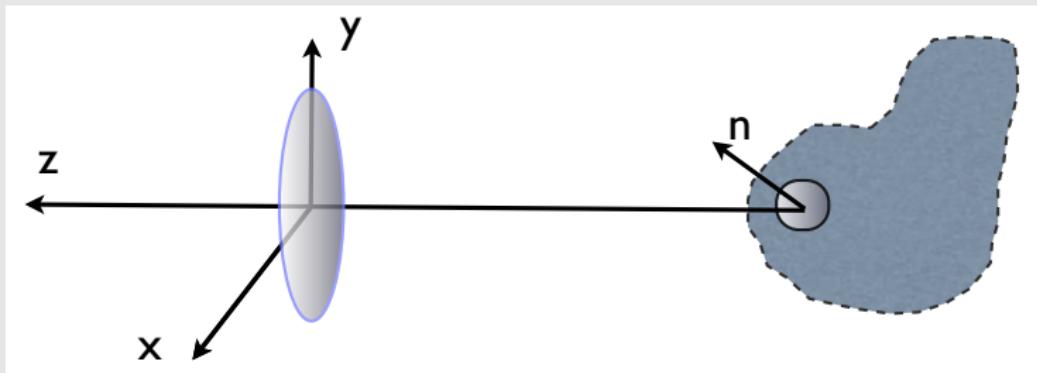
Geometrische Szenenrekonstruktion
Photometrische Szenenrekonstruktion
Das Generalitätsprinzip

Photometrische Bildentstehung (image formation)
Photometrisches Stereo
Shape from Shading

Teil XIX

Geometrische Szenenrekonstruktion

Oberflächenorientierung



- ▶ Wir gehen vom **objektzentrierten** Koordinatensystem der Reflexionsfunktion zum **betrachterzentrierten** Koordinatensystem über.
- ▶ Die Orientierung eines Punktes der Objektoberfläche wird durch den Normalenvektor \vec{n} der Tangentialebene dargestellt.

Die Tangentialebene

Sei die Gleichung der Tangentialebene gegeben durch

$$\begin{aligned} z &= px + qy + z_0 \\ \text{mit } p &= \frac{\partial z}{\partial x} \\ q &= \frac{\partial z}{\partial y} \end{aligned}$$

- ▶ p und q geben die Steigungen der Tangentialebene in x und y -Richtung an.
- ▶ $(p, q)^T$ ist der Gradient.
- ▶ Die p, q -Ebene heißt **Gradientenraum**.
- ▶ Der Normalenvektor \vec{n} zur Objektoberfläche ist

$$\vec{n} = (-p, -q, 1)^T$$

Orientierungen

Betrachterorientierung:

Wir nehmen an, daß der Blickwinkel α vernachlässigbar klein ist (sehr große Objektentfernung z), dann ist der Vektor \hat{v} in Betrachterrichtung:

$$\hat{v} = (0, 0, 1)^T$$

Beleuchtungsorientierung:

Wenn der Abstand der Lichtquelle vom Objekt sehr groß ist, ist die Beleuchtungsrichtung nahezu konstant:

$$\hat{s} = (-p_s, -q_s, 1)^T$$

Einfalls- und Ausfallwinkel

Berechnung des Einfallswinkels ϑ_i und Ausfallwinkel ϑ_e aus

- ▶ dem Normalenvektor $\vec{n} = (-p, -q, 1)^T$,
- ▶ der Betrachterorientierung $\vec{v} \approx \hat{v} = (0, 0, 1)^T$ und
- ▶ der Beleuchtungsorientierung $\vec{s} \approx \hat{s} = (-p_s, -q_s, 1)^T$:

$$\cos \angle(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$

$$\cos \vartheta_e = \cos \angle(\vec{n}, \vec{v}) = \frac{1}{\sqrt{1 + p^2 + q^2}}$$

$$\cos \vartheta_i = \cos \angle(\vec{n}, \vec{s}) = \frac{1}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}}$$

Die Reflektanzkarte

Reflektanzkarte $R(p, q)$

Die Reflektanzkarte $R(p, q)$ faßt für eine spezielle Aufnahmesituation

- ▶ die Betrachterorientierung,
- ▶ die Beleuchtungsorientierung
- ▶ und die Reflexionsfunktion

in einer Funktion zusammen.

Sie gibt die Bildintensität in Abhängigkeit von den Komponenten p und q der Oberflächennormalen \vec{n} an.

Beispiele

Lambert'scher Strahler:

$$R_L(p, q) = \frac{1 + p_s p + q_s q}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}}$$

Mondoberfläche:

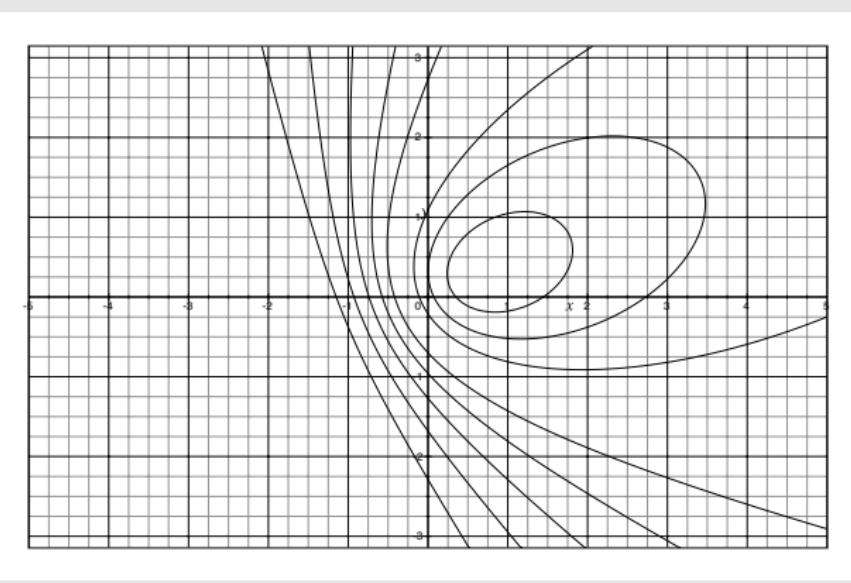
$$R(p, q) = \frac{1 + p_s p + q_s q}{\sqrt{1 + p_s^2 + q_s^2}}$$

Darstellung der Reflektanzkarte:

Die Reflektanzkarte $R(p, q)$ wird durch Isolinien gleicher Reflektanz im Gradientenraum (in der p, q -Ebene) dargestellt.

- ▶ Der Ursprung entspricht den Oberflächenorientierungen senkrecht zur Blickrichtung.
- ▶ Das Maximum liegt bei $(p, q) = (p_s, q_s)$,

Beispiel: $R_L(p, q)$ Lambert'scher Strahler

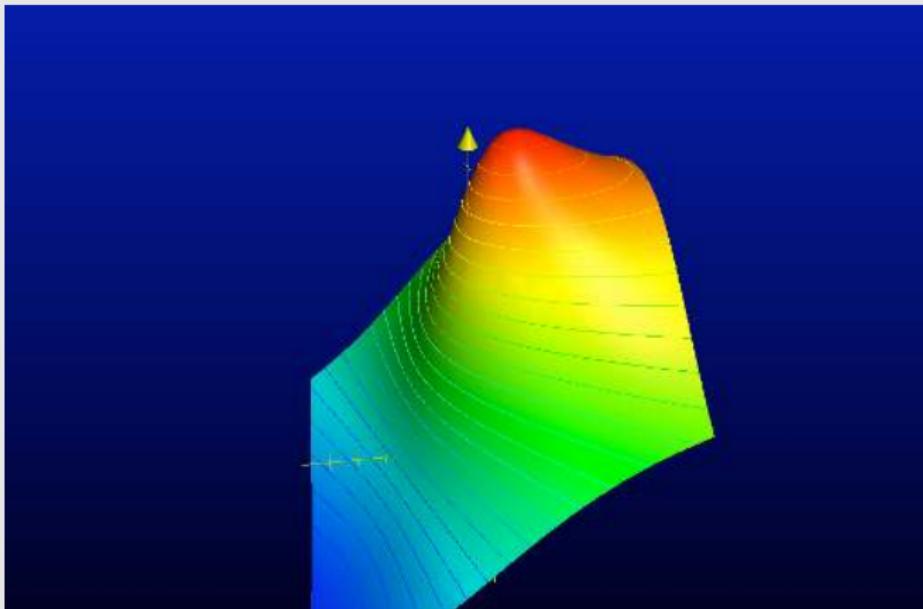


$R_L(p, q)$ mit $\rho = 1.0, p_s = 0.7, q_s = 0.3$

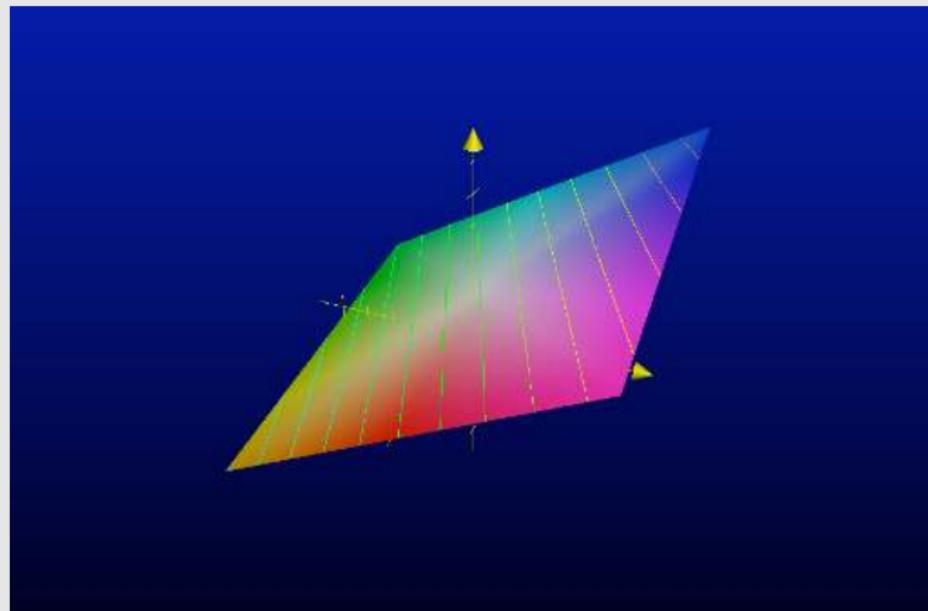
3D-Plot der Reflektanzkarte $R_L(p, q)$

Lambert'scher Strahler

3D-Plot der Reflektanzkarte $R_L(p, q)$



3D-Plot der Reflektanzkarte $R(p, q)$



3D-Plot der Reflektanzkarte $R(p, q)$

Mondgestein

Rekonstruktion der Oberflächenorientierung

Satz (Bildintensität und Reflektanzkarte)

Die Intensität $I(x, y)$ im Bild ist proportional zur Reflektanzkarte $R(p, q)$:

$$I(x, y) = c \cdot R(p, q)$$

- ☞ Die Abbildung der Oberflächenorientierung auf die Intensität I ist eindeutig, die inverse Abbildung ist es jedoch nicht.
- ☞ Es werden zusätzliche Einschränkungen benötigt, um die Oberflächen zu rekonstruieren.

Photometrisches Stereo

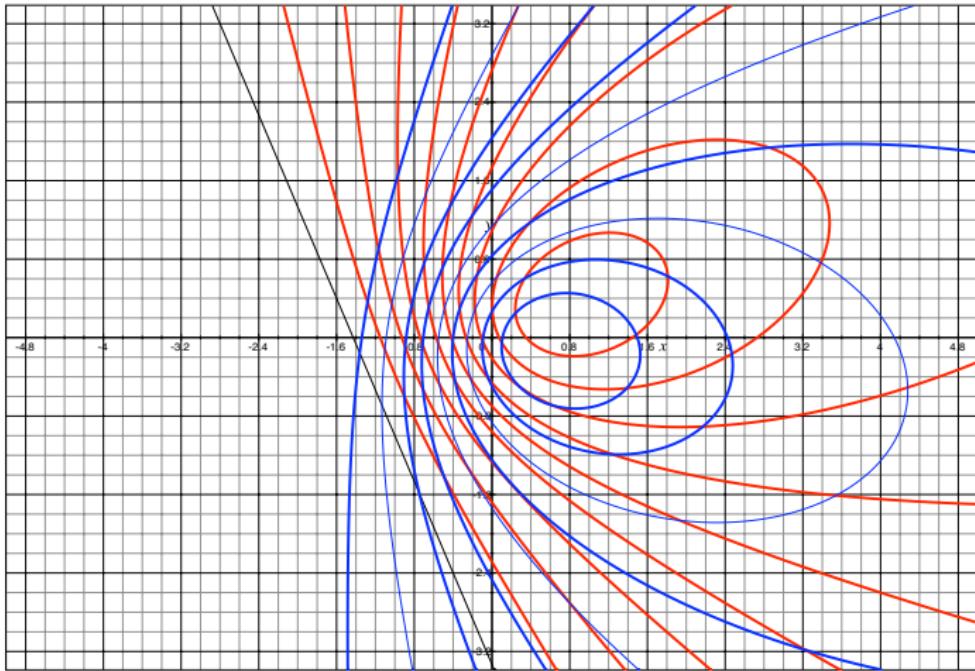
Zur Bestimmung von p und q werden mindestens zwei Gleichungen benötigt:

Lösung: Photometrisches Stereo

Es werden zwei Aufnahmen desselben Objektes mit unterschiedlicher Beleuchtungsorientierung aufgenommen. [Woodham-1981]

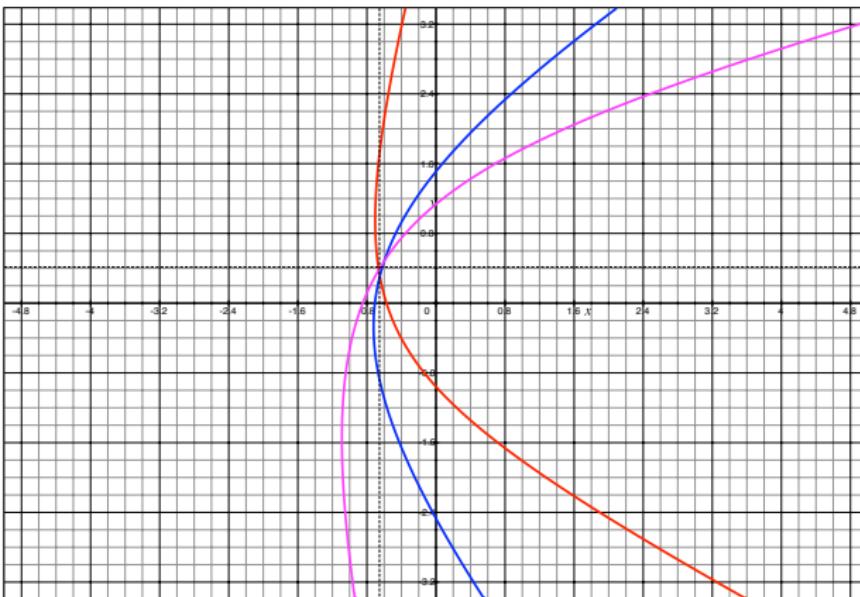
- ▶ Wenn die Reflektanzkarte linear ist und die Isolinien sich schneiden, existiert eine eindeutige Lösung.
- ▶ Bei nicht-linearen Reflektanzkarten können weitere Aufnahmen erforderlich sein.

Photometrisches Stereo



$$R_L(p, q) \text{ mit } \rho = 1.0, p_{s1} = 0.7, q_{s1} = 0.3, p_{s2} = 0.6, q_{s2} = -0.1$$

Eindeutige Lösung mit drei Aufnahmen



$R_L(p, q)$ mit $\rho = 1.0, p_{s1} = 0.7, q_{s1} = 0.3,$
 $p_{s2} = 0.6, q_{s2} = -0.1, p_{s3} = 0.5, q_{s3} = -0.3$

Beispiel: Eine Kugel mit drei Reflektanzkarten



$< 0, 0, -20 >$



$< -10, 0, -20 >$



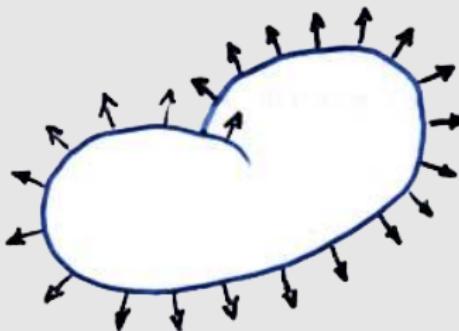
$< 0, 10, -20 >$



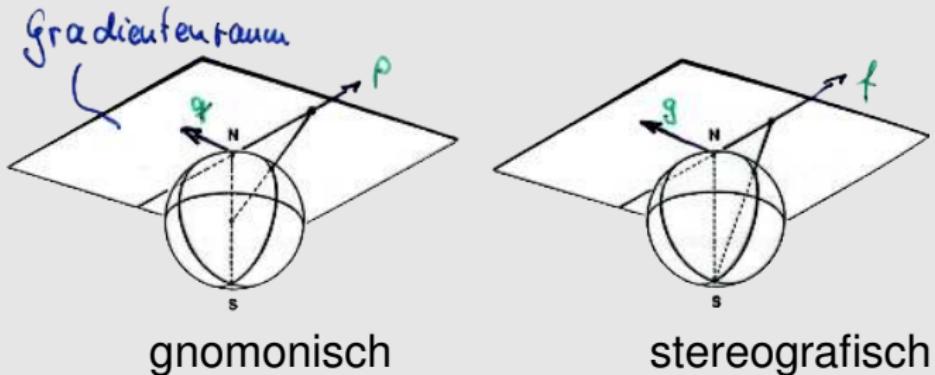
Als RGB-Mischung der drei Bilder:

Rekonstruktion mit nur einer Aufnahme

- ☞ Im Allgemeinen gibt es mit nur einem Bild keine eindeutige Rekonstruktion der Oberfläche: *ill posed problem*.
- ☞ Wir nehmen zusätzlich an, daß die Oberfläche glatt ist und die Objektränder Tangentialkanten sind.
- ☞ Dann ist die Oberflächenorientierung an den Rändern bekannt und kann als Startwert zur Rekonstruktion der Fläche genutzt werden.
- ☞ Im Gradientenraum ist allerdings ausgerechnet diese Orientierung an der Kontur nicht darstellbar.



Stereografische Projektion



Die Koordinaten f und g von (p,q) in stereografischer Projektion:

$$f = \frac{2p}{1 + \sqrt{1 + p^2 + q^2}}$$

$$g = \frac{2q}{1 + \sqrt{1 + p^2 + q^2}}$$

Image Irradiance Equation

Satz (Image Irradiance Equation)

$$I(x, y) = R(f, g)$$

Die „Image Irradiance Equation“ stellt den Zusammenhang zwischen

- ▶ den **Bildhelligkeiten I** als Funktion der Bildkoordinaten (x, y)
- ▶ und der **Reflektanzkarte R** als Funktion der Gradienten (f, g) im Gradientenraum her.

Regularisierung

Definition (**Regularisierung**)

$$I(x, y) = R(f, g)$$

Regularisierung

ill-posed Problem $\overbrace{\quad}^{\text{Regularisierung}}$ well- posed Problem

Die Regularisierung schränkt den Lösungsraum auf „glatte“ Lösungen ein.

Problem (Regularisierung:)

Aufgabe ist es, Funktionen $f(x, y)$ und $g(x, y)$ zu finden, die den Bildraum x, y so auf den Gradientenraum f, g abbilden,

- 1 daß die image irradiance equation möglichst gut erfüllt ist

$$I(x, y) = R(f, g)$$

- 2 und $f(x, y)$ und $g(x, y)$ möglichst glatt variieren.

Lösung

Eine mögliche Glattheitsforderung ist:

$$e_s = \int \int ((f_x^2 + f_y^2) + (g_x^2 + g_y^2)) dx dy \rightarrow \text{Minimum}$$

f_x, f_y, g_x, g_y : Ableitungen nach x, y .

Lösungsverfahren

Sowohl die Bildintensitäten als auch die Reflektanzkarte sind fehlerbehaftet. Daher suchen wir keine exakte Lösung sondern minimieren den quadratischen Fehler:

$$e_i = \int \int (I(x, y) - R(f, g))^2 dx dy \rightarrow \text{Minimum}$$

durch Minimierung des Integrals

$$e_i = \int \int F(f, g, f_x, f_y, g_x, g_y) dx dy$$

mit

$$F = (f_x^2 + f_y^2) + (g_x^2 + g_y^2) + \lambda(I(x, y) - R(f, g))^2$$

und λ als Wichtungsfaktor.

Der diskrete Fall

	i-1	i	i+1
j-1			
j			
j+1			

Glattheitsforderung s_{ij} am P. (i, j):

$$s_{ij} = \frac{1}{4}((f_{i+1,j} - f_{i,j})^2 + (f_{i,j+1} - f_{i,j})^2 + ((g_{i+1,j} - g_{i,j})^2 + (g_{i,j+1} - g_{i,j})^2)$$

Der diskrete Fall

	i-1	i	i+1
j-1			
j			
j+1			

Glattheitsforderung s_{ij} am P. (i, j):

$$s_{ij} = \frac{1}{4}((f_{i+1,j} - f_{i,j})^2 + (f_{i,j+1} - f_{i,j})^2 + ((g_{i+1,j} - g_{i,j})^2 + (g_{i,j+1} - g_{i,j})^2)$$

Quadratischer Fehler $r_{i,j}$, image irrad. equation:

$$r_{i,j} = (I_{i,j} - R(f_i, j, g_i, j))^2$$

Der zu minimierende Gesamtfehler e :

$$e = \sum_i \sum_j (s_{ij} + \lambda r_{i,j}) \rightarrow \text{Minimum}$$

Minimierung des Fehlers

Die Ableitung von e nach $f_{k,l}$ und $g_{k,l}$:

	-1	
-1	4	-1
	-1	

$$\frac{\partial e}{\partial f_{k,l}} = 2(f_{k,l} - \bar{f}_{k,l}) - 2\lambda(I_{k,l} - R(f_{k,l}, g_{k,l})) \frac{\partial R}{\partial f}$$

$$\frac{\partial e}{\partial g_{k,l}} = 2(g_{k,l} - \bar{g}_{k,l}) - 2\lambda(I_{k,l} - R(f_{k,l}, g_{k,l})) \frac{\partial R}{\partial g}$$

mit

$$\bar{f}_{i,j} = \frac{1}{4}(f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1})$$

$$\bar{g}_{i,j} = \frac{1}{4}(g_{i+1,j} + g_{i,j+1} + g_{i-1,j} + g_{i,j-1})$$

$f_{i,j} - \bar{f}_{i,j}$ ist eine Approximation des Laplaceoperators $\nabla^2 f$.

Iterative Lösung: Relaxation

Die Ableitungen werden gleich Null gesetzt und die Gleichungen umgestellt:

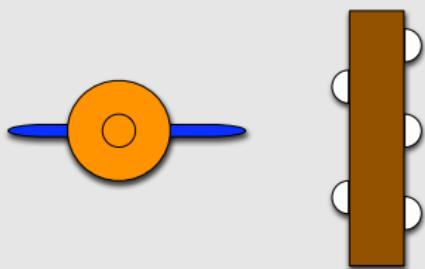
$$\begin{aligned} f_{k,I}^{n+1} &= \bar{f}_{i,j}^n + \lambda(I_{k,I} - R(\textcolor{red}{f}_{i,j}^n, \textcolor{blue}{g}_{i,j}^n)) \frac{\partial R}{\partial f} \\ g_{k,I}^{n+1} &= \bar{g}_{i,j}^n + \lambda(I_{k,I} - R(\textcolor{red}{f}_{i,j}^n, \textcolor{blue}{g}_{i,j}^n)) \frac{\partial R}{\partial g} \end{aligned}$$

- ☞ In jedem Iterationsschritt werden $\textcolor{red}{f}_{i,j}^n, \textcolor{blue}{g}_{i,j}^n$ durch $\bar{f}_{i,j}^n, \bar{g}_{i,j}^n$ an die Umgebung angepaßt.
- ☞ Die Startwerte an den Tangentialkanten werden so in die glatten Gebiete des Bildes propagiert.

Teil XIX

Geometrische Szenenrekonstruktion

Axiome zur Szenenrekonstruktion



48 Geometrische
Szenenrekonstruktion

49 Photometrische
Szenenrekonstruktion

50 Das Generalitätsprinzip

- Grundlegende Annahmen
- Szenenrekonstruktion als
Bootstrap-Problem

Axiome für die Szenenrekonstruktion

- ▶ Die grundlegende Schwierigkeit bei der Szenenrekonstruktion kann als **Inversionsproblem** betrachtet werden.
- ▶ Um die angesprochenen Tiefenhinweise auswerten zu können, müssen wir Annahmen über die Bildentstehung machen.
- ▶ Diese werden wir jetzt als Regeln explizit machen.[Neumann, 1982]

Annahme 1: Kohärenz

Nehme an, daß benachbarte Pixel zum selben Objekt gehören, solange es keine Hinweise für das Gegenteil gibt.

- ▶ Diese Annahme bildet die Grundlage für die Segmentierungsverfahren.
- ▶ Ein Hinweis auf eine Objektgrenze wäre insbesondere eine Kante.

Annahme 2: Persistenz

Ein Objekt, das an einer bestimmten Stelle in einer Bildfolge zu sehen ist, wird im Folgebild an einer Stelle in der Nähe zu sehen sein.

- ▶ Diese Annahme bildet die Grundlage für die Auswertung von Bildfolgen.
- ▶ Hinter der Persistenzannahme steckt die Tatsache, daß Materie wegen ihrer trägen Masse nicht beliebig schnell beschleunigt werden kann.

Voraussetzung: Der zeitliche Abstand der Einzelbilder der Folge ist klein genug, kein Zeitraffer.

Helligkeitsdiskontinuitäten

Annahme 3: Helligkeitsdiskontinuitäten

Objektgrenzen kommen nicht an Stellen vor, wo sich die Helligkeit gleichmäßig ändert.

- ▶ Eine andere Sicht auf die Kohärenzannahme.

Generalitätsprinzip

Annahme 4: Allgemeiner Standort

Lichtquellen und Betrachter befinden sich an einem allgemeinen Standort, wenn nicht anders bekannt.

- ▶ *Das Prinzip des allgemeinen Standortes:*
Strukturen im Bild kommen nicht durch eine spezielle Sicht des Beobachters auf die Szene zustande.
 - ▶ *Das Prinzip der allgemeinen Beleuchtung:*
Strukturen im Bild kommen nicht durch eine spezielle Position der Lichtquelle in der Szene zustande.
- ☞ Diese Annahme postuliert, daß die Aufnahmebedingungen so sind, daß eine **typische** Ansicht des Objekts entsteht.

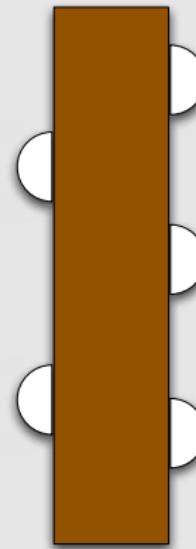
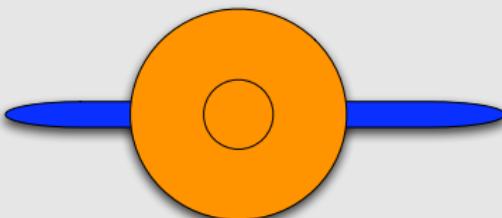
Ockham's razor

William of Ockham:

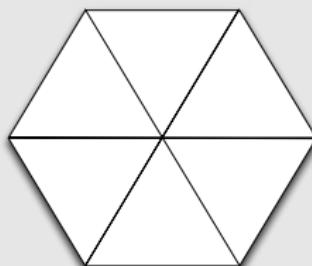
≈ 1285-1349 Die beste Erklärung eines Phänomens ist immer die einfachste, auf am wenigsten Annahmen beruhende.

Untypische Ansichten

Untypische Ansichten sind auch für Menschen schwer zu deuten:



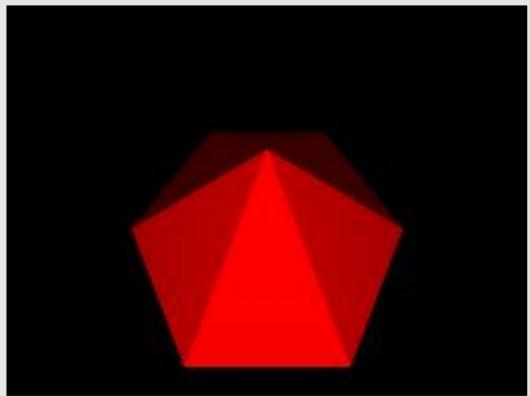
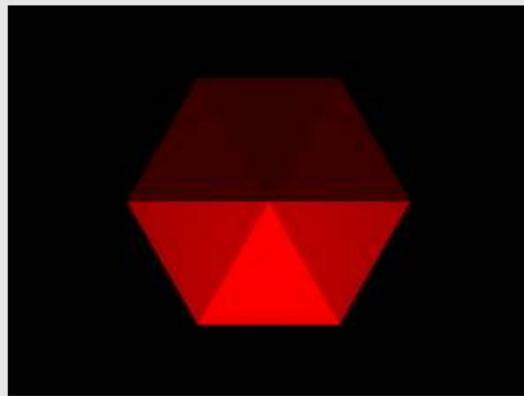
Beispiel: eine symmetrische Figur



Wenn dieses Bild von einem **allgemeinen Standort** aus aufgenommen wurde, kann es keine Pyramide sein:

- ☞ Nur wenn eine Pyramide senkrecht von oben aufgenommen wird, ist die Spitze im Mittelpunkt des Pyramidenbildes.
- ☞ Daher muß dieses Bild nach Annahme 4 als ebene Figur interpretiert werden.

Eine Pyramide in spezieller und in allgemeiner Ansicht



Annahme 5: Allgemeines Modell

Benutze zur Bilddeutung das allgemeinste Modell.

- ▶ Anders ausgedrückt: Wähle diejenige Interpretation, bei der die Parameter der Bildentstehung am wenigsten eingeschränkt werden [Binford, 1981].
- ▶ Die Annahme, daß wir eine Pyramide genau senkrecht betrachten, schränkt die Parameter stärker ein als die Annahme, daß das Objekt eben ist.
- ▶ Die Annahmen 4 und 5 sind Anwendungen des **Generalitätsprinzips**.

Annahme 6: Glattheit von Oberflächen

Physikalische Oberflächen sind i.A. glatt.

- ▶ Diese Annahme ist die Basis für die Bildsegmentierung über Homogenitätskriterien.

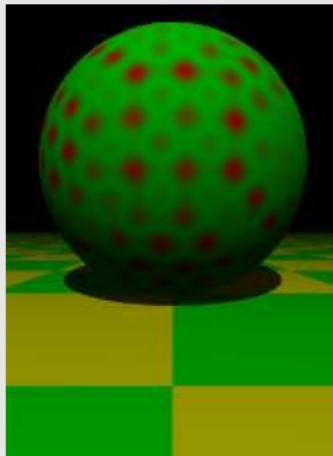
Annahme 7:

Parallele Kurven im Bild sind auch im Raum parallel.

- ▶ Diese Annahme folgt aus dem Generalitätsprinzip: Es erfordert einen speziellen Standort, damit uns nicht parallele Kurven als parallel erscheinen.
- ▶ Insbesondere die Ames-Raum-Illusion basiert darauf, daß diese Annahme verletzt wird.

Der Ames-Raum in spezieller Ansicht

Ames-Raum



48 Geometrische Szenenrekonstruktion

49 Photometrische
Szenenrekonstruktion

50 Das Generalitätsprinzip

- Grundlegende Annahmen
- Szenenrekonstruktion als Bootstrap-Problem

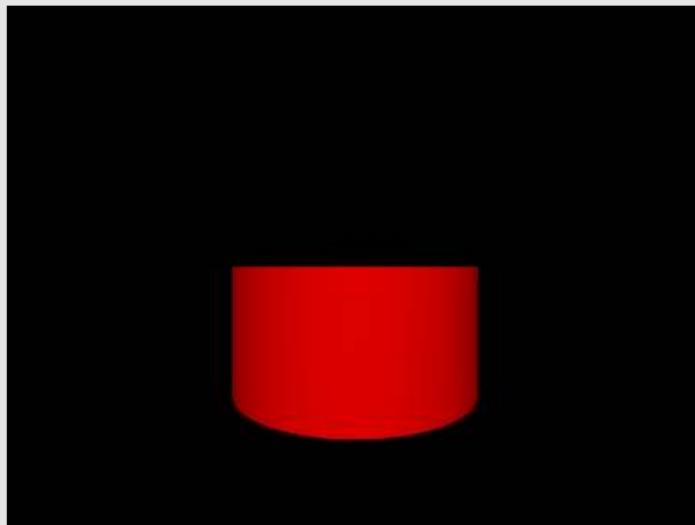
☞ Ein Texturgradient geht in der Regel auf perspektivische Verkürzung zurück, nicht auf spezielle Muster der Oberfläche.

Annahme 11: Geraden

Geraden im Bild sind auch Geraden im Raum.

- ▶ Diese Annahme folgt ebenfalls aus dem Generalitätsprinzip: Es erfordert einen **speziellen Standort**, damit uns gekrümmte Kurven als Gerade erscheinen.

Ein Zylinder in spezieller Ansicht



Der obere Deckel des Zylinders und die optische Achse der Kamera liegen in einer Ebene, so daß der Rand des Deckels als Gerade abgebildet wird.

Verdeckung und Vertices

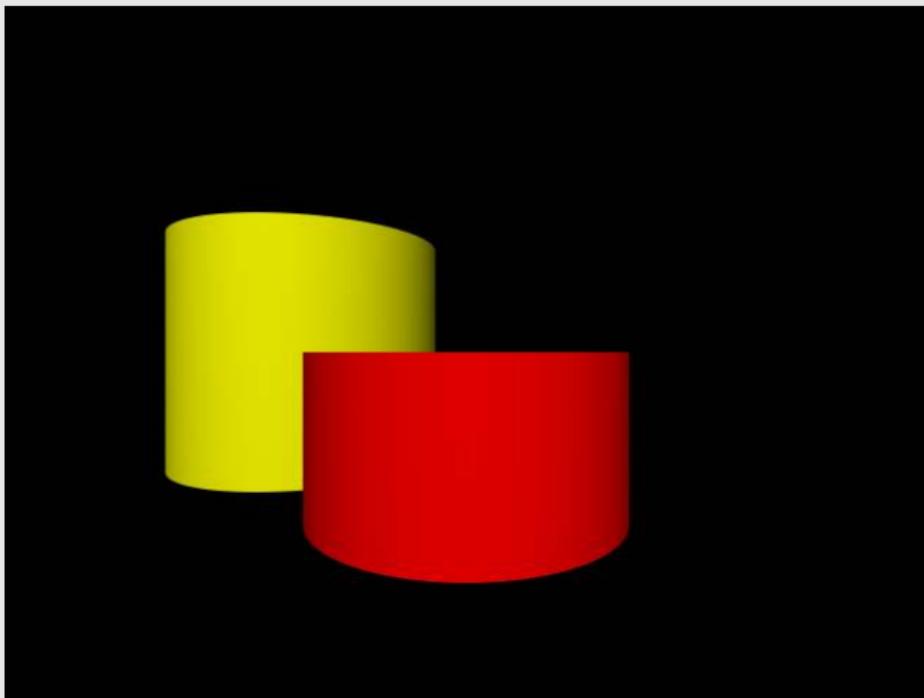


Annahme 13: T-Vertices

Der Stamm eines T-Vertex ist nicht dichter beim Betrachter als der Balken.

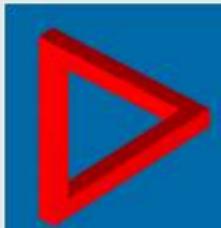
- ▶ Eine Linie (Stamm des T), die in einer durchgehenden Linie endet (Balken des T), wird als verdeckt interpretiert.

Beispiel: T-Vertices bei Verdeckung



T-Vertices an den verdeckenden Kanten

Verdeckung und Vertices



Annahme 14: Vertices

Zwei oder mehr Kurven, die im Bild am selben Ort terminieren, terminieren auch im Raum am selben Ort.

Beachte:

- ▶ Es erfordert einen speziellen Standort, damit mehrere Linienendpunkte in Deckung erscheinen.
- ▶ Die Illusion des unmöglichen Objektes entsteht, weil die Annahme 14 verletzt ist.

Das unmögliche Objekt in allgemeiner Ansicht

Ein seltsames Vogelnest



Annahme 15: Starrheit

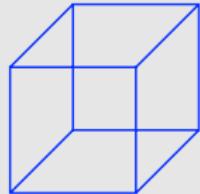
Physikalische Objekte sind starre Körper, solange nichts auf das Gegenteil hindeutet.

- ▶ Starrheit ist ein wichtiges Kriterium für Bildfolgen: Die relativen Abstände der Objektkomponenten im Raum bleiben von Bild zu Bild gleich.
- ▶ Die Starrheitsannahme gilt nur eingeschränkt in allgemeinen Szenen.

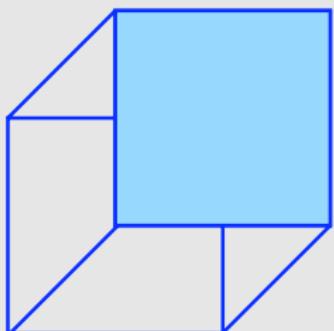
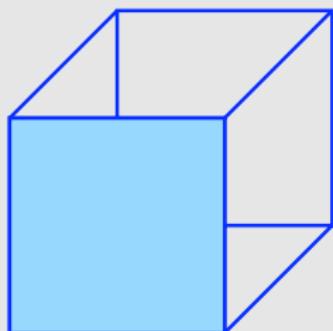
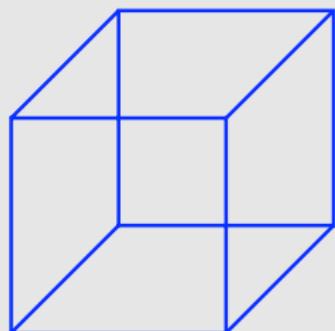
Der Necker-Würfel

Unser Sehsystem kann spontan zwischen verschiedenen Interpretationen von mehrdeutigen Bildern hin- und herwechseln.

- ▶ Ein berühmtes Beispiel ist der Necker-Würfel, die Zeichnung eines Drahtrahmenwürfels in Parallelprojektion, benannt nach dem Schweizer Naturforscher L. A. Necker (1832).
 - ▶ Dieses Bild hat zwei verschiedenen Interpretationen als Kubus.
 - ▶ Die Mehrdeutigkeit geht darauf zurück, daß die Flächen fehlen, sie liegt nicht an der Parallelprojektion.

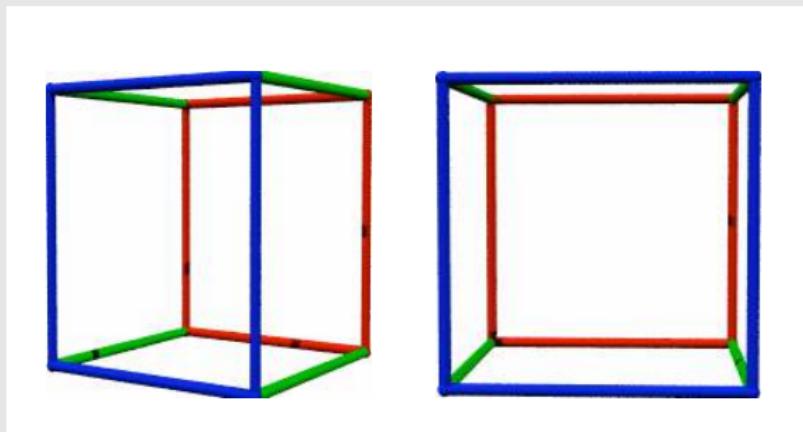


Mögliche Interpretationen



Ein Drahtrahmenwürfel in Zentralprojektion

Wenn wir die rote Fläche vorne liegen sehen, dann sehen wir anstelle des Würfels einen Pyramidenstumpf.



Der Necker-Würfel mit Pov-Ray animiert

Maske

Zur Starrheitsannahme

Beobachtung:

- ▶ Wenn wir den Drahtrahmenwürfel einäugig betrachten und den Kopf leicht bewegen, können wir eine sich verformende Pyramide sehen.
- ▶ Unser Sehsystem kann den Necker-Würfel als nicht-starres Objekt interpretieren, obwohl er ein starrer Körper ist.
- ▶ Für den menschlichen Sehsinn scheint also die Starrheitsannahme nicht immer relevant zu sein.

Annahme 16: Das Prinzip der allgemeinen Bewegung:

Bewegte Objekte bewegen sich so, daß spezielle Sichten nur selten in Folgen von allgemeinen Ansichten auftreten.[Dreschler-Fischer, 1987]

- ▶ Diese Annahme ermöglicht es, in Bildfolgen die Verletzungen des Generalitätsprinzips zu entdecken.

Zusammenfassung

Beachte:

Die meisten in der Szenenrekonstruktion angewendeten Regeln lassen sich auf die Annahmen 4 und 5 (Generalitätsprinzip) zurückführen.

Zusammenfassung: Verfahren und Annahmen

Shape from Texture: isotrope Textur, allgemeine Beleuchtung

Shape from Shading: bekannte Reflektanz, bekannte Beleuchtung

Shape from Contour: Tangentialkanten,
Diskontinuitätskanten

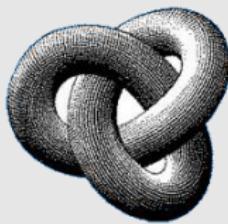
Geraden und Parallelität: allgemeiner Standort

Stereo, Bewegung: Korrespondierende Merkmale

Das Bootstrap-Problem

- ▶ Jedes der genannten Rekonstruktionsverfahren geht von einer **impliziten Modellvorstellung** von den zu rekonstruierenden Objektmerkmalen aus und ist daher nur unter bestimmten Voraussetzungen anwendbar.
- ▶ Bevor ein „shape-from...“-Verfahren eingesetzt werden kann, muß daher verifiziert werden, daß diese Voraussetzungen erfüllt sind.

Der Teufelskreis



Der Teufelskreis:

Um Objektstrukturen aus Bildstrukturen geometrisch zu rekonstruieren, müssen die erzeugenden Objektstrukturen zumindestens qualitativ schon **bekannt** sein.

Lösungsansatz: Bootstrap



Vorgehen:

Nehme ein vorläufiges Modell an und revidiere die Annahme,

- ▶ wenn spätere Ansichten der Annahme widersprechen
- ▶ oder wenn die Rekonstruktion benachbarter Objekte der Annahme widerspricht.

- ☞ Erfordert nicht-monotone Logik und ein Beschränkungsverwaltungssystem

Bewegungsstereo und optischer Fluß
Das Korrespondenzproblem bei Bildfolgen
Stereoskopie

Rekonstruktion starrer Körper
Bewegungsfeld einer Translation

Teil XX

Mehrfache Objektansichten

Bewegungsstereo und optischer Fluß



Ein Bewegungsfeld

51

Bewegungsstereo und optischer Fluß

- Rekonstruktion starrer Körper
- Bewegungsfeld einer Translation

52

Das Korrespondenzproblem bei Bildfolgen

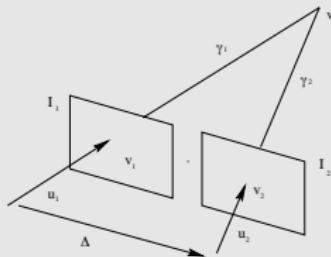
53

Stereoskopie

Literatur zum Motion Stereo

- ❖ Ullman, S. (1979).
The Interpretation of Visual Motion.
The MIT Press, Cambridge, Mass.
- ❖ Clocksin, W. (1978).
Determining the orientation of surfaces from optical flow.
In Sleeman, D., editor, *AISB/GI-78 on Artificial Intelligence*, pages 73–102, Hamburg.
- ❖ Horn, B. K. P. (1986).
Robot Vision.
The MIT Press, Cambridge, Mass. McGraw-Hill, New York.

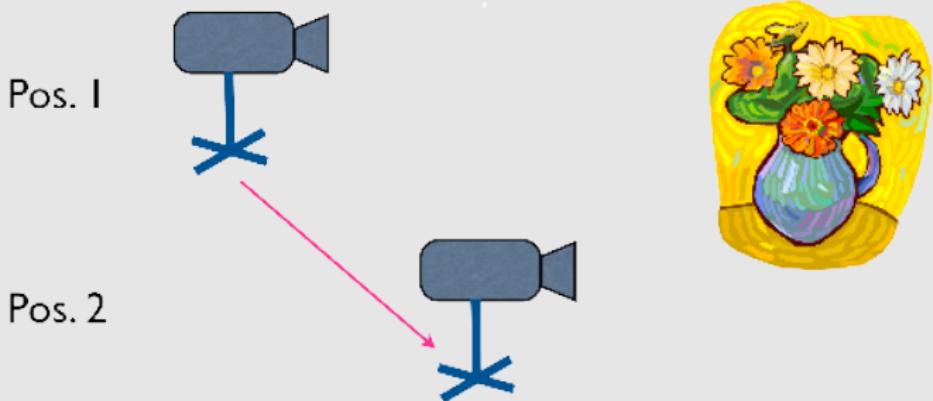
Stereoskopie



Wenn wir eine stereoskopische Aufnahme haben, können wir bei bekannter Basislinie mittels Triangulation die Tiefe messen.

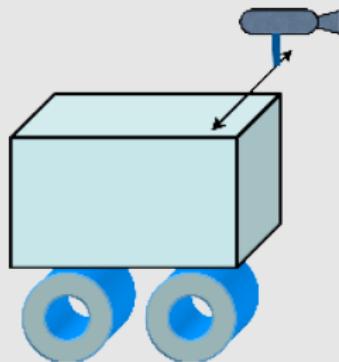
Triangulation

Stereo mit nur einer Kamera



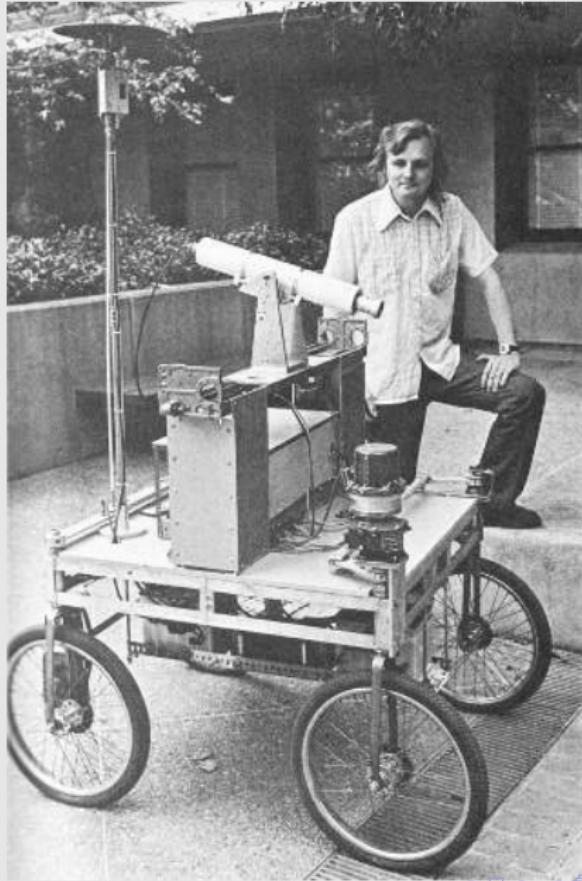
- ▶ Bei stationären Objekten kann die Kamera zur zweiten Position verschoben werden.

Erweiterung:

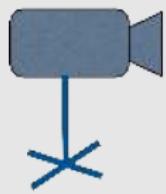


- ▶ Kamera bewegt sich ferngesteuert auf einer Laufschiene, z.B. am Stanford Cart (1980) von Moravec, 9 Positionen.
- ▶ **Vorteil:** Insgesamt große Basislinie, aber geringe Unterschiede von Position zu Position

Hans Moravec am Stanford Cart



Motion Stereo (Nevatia 1976)

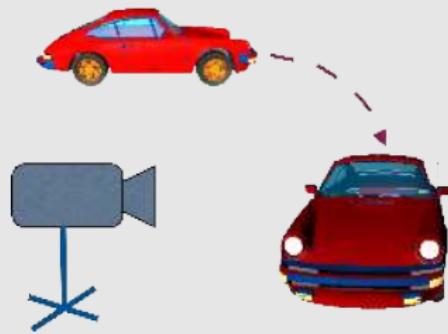


Objekt in kontrollierter Bewegung

Umkehrung der Relativbewegung:

- ▶ Das Objekt wird kontrolliert relativ zur Kamera bewegt.
- ▶ Mathematisch kein Unterschied zum vorherigen Verfahren.

Freie Objektbewegung



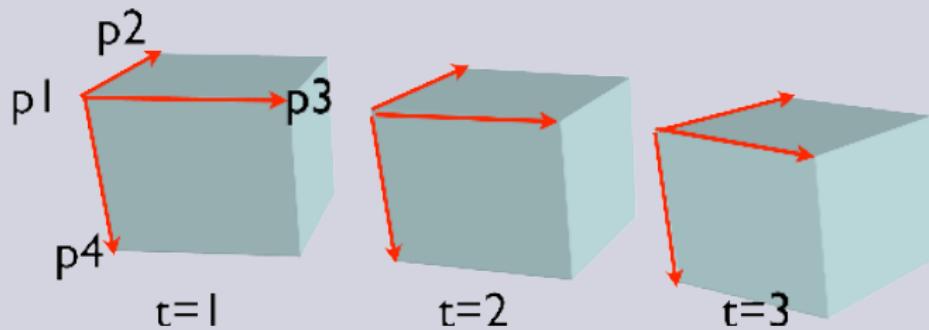
Objekt in freier
Bewegung

Für eigenständig bewegte
Objekte:

- ☞ Zusätzlich zur räumlichen Struktur muß die Objektbewegung relativ zur Kamera ermittelt werden.

Bewegungsstereo: Das Problem

Ein Objekt zu drei Zeitpunkten



Problem

- ▶ *starres, unbekanntes Objekt mit den Punkten $p_1 \dots p_4$,*
- ▶ *bewegungsbedingt unterschiedliche Ansichten,*
- ▶ *unbekannte Bewegungsparameter*

Homologe Punkte

Definition (Homologe Punkte)

Homologe Punkte sind Punkte in verschiedenen Bildern, die Abbildungen desselben Objektpunktes sind.

Bewegungsstereo liefert:

Lösung

Structure from Motion

- ▶ *Räumliche Struktur (für homologe Bildpunkte)*
- ▶ *räumliche Trajektorie (Translation, Rotation)*
- ▶ *falls notwendig: Gruppierung in Objekte*

Teilprobleme

Extraktion geeigneter Bildelemente: Körperfest,
lokalisierbar, verfolgbar,

Teilprobleme

Extraktion geeigneter Bildelemente: Körperfest,
lokalisierbar, verfolgbar,

Korrespondenzbestimmung: Korrespondenzketten,

Teilprobleme

Extraktion geeigneter Bildelemente: Körperfest,
lokalisierbar, verfolgbar,

Korrespondenzbestimmung: Korrespondenzketten,
Gruppierung in unabhängig bewegte Objekte oder
Objektteile,

Teilprobleme

Extraktion geeigneter Bildelemente: Körperfest,
lokalisierbar, verfolgbar,

Korrespondenzbestimmung: Korrespondenzketten,
Gruppierung in unabhängig bewegte Objekte oder
Objektteile,

3D-Analyse der bewegungsbedingten
Ortsveränderungen,

Teilprobleme

Extraktion geeigneter Bildelemente: Körperfest,
lokalisierbar, verfolgbar,

Korrespondenzbestimmung: Korrespondenzketten,
Gruppierung in unabhängig bewegte Objekte oder
Objektteile,

3D-Analyse der bewegungsbedingten
Ortsveränderungen,

Alternative:

Die Gruppierung und 3D-Analyse werden in einem Schritt
durchgeführt.

Anwendungen von Bewegungsstereo in der Robotik

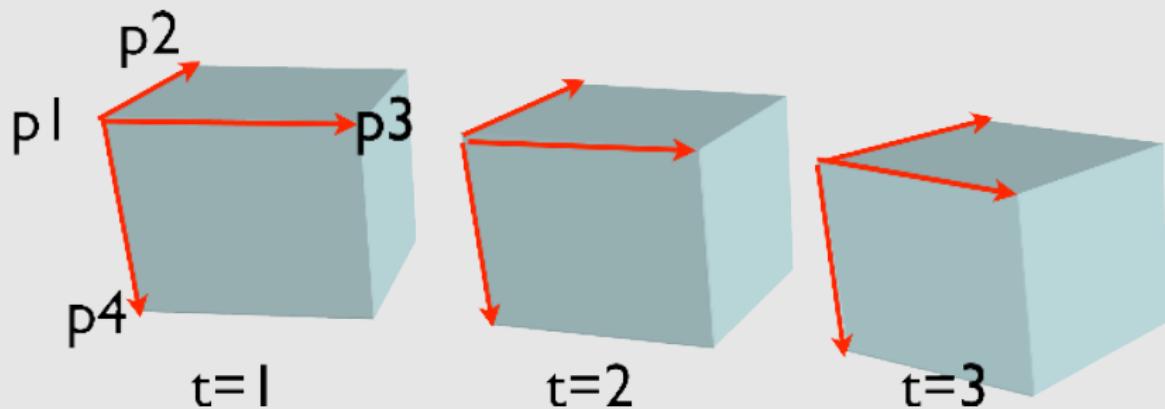
Vermessen eines unbekannten Objektes:

- ▶ Aufbau eines Objektmodells durch Beobachtung
- ▶ Werkstücke, Hindernisse
- ▶ Ermitteln von Eigen- und Objektbewegung

Navigatation: Hindernisvermeidung

Ergänzung für andere Shape From- Verfahren

3-D-Analyse von Punktverschiebungen

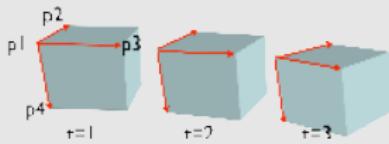


Problem (Wie viele Punkte in wie vielen Ansichten sind nötig?)

Annahme: *starrer Körper*

Gegeben: P Punkte, N Ansichten

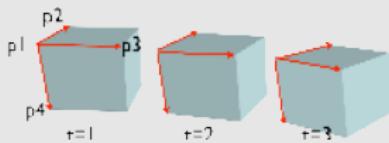
P Punkte, N Ansichten liefern:



Gegeben:

- ▶ $3 \cdot N \cdot P$ Bewegungsgleichungen
- ▶ $2 \cdot N \cdot P$ Positionen auf
Abbildungsstrahl
- ▶ Eine willkürliche Skalierung

P Punkte, N Ansichten liefern:



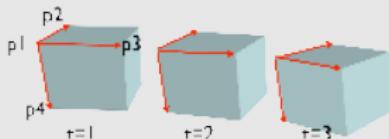
Gegeben:

- ▶ $3 \cdot N \cdot P$ Bewegungsgleichungen
- ▶ $2 \cdot N \cdot P$ Positionen auf
Abbildungsstrahl
- ▶ Eine willkürliche Skalierung

Unbekannt:

- ▶ $6 \cdot (N - 1)$ Bewegungsparameter
- ▶ $3 \cdot P$ Punktkoordinaten

P Punkte, N Ansichten liefern:



Gegeben:

- ▶ $3 \cdot N \cdot P$ Bewegungsgleichungen
- ▶ $2 \cdot N \cdot P$ Positionen auf
Abbildungsstrahl
- ▶ Eine willkürliche Skalierung

Unbekannt:

- ▶ $6 \cdot (N - 1)$ Bewegungsparameter
- ▶ $3 \cdot P$ Punktkoordinaten

Lösung

$$\Rightarrow P \geq 3 + \frac{2}{2N - 3}$$

Notwendige Zahl von Ansichten und Punkten

$$P \geq 3 + \frac{2}{2N - 3}$$

Algorithmen zur Lösung

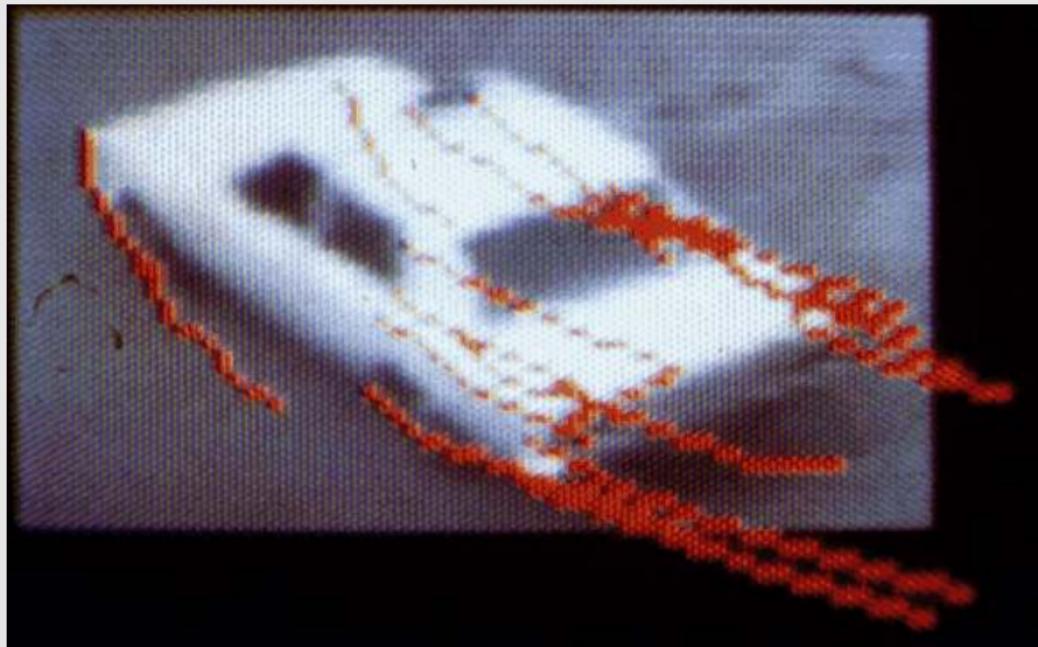
N	P	Lösung
2	$P \geq 5$	Tsai und Huang 81
3	4	Nagel und Neumann 81
4	4	Ullman-1979, ...

Einschränkungen

- ▶ Keine 3D-Information aus einer reinen Drehung um das Kamerazentrum.
- ▶ Keine 3D-Information bei einer reinen Translation entlang des Abbildungsstrahls eines Punktes.

Beispiel: MORIO-System

Bewegungsspuren von markanten Punkten über 21 Bilder einer Folge

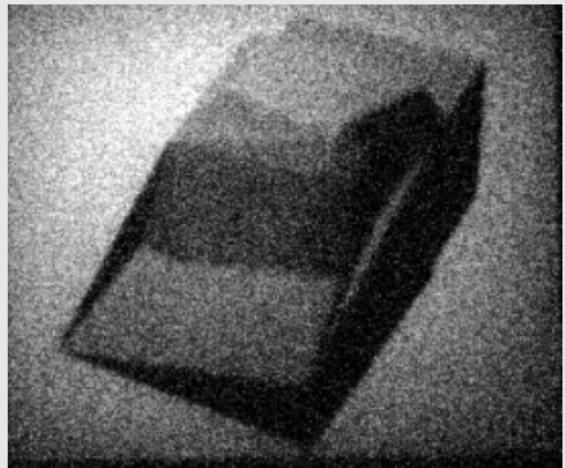


Beispiel: MORIO-System

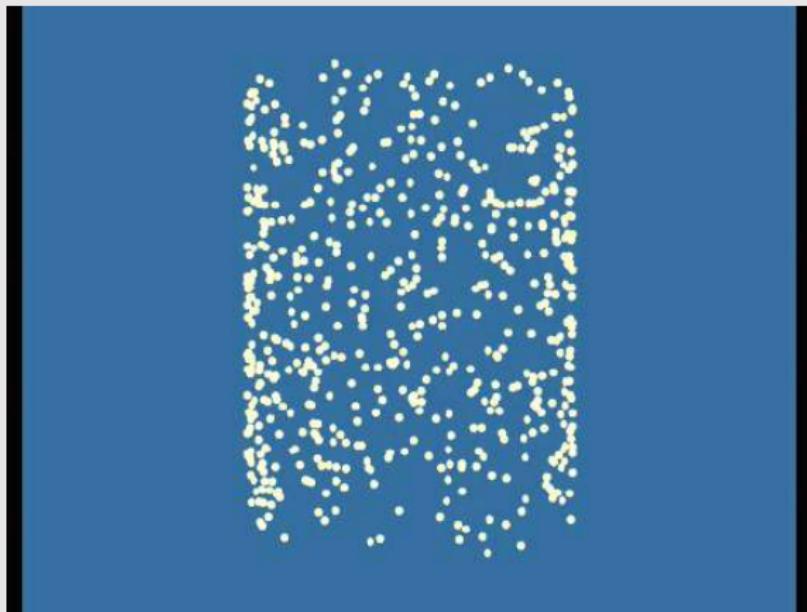
3D-Rekonstruktion der markanten Punkten als konvexe Hülle, dem ersten Bild der Folge überlagert.



MUSCAT: Modellierung von Konkavitäten



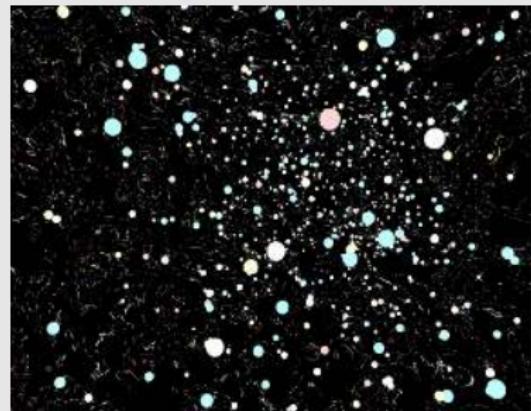
Ein transparenter Zylinder



Ein transparenter Zylinder

Zwei Bilder derselben Szene in Zentralprojektion

- ▶ Das zweite Bild wurde von einem höheren Standort aus aufgenommen als das erste.
- ▶ In schneller Folge abwechselnd projiziert entsteht ein räumlicher Eindruck.

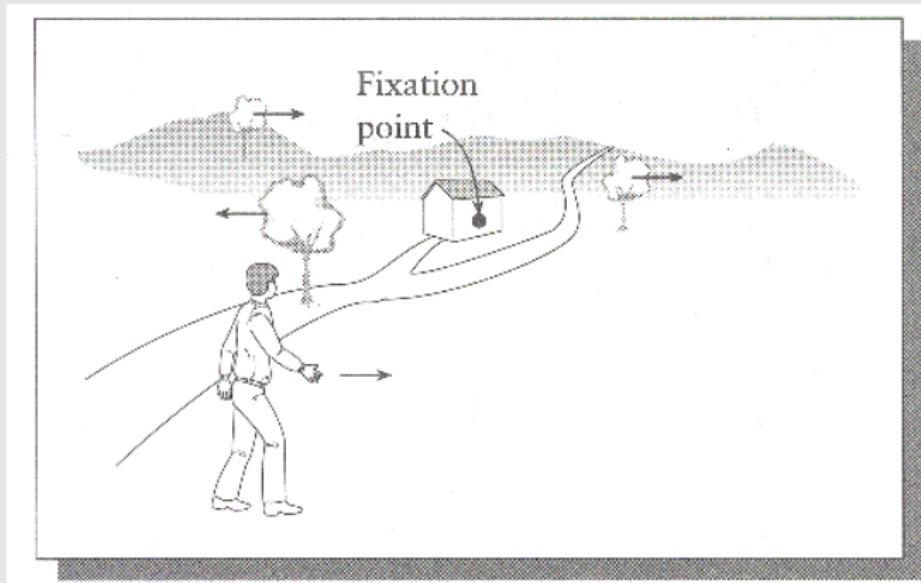


Beispiel: Bewegungsstereo

Bewegungsstereo

Ein bewegter Beobachter

und scheinbare Bewegungen der Umgebung



Das Bewegungsfeld

Definition (Bewegungsfeld)

Das Vektorfeld der Verschiebungen vom Zeitpunkt einer Bildaufnahme zum Zeitpunkt der nächsten Aufnahme (Verschiebungsvektoren) heißt „Bewegungsfeld“ (motion field).

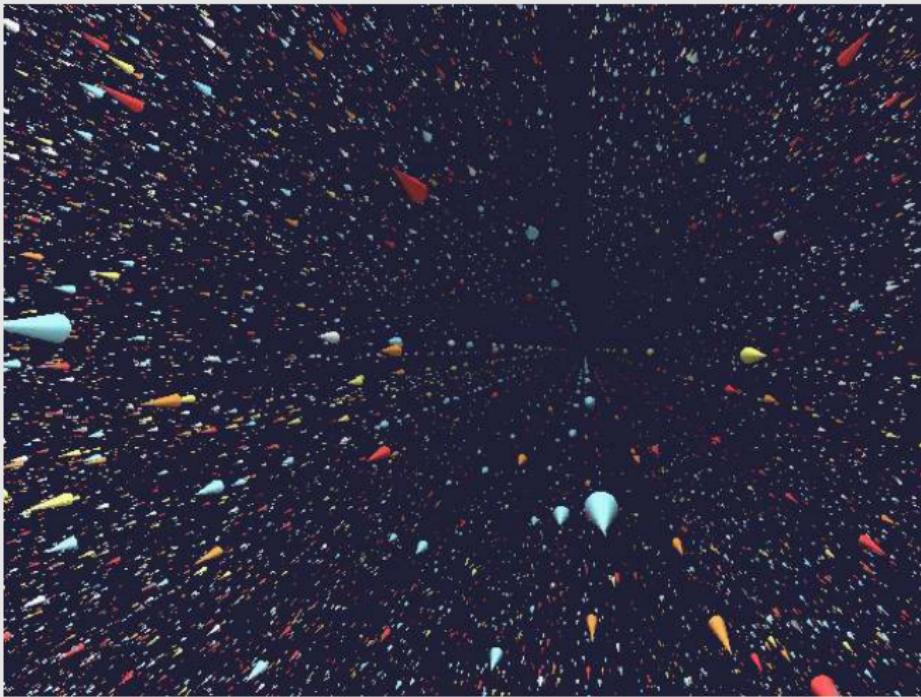
- ☞ Aus dem Bewegungsfeld kann auf die Entfernung und Form eines Objektes geschlossen werden.

Translation: Blick in Bewegungsrichtung

Das Bewegungsfeld einer Translationsbewegung

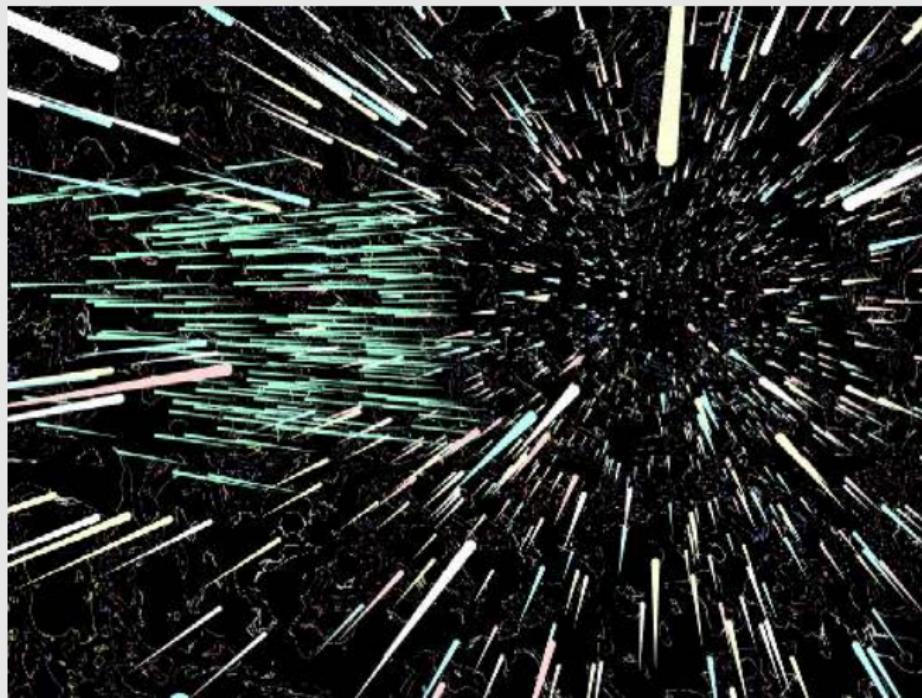
- ▶ Das Bewegungsfeld ist radial strukturiert.
- ▶ Alle Objekte bewegen sich im Bild auf Geraden, die sich in einem gemeinsamen Punkt schneiden.
- ▶ Dieser Punkt, aus dem die Welt hervor zu strömen scheint, heißt **Expansionspunkt** (FOE, focus of expansion).
- ▶ Der Expansionspunkt ist nichts anderes als der Fluchtpunkt der Geradenschar parallel zur Bewegungsrichtung.

Das Bewegungsfeld einer Translation, Blick seitwärts



Translationsbewegung, Blick seitwärts

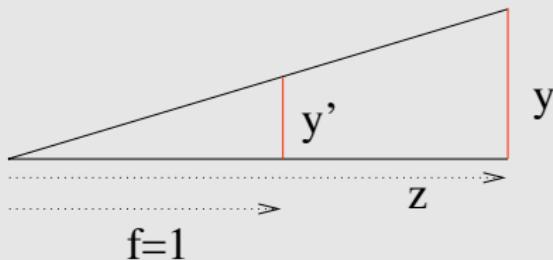
Expansionspunkte für zwei überlagerte Bewegungsfelder



Translationsbewegung, Blick seitwärts

Annahmen:

- Weltkoordinatensystem = Kamerakoordinatensystem
- Brennweite $f = 1$.



- Dann hat der Weltpunkt (x, y, z) den Bildpunkt (x', y') :

$$x' = \frac{x}{z} \quad \text{und} \quad y' = \frac{y}{z}$$

Der Beobachter bewege sich mit der Geschwindigkeit \vec{O}
und das Koordinatensystem wandere mit dem
Beobachter mit:

$$\vec{O} = \left(-\frac{dx}{dt}, -\frac{dy}{dt}, -\frac{dz}{dt} \right) = (-u, -v, -w)$$

Der Weltpunkt $P = (x_0, y_0, z_0)$ hat nach der Zeit t den
Bildpunkt P'_t mit

$$P'_t = (x', y') = \left(\frac{x_0 + u \cdot t}{z_0 + w \cdot t}, \frac{y_0 + v \cdot t}{z_0 + w \cdot t} \right)$$

Grenzwertbetrachtung:

Der Ausgangspunkt der Bewegung zum Zeitpunkt
 $t = -\infty$.

$$P'_{-\infty} = \left(\frac{u}{w}, \frac{v}{w} \right)$$

Expansionspunkt

Definition (Expansionspunkt)

Der **Expansionspunkt (focus of expansion) FOE** ist der Bildpunkt des scheinbaren Ausgangspunktes der Bewegung zum Zeitpunkt $t = -\infty$.

$$FOE = \left(\frac{u}{w}, \frac{v}{w} \right)$$

- ☞ Die Lage des Expansionspunktes hängt nur von der *Bewegungsrichtung* ab, nicht vom *Betrag der Geschwindigkeit!*

Tiefenmessung:

Sei $D(t)$ der Abstand des Bildpunktes P_t vom FOE, der sich mit der Geschwindigkeit $\vec{v}(t) = \frac{dD(t)}{dt}$ ändert, dann gilt:

$$\underbrace{\frac{D(t)}{|\vec{v}(t)|}}_{\text{Bild,2D}} = \frac{z(t)}{\underbrace{w}_{\text{Welt,3D}}} \quad \frac{\text{Distanz}}{\text{Geschwindigkeit}}$$

Satz

Der *Abstand im Bild* vom *Expansionspunkt* verhält sich zur *Verschiebungsgeschwindigkeit im Bild* wie die *Tiefe* zu der *Geschwindigkeitskomponente w* in z-Richtung.

Sei für einen Punkt P_1 die Tiefe $z_1(\textcolor{red}{t})$ zum Zeitpunkt $\textcolor{red}{t}$ bekannt, dann lässt sie sich für alle anderen Punkte P_i gleicher Geschwindigkeit berechnen:

$$\frac{D_1(\textcolor{red}{t})}{v_1(\textcolor{red}{t})} = \frac{z_1(\textcolor{red}{t})}{\textcolor{blue}{w}}$$

$$\frac{D_i(\textcolor{red}{t})}{v_i(\textcolor{red}{t})} = \frac{z_i(\textcolor{red}{t})}{\textcolor{blue}{w}} \mid \textcolor{blue}{w} \text{ eliminieren}$$

$$z_i(\textcolor{red}{t}) = z_1(t) \cdot \frac{D_i(\textcolor{red}{t})}{D_1(\textcolor{red}{t})} \cdot \frac{v_1(\textcolor{red}{t})}{v_i(\textcolor{red}{t})}$$

Für die x- und y-Koordinaten folgt dann aus den Projektionsgleichungen und der bekannten Tiefe z :

$$z(t) = w \cdot \frac{D(t)}{v(t)}$$

$$x(t) = x'(t) \cdot z(t) = x' \cdot w \cdot \frac{D(t)}{v(t)}$$

$$y(t) = y'(t) \cdot z(t) = y' \cdot w \cdot \frac{D(t)}{v(t)}$$

w ist ein Skalierungsfaktor. Auch wenn die Geschwindigkeit nicht bekannt ist, kann die Form von bewegten Objekten anhand des Bewegungsfeldes dreidimensional rekonstruiert werden. Es kann nur nicht der Maßstab bestimmt werden.

Szenenrekonstruktion

- ▶ Wenn die Beobachtergeschwindigkeit bekannt ist und das Bewegungsfeld gemessen werden kann, dann kann die Szene räumlich rekonstruiert werden.
- ▶ Wenn der Betrag der Geschwindigkeit allerdings nicht bekannt ist, kann nur die Form der Objekte (relative Größe) bestimmt werden;
- ▶ das Bewegungsfeld von nahen, kleinen und langsamen Objekten kann nicht vom Bewegungsfeld von entfernten, großen und schnellen Objekten unterschieden werden.

Animation

Beachte:

- ▶ Das Verfahren ist nur anwendbar, wenn eine reine Translation vorliegt.
- ▶ Bei sich drehenden Objekten kann das Verfahren nicht angewendet werden.

Allgemeine Bewegung

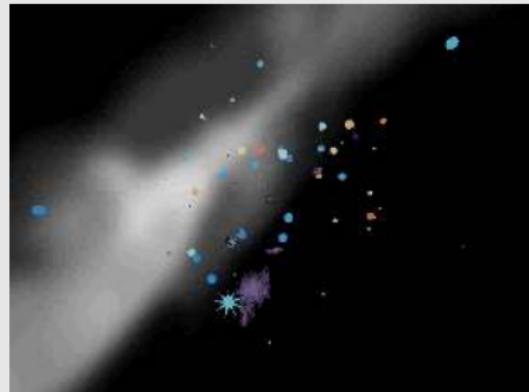
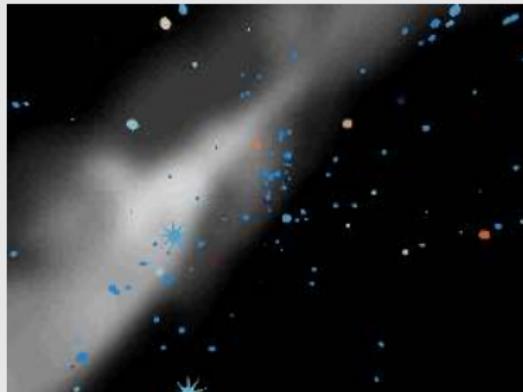
Relativistische Effekte

Wenn die Eigenbewegung sehr schnell ist, sind zusätzlich relativistische Effekte zu berücksichtigen.

- ▶ Die scheinbare **Position** verändert sich (Aberration).
- ▶ Die scheinbare **Farbe** verändert sich (relativistischer Dopplereffekt).

Ein Flug mit nahezu Lichtgeschwindigkeit zum großen Orionnebel M42

Ein Flug mit nahezu Lichtgeschwindigkeit zum großen Orionnebel M42



- ▶ Bei relativistischen Geschwindigkeiten scheinen sich durch die Aberration die seitlichen Objekte in Bewegungsrichtung zu verschieben.
- ▶ Die Farben verschieben sich zu Farben höherer Frequenzen.

Konsequenzen für die Animation

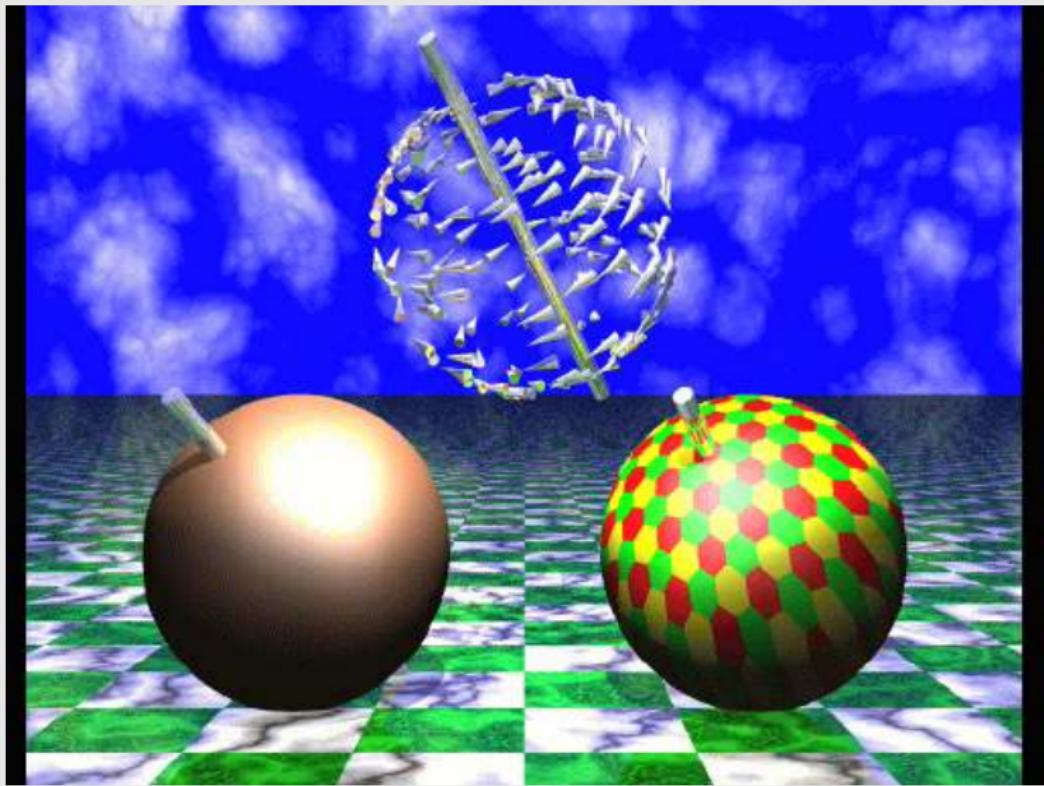
Wenn eine Animation einen realistischen Tiefeneindruck ergeben soll, dann müssen die Objekte so abgebildet werden, daß das Bewegungsfeld korrekt ist:

- ▶ also nahe Objekte müssen sich stärker verschieben als weiter entfernte Objekte.
- ▶ Sonst erscheint die Szene unnatürlich flach, wie in vielen Zeichentrickfilmen zu sehen, wo sich die Figuren vor einen starren Hintergrund bewegen.

Bewegungsfeld vs. optischer Fluß

- ▶ Nicht jedes Bewegungsfeld ist für das Auge sichtbar, denn wenn sich unstrukturierte Objekte mit einer unstrukturierten Oberfläche verschieben, kann das Bild unverändert bleiben.
- ▶ Umgekehrt aber können Beleuchtungsveränderungen das Bild so verändern, daß es wie ein Bewegungsfeld erscheint.

Drei rotierende Kugeln



Alle drei Kugeln rotieren synchron

Ein vorgetäuschtes Bewegungsfeld

Das Korrespondenzproblem bei Bildfolgen

Definition (Korrespondenzproblem)

Das Problem in einer Menge von Bildern **homologe Punkte** zu finden, heißt **Korrespondenzproblem**.

Das Korrespondenzproblem tritt auf

- ▶ beim Berechnen des Bewegungsfeldes,

- ☞ Das Korrespondenzproblem ist nicht immer eindeutig zu lösen.

Das Korrespondenzproblem bei Bildfolgen

Definition (Korrespondenzproblem)

Das Problem in einer Menge von Bildern **homologe Punkte** zu finden, heißt **Korrespondenzproblem**.

Das Korrespondenzproblem tritt auf

- ▶ beim Berechnen des Bewegungsfeldes,
 - ▶ **bei der Stereoskopie,**
-
- ☞ Das Korrespondenzproblem ist nicht immer eindeutig zu lösen.

Das Korrespondenzproblem bei Bildfolgen

Definition (Korrespondenzproblem)

Das Problem in einer Menge von Bildern **homologe Punkte** zu finden, heißt **Korrespondenzproblem**.

Das Korrespondenzproblem tritt auf

- ▶ beim Berechnen des Bewegungsfeldes,
 - ▶ bei der Stereoskopie,
 - ▶ **beim Registrieren von Bildern.**
- ☞ Das Korrespondenzproblem ist nicht immer eindeutig zu lösen.

Optischer Fluß

Definition (Optischer Fluß)

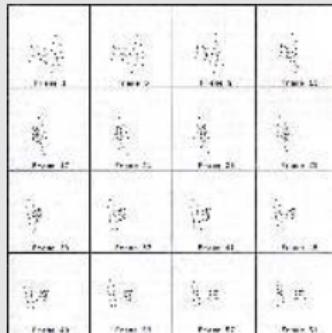
Das Vektorfeld der vom visuellen System wahrgenommenen Verschiebungen heißt **optischer Fluß** (optical flow).

Definition (Grauwertverschiebungen)

Grauwertverschiebungen werden diejenigen Approximationen an das Bewegungsfeld genannt, die von Algorithmen zur Lösung des Korrespondenzproblems berechnet werden.

Sie heißen so, da die klassischen Algorithmen lokal die Verschiebungen von Grauwertkonstellationen von einem Bild zum nächsten berechnen.

Moving Light Displays MLD

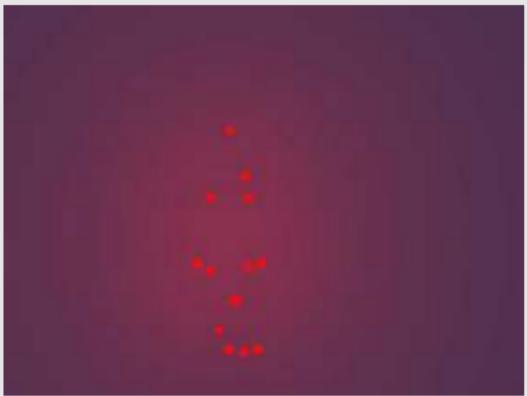
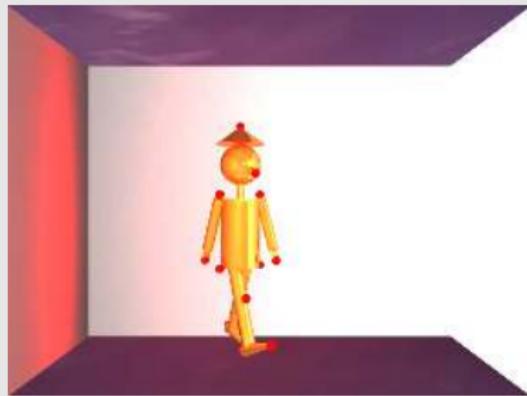


Das menschliche Sehsystem kann sehr gut die Bewegungen einzelner Punkte deuten und die Szene räumlich rekonstruieren.

Beispiel: MLD Herrchen mit Hund

MLD

Ein simuliertes MLD

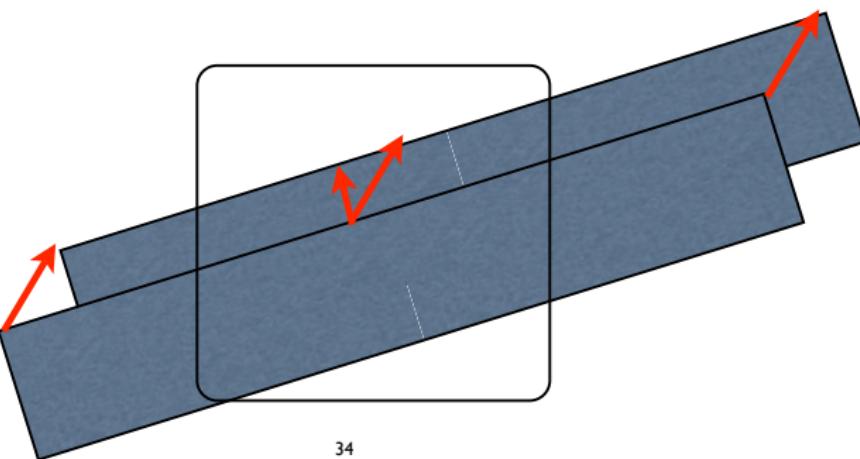


Am simulierten Roboter wurden an allen strategischen Stellen Lichter angebracht.

Ein simuliertes MLD

Ein simuliertes MLD

Das Blendenproblem



34

Die Grauwertverschiebungen können generell nur in Richtung des Gradienten bestimmt werden, wenn nur eine lineare, durchgehende Kante im Bild zu sehen ist.

Blendeneffekt

- ▶ Unser Sehsystem kann den optischen Fluß nur dort messen, wo eine Struktur in zwei Richtungen variiert. An glatten Kanten, die im gesamten Bildfeld keine markanten Punkte aufweisen, kann nur die Komponente des Flusses senkrecht zur Kante festgestellt werden.
- ▶ Wir sehen dann eine Bewegung in diese Richtung, auch wenn die Bewegung in Wahrheit noch eine weitere Komponente in Kantenrichtung hat.

Barber pole illusion



Barber pole illusion

Barber pole illusion

Barber pole illusion

Teil XX

Mehrfache Objektansichten

Stereoskopie



51 Bewegungsstereo und optischer Fluß

52 Das Korrespondenzproblem bei Bildfolgen

53 Stereoskopie

- Binokulares Sehen
- Stereoskopische Tiefenrekonstruktion
- Zufallspunktstereogramme
- Autostereogramme

Literatur zur Stereoskopie



Stereogramm.

Ars Edition 1994.



Sonka, M., Hlavac, V., and Boyle, R. (1999).

Image Processing, Analysis, and Machine Vision.

PWSPublishing.



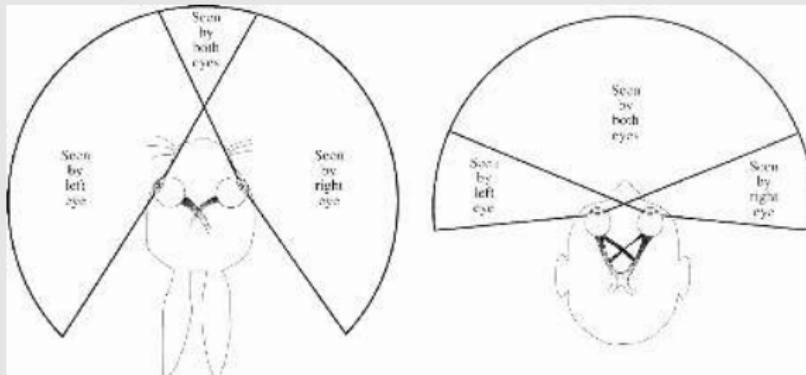
Marr, D. and Poggio, T. (1976).

Cooperative computation of stereo disparity.

Science, pages 283–287.

Stereoskopie: Binokulares Sehen

- ☞ Das Gehirn ermittelt aus zwei Bildern, die aus zwei leicht verschiedenen Positionen aufgenommen werden, einen räumlichen Eindruck.
- ☞ Für die Stereoskopie ist es wichtig, daß sich das Gesichtsfeld der beiden Augen überlappt.



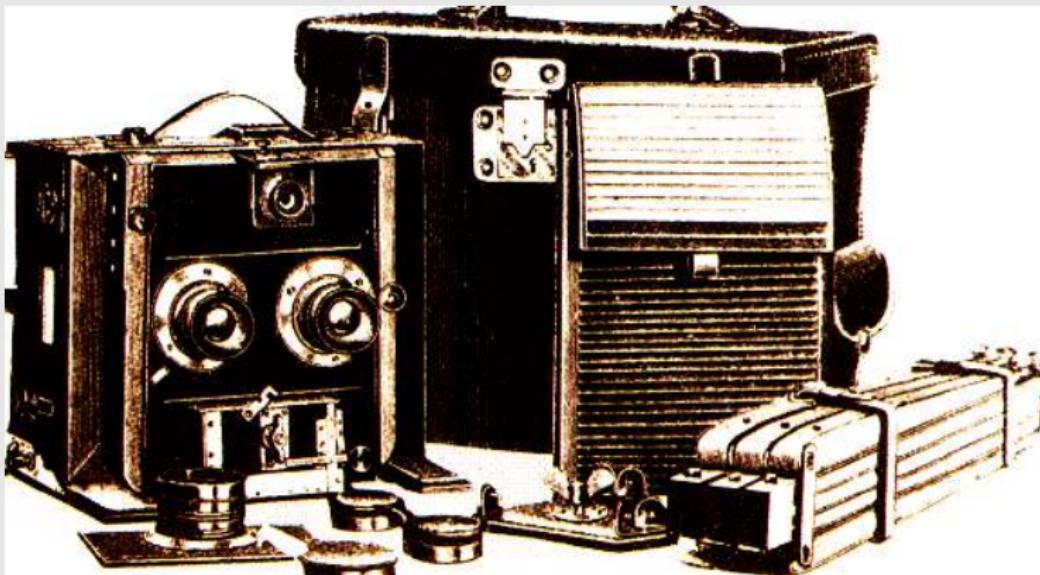
Anordnung der Augen: Kaninchen und Mensch.

Entdeckung der Stereoskopie

- ☞ 1838 Charles Wheatstone entwickelt das Stereoskop.
- ☞ 1839 Die Photographie und die Stereokamera werden erfunden.
- ☞ 1856 Stereoskope werden für den freien Markt produziert. Während der „Great exhibition“ werden in London mehrere tausend Stereoskope verkauft.
- ☞ ca. 1950: Science Fiction Filme werden bezugt als Stereofilme gedreht.
- ☞ 1959: Bela Julesz entwickelt die Zufallspunktstereogramme.
- ☞ 1979: Christopher Tyler erfindet das Autostereogramm.

Eine Stererokamera

Behälter für Photoplatten (1895)



Historische Stereoaufnahmen

Ein Stereobildpaar aus dem Glaspalast in London (um 1851–1853)

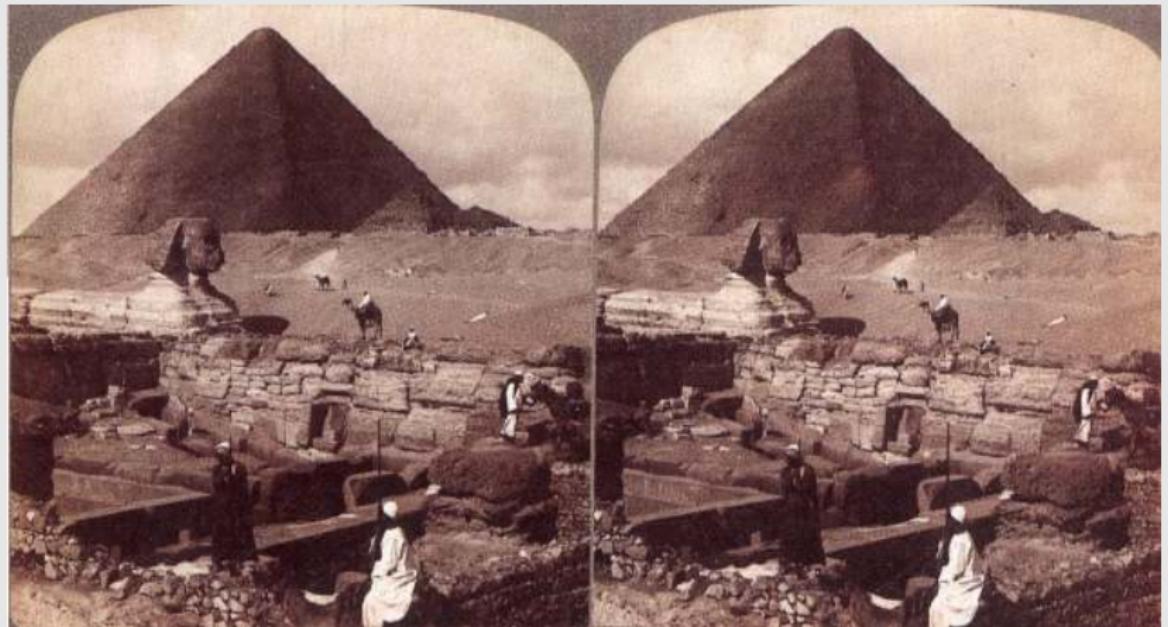


Historische Stereoaufnahmen



Elefanten an einer Wasserstelle auf Ceylon, spätes 19.
Jahrhundert

Historische Stereoaufnahmen



Cheops-Pyramide mit Sphinx, spätes 19. Jahrhundert

Beispiel: Glaspalast

Beispiel: Cheopspyramide und Sphinx

Beispiel: Elefantenherde

Beispiel: Schneemann

Stereoskopie-Anwendungen

- ▶ Kartographie
- ▶ Robotik
- ▶ Medizin
- ▶ Raumfahrt
- ▶ Computergrafik
- ▶ Kino ...
- ▶ ...

Präsentation von Stereobildpaaren

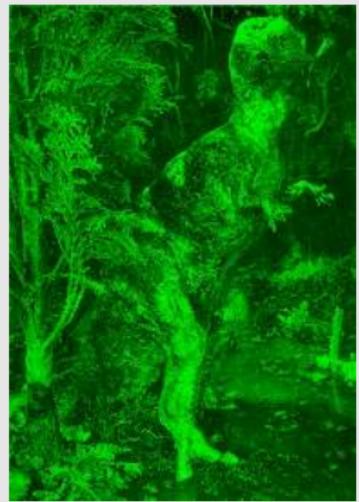
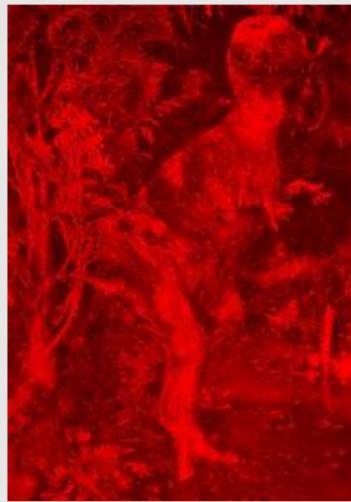
Anaglyphendarstellung: Nur für **einfarbige** Bilder geeignet; das Bild für das linke Auge und das rechte Auge werden übereinander in unterschiedlichen Farben präsentiert.

Anaglyphenbrille: Eine Brille mit passend gefärbten Gläsern selektiert die Bilder für das linke und rechte Auge.

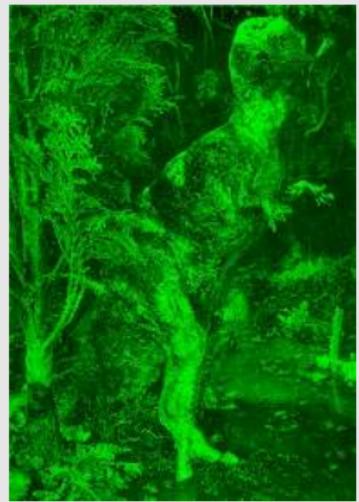
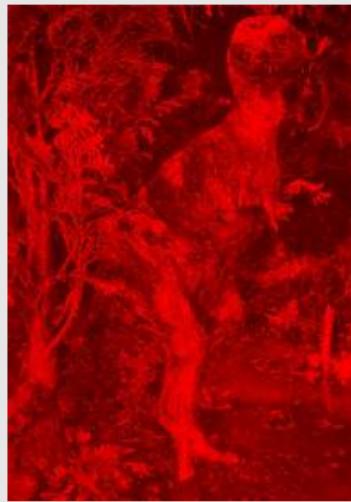
Tyrannosaurus Rex in Anaglyphendarstellung



Tyrannosaurus Rex in Anaglyphendarstellung



Tyrannosaurus Rex in Anaglyphendarstellung



T-Rex

Rhamphorhynchus gemmingi in Anaglyphendarstellung



Rhamphorhynchus gemmingi in Anaglyphendarstellung

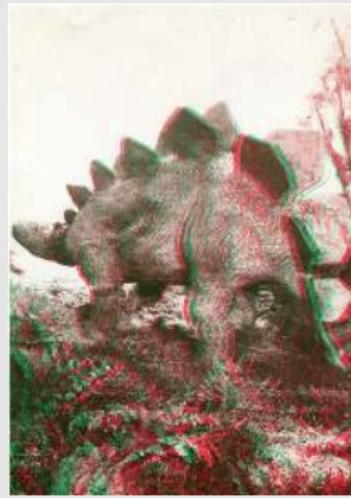


Rhamphorhynchus gemmingi in Anaglyphendarstellung

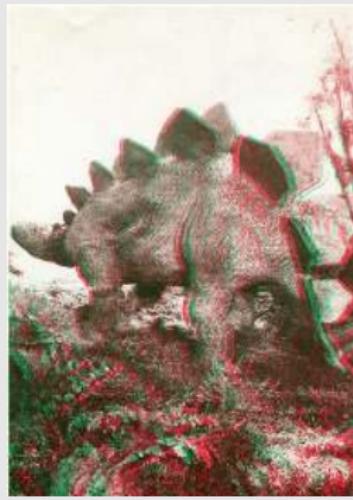


Flugsaurier

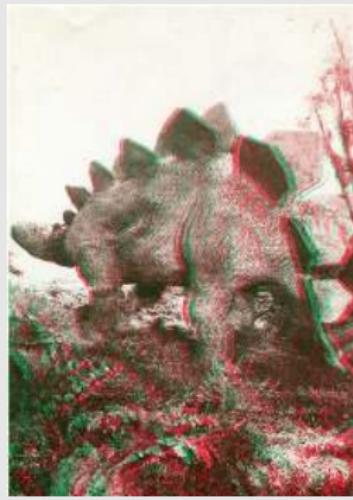
Stegosaurus stenops



Stegosaurus stenops



Stegosaurus stenops



Stegosaurus

Präsentation von Stereobildpaaren

Stereoskop : Jedes Auge sieht das passende Bild über ein separates Okular, auch für Farbbilder geeignet.

Polarisierender Projektor und Brille mit Polfiltern:

Wie Anaglyphendarstellung, aber für Farbe geeignet; jedoch nicht für Bücher.

In Brillen eingebaute Displays, die für jedes Auge einen separaten Bildschirm haben, auch für Animationen geeignet.

Arbeitstisch und Brillen mit Polarisationsfiltern

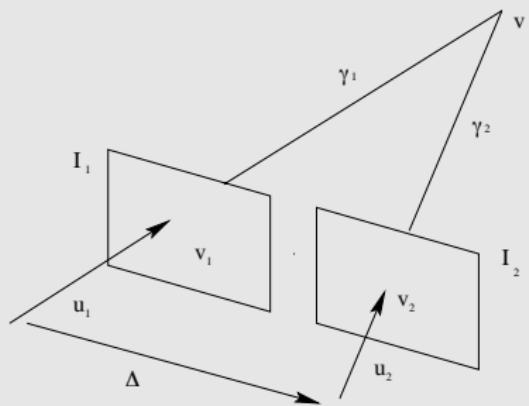


Mit viel Übung und Geduld

Stereogramme: Wenn die beiden Augen richtig ausgerichtet werden, entweder parallel oder über Kreuz (Schielen), sieht man zwischen den beiden Stereobildern ein räumliches Bild.

Zufallspunktstereogramme: Testbilder aus der kognitiven Psychologie, die zeigen, daß für das räumliche Sehen keine Bildstrukturen nötig sind.
⇒ Vorläufer des „Magischen Bildes“.

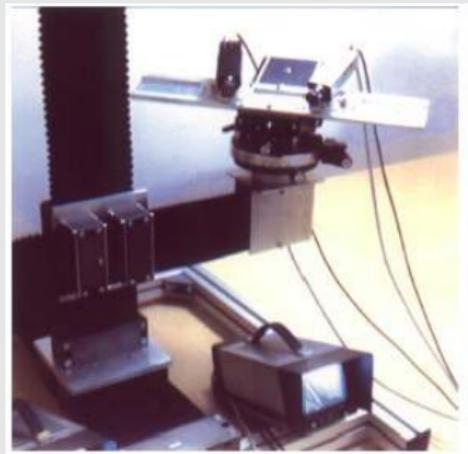
Stereoskopische Parallaxe



Bildebenen: I_1, I_2

Optische Zentren: L_1, L_2

Projektionsstrahlen: γ_1, γ_2



Objektpunkt: v

Einheitsvektoren

in Projektionsrichtungen: \vec{u}_1, \vec{u}_2

Basislinie: Δ

Stereo-Algorithmus:

- 1 Suche homologe Bildpunkte v'_1 und v'_2 .
 - 2 Bestimme v als Schnittpunkt der beiden Geraden γ_1 und γ_2 .
- ☞ In der Praxis werden sich γ_1 und γ_2 nicht schneiden (Rundungsfehler!), so daß es günstiger ist, den Punkt des kürzesten Abstands zu suchen.

Triangulation

$$\vec{v} = \frac{a_0 \cdot \vec{u}_1 + \Delta + b_0 \cdot \vec{u}_2}{2} + \vec{L}_1$$

a_0 und b_0 werden so bestimmt, daß

$$\ell(a, b) = \|a\vec{u}_1 - (\Delta + b\vec{u}_2)\|^2$$

minimal ist.

Wenn γ_1 und γ_2 sich schneiden, ist $\ell(a_0, b_0) = 0$
sonst:

$$a_0 = \frac{\vec{u}_1 \cdot \Delta - (\vec{u}_1 \cdot \vec{u}_2)(\vec{u}_2 \cdot \Delta)}{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}$$
$$b_0 = \frac{(\vec{u}_1 \cdot \vec{u}_2)(\vec{u}_1 \cdot \Delta) - \vec{u}_2 \cdot \Delta}{1 - (\vec{u}_1 \cdot \vec{u}_2)^2}$$

Vorgehensweise

Bestimmen von \vec{u}_1 und \vec{u}_2 :

Sei

\vec{v}_{pi} : Lage des Bildpunktes von P_i ,

\vec{L}_1 : optisches Zentrum, dann ist

$\vec{v}_{pi} - \vec{L}_1$ ein Vektor in Richtung des Projektionsstrahls
und

$$\vec{u}_1 = \frac{\vec{v}_{pi} - \vec{L}_1}{\|\vec{v}_{pi} - \vec{L}_1\|}$$

\vec{L}_2 und \vec{v} folgen aus der inversen perspektivischen
Transformation.

Diskretisierungsfehler

Bei der Stereorekonstruktion aus Digitalbildern entstehen typische Diskretisierungsfehler, wenn die Punkte nicht mit Subpixelgenauigkeit vermessen werden können:

- ▶ Der Rekonstruktionsfehler zeigt ein typisches Zickzakmuster.
- ▶ Der Rekonstruktionsfehler ist umso größer, je weiter das Objekt entfernt ist.

Rekonstruktionsfehler

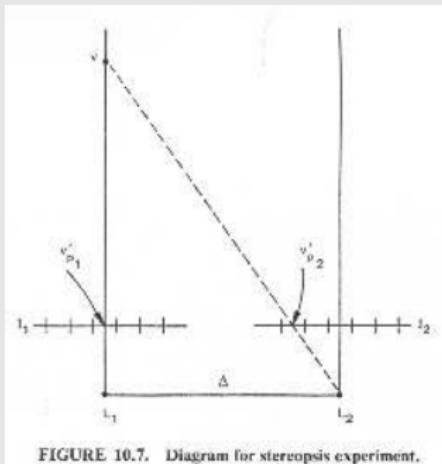
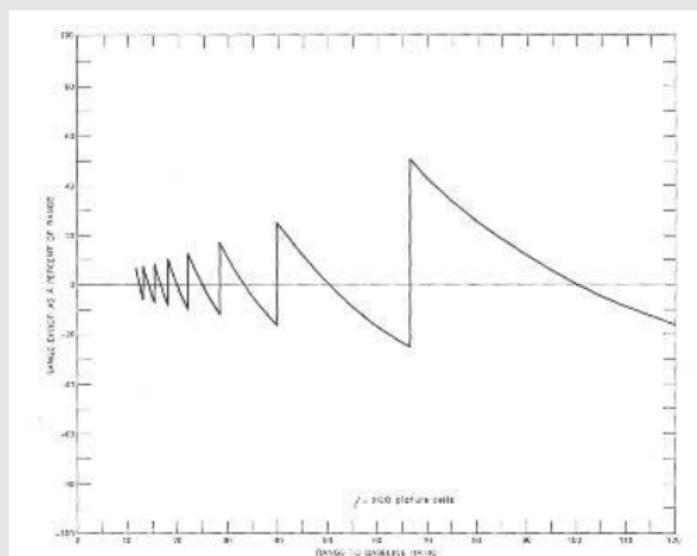


FIGURE 10.7. Diagram for stereopsis experiment.

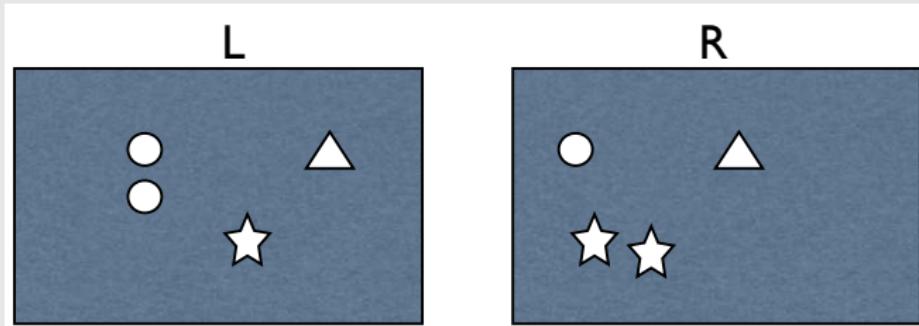
Die Bildunkte v'_{p1} und v'_{p2} werden nur begrenzter Genauigkeit bestimmt.

Einfluß der Auflösung

Tiefenfehler in Prozent der Entfernung

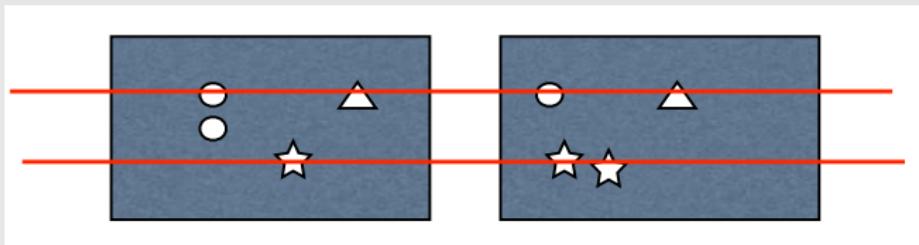


Das Korrespondenzproblem



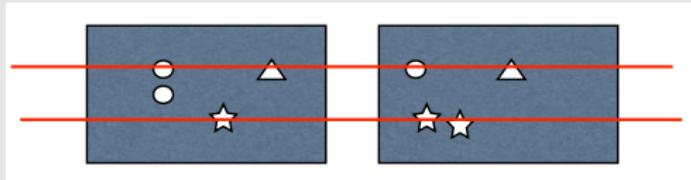
- ▶ Welche Paare von Punkten aus dem linken und rechten Bild sind Bilder desselben Punktes im Raum (**homologe** Punkte)?
- ▶ Problem: nicht eindeutig, solange keine einschränkenden Annahmen gemacht werden können.

Einschränkungen durch die Geometrie



- ▶ Korrespondierende Punkte liegen auf Geraden (epipolare Linien).
- ▶ Korrespondierende Punkte liegen im rechten Bild weiter links als im linken Bild.

Naiver Lösungsansatz



- ▶ Wähle Punkt im linken Bild.
- ▶ Suche im rechten Bild auf der epipolaren Linie nach der ähnlichsten Grauwertkonstellation.

Kameraanordnung

- ▶ Die Bildebenden der beiden Kameras sind koplanar und liegen in der xy-Ebene des Kamerakoordinatensystems.
- ▶ Beide optischen Zentren liegen auf der x-Achse.
- ▶ Dann sind epipolare Linien parallel zur x-Achse und homologe Punkte liegen in derselben Bildzeile.

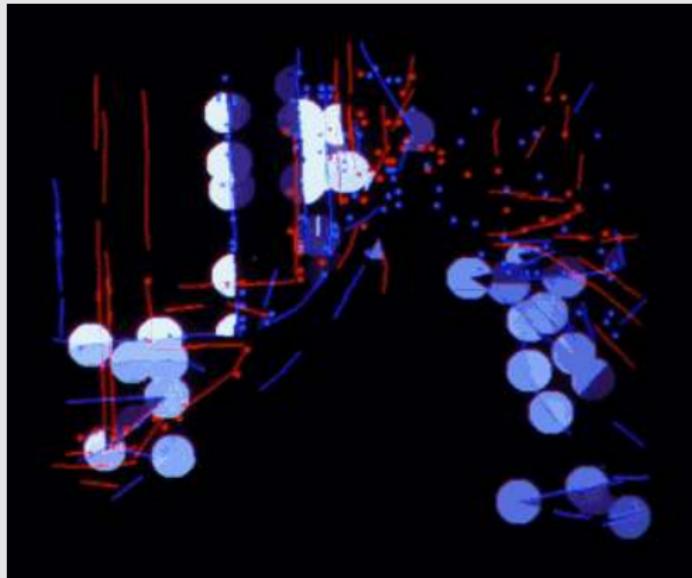
Stanford Autonomous Land Vehicle

Stereo-Aufnahme einer Straße mit detektierten Kanten
(Triendl-Operator)



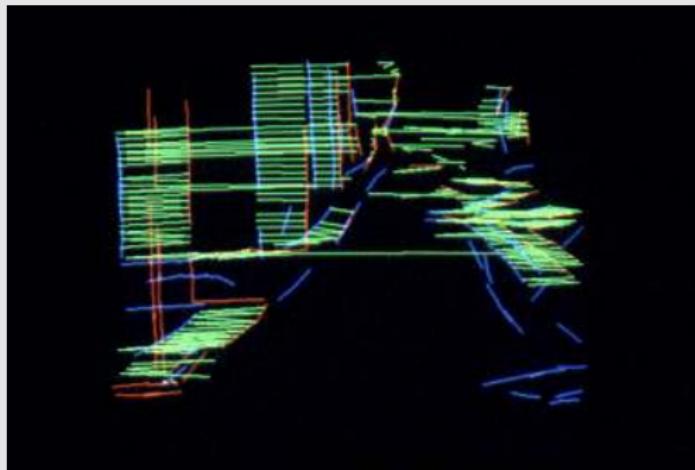
Stanford Autonomous Land Vehicle

Merkmale im linken und rechten Bild



Stanford Autonomous Land Vehicle

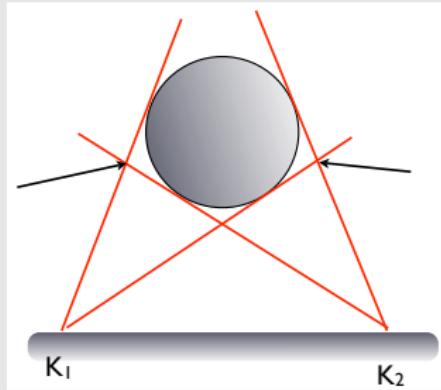
Zuordnung von homologen Punkten



Schwierigkeiten bei der Ähnlichkeitsmessung

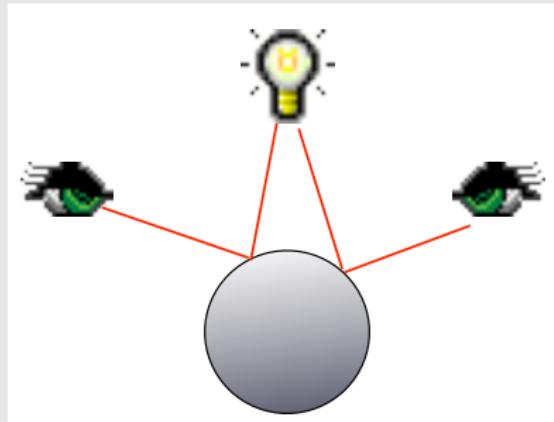
- ▶ Unterschiedlicher Blickwinkel:
Perspektive, Licht, Winkel,
- ▶ Unterschiedlicher Abstand,
- ▶ Unterschiedliches Gesichtsfeld,
- ▶ Kontrastumkehrung.

Probleme bei gewölbten Flächen



- ▶ Beide Kameras bilden eine Kreischeibe ab, aber die sichtbaren Konturen sind nicht homolog, da die Kernstrahlen an anderen Punkten die Kugeln tangential streifen.
- ▶ Tangentialkanten eignen sich daher nicht zur Stereorekonstruktion.

Glanzlichter



Glanzlichter

- ▶ Glanzlichter sind die Spiegelbilder von hellen Lichtquellen.
- ▶ Auf gekrümmten Flächen hängt ihre sichtbare Position vom Standort des Betrachters ab.
- ▶ Glanzlichter eignen sich daher nicht zur Stereorekonstruktion.

Spiegelungen



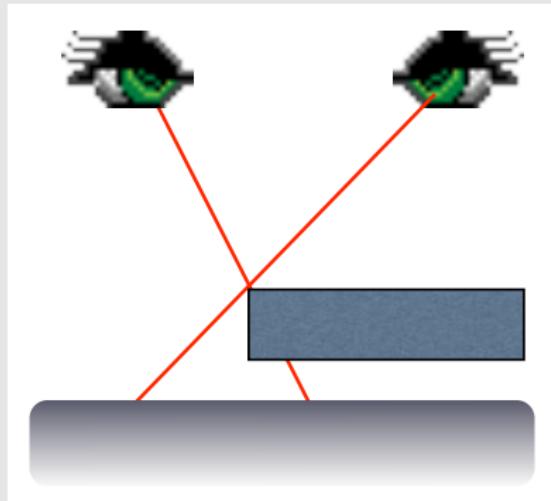
- ▶ Spiegelungen sind keine ortsfesten Merkmale.
- ▶ Ihre sichtbare Position hängt vom Standort des Betrachters ab.
- ▶ Spiegelungen eignen sich daher nicht zur Stereorekonstruktion.

Spiegelungen



- ▶ Gespiegelte Merkmale sind Merkmale der Umgebung und nicht der spiegelnden Fläche.
- ▶ Wenn wir gespiegelte Merkmale rekonstruieren, rekonstruieren wir die gespiegelte Umgebung und nicht die Oberfläche des Spiegels.
- ▶ Beim Lösen des Korrespondenzproblems muß daher geprüft werden, ob die Merkmale zur Objektoberfläche gehören oder ob es Spiegelungen sind.

Verdeckung



- ▶ Bei Verdeckung wandert die sichtbare Kante standortabhängig auf der Oberfläche des verdeckten Objektes.
- ▶ Verdeckungskanten eignen sich daher nicht zur Stereorekonstruktion.

Fazit

- ▶ Homologe Punkte sehen sich nicht notwendigerweise ähnlich.
- ▶ Ähnlich aussehende Punkte korrespondieren nicht notwendigerweise.
- ▶ Dennoch werden sehr häufig Ähnlichkeitsmaße zur Korrespondenzanalyse eingesetzt.
 - ▶ Grauwertvergleich (Korrelation)
 - ▶ Strukturvergleich (graph matching)

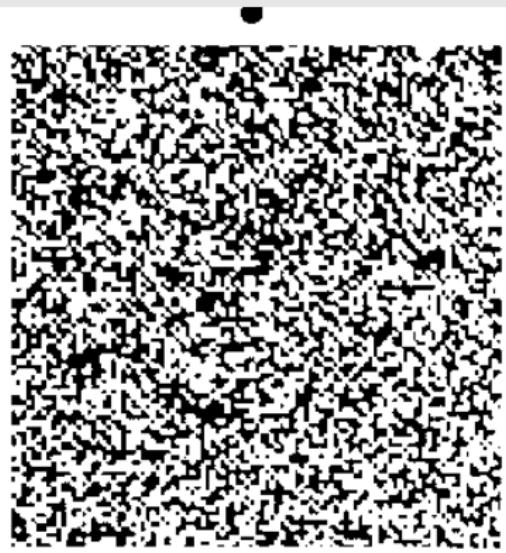
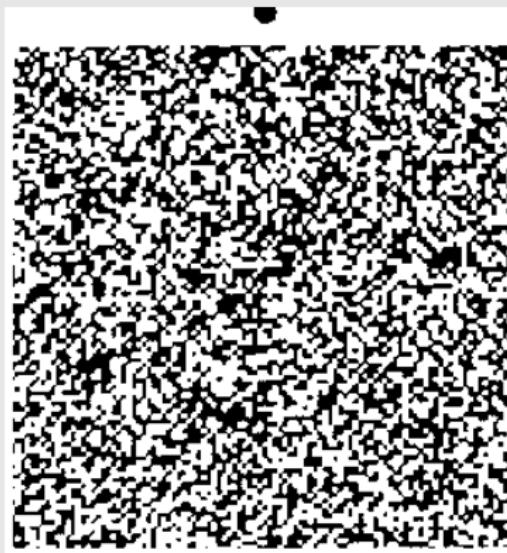
Andere Möglichkeit

Wenn die Aufnahmebedingungen frei gestaltet werden können, kann das Korrespondenzproblem durch technische Hilfsmittel gelöst werden:

- ▶ Markierungspunkte auf der Objektoberfläche
- ▶ Markieren mit Lichtpunkten usw.

Zufallspunktstereogramme von Bela Julesz (1959)

(Random-dot-stereogram)



Beispiel: Stereogramm 1

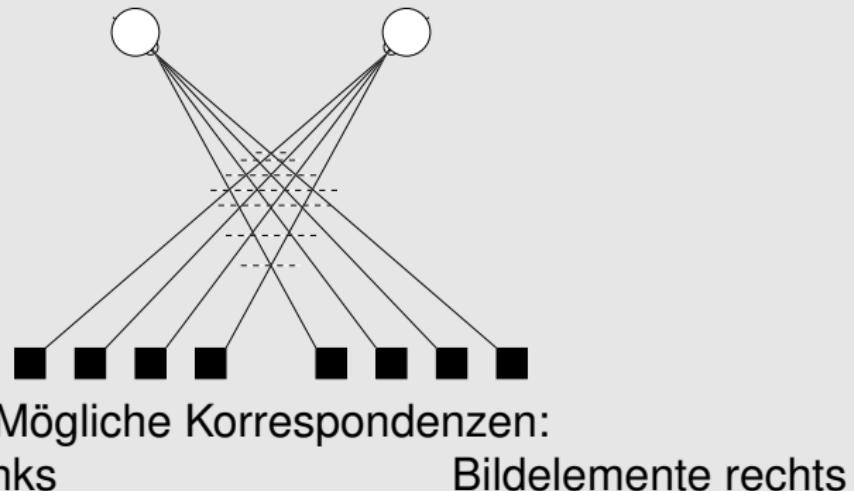
Stereogram 1

Zur Bedeutung der Zufallspunktstereogramme

- ▶ Zufallspunktstereogramme sind ein Beweis, daß der Mensch das Korrespondenzproblems allein auf der Basis von Texturen lösen kann.
- ▶ Es ist weder notwendig, die abgebildeten Objekte zu erkennen noch die Bilder zu segmentieren.
- ▶ Texturen allein sind für einen räumlichen Eindruck ausreichend.

Ansatz von Marr und Poggio 1976

Orientierung am visuellen System des Menschen:
[Marr and Poggio, 1976]



- ▶ Das visuelle System des Menschen wählt vermutlich die Lösung, bei der benachbarte Bildelemente eine ähnliche Tiefe haben.

Annahmen

Annahmen:

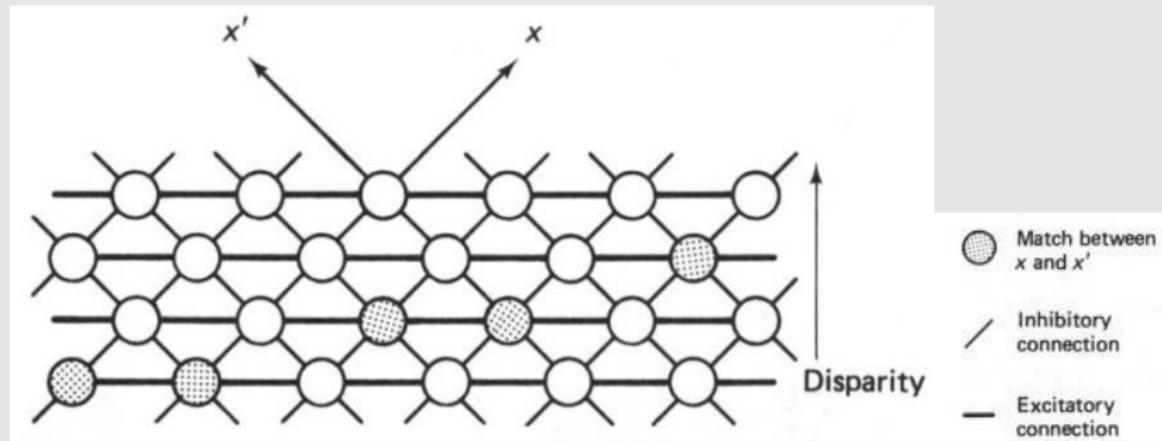
- ▶ Physikalische Objekte haben in der Regel glatte Oberflächen, bei denen die Tiefe von benachbarten Punkten ähnlich ist.
- ▶ Physikalische Objekte sind meistens nicht transparent.

Regeln:

Daraus leiten sich folgende Regeln ab:

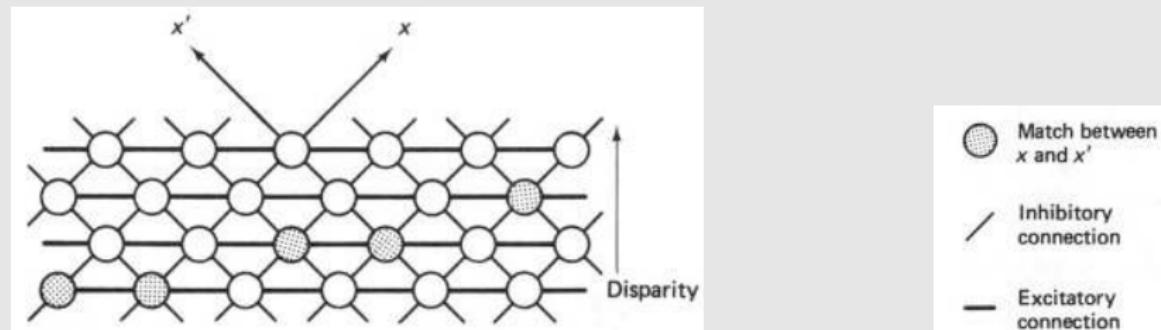
- ▶ Jeder Punkt im Bild kann nur einen Tiefenwert haben.
- ▶ Ein Punkt hat ziemlich sicher den Tiefenwert seines Nachbarn oder einen ähnlichen.

Implementation als Relaxationsverfahren



- ▶ Netzwerk von Zuständen (binär), die anzeigen, ob eine Zuordnung zwischen zwei Punkten stattgefunden hat.
- ▶ Zuordnungen gleicher Tiefe verstärken sich, konkurrierende Zuordnungen hemmen sich.

Implementation des Netzwerks



Implementation des Netzwerks als binäre Matrix

$C(x, y, d)$,

- ▶ wobei $C(x_1, y_1, d)$ eine Zuordnung des Bildpunkts (x_1, y_1) in Bild 1 zu einem Punkt (x_2, y_2) bedeutet,
- ▶ die den Tiefenwert d hat.

Iteration

- ▶ Initialisiere $C_0(x, y, d)$ mit „1“, wo eine Zuordnung zwischen Texeln möglich ist, sonst mit „0“.
- ▶ Iteriere die Matrix C , bis ein Konvergenzkriterium erfüllt ist:

$$C_{n+1}(x, y, d) = \left\{ \sum_{(x', y', d') \in S} C_n(x', y', d') - \sum_{(x', y', d') \in \Theta} C_n(x', y', d') + C_0(x', y', d') \right\}$$

S = support set: unterstützende Zuordnungen

Θ = inhibitory set: konkurrierende Zuordnungen

Normalisierung

Nach jeder Iteration normalisiere alle Elemente der Matrix $C(x, y, d)$ zu „1“ oder „0“ in Abhangigkeit von einer Schwellen T .

Resultate: Ein Bild mit zwei Tiefenebenen

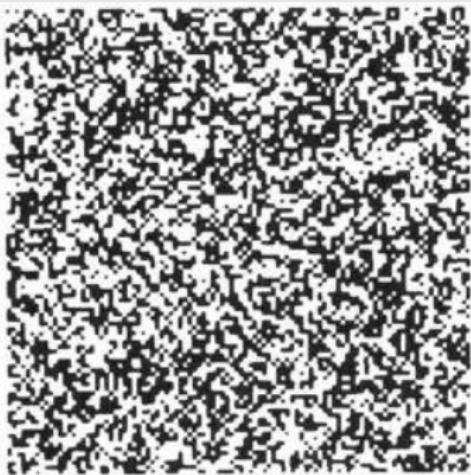
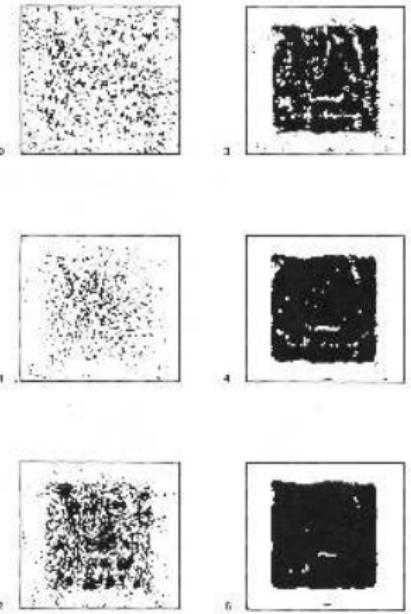


Fig. 3.23 A random-dot stereogram.

Resultate

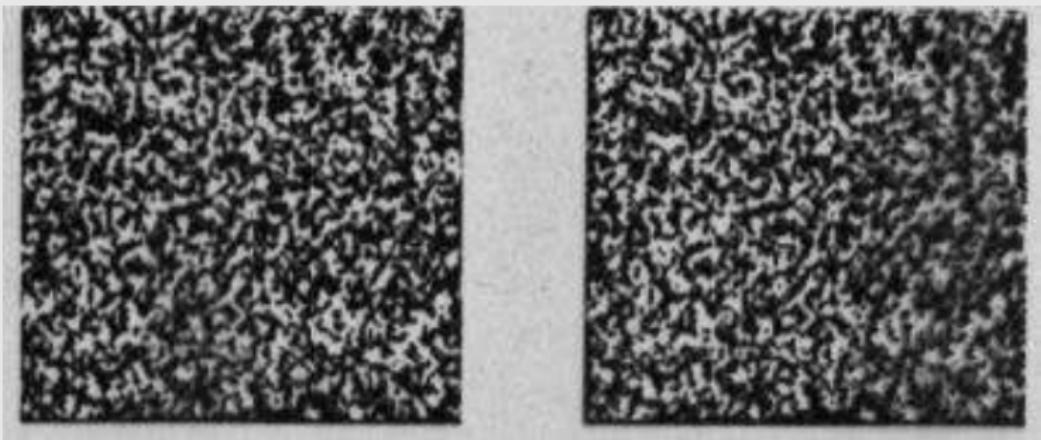


Sechs Iterationen des Stereo-Algorithmus

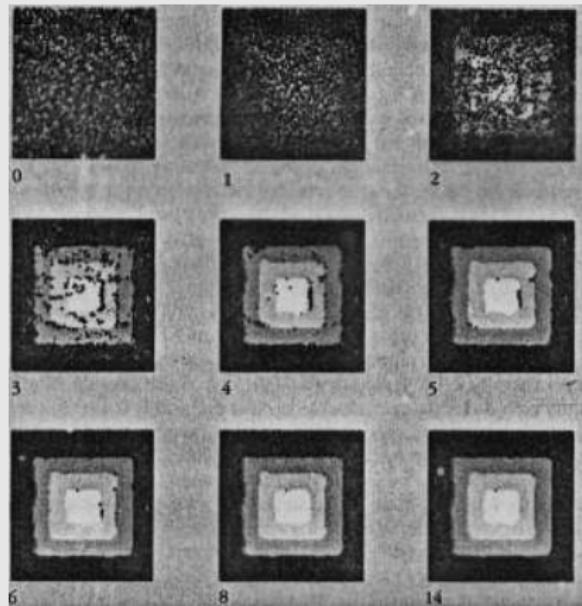
Für das Stereogramm von Folie 1359.

- ▶ Die Helligkeit ist proportional zur Disparität d und damit ein Maß für die Tiefe.
- ▶ Die Konvergenzgeschwindigkeit ist umso größer, je dichter die Texel liegen.

Beispiel 2: Bild mit vier Ebenen



Beispiel 2: Bild mit vier Ebenen



Das erste Stereogramm von Kompaneysky

1939:

Die Venus



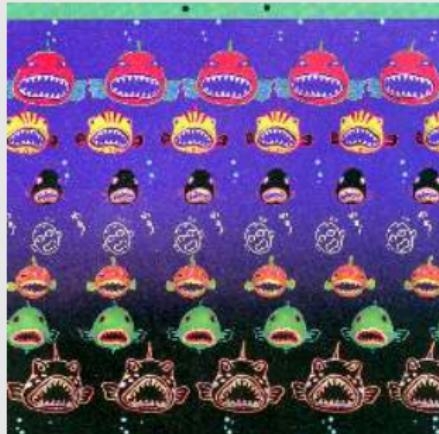
Abb. 1

UX

Kompaneysky 1939: Venus

Venus

Autostereogramme



Die Abenteuer des
Herrn Q., (Naoyuki und
Miyuki Kato)

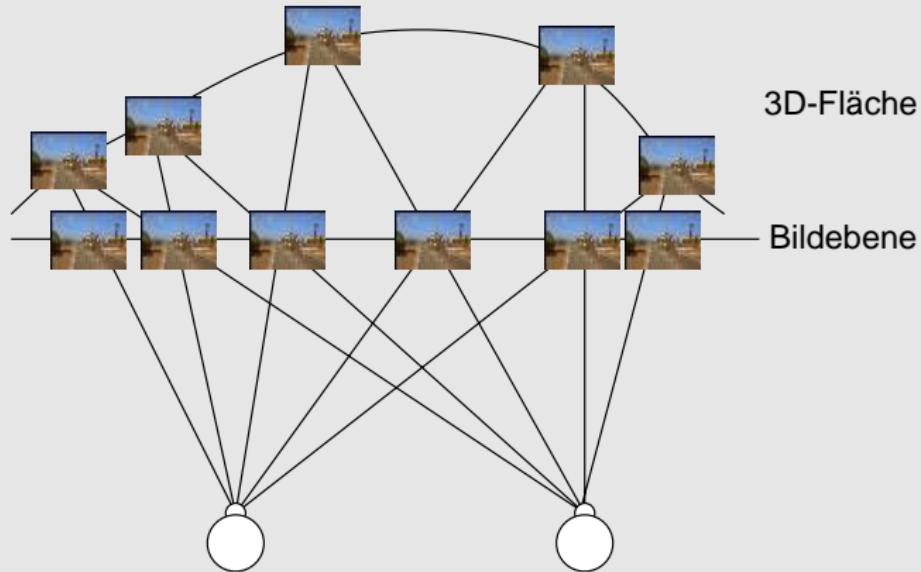
- ▶ Christopher Tyler (1979) hat die Zufallspunktstereogramme so weiter entwickelt, daß ein einzelnes Bild beide Teilbilder enthält (Autostereogramme, magische Bilder).
- ▶ Diese Autostereogramme können ohne Stereoskop betrachtet werden.

Die Abenteuer des Herrn Q., (Naoyuki und Miyuki Kato)

Konstruktion eines „Magischen Bildes“

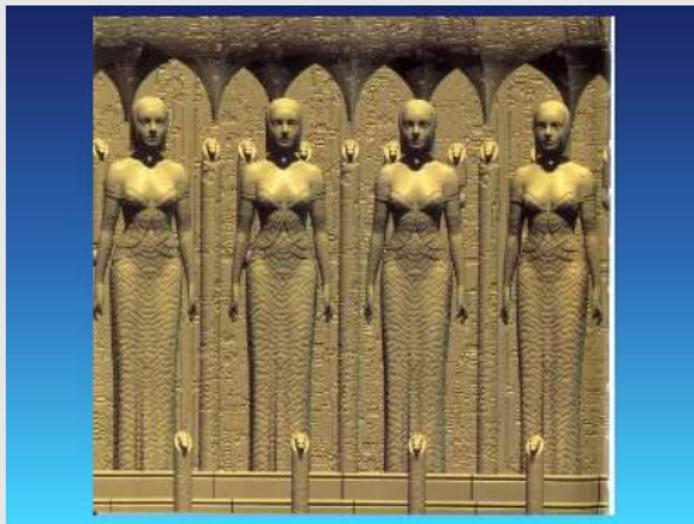
- ☞ Stark vereinfachte Stereogeometrie:
Die Augen sind parallel gestellt, als solle ein unendlich entferntes Objekt betrachtet werden.
- ☞ Die epipolaren Linien sind parallel zur x-Achse.
- ☞ Die Bildelemente (Texel) wiederholen sich periodisch entlang der epipolaren Linien. Dabei werden sie leicht verschoben, so daß die Entfernungswahrnehmung variiert.
- ☞ Form und Farbe der Texel können für jede Zeile willkürlich gewählt werden.

Die Tapetentechnik



Konstruktion eines Autostereogramms mit der
Tapetentechnik

Tapetentechnik: Die drei Grazien



Schweigende Schönheiten, Naoyuki Kato

Schweigende Schönheiten, Naoyuki Kato

Farbkorrespondenz



Farbkorrespondenz

Künstler-Stereogramme



Seisaku Kano

Seisaku Kano

The Space: Seisaku Kano

Der Stuhl: Salvador Dali 1976

Salvador Dali 1972-73, Gala von hinten

unvollendet

Künstler-Stereogramme: Salvador Dalí

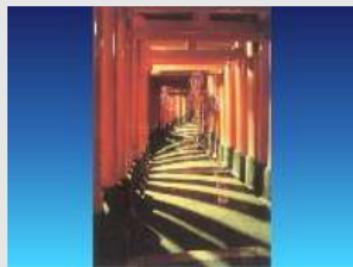


Der Stuhl 1976



Gala von hinten 1972-73, unvollendet

Lichtstiftbilder: McNelly

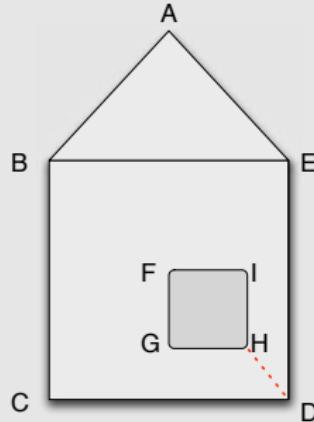


Lichtstiftbilder McNelly

Lichtstiftbilder McNelly

Teil XXI

Bildverstehen



54

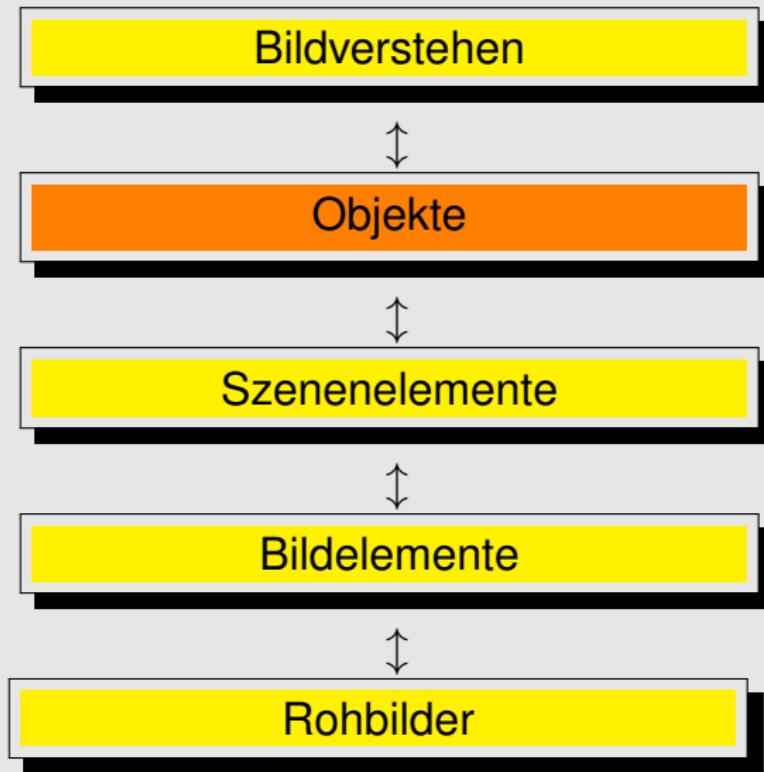
Objekterkennung

- Ansätze zur Objekterkennung
- Gemeinsame Eigenschaften
- Häufigkeitsanalysen
- Schablonentest
- Relationalstrukturen

55

Ereigniserkennung

Objekterkennung



Objekterkennung

Objekterkennung ist eine typische Aufgabe der Bildanalyse.

Objekterkennung umfaßt:

Objektidentifizierung: Physikalische Identität feststellen.

Objektklassifikation: Das Objekt einer Klasse von endlich vielen vorbestimmten Klassen zuordnen.

Objektkategorisierung: Das Objekt einer Kategorie von biologisch visuellen Kategorien zuordnen.

Beispiel: The Chair Room

(H. Bülfhoff, MPI Tübingen)



Ansätze zur Objekterkennung

Gemeinsame Eigenschaften: Merkmalsvergleich.

Häufungsgebiete: Abstandsmessung, Ähnlichkeit (siehe Bildverarbeitung 1 und 2).

Prototypen: Schablonentest.

Relationalstrukturen: Graphzuordnung.

Lineare Maschinen: Perceptrons, Neuronale Netze (siehe Bildverarbeitung 1).

Modellbasiert: 2-D, 3-D Modellvergleich.

Klassifikation über gemeinsame Merkmale:

Gegeben:

Drei Klassen von Objekten: *Pinguin*, *Hund*, *Goldfisch*.

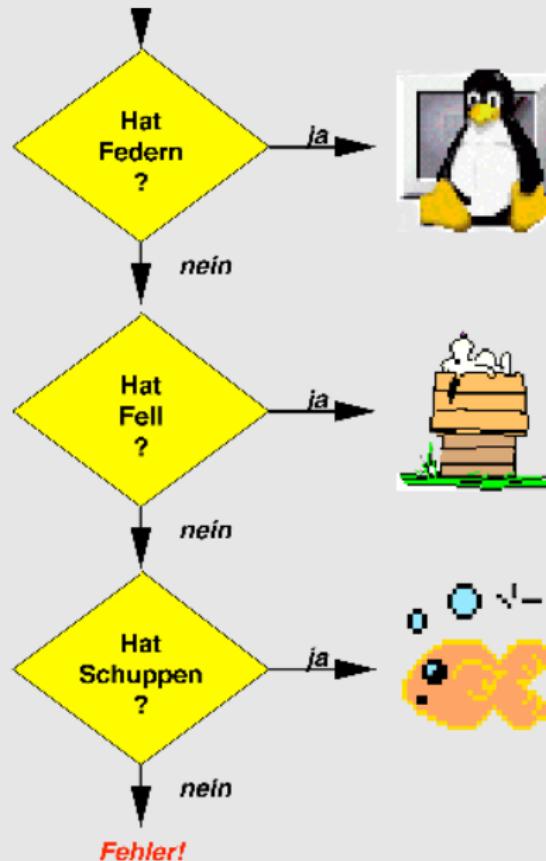


Gesucht:

Merkmale, die zur Unterscheidung dienen.

Ein Algorithmus, der Objekte dieser drei Klassen unterscheiden kann.

Klassifikation über gemeinsame Merkmale:



Ein einfacher
Klassifikationsalgorithmus:
Eine Folge von **Tests** auf
gemeinsame Merkmale.

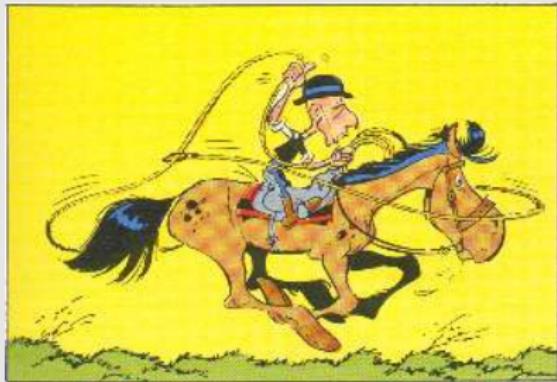
Grenzen des Verfahrens

Solche Entscheidungsverfahren sind nur begrenzt anwendbar.

Die wichtigsten Probleme sind:

- ☞ Häufig sind die Merkmale nicht binär (oder diskret).
- ☞ Nur schwer auf neue Klassen erweiterbar.
- ☞ Großer Rechenaufwand, wenn viele binäre Merkmale geprüft werden müssen.

Gemeinsame Merkmale: Häufigkeitsanalysen



Gemeinsame Merkmale: Häufigkeitsanalysen

Problem (Problem):

Wie können wir erkennen, welchen Sport eine Person ausübt, beispielsweise Rugby-Spieler von Jockeys unterscheiden?

Wir beobachten:

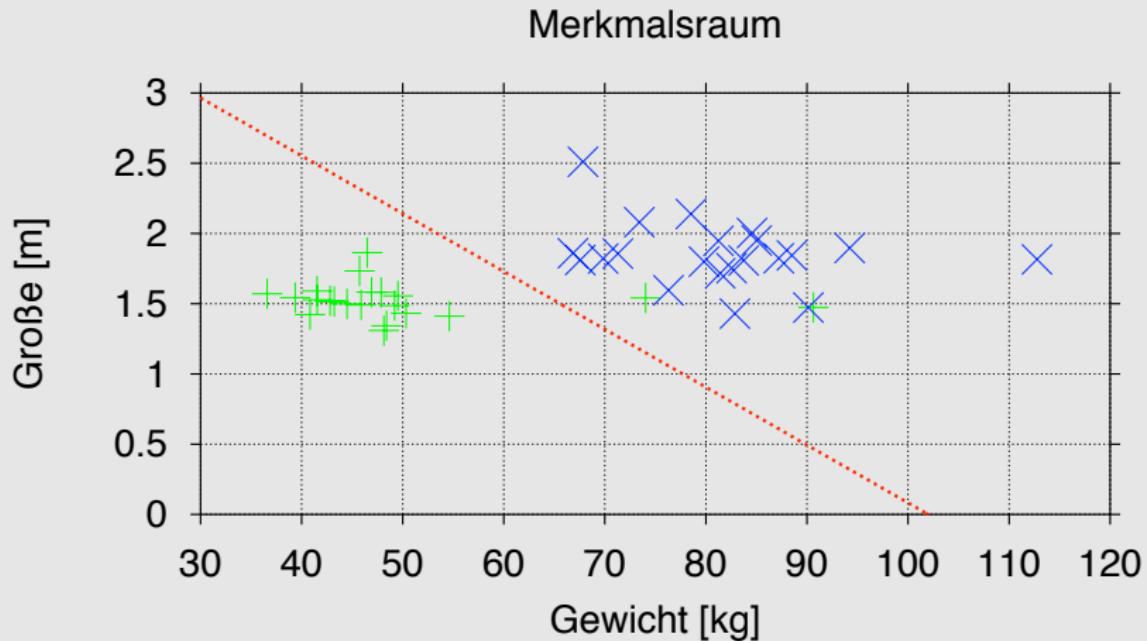
Jockeys sind meistens klein und leicht.

Rugbyspieler sind meistens größer und schwerer.

☞ *Aber was heißt hier klein oder groß, leicht oder schwer?*

Klassifikation über Häufigkeitsanalysen:

Verteilung von zwei Stichproben im *Merkmalsraum*.



Jockeys	+	trenn(x)
Rugby-Spieler	X		

Prinzipielle Probleme

Dieses Beispiel zeigt einige wichtige Probleme:

Dimension des Merkmalsraums: Wieviele Merkmale sind für eine zuverlässige Klassifikation nötig?

Art der Merkmale: Welche Merkmale sollen gewählt werden?

Fehlerminimierung: Wie sollen die Entscheidungsgrenzen gezogen werden, um die Fehlerrate zu minimieren?

Lernstichprobe: Wie groß soll die Lernstichprobe gewählt werden?

Schablonentest durch Grauwertkorrelation



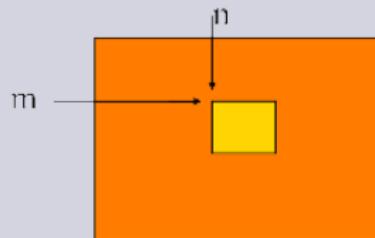
Originalbild

Erkennen von
Objekten durch
Vergleich mit einer
Schablone (engl.
template).



Vorgehensweise

Gegeben:



Gegeben sei als **Schablone t** ein Bild des gesuchten Objekts,

- ▶ im richtigen Maßstab und
- ▶ in der richtigen Orientierung.

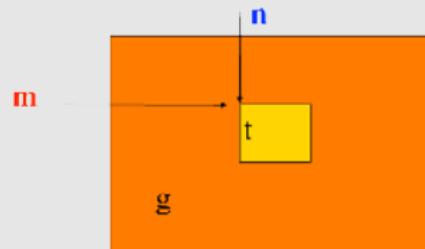
Vorgehen:

Die Schablone wird in allen Positionen m, n , an denen sie vollständig ins Bild paßt, mit dem darunterliegenden Bildausschnitt verglichen und ein Anpassungsfehler bestimmt.

Ansatz:

Sei

- ▶ $g(i, j)$ die Bildfunktion,
- ▶ $t(i, j)$ die Schablone,
- ▶ D der Definitionsbereich
der Schablone:



Das Fehlermaß E

$$E(\textcolor{red}{m}, \textcolor{blue}{n}) = \sqrt{\sum_i \sum_j [g(i, j) - t(i - \textcolor{red}{m}, j - \textcolor{blue}{n})]^2}$$

mit $(i - \textcolor{red}{m}, j - \textcolor{blue}{n}) \in D$

$$E(\textcolor{red}{m}, \textcolor{blue}{n}) = \sqrt{\sum_i \sum_j [g(i, j) - t(i - \textcolor{red}{m}, j - \textcolor{blue}{n})]^2}$$

mit $(i - \textcolor{red}{m}, j - \textcolor{blue}{n}) \in D$

$$= \sqrt{\sum_i \sum_j [g(i, j)^2 - 2g(i, j)t(i - \textcolor{red}{m}, j - \textcolor{blue}{n}) + t^2(i - \textcolor{red}{m}, j - \textcolor{blue}{n})]}$$

$$\begin{aligned} E^2(\textcolor{red}{m}, \textcolor{blue}{n}) &= \sum_i \sum_j g(i, j)^2 \\ &\quad - \sum_i \sum_j 2g(i, j)t(i - \textcolor{red}{m}, j - \textcolor{blue}{n}) \\ &\quad + \sum_i \sum_j t^2(i - \textcolor{red}{m}, j - \textcolor{blue}{n}) \end{aligned}$$

$$\begin{aligned}
 E^2(m, n) &= \underbrace{\sum_i \sum_j g(i, j)^2}_{\text{Autokorrelation des Bildausschnitts}} \\
 &- \underbrace{\sum_i \sum_j 2g(i, j)t(i - m, j - n)}_{\text{Korrelation zwischen Bild und Schablone}} \\
 &+ \underbrace{\sum_i \sum_j t^2(i - m, j - n)}_{\text{Autokorrelation der Schablone, konstant}}
 \end{aligned}$$

$$R_{gt}(m, n) = \sum_i \sum_j g(i, j)t(i - m, j - n) \quad \text{zu maximieren}$$

Normalisierung

Korrektur für variierende Bildenergie

$$R_{gt}(\textcolor{red}{m}, \textcolor{blue}{n}) = \sum_i \sum_j g(i, j) t(i - \textcolor{red}{m}, j - \textcolor{blue}{n}) \quad \text{zu maximieren}$$

$$N_{gt}(\textcolor{red}{m}, \textcolor{blue}{n}) = \frac{R_{gt}(\textcolor{red}{m}, \textcolor{blue}{n})}{\sqrt{g(i, j)^2}}$$

$$N_{gt}(\textcolor{red}{m}, \textcolor{blue}{n}) \leq \underbrace{\sqrt{t^2(i - \textcolor{red}{m}, j - \textcolor{blue}{n})}}_{= R_{tt}(\textcolor{red}{m}, \textcolor{blue}{n})}$$

Cauchy-Schwarz'sche Ungleichung

$$\frac{N_{gt}(\textcolor{red}{m}, \textcolor{blue}{n})}{R_{tt}(\textcolor{red}{m}, \textcolor{blue}{n})} \leq 1 \quad \text{Normierte Kreuzkorrelation}$$

Normierte Kreuzkorrelation

Definition (Normierte Kreuzkorrelation)

Das Verhältnis $\frac{N_{gt}(m,n)}{R_{tt}(m,n)}$ heißt normierte Kreuzkorrelation zwischen Bild g und Schablone t .

Satz

Die normierte Kreuzkorrelation zwischen Bild und Maske ist genau dann gleich 1, wenn Bild und Maske identisch sind, sonst niedriger.

Lösungsansatz

Vorgehen:

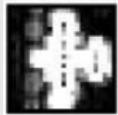
Für ein gegebenes Bild und eine gegebene Schablone

- ▶ berechne für jede Stelle im Bild die normierte Kreuzkorrelation.
- ▶ Lokale Maxima der normierte Kreuzkorrelation, deren Höhe dicht bei „1“ liegt, deuten auf erkannte Objekte hin.

Probleme

- ▶ Nebenmaxima der Korrelation
- ▶ Nur für echte Translation der Schablone geeignet, keine Drehung oder perspektivische Verzerrung zulässig.
- ▶ Aufwand proportional zu $N \cdot M \cdot I \cdot J$
- ▶ Rechenzeit sparen durch Faltung im Frequenzraum (Faltungstheorem)

Resultat für das Flughafenbild



Objektverfolgung durch Grauwertkorrelation

Problem

Die Aufgabe: Ein Objekt soll in einer Bildfolge (Film) von Bild zu Bild verfolgt werden.

Lösung

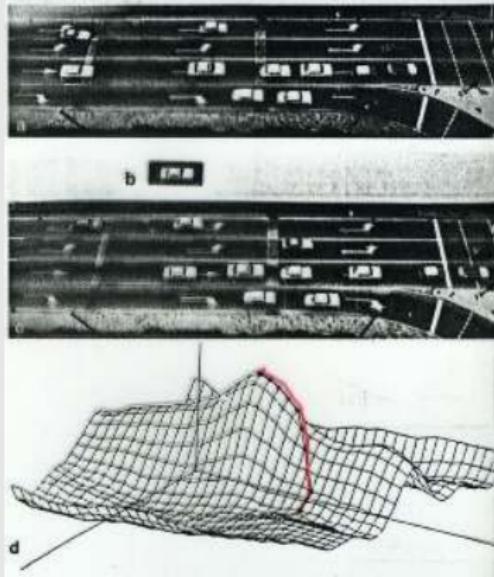
Vorgehensweise:

- ▶ Wähle Ausschnitt um das zu verfolgende Objekt als Schablone.
- ▶ Suche Maximum der Kreuzkorrelation zwischen Schablone und Folgebildern.

Probleme

- Rechenzeit:
- ▶ kleine Maskengröße: viele lokale Maxima
 - ▶ große Maskengröße: Rechenaufwand hoch

- Suchstrategie:
- ▶ erschöpfende Suche
 - ▶ größere Schrittweite (abhängig von der Form der Autokorrelation der Schablone)
 - ▶ Mehrstufige Suche
 - ▶ Gradientensuche



- a-c) Drei Ausschnitte einer Straßenszene, zeitlicher Abstand je eine Sekunde.
- b) Ausschnitt als Schablone
- d) Verlauf der normierten Kreuzkorrelation und Gradientenaufstieg

FIM-Tracker

Anwendungsbeispiel: FIM-Tracker

Beispiel

FIM-Tracker Der FIM-Tracker war eines der ersten Systeme zur Echtzeitverfolgung von Objekten.

Probleme bei Realwelt-Vorlagen:

- ▶ Maßstabsänderung
- ▶ Verdeckung
- ▶ Drehung
- ▶ Echtzeitverarbeitung

Anwendungsbeispiel: FIM-Tracker

Beispiel

FIM-Tracker Der FIM-Tracker war eines der ersten Systeme zur Echtzeitverfolgung von Objekten.

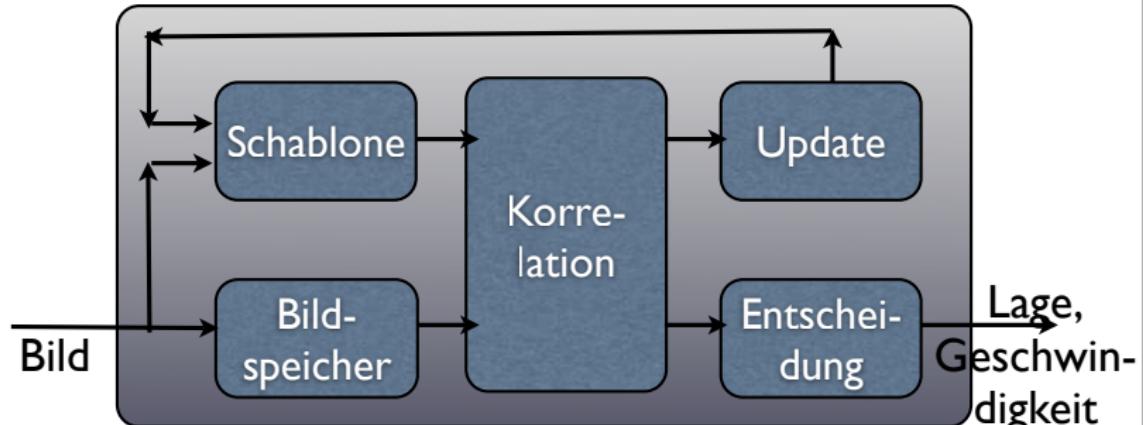
Probleme bei Realwelt-Vorlagen:

- ▶ Maßstabsänderung
- ▶ Verdeckung
- ▶ Drehung
- ▶ Echtzeitverarbeitung

Lösung

Laufende Anpassung der Schablone

Systemaufbau



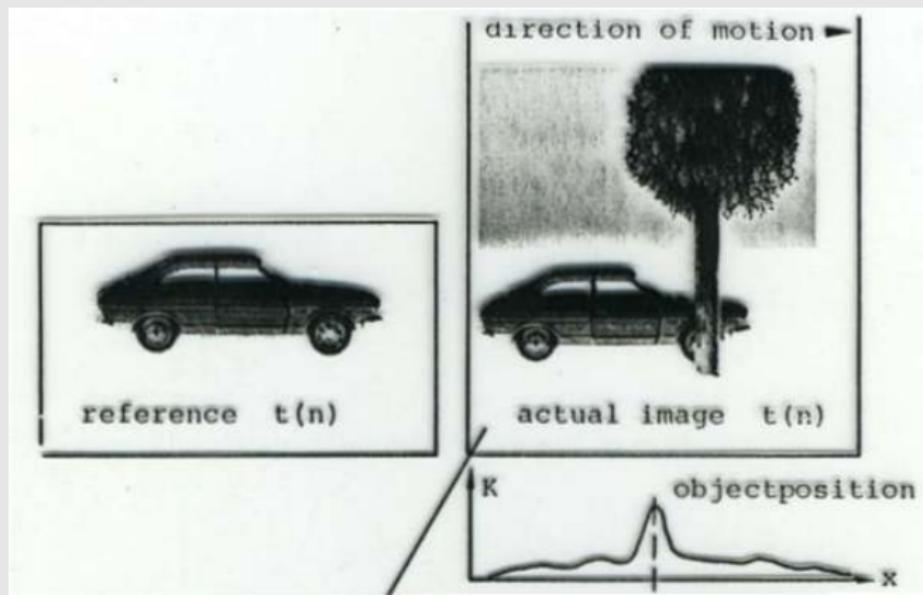
Aktualisierung der Schablone

Die Schablone wird in jedem Schritt ausgetauscht und durch das aktuelle Bild des gefundenen Objektes ersetzt. Damit paßt sich die Schablone laufend

- ▶ der veränderten Perspektive,
- ▶ der veränderten Beleuchtung und
- ▶ der veränderten Entfernung an.

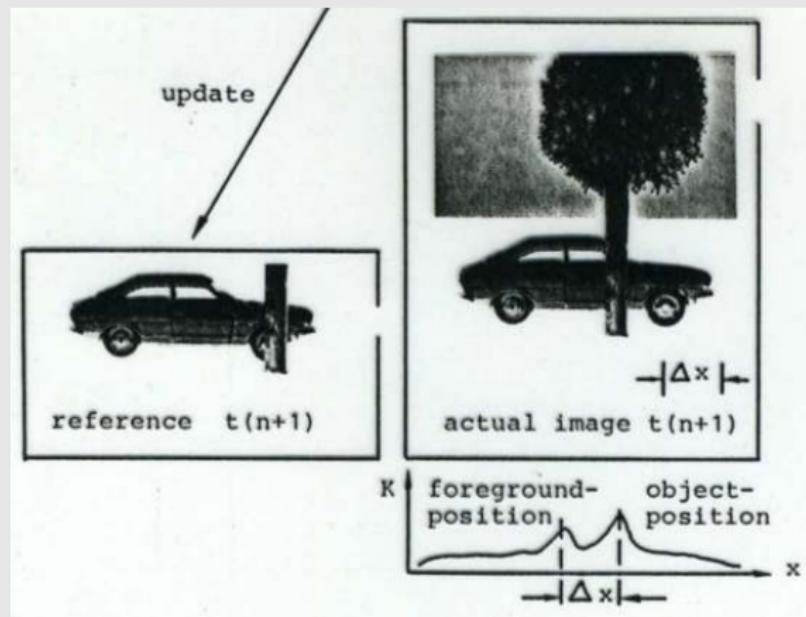
Aktualisierung der Schablone:

Probleme bei Verdeckung



Aktualisierung der Schablone:

Korrelationsmaxima

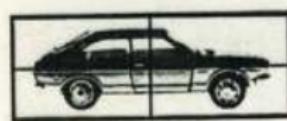


Vorsicht! Das verdeckende Objekt wird mit in die Schablone eingebaut. Es ergeben sich zwei Korrelationsmaxima.

Maßstabsanpassung

Abhilfe gegen Maßstabsänderungen:

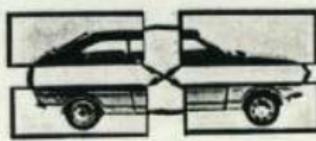
Ein Teil der Probleme kann gelöst werden, wenn Teile des Objektes separat mit Telschablonen gesucht werden.



$t(n)$



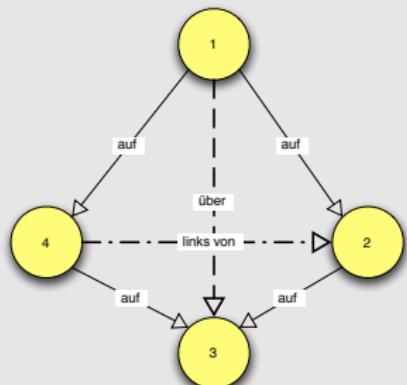
$t(n+1)$



$t(n+1)$

Fig. 4. Determination of the scaling factor

Relationalstrukturen



54

Objekterkennung

- Ansätze zur Objekterkennung
- Gemeinsame Eigenschaften
- Häufigkeitsanalysen
- Schablonentest
- Relationalstrukturen

55

Ereigniserkennung

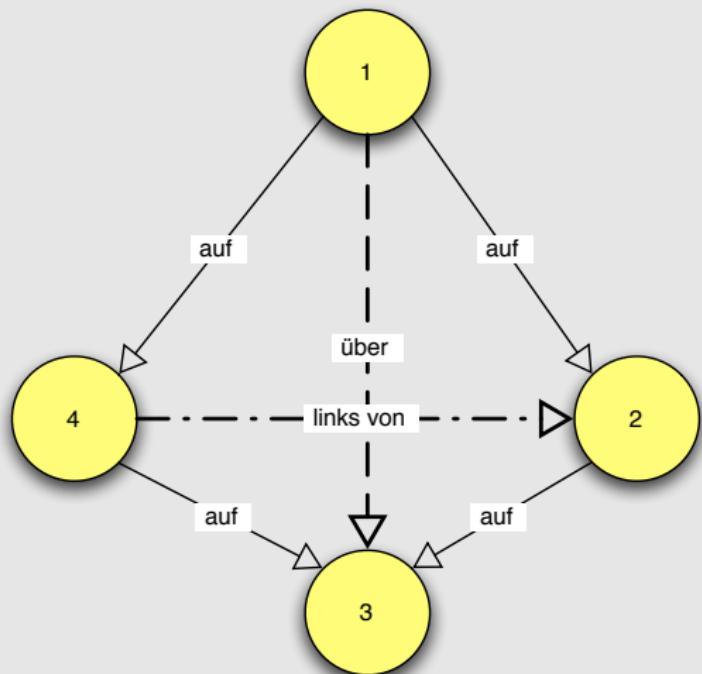
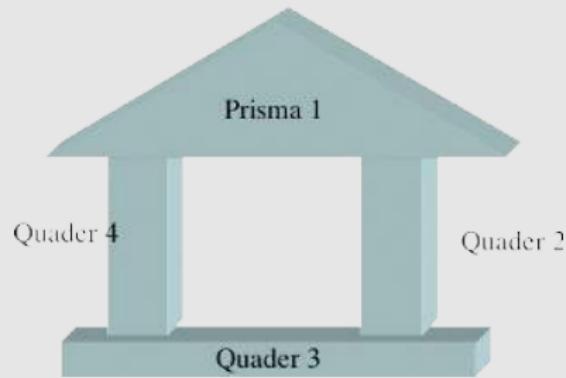
Objekterkennung durch symbolischen Bildvergleich

Zielsetzung: Löse das Korrespondenzproblem oder erkenne Objekte durch Vergleich von symbolischen Beschreibungen der Bilder.

Vorgehensweise:

- ▶ Führe für jedes Bild die Vorverarbeitungs- und Segmentationsprozesse durch.
- ▶ Setze für jedes Bild die Segmentationsergebnisse in eine symbolische Beschreibung um.
- ▶ Vergleiche die symbolischen Beschreibungen.

Beispiel



relationale Beschreibung

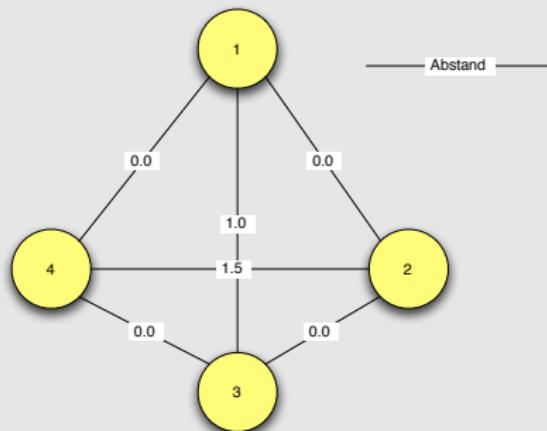
Anforderung an eine relationale Beschreibung

Ein Graph, der die Objekte eines Bildes (einer Szene) und ihre Anordnung beschreiben soll, muß folgende Eigenschaften haben :

- ▶ Die Knoten des Graphen müssen benannt werden können.
- ▶ Die Kanten des Graphen müssen benannt werden können.
- ▶ Die Knoten und Kanten müssen Attribute beinhalten können.
- ▶ Die Kanten des Graphen müssen gerichtet sein.

Beispiel: Abstandsrelation

Eine alternative relationale Repräsentation des Blockbildes: Abstandsrelation.



Definition: Bildbeschreibungs-Graph

Definition (Bildbeschreibungs-Graph)

Ein Bildbeschreibungs-Graph ist ein 4-(6-)Tupel
 $(NEVW\alpha\beta)$ mit:

N : eine endliche, nichtleere Menge von **Knoten**

E : eine endliche, nichtleere Menge von gerichteten Kanten ($\subseteq N \times N$)

V : die Menge der Knotenmarkierungen

W : die Menge der Kantenmarkierungen

$\alpha : N \rightarrow 2^A$: eine Funktion, die einem Knoten eine Menge von Attributen zuordnet.

$\beta : E \rightarrow 2^A$: eine Funktion, die einer Kante eine Menge von Attributen zuordnet.

Relationengebilde

Die Knoten eines bildbeschreibenden Graphen repräsentieren **Objekte** oder Objektteile in eine Szene.

Die Kanten des Graphen repräsentieren **Beziehungen** (Relationen) zwischen den Objekten.

- ☞ Daher kann ein Graph auch als spezielles Relationengebilde interpretiert werden.

Graph vs. Relationalstruktur

- ▶ Im Gegensatz zur allgemeinen Relationalstruktur sind beim Graphen nur binäre Relationen zugelassen.
- ▶ Andererseits kann jede n-stellige Relation in n-1 binäre Relationen überführt werden.
- ▶ Daraus folgt:
 - ☞ Graphen und Relationengebilde sind gleich mächtig und ineinander umwandelbar.

Relationengebilde

Definition (Relationengebilde)

Ein Relationengebilde (synonym Relativ) ist ein Paar

$$RS = (C, R).$$

C: ist die Trägermenge (Carrier), d.h. die Menge der Symbole und Werte, die innerhalb der Relationstupel des Gebildes auftreten können.

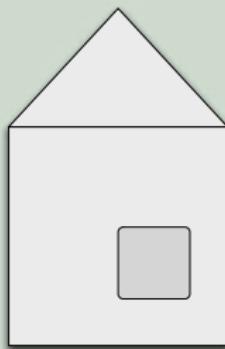
R: $R = (R_1, \dots, R_n)$ ist ein n-Tupel von Relationen über *C*. Jede Relation R_i ist eine Menge von k_i -Tupeln über der Trägermenge *C*. Die Relation R_i hat dann also k_i Stellen.

R: R heißt Relationalstruktur über (auf) *C*.

Ein Beispiel

Beispiel (Hausszene:)

Was ist zur Beschreibung nötig?

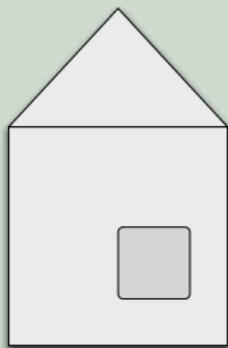


- ▶ Eine Relation, die ein Objekt **Kante** beschreiben kann.

Ein Beispiel

Beispiel (Hausszene:)

Was ist zur Beschreibung nötig?

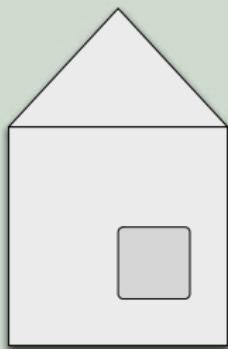


- ▶ Eine Relation, die ein Objekt **Kante** beschreiben kann.
- ▶ Eine Relation, die den **Zusammenhang** zweier Kanten beschreiben kann.

Ein Beispiel

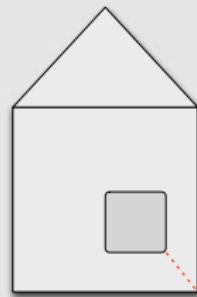
Beispiel (Hausszene:)

Was ist zur Beschreibung nötig?



- ▶ Eine Relation, die ein Objekt **Kante** beschreiben kann.
- ▶ Eine Relation, die den **Zusammenhang** zweier Kanten beschreiben kann.
- ▶ Eine Relation, die die **relative Position** zweier Polygonzüge zueinander beschreiben kann.

Die Relationen



- ▶ Die Kantenrelation:

$$(< \text{KantenID} > \quad < \text{Punkt} > \quad < \text{Punkt} > \\ < \text{Laenge} >)$$

- ▶ Die Zusammenhangsrelation:

$$(< \text{WinkelID} > \quad < \text{Kante} > \quad < \text{Kante} > \\ < \text{Punkt} > \quad < \text{Winkel} >)$$

- ▶ Relative Position:

$$(< \text{Pos.ID} > \quad < \text{virt.Kante} > \\ < \text{Winkel} > \quad < \text{Winkel} >)$$

Anforderungen an die relationale Bildbeschreibung

Welche Eigenschaften sollte die Beschreibung haben?

Lageinvarianz:

Die Beschreibung sollte **invariant** sein gegen

- ▶ Rotation,
- ▶ Translation
- ▶ und Skalierung.

Reversibilität:

Die Beschreibung sollte das Objekt eindeutig beschreiben, so daß aus der Beschreibung das Objekt rekonstruiert werden kann.

Konsequenzen für die Relationen

Aus den Anforderungen folgt,

- ▶ daß die Punkte nicht durch ihre Koordinaten, sondern durch Bezeichner repräsentiert werden sollten,
- ▶ daß die Längen normiert werden müssen
- ▶ und daß die Winkel zwischen Kanten nur relativ zu den beteiligten Kanten angegeben werden können.

Das Relationengebilde

$C = \text{Kantenbezeichner}$

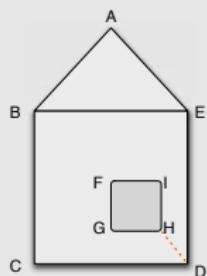
$\cup \text{Punktbezeichner}$

$\cup \text{Winkelbezeichner}$

$\cup (0, 1) \cup (-180^\circ, 180^\circ)$

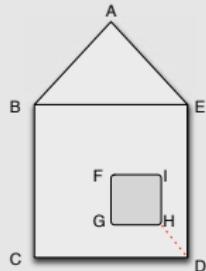
$R = (\text{Kantenrelation}, \text{Winkelrelation}, \text{Positionsrelation})$

Die Kantenrelation



(Kante1	A	B	0.7)
(Kante2	B	C	1.0)
(Kante3	C	D	1.0)
(Kante4	D	E	1.0)
(Kante5	E	A	0.7)
(Kante6	E	B	1.0)
(Kante7	F	6	0.4)
(Kante8	G	H	0.4)
(Kante9	H	I	0.4)
(Kante10	I	F	0.4)
(VKante1	H	D	0.37)

Die Winkelrelation



(Winke1	Kante1	Kante2	B	135°)
⋮				
(Winke13	Kante7	Kante10	F	90°)
(Winke14	VKante1	Kante3	D	60°)
(Winke15	VKante1	Kante9	H	120°)

{ (Position1 VKante1 Winke14 Winke15) }

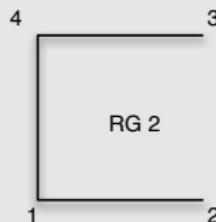
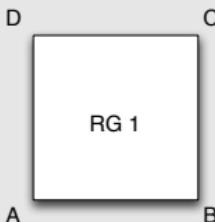
Vergleich von Objektmodell und Bild

Gegeben seien zwei Relationengebilde RG^1 und RG^2 , die ein Objektmodell und Bildstrukturen beschreiben.

$$\begin{aligned} RG^1 &= (C^1, R^1) \\ RG^2 &= (C^2, R^2) \end{aligned}$$

Um das Objektmodell mit den Bildstrukturen zu vergleichen, wird eine Abbildung von RG^1 auf RG^2 gesucht.

Bild und Objektmodell



(Kante1	A	B	1.0)	(Edge1	2	1	1.0)
(Kante2	B	C	1.0)	(Edge2	1	4	1.0)
(Kante3	C	D	1.0)	(Edge3	4	3	1.0)
(Kante4	D	A	1.0)				
(Winkel1	Kante1	Kante2	-90°)	Angle1	Edge3	Edge2	-90°
(Winkel2	Kante2	Kante3	-90°)	Angle2	Edge2	Edge1	-90°
(Winkel3	Kante3	Kante4	-90°)				
(Winkel4	Kante4	Kante1	-90°)				

Abbildung von Relationengebilden

Definition (Abbildung)

Eine Abbildung ist eine Menge von Paaren
 $\{(r_1 r_2)\} \subseteq RG_1 \times RG_2.$

Jedes Paar beschreibt eine Zuordnung eines
Relationstupels aus RG_1 zu einem Relationstupel aus
 $RG_2.$

Einschränkung der Zuordnungen

Nicht jede Zuordnung ist sinnvoll. Daher kann die Menge der möglichen Zuordnungen durch folgende Bedingungen eingeschränkt werden:

- ▶ Beide Relationstupel haben die gleiche Stellenzahl.
 ☞ Vergleichbarkeit der „Syntax“.
- ▶ Beide Relationstupel gehören zu einer Relation.
 ☞ Vergleichbarkeit der „Semantik“.

Auf unser Beispiel angewandt, heißt das, daß nur Kanten auf Kanten und Winkel auf Winkel abgebildet werden dürfen.

Familien von Abbildungen

Eine Abbildung

$$F : (C^1, R_1^1, R_2^1, \dots) \mapsto (C^2, R_1^2, R_2^2, \dots)$$

besteht also aus einer **Familie** von Abbildungen

$$f_i : R_i^1 \mapsto R_i^2$$

der einzelnen Relationen und der Abbildung

$$f_0 : C^1 \mapsto C^2$$

der Trägermengen.

R-Morphismen

Für die Abbildungen muß gelten:

- 1 Für die Trägermengen:

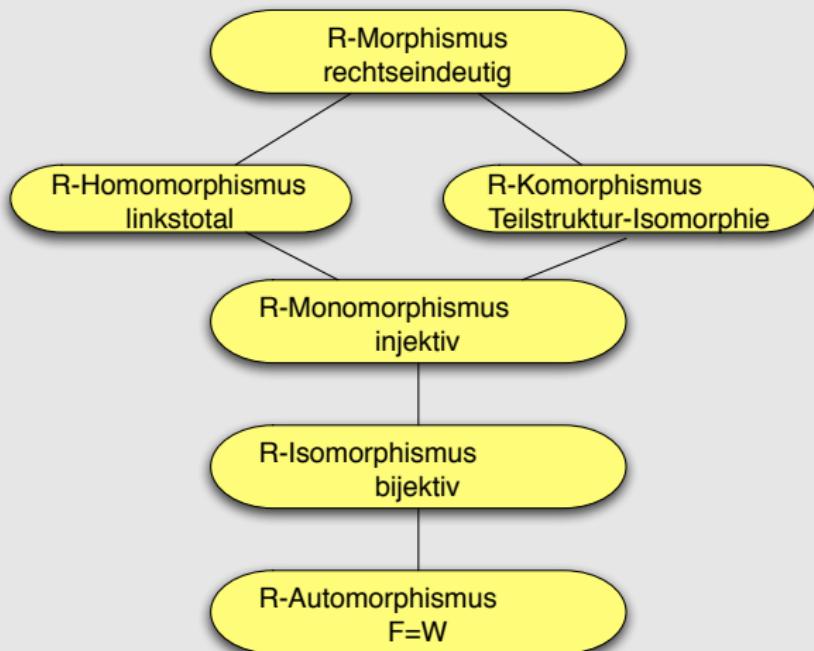
$$((m_1^1, m_2^1, m_3^1, \dots) (m_1^2, m_2^2, m_3^2, \dots)) \Rightarrow \forall_j : (m_j^1, m_j^2) \in f_0$$

- 2 Jede Abbildung f_i ist rechtseindeutig (und ist damit eine Funktion).

Definition (R-Morphismus)

Eine Abbildung, die diesen Eigenschaften genügt, heißt R-Morphismus

Taxonomie von R-Morphismen



Konstruktion von R-Morphismen

Die einfachste Konstruktionsvorschrift lautet:

- ▶ Betrachte alle möglichen Zuordnungen
- ▶ und streiche alle die weg, die nicht zu einem R-Morphismus gehören.

Problem:

Wann gehört eine Zuordnung nicht zu einem R-Morphismus?

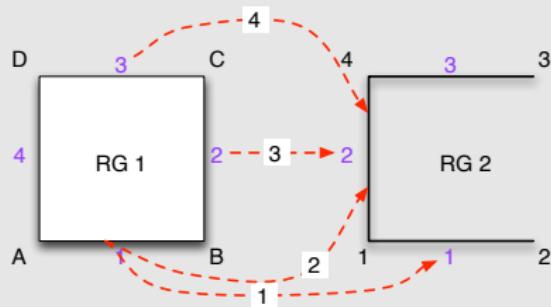
Nicht-kompatible Zuordnungen

- ▶ Es genügt, alle **Paare** von Zuordnungen zu betrachten. Können beide zu einem R-Morphismus gehören, so sind die beiden Zuordnungen kompatibel.
- ▶ Ein R-Morphismus wird also durch eine Menge von Zuordnungen beschrieben, in der alle Zuordnungen paarweise kompatibel sind.

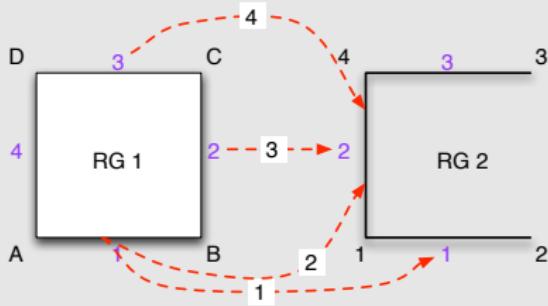
Problem:

Wann sind zwei Zuordnungen nicht kompatibel?

Bild und Objektmodell



$(Kante1 \quad A \quad B \quad 1.0)$	$(Edge1 \quad 2 \quad 1 \quad 1.0)$
$(Kante2 \quad B \quad C \quad 1.0)$	$ \quad (Edge2 \quad 1 \quad 4 \quad 1.0)$
$(Kante3 \quad C \quad D \quad 1.0)$	$(Edge3 \quad 4 \quad 3 \quad 1.0)$
$(Kante4 \quad D \quad A \quad 1.0)$	



$$z_1 = \{((Kante1 \ A \ B \ 1.0) \ (Edge1 \ 2 \ 1 \ 1.0))\}$$

$$z_2 = \{((Kante1 \ A \ B \ 1.0) \ (Edge2 \ 1 \ 4 \ 1.0))\}$$

$$z_3 = \{((Kante2 \ B \ C \ 1.0) \ (Edge2 \ 1 \ 4 \ 1.0))\}$$

$$z_4 = \{((Kante3 \ C \ D \ 1.0) \ (Edge2 \ 1 \ 4 \ 1.0))\}$$

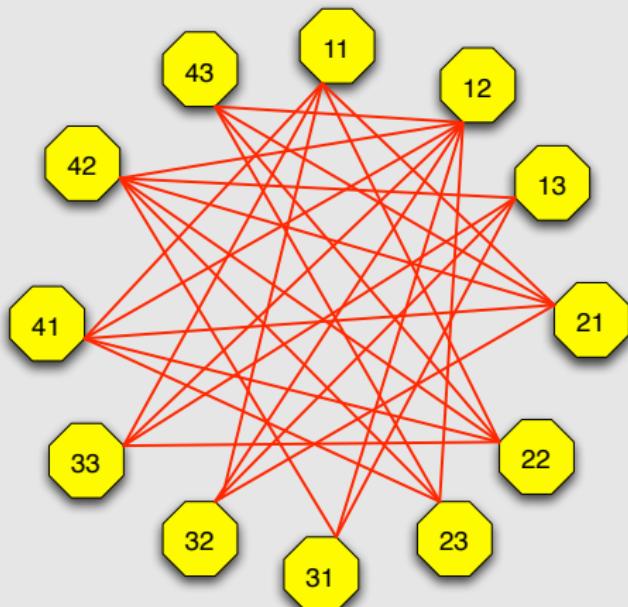
z_1, z_2 : nicht kompatibel, *nicht linkseindeutig*

z_1, z_3 : kompatibel

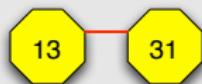
z_2, z_3 : nicht kompatibel, *nicht rechtseindeutig*

z_2, z_3 : kompatibel als *R-Morphismus*, nicht kompatibel als *R-Mono-, Iso-, Komorphismus*

Der Kompatibilitätsgraph



12
Zuordnung zwischen
Kante 1 und Edge 1



Zuordnung 13 ist
kompatibel zu 31

R-Morphismen und Cliques in einem Graph

Satz

In einem R-Morphismus sind alle Zuordnungen paarweise kompatibel.

Der Umkehrschluß ist ebenfalls richtig :

Satz

Eine Menge von paarweise kompatiblen Zuordnungen bilden einen R-Morphismus.

Vorgehen

Lösungsansatz:

Um R-Morphismen für Zuordnungen zwischen zwei Relationengebilden (z.B. für Bild und Modell) zu finden, suche maximal zusammenhängende Teilgraphen des Kompatibilitätsgraphen (**Cliquen**).

Vorgehen

Lösungsansatz:

Um R-Morphismen für Zuordnungen zwischen zwei Relationengebilden (z.B. für Bild und Modell) zu finden, suche maximal zusammenhängende Teilgraphen des Kompatibilitätsgraphen (**Cliquen**).

Problem

Wie findet man alle Cliquen in einem (Kompatibilitäts-)Graphen?

Vorgehen

Lösungsansatz:

Um R-Morphismen für Zuordnungen zwischen zwei Relationengebilden (z.B. für Bild und Modell) zu finden, suche maximal zusammenhängende Teilgraphen des Kompatibilitätsgraphen (**Cliquen**).

Problem

Wie findet man alle Cliquen in einem (Kompatibilitäts-)Graphen?

Lösung (Cliquensuche):

Rekursives Zerlegen der Menge der Zuordnungen

Ein Cliquenalgorithmus

	11	12	13	21	22	23	31...	
1	1	1	1	1	1	1	1	11 n. komp. zu 12
1.1	1	0	1	1	1	1	1	11 n. komp. zu 13
1.2	0	1	1	1	1	1	1	12 n. komp. zu 13
1.1.1	1	0	0	1	1	1	1	
1.1.2	0	0	1	1	1	1	1	
1.2.1	0	1	0	1	1	1	1	
1.2.2	0	0	1	1	1	1	1	
		⋮						

Vorgehen

- ▶ Nehme zunächst an, alle Zuordnungen wären kompatibel.
- ▶ Prüfe paarweise die Zuordnungen auf Kompatibilität.
- ▶ Wenn ein Paar von inkompatiblen Zuordnungen gefunden wird, bilde zwei neue Zuordnungsmengen, die jeweils nur eine der beiden inkompatiblen Zuordnungen enthalten.
- ▶ Prüfe rekursiv beide Mengen auf Kompatibilität, und zerlege rekursiv solange weiter, bis keine Konflikte mehr auftreten.

Beachte

- ▶ Das Problem Cliquen in einem Graph zu finden, gehört in die Klassen der NP-vollständigen Probleme.
- ▶ Graph-matching über R-Morphismen ist daher nur für kleine Graphen praktikabel.
- ▶ Bei größeren Aufgabenstellungen sind daher weiterreichende Mechanismen, wie beispielsweise hierarchische Modelle, erforderlich.

Teil XXI

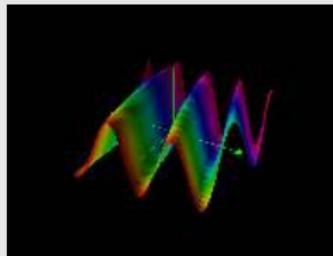
Bildverstehen

Teil XXII

Übungen: Werkzeuge, Handbücher und Beispiele

Übungen

Werkzeuge, Handbücher und Beispiele



56

Die Syntax von POV-Ray

- Definition von Standardobjekten
 - Kugel
 - Zylinder
 - Kegel
 - Torus
 - Ebenen
 - Blöcke und Quader
- Kontrollstrukturen in POV-Ray
- Vektorfunktionen

57

Strukturierung von Povray-Programmen

58

Video-Schnitt

Definition einer Kugel

Die Kugel wird beschrieben
durch:

- ▶ Vektor zum Mittelpunkt
- ▶ Radius
- ▶ Eigenschaften der
Oberfläche

movie

Eine Kugel im Ursprung

Beispiel (Definition einer Kugel)

Eine Kamera, eine Lichtquelle und eine grüne Kugel (Radius 1).

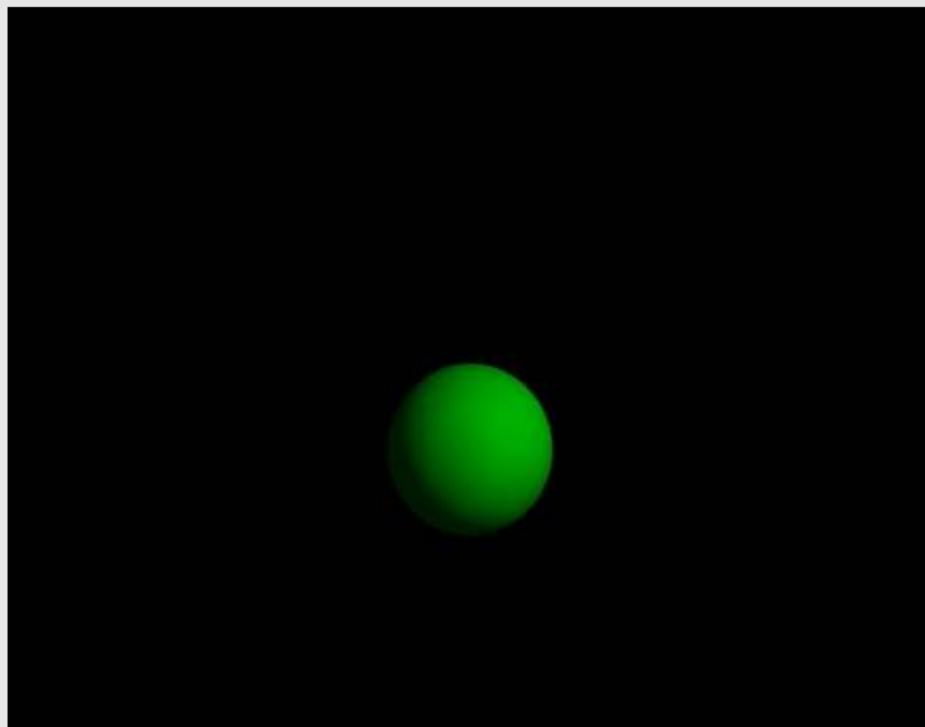
```
#include "colors.inc"

camera { angle 20
    location <2, 10, -30>
    look_at <0, 1, 0>}
light_source { <20, 20, -20>
    color White }

sphere {
    0, // center: the origin , 0 =<0,0,0>
    1 // radius
    pigment { Green }}
```

Das von POV-Ray erzeugte Bild

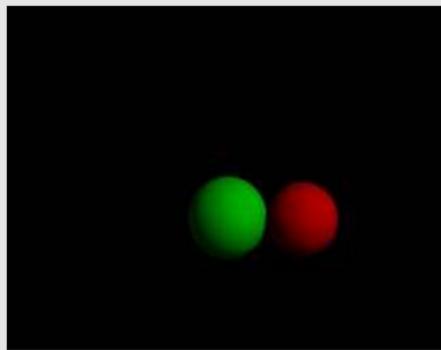
Beisp. 144



Rechnen in POV-Ray

Problem (Relative Anordnung von Objekten:)

Wie können Koordinaten in Abhängigkeit von anderen Werten angegeben werden?



Z.B. eine zweite Kugel genau neben der ersten angeordnet werden?

Beispiel

Der Radius der zweiten Kugel „radiusB“ wird abhängig vom Radius der Kugel „A“ definiert.

Variable und arithmetische Ausdrücke

```
#declare radiusA=1;  
#declare radiusB=radiusA*0.9;  
#declare centerA=0;  
#declare centerB=centerA+x*(radiusA+radiusB);  
  
sphere {  
    centerA, // center  
    radiusA // radius  
    pigment { Green }  
}  
sphere {  
    centerB, // center  
    radiusB // radius  
    pigment { Red }  
}
```

Definition eines Zylinders

Ein Zylinder wird beschrieben durch:

- ▶ Vektor zum Mittelpunkt des Deckels
- ▶ Vektor zum Mittelpunkt des Bodens
- ▶ Radius
- ▶ Eigenschaften der Oberfläche

movie

Zwei Zylinder in POV-Ray

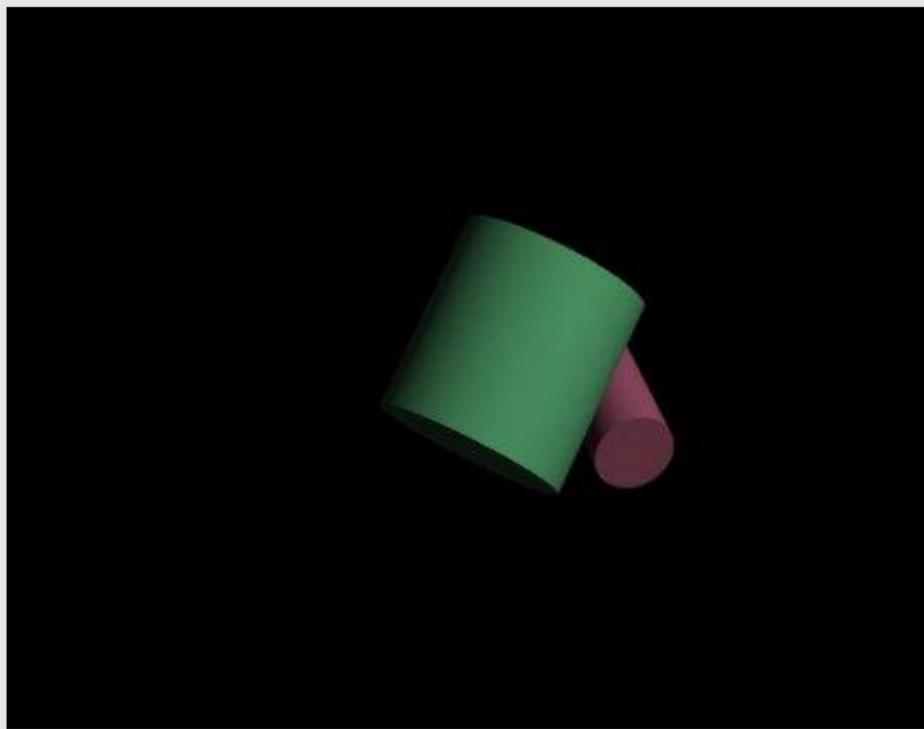
Beispiel

Definition zweier Zylinder

```
cylinder {0,          // endpoint 1 of axis
           <1, 2, 1> // endpoint 2 of axis
           1.2        // radius
pigment {Aquamarine}
}
cylinder {2*x,      // endpoint 1 of axis
           x+y+3*z // endpoint 2 of axis
           0.5       // radius
pigment {
color red 0.858824
green 0.439216
blue 0.576471}}
```

Das von POV-Ray erzeugte Bild

Beisp. 147



Definition eines Kegels

Ein Kegel wird beschrieben durch:

- ▶ Vektor zum Mittelpunkt des Bodens
- ▶ Radius des Bodens
- ▶ Vektor zum Mittelpunkt des Deckels
- ▶ Radius des Deckels
- ▶ Eigenschaften der Oberfläche

movie

Zwei Kegel in POV-Ray

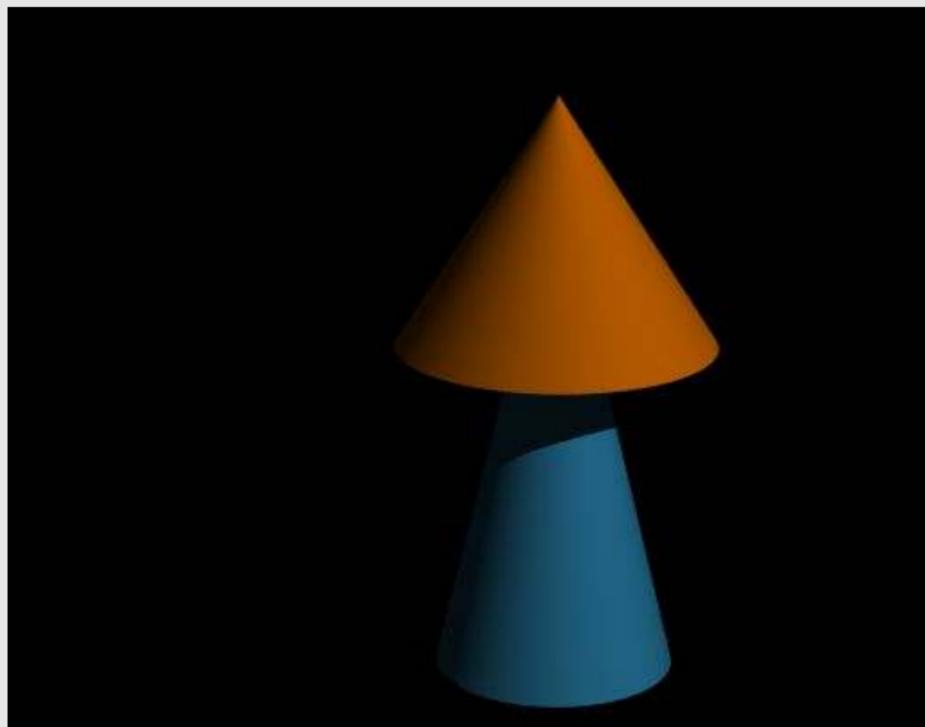
Beispiel

Definition zweier Kegel

```
cone {<1,1,1>,    // endpoint 1 of axis
      2,          // radius 1
      <1, 4, 1>, // endpoint 2 of axis
      0           // radius 2
  pigment {Coral}
}
cone {1, // endpoint 1 of axis
      0.5, // radius 1
      <1,-3,1>, // endpoint 2 of axis
      1.5 // radius 2
  pigment {SkyBlue}
}
```

Das von POV-Ray erzeugte Bild

Beisp. 148



Definition eines Torus

Ein Torus wird beschrieben durch:

- ▶ Hauptradius (major radius)
- ▶ Nebenradius (minor radius)
- ▶ Lage (nicht in POV-Ray)
- ▶ Eigenschaften der Oberfläche

movie

Ein großer und ein kleiner Torus

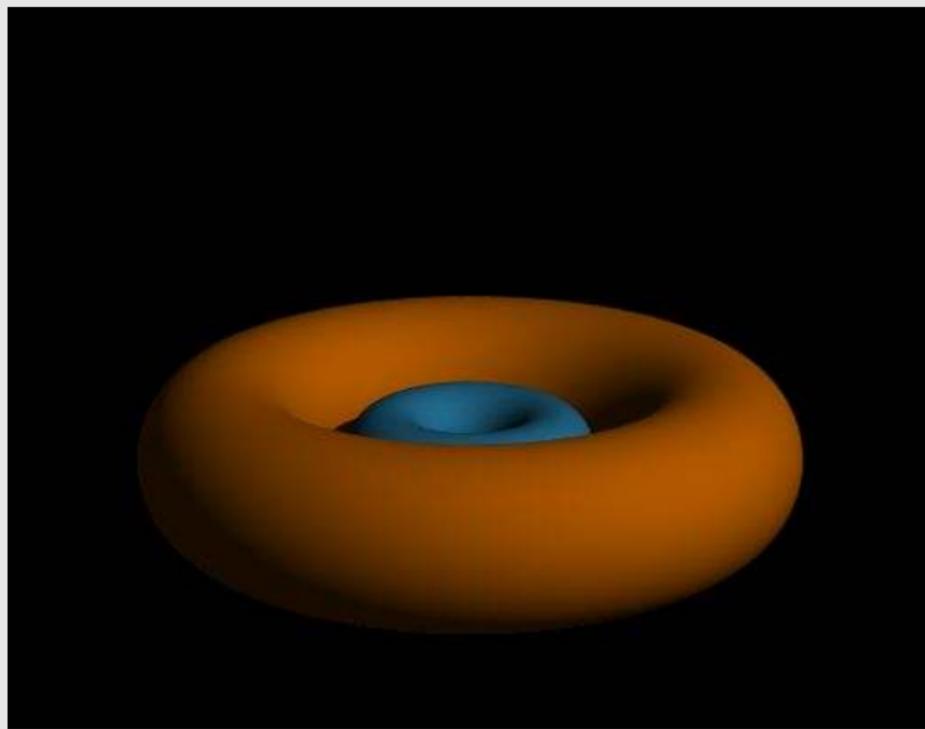
Beispiel

Definition zweier Ringe als Torus

```
torus {3,    // major radius
       1    // minor radius
       pigment {Coral}
}
torus {1,    // major radius
       0.5 // minor radius
       pigment {SkyBlue}
}
```

Das von POV-Ray erzeugte Bild

Beisp. 149



Definition einer Ebene

Eine Ebene wird
beschrieben durch:

- ▶ Die Orientierung im Raum (Normalenvektor)
- ▶ Den Ort (ein Punkt in der Fläche, angegeben durch den Abstand vom Ursprung)

movie

Zwei Ebenen senkrecht zueinander

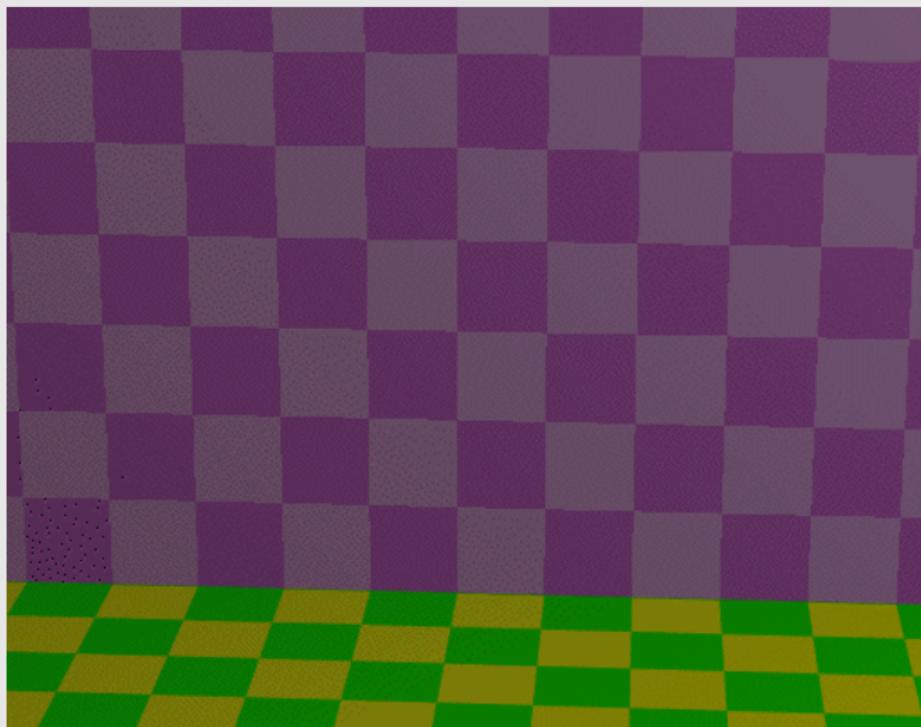
Beispiel

Definition zweier Ebenen in den Grundebenen des Koordinatensystems

```
plane {y,      // normal vector
       -1     // offset
       pigment {checker Yellow ,Green}}  
  
plane {-z,    // normal vector
       2      // offset
       pigment {checker colour Orchid
                 colour Plum}}}
```

Das von POV-Ray erzeugte Bild

Beisp. 150



Definition eines Quaders

Beispiel

Definition eines animierten Quaders (box)

- ▶ Ein Quader wird beschrieben durch zwei gegenüberliegende Eckpunkte:
- ▶ Die Seitenflächen des Quaders sind parallel zu den Grundebenen des Koordinatensystems.

movie

Ein animierter Quader

```
// Definition of a box
#include "colors.inc"
#include "metals.inc"
light_source {
    <sin(clock*2*pi)*20, 20, -20>
    color White }
background { color SkyBlue }
#declare PA=-0.5;
#declare PB=PA+2*clock;
box {PA, // corner
    PB // opposite corner
    texture{ pigment {
        checker colour Red
        colour Yellow}}
    finish{F_MetalE}}
}
```

Animation eines Quaders

Pulsierender Quader

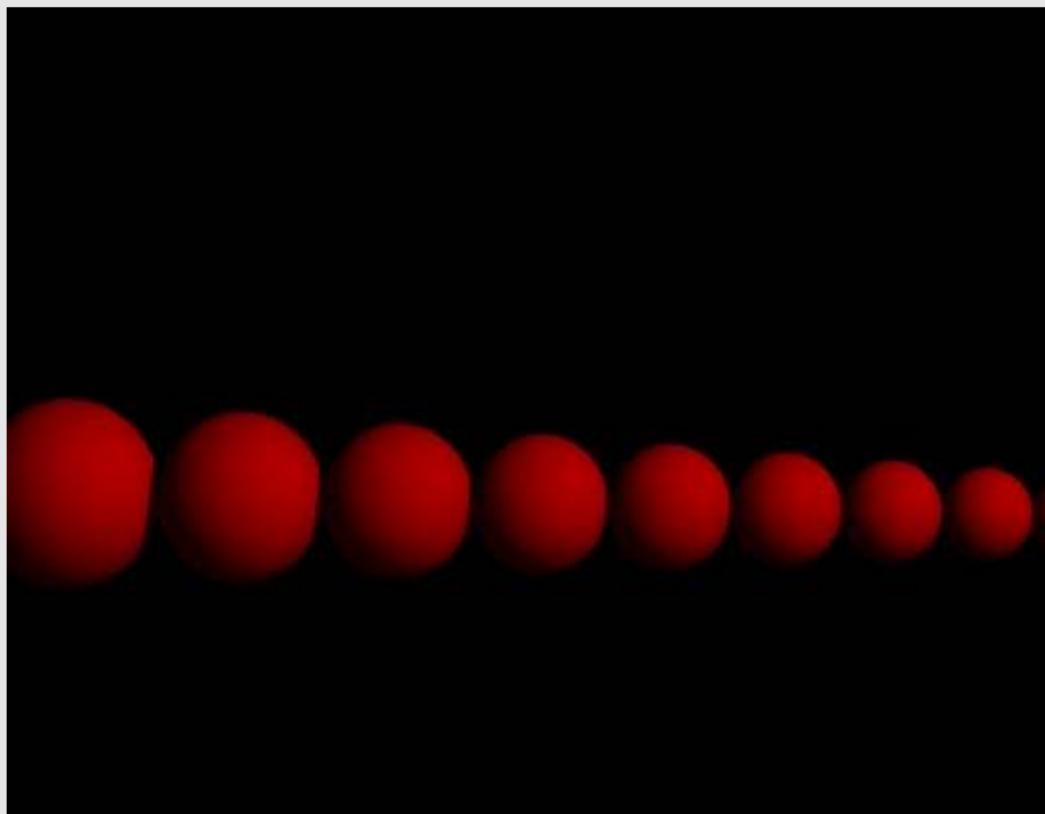
Wiederholung: While-Schleife

Beispiel

Eine lange Reihe von Kugeln: In jedem Iterarationsschritt wird die Kugel um den Faktor 0.9 kleiner und nach rechts verschoben.

```
#declare Radius=1.0;
#declare Center=<-5,0,0>;
#declare smaller=0.9;
#while (Radius>0.3)
    sphere {Center, Radius
            pigment { Red }
    #declare Center=Center
                +(1+smaller)*<Radius,0,0>;
    #declare Radius=Radius*smaller;
#end
```

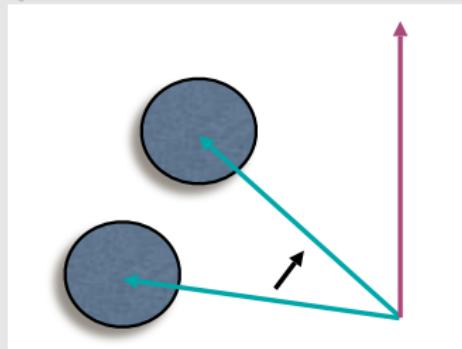
Das erzeugte Bild zu Beispiel 152



Anordnung der Kugeln im Kreis

Beispiel

Um die Kugeln im Kreis anzurichten, wird der Radiusvektor des Kreises bei jedem Iterationsschritt um einen festen Winkel weitergedreht.

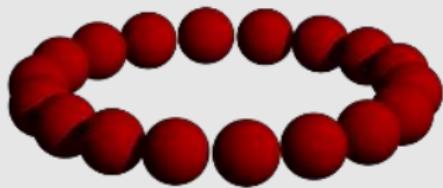


Anordnung im Kreis durch Rotation

```
#declare SRadius=0.5;
#declare SCenter=3*x;
#declare Sangle=0;
#declare stepAngle=20;
#while (Sangle<360.0)
    sphere {
        vaxis_rotate(SCenter,y,Sangle),
        SRadius
        pigment { Red }
        #declare Sangle=Sangle+stepAngle;
    }
#end
```

Das von POV-Ray erzeugte Bild

Beisp. 153



Vektorfunktionen

Kreuzprodukt: **vcross** (*vector*, *vector*)

Normalisierung: **vnormalize** (*vector*)

Rotation: Drehe den Punkt *vector* um die Achse *axis* mit dem Drehwinkel *angle*, angegeben im Bogenmaß.

vaxis_rotate (*vector*, *axis*, *angle*)

- ▶ Koordinatenangaben sind Vektoren.
- ▶ Farbangaben sind Vektoren.

Vordefinierte Vektoren:

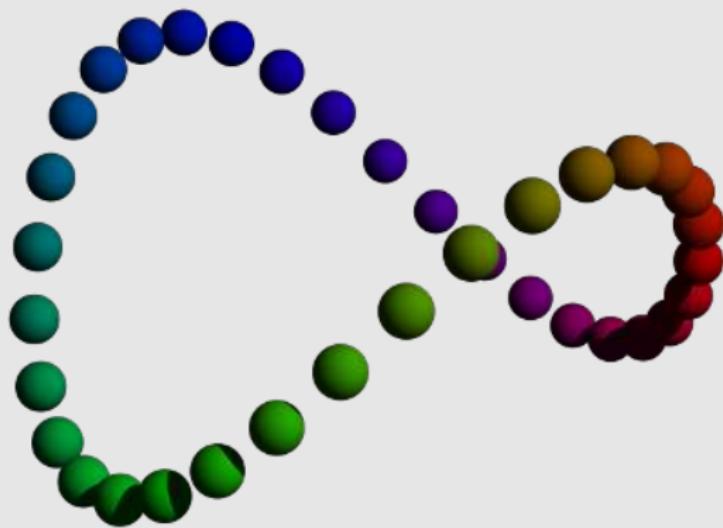
- x: Der Einheitsvektor in Richtung der X-Achse:
 $<1,0,0>$.
- y: Der Einheitsvektor in Richtung der Y-Achse:
 $<0,1,0>$.
- z: Der Einheitsvektor in Richtung der Z-Achse:
 $<0,0,1>$.
- 1: Der Vektor $<1,1,1>$
- 0: Der Nullvektor $<0,0,0>$

Trigonometrische Funktionen, Farbvektoren

Beispiel (Rotiere den Farbvektor im Farbkreis)

```
#declare SRadius=0.3;
#declare SCenter=4*x;
#declare Sangle=0;
#declare stepAngle=10;
#while (Sangle<360.0)
    sphere {
        vaxis_rotate(SCenter,y,Sangle)
        +2*y*(sin(2*pi*Sangle/180.0)),
        SRadius
        pigment{ color
            vnormalize(
                vaxis_rotate(x,x+y+z,Sangle))}}
#declare Sangle=Sangle+stepAngle;
#end
```

Die Farbe rotiert durch den Farbkreis



Simulationsuhr und Animation

```
#declare SRadius=0.3;
#declare SCenter=4*x;
#declare Sangle=0;
#declare stepAngle=10;
#declare Sphase=360*clock
#while (Sangle<360.0)
    sphere {
        vaxis_rotate(SCenter,y,Sangle+Sphase)
        +2*y*(sin(2*pi*(Sangle+Sphase)/180.0)),
        SRadius
        pigment{ color
            vnormalize(vaxis_rotate(x,x+y+z,
                Sangle+Sphase))} }
#declare Sangle=Sangle+stepAngle;
#end
```

Pulsierender Quader

Modularisierung des Programms

- ▶ Makros, Funktionen
- ▶ Include-Files:
 - ▶ für vordefinierte Objekte,
 - ▶ für gemeinsam benutzte Parameter
 - ▶ und als Algorithmenbibliothek.

Bibliotheken

- ▶ Die Standardbibliotheken bieten vordefinierte
 - ▶ Objekte
 - ▶ Farben, Texturen, Materialien
 - ▶ Funktionen und Macros
- ▶ Im Informatik-Rechenzentrum finden Sie die Povray-Bibliotheken unter
/opt/pkg/povray/lib/povray
- ▶ Verwendet werden diese mit **#include "filename"**

Standardbibliotheken

colors.inc: Vordefinierte Farben

textures.inc: Vordefinierte Materialien

finish.inc: Reflektanzen

stones.inc: Gesteinstexturen

skies.inc: Wolkentexturen

stars.inc: Sternenhimmel

woods.inc: Holzmaserungen

skies.inc: Himmelstexturen

Beispiel (Simulierte Cirrostratus-Wolken)

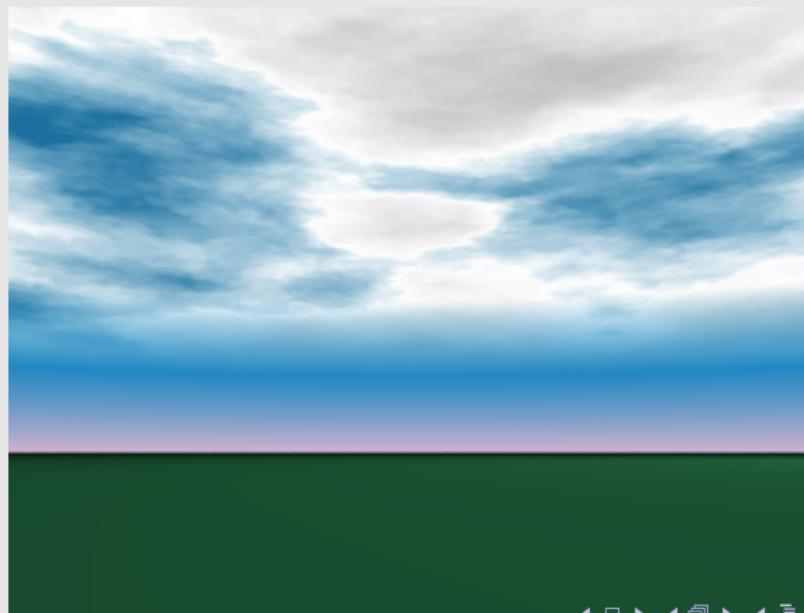
```
#include{ skies.inc }
sky_sphere{ S_cloud1 }
```



skies.inc: Himmelstexturen

Beispiel (Blauer Himmel mit Schäfchenwolken)

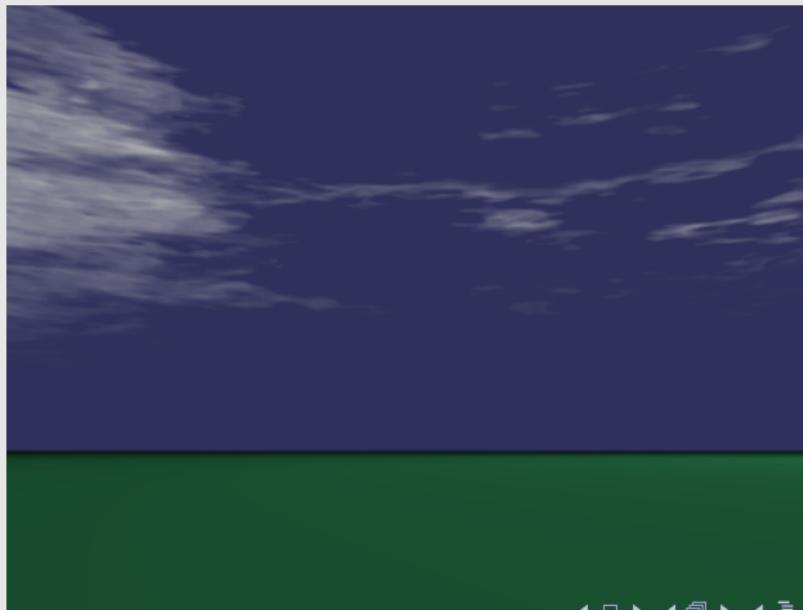
```
#include{ skies.inc }
sky_sphere{ S_cloud2 }
```



skies.inc: Himmelstexturen

Beispiel (Geringe Cirrusbewölkung)

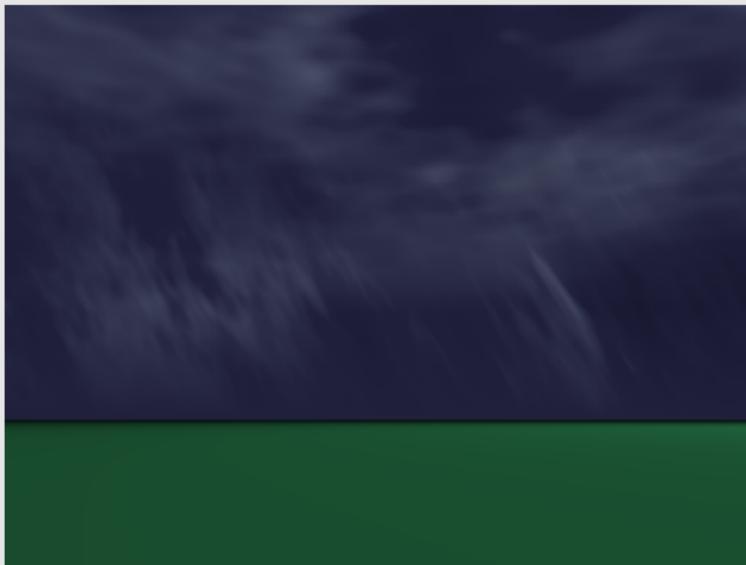
```
#include{ skies.inc }
sky_sphere{ S_cloud3 }
```



skies.inc: Himmelstexturen

Beispiel (Regenwolken)

```
#include{ skies.inc }
sky_sphere{ S_cloud4 }
```



skies.inc: Himmelstexturen

Beispiel (Dichte Cirruswolken)

```
#include{ skies.inc }
sky_sphere{ S_cloud5 }
```



skies.inc: Himmelstexturen

Beispiel (Ein flammendes Inferno)

```
#include{ skies.inc }
sky_sphere{ pigment{ Apocalypse } }
```



Himmelsobjekte

Himmelsobjekte haben besondere Eigenschaften:

- ▶ Sie sind unendlich weit entfernt.
- ▶ Sie haben ihre eigene Lichtquelle.
- ▶ Sie können rotiert werden.
- ▶ Ihre Pigmente können skaliert werden.

```
sky_sphere {  
    S_cloud2  
    rotate y*clock*360}
```

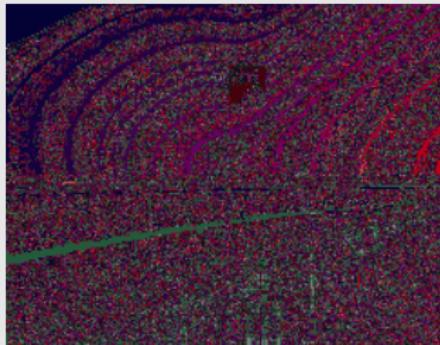
Animation der Wolken

cloud1

Ton aus

Start Ton

Beispiel: Flagge mit Bild



Universitätsflagge

Die Komponenten:

- ▶ Flaggenmast: Pfahl, Knopf
- ▶ Flaggentuch mit Bild
- ▶ Himmel
- ▶ Rasen

Zu ladende Standardbibliotheken

```
#include "colors.inc"
#include "skies.inc"
#include "textures.inc"
#include "golds.inc"
```

Ausstattung der Szene: Lichtquellen, Himmel

```
#declare flagsize= 1.5;
#declare topofflagpole=6*flagsize*y;
#declare radiusofpole=0.06;
camera {
    location <0, 0, -17> + 0.5*topofflagpole
    look_at 0.5*topofflagpole
    angle 54}
light_source { <100, 100, -50> White }
light_source { <-20, 20, -50> White
              shadowless}
sky_sphere { pigment{Apocalypse} }
plane { y, 0
        pigment { color red 0.3 green 0.75 blue 0.5}}
```

Modellierung der Flagge



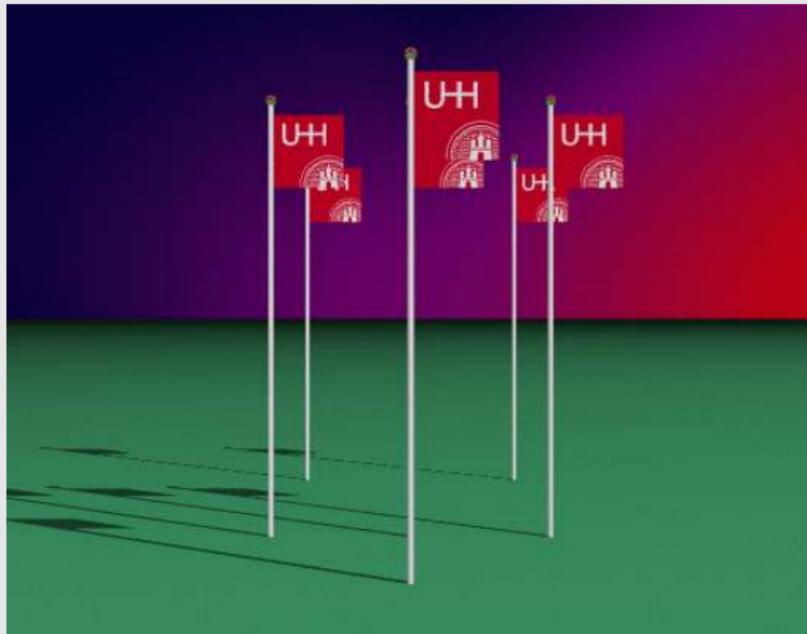
"UHH_ROT.png"

- ▶ Das Flaggentuch wird als Polygon modelliert.
- ▶ Das Bild auf der Flagge wird als **image_map**-Textur auf das Polygon projiziert.
- ▶ Das Bild hat in der X , Y -Ebene die Eckpunkte $(0, 0)$ und $(1, 1)$.

Komponenten der Flagge: Mast, Knauf, Tuch

```
#declare flag=union{
    // Mast
    cylinder{0,topofflagpole ,radiusofpole
        pigment{White}
        finish{Shiny}}
    // Kopf des Mastes
    sphere{topofflagpole , 2*radiusofpole
        texture{T20}}
    // Das Flaggentuch: Ein Rechteck
    polygon {5, 0, x, x+y, y, 0
        scale flagsize
        pigment{image_map{png "UHH_ROT.png"}}
        scale flagsize}
    translate 0.8*topofflagpole+x*radiusofpole }}
```

Mehrfachnutzung von Objekten (Beisp. 161)



Erzeugen mehrerer Instanzen des Modells

Beispiel (Instanzen des Objektmodells „flag“):

Das Modell wird mittels Translation transformiert und an vier verschiedenen Positionen in die Szene eingesetzt.

```
object { flag }
object { flag  translate 2*flagsize*x}
object { flag  translate -2*flagsize*x}
object { flag  translate 2*flagsize*(x+2*z)}
object { flag  translate 2*flagsize*(-x+2*z)}
```

Die Fachbereichsflagge

Beispiel („fbilogos“):

Eine zweite Flagge mit anderem Muster:

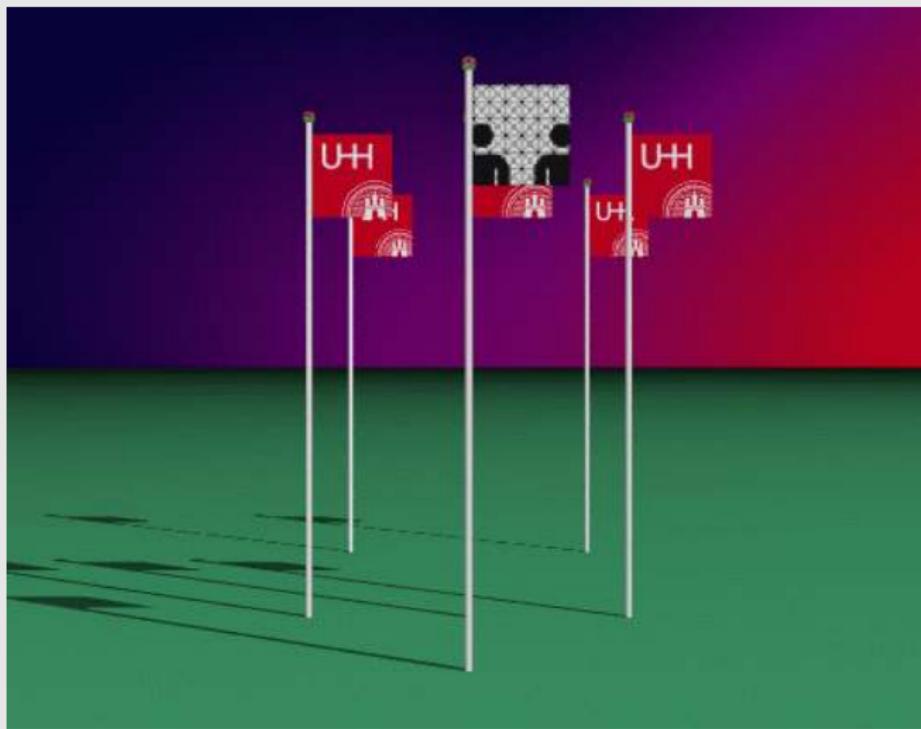


"fbilogos.png"

```
#declare fbiflag=union{
    cylinder{0,topofflagpole ,radiusofpole
    pigment{White}
    finish{Shiny}}
    sphere{topofflagpole , 2*radiusofpole
    texture{T20}}
    polygon {5, 0, x, x+y, y, 0
    scale flagsize
    pigment{image_map{png "fbilogos.png"}}
        scale flagsize}
    translate 0.8*topofflagpole+x*radiusofpole}
}
```

Flaggenwald mit Departmentflagge

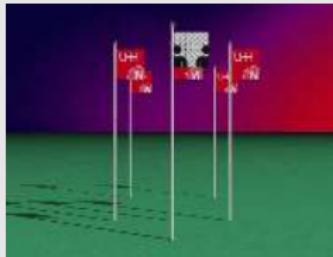
Beisp. 162



Flaggenwald mit Departmentflagge

```
object { flag }
object { flag translate 2*flagsize*x }
object { flag translate -2*flagsize*x }
object { flag translate 2*flagsize*(x+2*z) }
object { flag translate 2*flagsize*(-x+2*z) }
object { fbiflag translate -2*flagsize*z }
```

Macros, Parameter



Unsere Fachbereichsflagge

Anweisungsfolgen können zu Makros zusammengesetzt und parametrisiert werden:
So können Modelle parametrisiert werden.

Ein Makro für eine Flagge

```
#macro flagB ( bild , ort , breite , hoehe )
union{
    cylinder{0,topofflagpole , radiusofpole
        pigment{ White }
        finish{ Shiny }}
    sphere{ topofflagpole , 2*radiusofpole
        texture{T20}}
    polygon {5, 0, x*breite , x*breite+y*hoehe ,
              y*hoehe, 0
        scale flagsize
        pigment{image_map{png bild}}
        translate -y*(breite-hoehe)/(breite)
        scale flagsize * breite}
    translate 0.8*topofflagpole+x*radiusofpole}
    translate ort}
#end
```

Eine Instanz der Departmentflagge

```
flagB ("fbilogo.png",  
<0,0,0>, 1.5*flagsize , flagsize) // Aufruf
```



Beachte bei Makros:

Verzögerte Auswertung:

Makros werden durch textuelle Ersetzung der Parameter im Rumpf des Makros ausgewertet.

- ▶ Namen können dadurch in einen anderen **lexikalischen Kontext** geraten!
- ▶ Ausdrücke können **mehrfach** ausgewertet werden.
- ▶ **Operatorpräzedenzen** können sich ändern.
- ▶ Es wird **kein Rekursionsstack** aufgebaut.
Daher sollte Rekursion allenfalls für **endrekursive** Makros eingesetzt werden.

Fallstricke: Falsche Operandenbindung

Beispiel (Infixnotation und Operatorpräzedenz)

Wenn die Aktualparameter arithmetische Ausdrücke sind, können Bindungsfehler auftreten, wenn die benachbarten Operatoren stärker binden als die Operatoren im Ausdruck.

```
#macro square (a)
```

```
    a*a
```

```
#end
```

```
#declare c = square (2+3);
```

expandiert zu:

2 + 3*2 + 3 → 11

Fallstricke: Falsche Namensbindung

Beispiel (Falscher lexikalischer Kontext)

Wenn die Aktualparameter Variablennamen enthalten, können Bindungsfehler auftreten, wenn die Variablen im Rumpf des Makros durch lokale Variablen verschattet werden.

```
#macro falscheNamen (a)
    #local xx = 3;
    #declare resultat= 3*a+xx ;
#end
```

```
#declare xx = 1;
falscheNamen (xx)
```

expandiert zu:

$3*a+xx \rightarrow 3*xx+xx \rightarrow 3*3+3 \rightarrow 12$

Definition von Funktionen

Funktionen werden mittels des Schlüsselworts **function** definiert:

Beispiel (Der Mittelwert zweier Zahlen)

Z.B.:

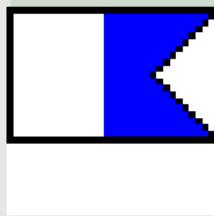
```
#declare mitte = function (a, b) {  
    (a+b)/2}
```

- ▶ Bei Funktionen ist Rekursion verboten.
- ▶ Der Resultattyp ist eingeschränkt auf numerische Werte und auf Materialien.
- ▶ Die Argumente werden applikativ ausgewertet.

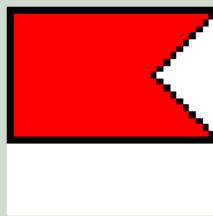
Arrays

Arrays können als Elemente beliebige Objekte enthalten:

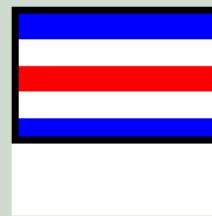
Beispiel (Ein Array von Bildnamen:)



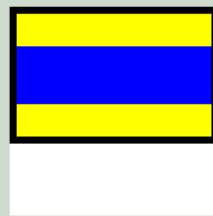
A.png



B.png



C.png



D.png

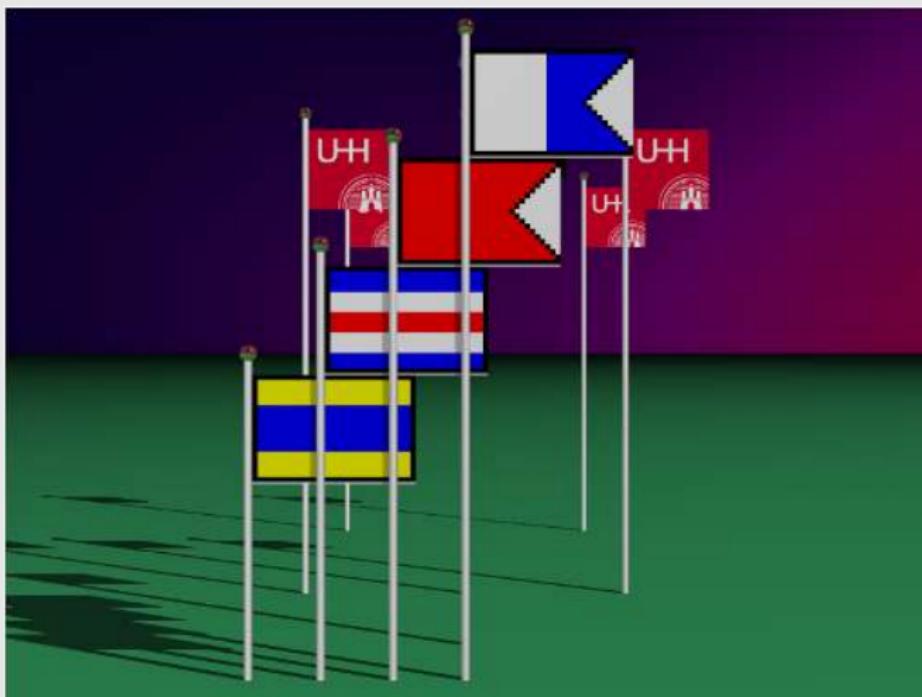
Hier werden vier Bilder als Texturen für Flaggen verwendet.

Eine Schleife über die Elemente eines Arrays

Jede der vier Flaggen erhält ein anderes Bild als Textur.

```
#declare bilder = array [4] {  
    "A.png", "B.png", "C.png", "D.png" }  
  
#declare i = 0;  
#while ( i < dimension_size( bilder , 1 ) )  
    flagB ( bilder [i],  
            -3*flagsize*z-i*x-i*y*1.5, 1.5, 1)  
    #declare i= i+1;  
#end
```

Vier zusätzliche Flaggen des Flaggenalfabets (Beisp. 166)



Zufallszahlen

- ▶ Zufallszahlen können genutzt werden, um Objekte zufällig zu parametrisieren, z.B.
 - ▶ Farbe,
 - ▶ Größe,
 - ▶ Orientierung,
 - ▶ Position...
- ▶ Die Funktion **rand** erzeugt gleichverteilte Zahlen zwischen 0.0 und 1.0.
- ▶ Mit **seed** wird ein Zufallszahlengenerator erzeugt und initialisiert.

Zufällige Kugeln

Beispiel (Kugeln in zufälliger Lage:)

```
#declare R1 = seed(0);
// neuer Zufallszahlengenerator

object {sphere 0, 1
    scale (0.5+ rand(R1)) // zufällige Größe
    translate y*rand(R1)} // zufällige Position
```

schnemann
Ton aus

Start Ton

Kameraführung

- ▶ Alle Kameraparameter können in Abhängigkeit von der clock gesetzt werden.
- ▶ So kann die Kamera
 - ▶ in die Szene hineinfahren,
 - ▶ einem Objekt folgen,
 - ▶ auf bestimmte Objekte fokussieren usw.

Die Kamera fährt mit dem Schneemann mit.

cloud1

Bedingte Ausdrücke: IF_DIRECTIVE:

```
# if ( Cond ) TOKENS . . .
[ # else TOKENS . . . ]
#end
```

- ▶ Ein Wert von 0.0 wird als false interpretiert, jeder float-Wert verschieden von 0.0 als true.
- ▶ Bedingte Ausdrücke können bis zu einer Schachtelungstiefe von 200 verschachtelt werden.

Beispiel (#if: Würfel oder Kugel?)

Abhängig von der Bedingung „Which“ wird eine Kugel oder ein Kubus modelliert.

```
#declare Which=1;  
#if (Which)  
box { 0, 1 }  
#else  
sphere { 0, 1 }  
#end  
// oder  
#declare Which=1;  
#if (Which)  
box { 0, 1 }  
#end
```

Test, ob eine Variable definiert ist.

Die folgenden Ausdrücke testen, ob Namen definiert sind,
sehr nützlich in include-Files:

```
#ifdef (User_Thing) // This section is parsed
                  // identifier "User_Thing"
                  // previously declared
    object{User_Thing} // invoke identifier
#else           // This section is parsed if the
                  // identifier "User_Thing" was not
                  // previously declared
    box{<0,0,0>,<1,1,1>} // use a default
#endif
```

Ist ein Array-Element definiert?

Beispiel (#ifdef:)

Mit **#ifdef** und **#ifndef** kann geprüft werden, ob ein Array-Element initialisiert ist.

```
#declare MyArray = array[10];
#ifndef(MyArray[0])
    #debug "first_element_is_assigned\n"
#else
    #debug "first_element_is_not_assigned\n"
#endif

#debug drückt den Eingabestring auf der Konsole aus.
```

Mehrfachverzweigungen: SWITCH

```
#switch ( Switch_Value )
    SWITCH_CLAUSE ... [#else TOKENS...]
#end

SWITCH_CLAUSE:
    #case( Case_Value ) TOKENS... [#break] |
    #range( Low_Value , High_Value ) TOKENS ... [
```

Zwei Werte werden als gleich gewertet, wenn der Unterschied kleiner als 1e-10 ist.

switch: Schema

```
#switch (VALUE)
  #case (TEST_1)
    // This section is parsed if VALUE=TEST_1
  #break // First case ends
  #case (TEST_2)
    // This section is parsed if VALUE=TEST_2
  #break // Second case ends
  #range (LOW_1,HIGH_1)
    // This section is parsed if LOW_1 <= VALUE <
  #break // Third case ends
  #range (LOW_2,HIGH_2)
    // This section is parsed if LOW_2 <= VALUE<=
  #break // Fourth case ends
  #else // This section is parsed if no other c
    // range is true.
  #end // End of conditional part
```

Beispiel: Drei Uhren

Die Szenenbestandteile haben unterschiedliche Perioden

- ▶ Eine Uhr für die Himmelsdrehung: sehr langsam
- ▶ Eine Uhr für die Wellen: schnell
- ▶ Eine Uhr für den Aufbau des Ammoniten: im Takt der Musik

Im Takt zur Musik

- Alonso, M. and Finn, E. J. (1972).
Physics, 2nd ed.
Addison-Wesley, Reading, Mass.,.
- Arnold, R. D. and Binford, T. O. (1980).
Geometric constraints in stereo vision.
In *Proc. SPIE*, volume 238, pages 281–292, San Diego/CA.
- Ayache, N. and Faverjon, B. (1987).
Efficient registration of stereo images by matching graph descriptions of edge segments.
International Journal of Computer Vision, 1:107–131.
- Baker, H. H. (1982).
Depth from edge and intensity based stereo.
Technical Report STAN-CS-82-930, Department of Computer Science, Stanford University, Stanford, CA.
- Ballard, D. H. and Brown, C. M. (1982).
Computer Vision