IVC Interactive Visual Computing

Leonie Dreschler-Fischer

Prüfungsunterlagen Band 1:

Querschnittsthemen und Grundlagen

Universität Hamburg
Fachbereich Informatik

WS 2014/2015

Inhaltsverzeichnis

Band I: Querschnittsthemen und Grundlagen 1

I Einführung 15

1 Einführung und Grundbegriffe 15
1.1 Entwicklung 15
1.2 Computergrafik vs. Bildverarbeitung 17
1.3 Methoden und Modelle 21

2 Anwendungen und Beispiele 23
2.1 Konstruktion und Fertigung 23
2.2 Anwendungen in der Bildverarbeitung 26
2.3 Echtzeitbilddarstellungen 27
2.4 Visualisierung 28
2.5 Fotorealismus 30

3 Organisatorische 40
3.1 Übungsaufgabe 40
3.2 Literatur 42
3.2.1 Literaturempfehlungen zu Bildverarbeitung und Bildsynthese 42

II Geometrische Grundlagen: Koordinaten und Transformationen 45

4 2D-Koordinatensysteme und Transformationen 45
4.1 2D-Koordinatensysteme 45
4.2 2D-Transformationen 49
4.2.1 2D-Translation 49
4.2.2 2D-Skalierung 50
4.2.3 2D-Rotation 52
4.3 Zusammengesetzte Transformationen 55

5 3D-Koordinatensysteme 61
5.1 Händigkeit eines Koordinatensystems 61
5.2 3D-Transformationen 63
5.2.1 Transformationen in POV-Ray 64

5.2.2 3D-Translation 68
5.2.3 3D-Skalierung 68
5.2.4 3D-Rotation 69

6 Transformationen in POV-Ray 74
6.1 Affine Transformationen in POV-Ray 74
6.1.1 3D-Scherung 74
6.2 Koordinatentransformation und Frames 78
6.3 Ein virtueller Schneemann 82
6.4 Iterative Konstruktion von Objekten 92

III Geometrische Grundlagen: Geometrisches Modellieren 97

7 Einführung: Zweck und Anwendungen des geometrischen Modellierens 97

8 Repräsentation geometrischer Modelle 98
8.1 Volumenbasierte Repräsentationsformen 98
8.1.1 Octtree 98
8.1.2 Constructive Solid Geometry (CSG) 99

IV Geometrische Grundlagen: Abbildung des Raumes 102

9 Projektionen 102
9.1 Grundbegriffe 102
9.2 Lochkamera 103

10 Parallelprojektion 106
10.1 Orthogonale Projektion 108
10.1.1 Transformationsgleichungen für die orthogonale Projektion 110
10.2 Schiefe Parallelprojektion 111
10.2.1 Projektionsgleichungen für den Schrägriß 111

11 Zentralprojektion 113
11.1 Eigenschaften der Zentralprojektion 113
11.2 Projektionsgleichungen 117

11.2.1 Die inverse Zentralprojektion 120

V Perspektive 122

12 Gesetze der Perspektive: Fluchtpunkte 122
12.1 Projektionsgleichung für Weltkoordinaten 122
12.2 Hauptfluchtpunkte 124
12.3 Perspektivische Elemente in der Malerei 133

13 Illusionen 136
13.1 Interpretation von perspektivischen Effekten durch das visuelle System 136
13.2 Scheinriesen und Scheinzwerge — der Ames-Raum 138
13.3 Unmögliche Objekte 139

14 Virtuelle und reale Kameras 144
14.1 Die Linsenkamera 144
14.2 Virtuelle Kameras in POV-Ray 152
14.2.1 Verzerrung der Zentralprojektion 159
14.2.2 Fokussierungsunschärfe 160
14.3 Kamerakalibrierung 161
14.4 Betrachtungstransformation, 3D-Clipping 163

15 Höherdimensionale Räume 168
15.1 Reduktion der Anzahl der Dimensionen 168
15.2 Abbildung der räumlichen Dimensionen auf andere Parameter 171
15.3 Projektion in Unterräume 172

VI Das Licht 181

16 Elektromagnetische Strahlung 181
16.1 Das Elektromagnetisches Spektrum 181
16.2 Interferenz und Brechung 185
16.3 Strahlungsgesetze 209

17 Licht und Oberflächen 214
17.1 Die Farbe des Umgebungslichtes 214
17.2 Pigmente und Licht 215

18 Umgebungseffekte 220

18.1 Lichtquellen 220
18.2 Transmission, atmosphärische Effekte 229
18.2.1 Atmosphärische Trübung 229
18.3 Atmosphärische Effekte in POV-Ray 233
18.3.1 Nebel und Bodennebel 233
18.3.2 Fortgeschrittene Transmissions-Effekte 241

VII Farbmodelle 250

19 Farbmodelle 250
19.1 Farbparameter 250
19.2 Mischen von Farben 252
19.3 Farbsysteme 255
19.3.1 Das CIE-System 255
19.3.2 RGB-System und CMY-System 257
19.3.3 HSV-System 260

20 Physiologie des Farbsehens 265
20.1 Anatomie des Auges 265
20.1.1 Die Linse 267
20.1.2 Die Netzhaut 269
20.1.3 Farbwahrnehmung 273
20.2 Chromatec-Verfahren 279

21 Farbgestaltung 282

VIII Texturen und Fraktale 283

22 Modellierung von Texturen in POV-Ray 283
22.1 Pigmente und Muster 287
22.2 Modifikatoren 292
22.3 Normalentexturen 294

23 Materialien und Transparenz 305
23.1 Texturkarten: 305

23.2 Lichtdurchlässige Materialien 314

IX Fraktale 318

24 Fraktale 318
24.1 Diskrete Fraktale 319
24.2 Die Hausdorff-Dimension 321
24.3 Anwendung fraktaler Strukturen 325
24.3.1 Modellierung von Oberflächen 325
24.3.2 Modellierung vonPflanzen 328

25 Chaotische Folgen 336
25.1 Topologische Dynamik 336
25.2 Julia-Mengen 342
25.3 Die Mandelbrot-Menge 344

26 Fraktale Objekte und Texturen 349
26.1 Julia-Mengen als fraktale Oberflächen 350

Band II: Computer Vision 355

X Bilder und Bildfunktionen 357

27 Analoge Bildfunktionen 358

28 Digitale Bildfunktionen 365
28.1 Diskretisierung 365
28.2 Rasterung 366
28.3 Quantelung 371

29 Repräsentation von Bildern 375
29.1 Speicherbedarf der Bildmatrix 375
29.1.1 Externe Bildformate 377

XI Repräsentation im Frequenzraum 379

30 Auflösung eines Bildes 379
30.1 Fourier-Reihenentwicklung 381

30.2 Fourier-Transformation (1-dimensional) 386

31 Abtasttheorem (sampling theorem) 390
31.1 Ein Beweis 391
31.2 Scheinfrequenzen: Aliasing 395
31.3 Abtasten in der Zeit 404

XII Verarbeitung im Frequenzraum 408

32 2D-Fourier-Transformation 408
32.1 2D-Fourier-Transformation: Kontinuierliche Signale 408
32.1.1 Interpretation des Spektrums 408
32.1.2 Eigenschaften 410

33 Filter 414
33.1 Linearität von Filtern 414
33.2 Filtern von Bildern 416
33.2.1 Tiefpaßfilter 416
33.2.2 Hochpaßfilter 420
33.2.3 Bandsperre 422
33.3 Audio-Filter 424

XIII Ikonische Algorithmen 425

34 Globale Bildeigenschaften 425
34.1 Statistische Merkmale 425
34.1.1 Empirischer Mittelwert und Varianz 426
34.1.2 Rauschen 428
34.1.3 Einfache Rauschfilter 432
34.1.4 Grauwerthistogramme 435
34.2 Profile und Projektionen 438

35 Geometrische Operationen auf Bildern 439
35.1 Abstandsmaße für Bildpunkte 439
35.1.1 Diskrete Metriken 439
35.1.2 Keine diskrete Metriken 441
35.2 Geometrische Transformationen 442
35.2.1 Polynomiale Koordinatentransformation 442

35.2.2 Grauwertinterpolation 445

36 Lokale Bildeigenschaften **447**
36.1 Scharfzeichnungsfilter 447

XIV Binarisieren **452**

37 Grundlagen der Segmentierung **452**
37.1 Ziele der Segmentierung 452
37.2 Uniformitätsannahme 455

38 Binarisieren **457**
38.1 Voraussetzungen für Schwellenwertverfahren 457
38.2 Schwellenwertwahl 461
38.3 Bei stark verrauschten Bildern 468

XV Komponentenanalyse **471**

39 Zusammenhang und Komponentenanalyse **471**
39.1 Zusammenhangsrelationen 471
39.2 Komponentenfärbung 473

40 Konturverfolgung: **476**

XVI Bereichsfinder **479**

41 Bereichsverschmelzung (Region Growing) **479**
41.1 Histogramverfahren 481
41.2 Teilen und Verschmelzen (Split-and-Merge) 482

42 Ballungsanalyse (Clustering) **483**
42.1 Segmentierung als Klassifikationsproblem 483
42.1.1 Der Merkmalsraum 483

XVII Kantenfinder **485**

43 Objektgrenzen und Kanten **485**

43.1 Entstehung von Kanten 485

44 Gradientenbasierte Verfahren **488**
44.1 Stufenkanten 488
44.2 Digitale Approximationen des Gradienten 491
44.2.1 Roberts-Cross-Operator 492
44.2.2 Sobel-Operator 493

45 Laplace-basierte Operatoren **496**
45.1 Der Laplace-Operator 497
45.2 Der Mexican-Hat Operator 498
45.3 Der Canny-Operator 508

XVIII Gruppierung von Kanten **512**

46 Hough-Transformation **512**
46.1 Beispiele für Parzen-Fenster-Schätzung 517
46.2 Senkrechte Geraden 522

XIX Geometrische Szenenrekonstruktion **527**

47 Geometrische Szenenrekonstruktion **527**
47.1 Tiefen- und Formhinweise 530
47.1.1 Motorische Hinweise 531
47.1.2 Scheinbare Größe 532
47.1.3 Texturgradienten 537
47.2 Shape from Contour 540
47.2.1 Rekonstruktion von Objektkonturen 540
47.2.2 Rekonstruktion von Oberflächen 546
47.2.3 Klassifikation von Konturen 549

48 Photometrische Szenenrekonstruktion **552**
48.1 Photometrische Bildentstehung (image formation) 552
48.1.1 Messen von Strahlungswerten 554
48.1.2 Modellierung des Sensors 557
48.1.3 Modellierung der Oberfläche 561
48.1.4 Beispiele für Reflexionsfunktionen 564
48.1.5 Die Reflektanzkarte 567
48.2 Photometrisches Stereo 571

48.3 Shape from Shading 573

49 Das Generalitätsprinzip **578**
49.1 Grundlegende Annahmen 578
49.2 Szenenrekonstruktion als Bootstrap-Problem 590

XX Mehrfache Objektansichten **594**

50 Bewegungsstereo und optischer Fluß **594**
50.1 Rekonstruktion starrer Körper 594
50.2 Bewegungsfeld einer Translation 603

51 Das Korrespondenzproblem bei Bildfolgen **609**
51.1 Bewegungsfeld vs. optischer Fluß 609

52 Stereoskopie **614**
52.1 Binokulares Sehen 614
52.2 Stereoskopische Tiefenrekonstruktion 620
52.3 Zufallspunktstereogramme 629
52.4 Autostereogramme 635

53 Das Korrespondenzproblem **639**
53.1 Das Korrespondenzproblem 643
53.2 Geometrische Einschränkungen aus der Bildentstehung . . 647
53.3 Die Ausnutzung von Objekteigenschaften 658
53.4 Zusammenfassung 667

XXI Bildverstehen **670**

54 Objekterkennung **670**
54.1 Ansätze zur Objekterkennung 670
54.2 Gemeinsame Eigenschaften 672
54.3 Häufigkeitsanalysen 673
54.4 Schablonentest 676

54.5 Relationalstrukturen 686

XXII Übungen: Werkzeuge, Handbücher und Beispiele 700

55 Die Syntax von POV-Ray 700
55.1 Definition von Standardobjekten 700
55.1.1 Kugel 701
55.1.2 Zylinder 704
55.1.3 Kegel 705
55.1.4 Torus 707
55.1.5 Ebenen 708
55.1.6 Blöcke und Quader 710
55.2 Kontrollstrukturen in POV-Ray 711
55.2.1 Wiederholung: While-Schleife 711
55.3 Vektorfunktionen 713

56 Strukturierung von Povray-Programmen 716
56.1 Bibliotheken 716
56.1.1 Definition von Objektmodellen 720
56.2 Macros und Funktionen 724
56.3 Datenstrukturen 727
56.3.1 Arrays 727
56.4 Fallunterscheidungen 730
56.4.1 Synchronisation einer periodischen Bewegung mit der Musik 733

57 Video-Schnitt 736
57.1 Eine Folge von Einzelbildern zum Quicktime Movie verbinden 736
57.2 iMovie 738
57.3 Final cut pro 738
57.4 Quartz composer 738

58 Vertonen 739
58.1 Garageband 739

58.2 Finale 739

59 Bildverarbeitung **739**

60 Zur Prüfungsvorbereitung: Verständnisfragen **740**

Index **743**
Allgemeiner Index 743

Glossary **759**

Literatur **772**

Das Modul IVC: Das Modul „IVC Interactive Visual Computing“verknüpft die gemeinsamen visuellen Aspekte der Teilbereiche „Computer Vision“ und „Computer Graphics“mit der darüber hinaus gehenden Interaktivität der „Real-time Interactive Media“. Entsprechend dieser Grundidee führt das Modul in die wesentlichen Aspekte der Bildverarbeitung, der Computergrafik und echtzeitfähiger, interaktiver Systeme ein. Es werden die konzeptionellen und mathematischen Grundlagen, sowie Algorithmen und Werkzeuge für die Erstellung und das Verstehen grafischer Darstellungen behandelt. Im interaktiven Teil behandelt das Modul die Mensch-Computer Interaktion von der maschinellen Seite und befasst sich darüber hinaus mit Aspekten von Echtzeitsystemen und Echtzeit Computergrafik. Lernziele sind Kenntnisse der mathematischen und technischen Grundlagen und Erfordernisse der Bildverarbeitung und Bilderzeugung für statische und dynamische, interaktiv erzeugte Bilder, sowie der Methoden der geometrischen, photometrischen und dynamischen Modellierung und deren Anwendungen in der Bildverarbeitung, Computergrafik, Echtzeit-Computergrafik sowie Methoden zur Erzeugung Virtueller Realität.

Sie haben sich mit diesem Modul für ein vielseitiges und spannendes Teilgebiet der Informatik entschieden. Uns hat die Vorbereitung dieser Unterlagen zur Vorlesung viel Spaß gemacht, und ich wünsche mir, daß sich meine Begeisterung für die Bildverarbeitung und Computergrafik auf Sie übertragen wird. Ich freue mich auf unsere Zusammenarbeit und wünsche mir, daß Sie mit Spaß, Engagement und Erfolg dabei sein werden und die Faszination der virtuellen Welten kennenlernen und erleben können.
Viel Vergnügen in den virtuellen Welten!

Steven Klug und Omar Nuri haben beim Scannen der Bilder geholfen. Jochen Scholz und Fredrik Winkler haben viele wertvolle Anregungen und Korrekturhinweise für dieses Skript gegeben. Meinen herzlichen Dank an alle!
Hamburg, im Oktober 2012

Ihre Leonie Dreschler-Fischer

IVC Interactive Visual Computing

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 1:

Einführung

Universität Hamburg
Fachbereich Informatik

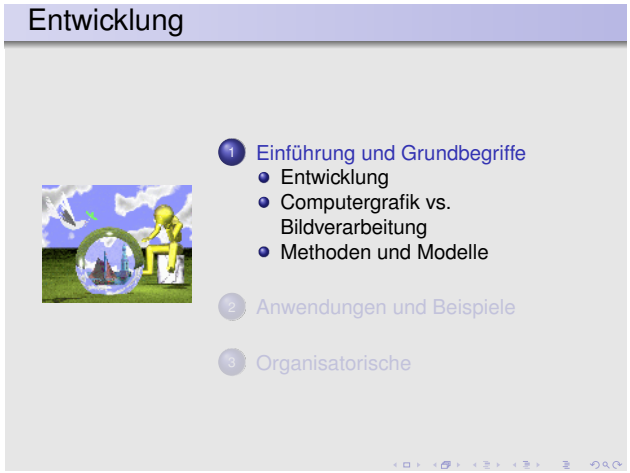
WS 2014/2015

Teil I

Einführung

1 Einführung und Grundbegriffe

1.1 Entwicklung



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. . .)

✎ Anmerkung: **Vektorgrafik versus Rastergrafik:**

Vektorgrafiken sind Linienzeichnungen, bei denen Objekte durch ihre Konturen repräsentiert werden. Helle und dunkle Bildbereiche werden wie in technischen Zeichnungen durch Schraffuren abgesetzt. Die dazugehörigen Datenstrukturen beschreiben punktförmige oder linienhafte grafische Primitive, wie Geraden, Kreise, Vektoren usw. Ausgabegeräte zeichnen diese Primitive direkt nach, entweder mit dem Elektronenstrahl auf dem Bildschirm oder mit Tuschestiften auf dem Papier. Die Ausgabezeit für Vektorgrafiken ist daher ungefähr proportional zur Länge der zu zeichnenden Linien, ebenso der Speicheraufwand für die Grafiken.

Bei der *Rastergrafik* entsteht das Bild als ein Mosaik von Rasterpunkten unterschiedlicher Farbe und Helligkeit. Das Bild wird in einem Bildwiederholspeicher zwischengespeichert und zeilenweise auf das Papier oder den Bildschirm übertragen. Druckzeit und Speicheraufwand (für unkomprimierte Daten) sind proportional zu den Abmessungen des Bildes und zur Dichte der Rasterpunkte (Auflösung). Mit hochauflösender Rastergrafik können fotorealistische Bilder erzeugt werden.

Zur Aussagekraft von Bildern

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.

1.2 Computergrafik vs. Bildverarbeitung

Computergrafik

Definition: 1 (*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

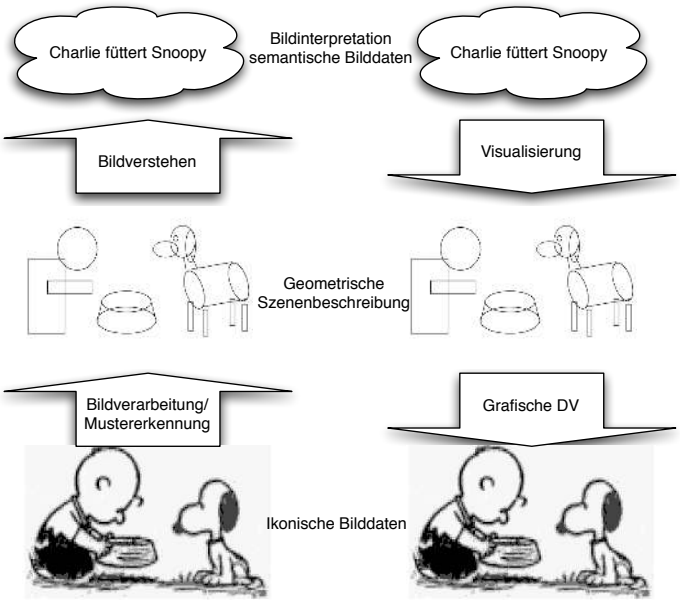
Definition: 2 (*Grafische Datenverarbeitung*)
Die grafische Datenverarbeitung umfaßt Verfahren zur Eingabe und Manipulation von Bildbeschreibungen sowie deren Umwandlung in grafische Darstellungen.

Probleme:

Grafische Datenverarbeitung: Komplexität des Problems

Bildverstehen: Mehrdeutigkeit von Bildern

Prozesse der grafischen DV



✎ **Anmerkung:** Ebenen der Bildbeschreibung

Die vorherige Folie zeigt verschiedene Ebenen der Bildbeschreibung, sowie die Verarbeitungsprozesse, die zwischen diesen Ebenen die Repräsentationen transformieren:

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?

Die Teilgebiete der Computergrafik betreffen unterschiedliche Ebenen dieser Hierarchie und unterschiedliche Verarbeitungsrichtungen innerhalb der Hierarchie.

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.

Visualisierung ist die Aufgabe, eine gegebene Bedeutung in einem geeigneten Bild darzustellen, während es beim Bildverstehen darum geht, die Bedeutung einer abgebildeten Szene zu erfassen, d.h. die Situation zu verstehen, in der sich die abgebildeten Personen oder Objekte befinden.

Verarbeitungsrichtung

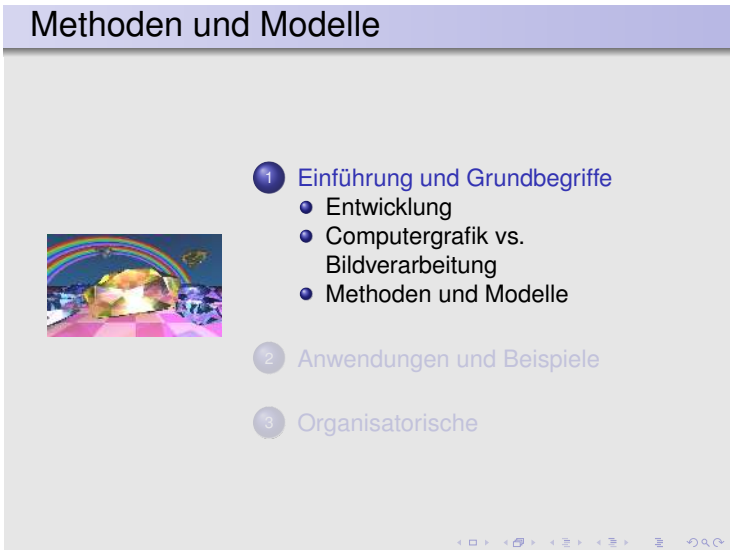
- Die Prozesse der graphischen Datenverarbeitung sind also im Wesentlichen *„top-down“* Prozesse — von der Beschreibung zum Bild,
- während die Prozesse des Bildverstehens *„bottom up“* Prozesse sind — vom Bild zur Bildbeschreibung.
- Beide Teilgebiete der Computergrafik nutzen als gemeinsames methodisches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

Probleme

Bildverstehen und Computergrafik haben mit sehr unterschiedlichen Schwierigkeiten zu kämpfen:

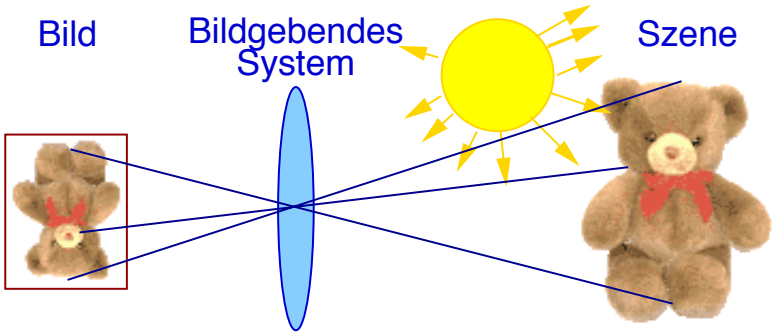
- Das größte Problem beim *Bildverstehen* ist die mangelnde Eindeutigkeit: Jedes Bild kann das Abbild von unendlich vielen verschiedenen Szenen sein.
- Bei der *grafischen DV* dagegen ist in der Regel die Lösung eindeutig, aber die Aufgabe besteht darin, Algorithmen zu entwickeln, die diese eindeutigen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

1.3 Methoden und Modelle



Bildentstehung

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



$$\text{Bild} \xRightarrow{\textit{Bildverstehen}} \text{Szenenbeschreibung}$$

Computergrafik

$$\text{Bild} \xLeftarrow{\textit{Bildsynthese}} \text{Szenenbeschreibung}$$

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.

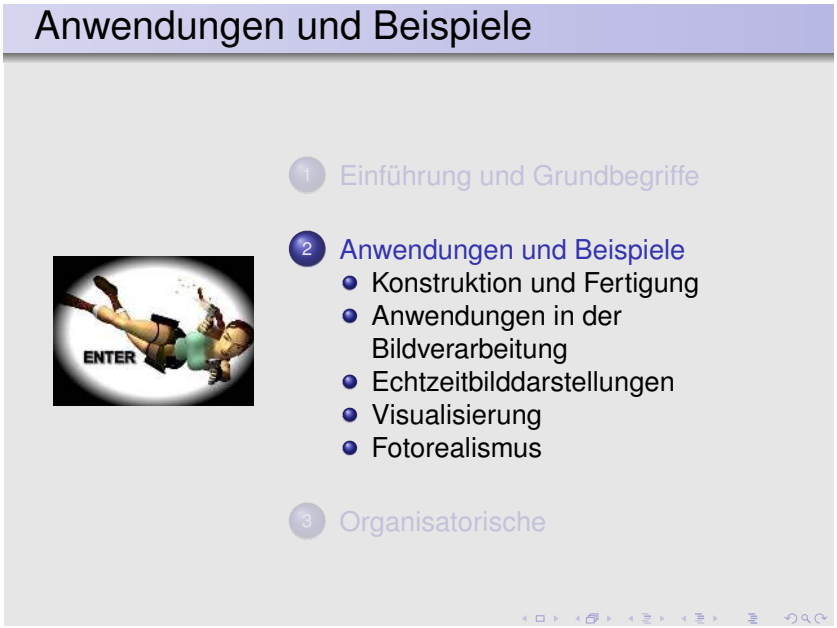
Digitalisierungsmodelle beschreiben den Prozeß der Diskretisierung von analogen Bildfunktionen.

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).

2 Anwendungen und Beispiele

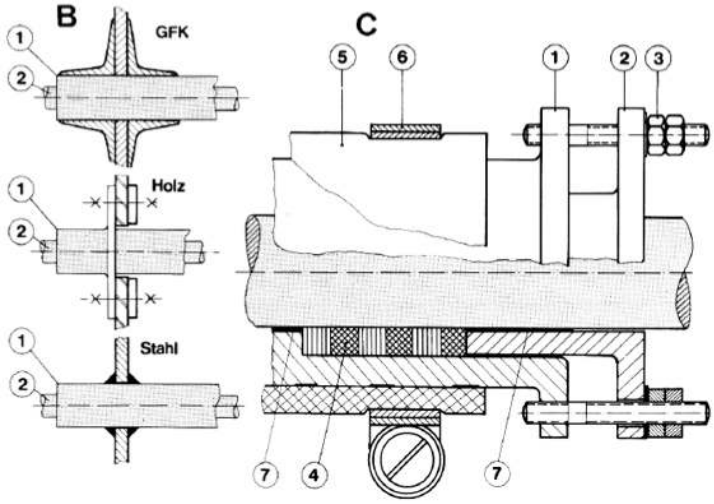
2.1 Konstruktion und Fertigung



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:

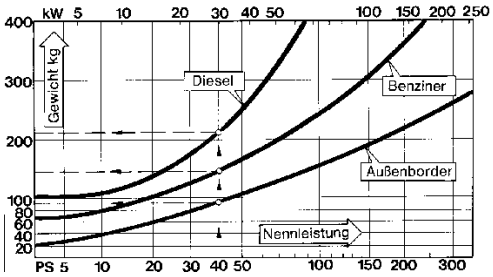


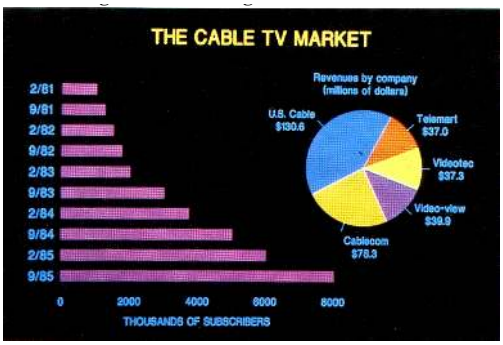
✎ **Anmerkung:** Technische Zeichnungen waren eine der ersten Anwendungen der Computergrafik. Die Programmierung und Arbeitsweise der klassischen Plotter als Ausgabegeräte lehnte sich eng an die Arbeit mit Zeichenbrett und Schablone an. Inzwischen steht nicht mehr das Zeichnen im Vordergrund, sondern der Rechner als Werkzeug zur Konstruktion und Steuerung des Fertigungsprozesses, *computer aided design, computer aided manufacturing* (CAD/CAM).

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.

2.2 Anwendungen in der Bildverarbeitung

Anwendung: Bildverarbeitung:

- Bildrestauration
- Robotik,
- Fernerkundung,
- Qualitätskontrolle, Überwachung

Bildrestauration

Beseitigung von Bewegungsunschärfe

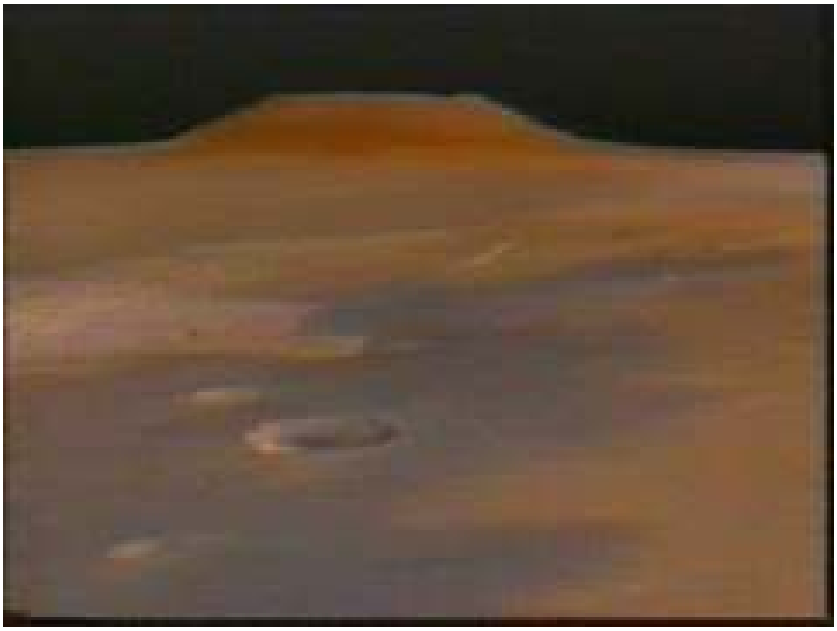
✎ **Anmerkung:** Zur Beseitigung der Bewegungsunschärfe wurde in diesem Beispiel die inverse Wiener Filterung verwendet, siehe Gonzales and Wood, 2008.

2.3 Echtzeitbilddarstellungen

Echtzeitbilddarstellungen:

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

Fernerkundung



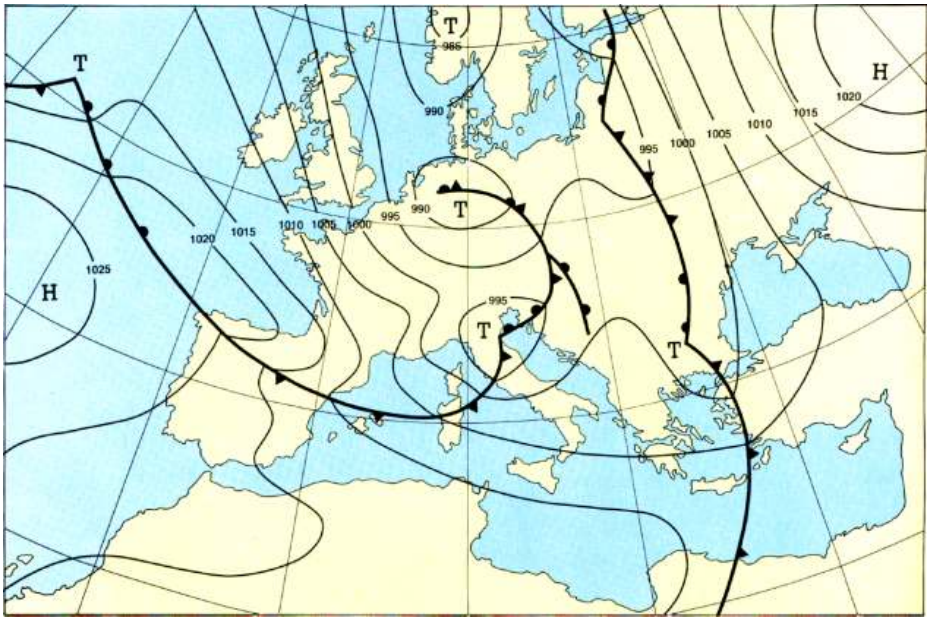
Simulierter Flug über den Mars (Olympus Mons)

2.4 Visualisierung

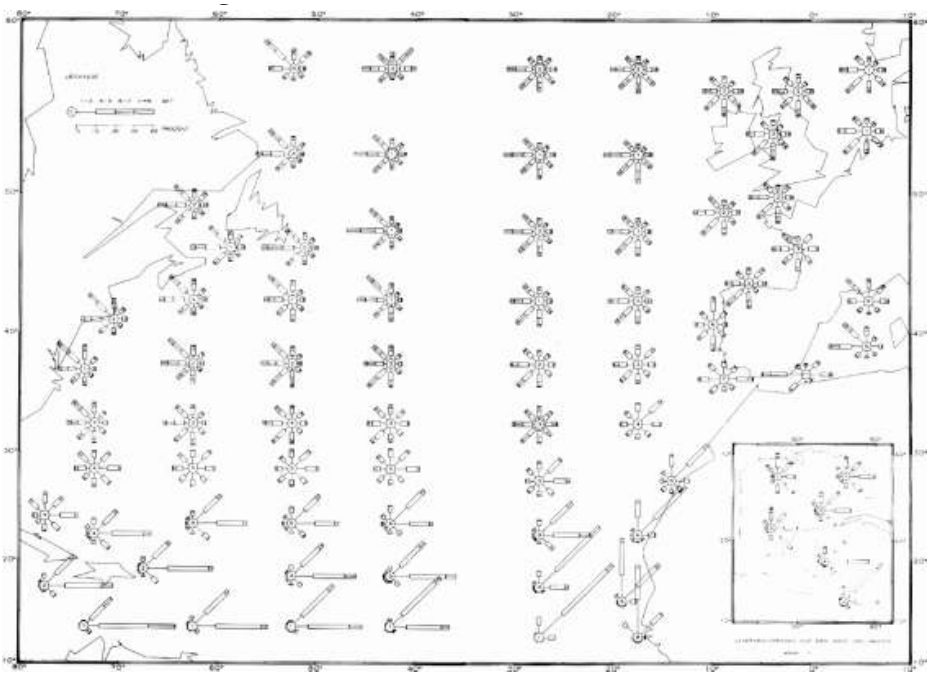
Visualisierungen

- 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



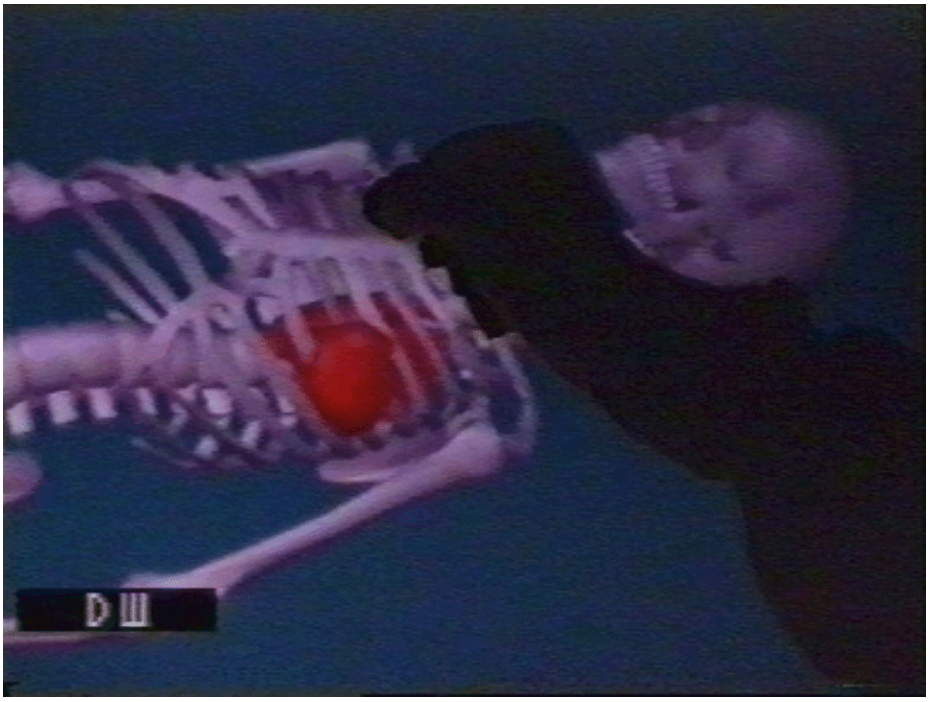
Visualisierung der Windverhältnisse



2.5 Fotorealismus

Visualisierung von Strömungen

Visualisierung von medizinischen Daten

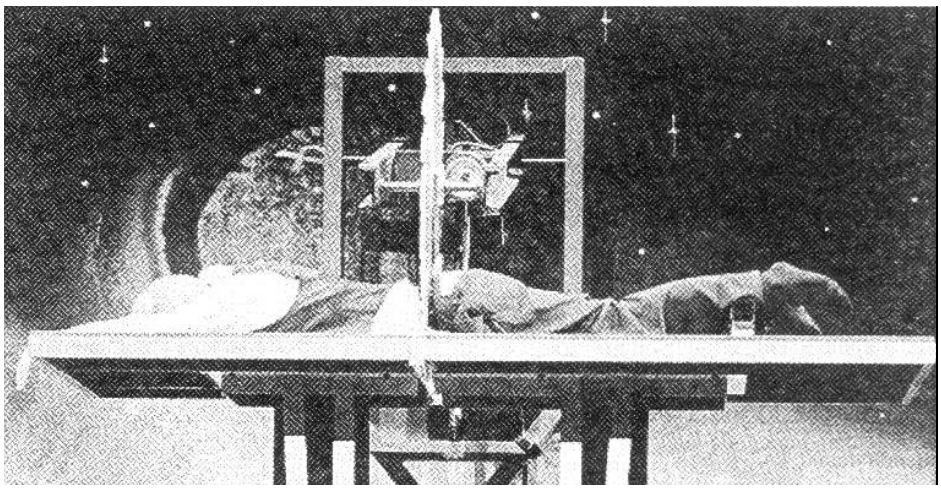


Fotorealismus

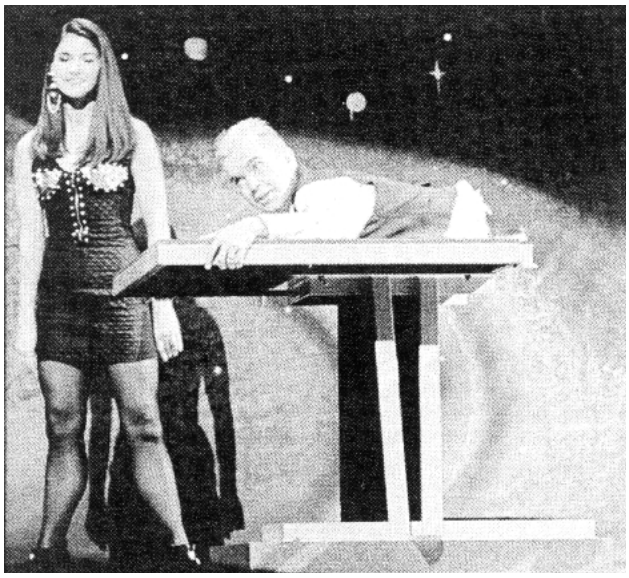
Erzeugung von Bildern oder Filmen, die von natürlichen nicht zu unterscheiden sind.

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

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



Nachher:



✎ **Anmerkung:** Früher wurden Zauberkunststücke mit Spiegeln durchgeführt; heute können solche Effekte mit Computern simuliert werden.

Animationen für Computerspiele

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.

✎ **Anmerkung:** Beispiele für digital nachbearbeite Spielfilme:

Die digitale Aufnahme von Spielfilmen ist heutzutage auch für solche Filme sinnvoll, die keine spektakulären Tricks erfordern. Ca. 20% der Aufnahmen im Film „Der erste Ritter“ wurden digital nachbearbeitet. Auf diese Weise wurden die Kosten für Kulissen und Statisten drastisch gesenkt: In einer Szene ist ein gewaltiges Ritterheer zu sehen, das einen Abhang hinunterstürmt. Diese Szene wurde mit wenigen Statisten aufgenommen, die dann mehrfach in den Film hineinkopiert wurden. Ebenso wurden für ein Zeltlager nur wenige Zelte aufgebaut, die dann durch Bildverarbeitung vervielfältigt wurden.

Die eindrucksvollen Burganlagen sind heute nur noch Ruinen. Fehlende Teile wurden im Computer ergänzt.

Bei den dramatischen Schwertkämpfen halten Sean Connery und Richard Gere im Nahkampf nur die Hefte der Schwerter in den Händen. Die scharfen, blitzenden Klingen wurden erst im Computer hinzugefügt. Das senkt die Verletzungsgefahr und die Versicherungskosten für die Stars.

Nachtaufnahmen mit Lagerfeuern und Fackelschein wurden teilweise bei Tage aufgenommen und erst später am Rechner als Nachtaufnahmen gestaltet.

Eine wichtige Anwendung von Bildverarbeitung ist die Retouche: Zu Zeiten von König Artus gab es noch keine Überlandleitungen oder asphaltierten Straßen: Solche Spuren des 20. Jahrhunderts lassen sich am Rechner leicht aus historischen Szenen tilgen.

Im Film „Der englische Patient“ ist ein spektakulärer Flugzeugabsturz in einer kargen Wüstenlandschaft zu bewundern. Der Flugzeugabsturz wurde auf dem Studiogelände gedreht und die Wüste und der blaue Himmel vom Rechner simuliert.

Die Bildverarbeitung hilft auch bei der täuschend echten Verbindung von Modellen, Animationen und natürlichen Szenen, wie beispielsweise im Film „Independence Day“. Es folgen Ausschnitte aus „Independence Day“ ? Regie: Roland Emmerich ? Drehbuch: Dean Devlin und Roland Emmerich ? Kamera: Karl Walter Lindenlaub ? Schnitt: David Brenner ? Musik: David Arnold) von 1996 und „Der erste Ritter“ (Regie: Jerry Zucker - Drehbuch: Lorne Cameron, David Hoselton und William Nicholson - Musik: Jerry Goldsmith - Kamera: Adam Greenberg Schnitt: Walter Murch) von 1995.

Retouche von Zivilisationsspuren

Kaschieren von Ruinen

Entschärfen gefährlicher Szenen

Entschärfen gefährlicher Szenen

Entschärfen gefährlicher Szenen

Massenszenen

Vervielfältigung von Statisten

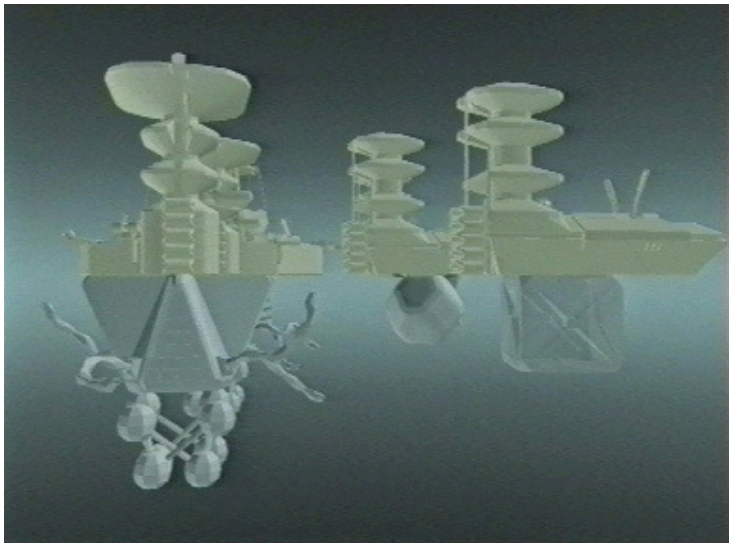
Vervielfältigung von Kulissen

Einbringen von Modellen

Virtuelle Kulissen

Die Darsteller agieren im leeren Raum vor einem neutralen Hintergrund. Die Kulissen werden in Echtzeit virtuell generiert und im Blue-Screen-Verfahren hinzugefügt.

Eine virtuelle Welt, Computerkunst



aus dem Film ‘Neulich in Flockville“ von Christine Heubi

3 Organisatorische

3.1 Übungsaufgabe

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/
- MacMegaPOVhttp://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.

Zwei Beispiele für erfolgreiche CGB/IVC-Filme:

- http://kogs-www.informatik.uni-hamburg.de/~dreschle/informatik/Abschlussfilme_CGB/Pages/2006_The_MaKink_Of.html
- http://kogs-www.informatik.uni-hamburg.de/~dreschle/informatik/Abschlussfilme_CGB/Pages/2006_Fermats_letzter_Satz.html

3.2 Literatur

Online-Materialien

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
- Beispielfilme aus den Übungen: http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Lehre/Entries/2009/12/23_Studentische_Filme_zu_CGB_und_IVC.html

3.2.1 Literaturempfehlungen zu Bildverarbeitung und Bildsynthese

- Zu den Grundlagen der Bildentstehung siehe:
  - (Glassner, 1995): Sehr ausführliche physikalische Modellierung der Bildsynthese.
  - (Fellner, 2004): Eine knappe, leichte Einführung
  - (Nischwitz and Haberäcker, 2004): Ein ausführliches Vorlesungsskript mit vielen Beispielen zur Echtzeitgrafik in OpenGL.

- Zu den Grundlagen der Bildverarbeitung siehe:
  - (Steinmüller, 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 (Nur für Angehörige der Universität Hamburg: Volltextzugang von ausserhalb des Campus mit Bibliotheksausweis)

    ✎ **Anmerkung:** Vervielfältigungen (z.B. Kopien, Downloads) sind nur von einzelnen Kapiteln oder Seiten und nur zum eigenen wissenschaftlichen Gebrauch erlaubt. Keine Weitergabe an Dritte. Kein systematisches Downloaden durch Robots.
  - Aufnahmetechnik und Darstellung: (Reinhard et al., 2010)
  - (Gonzales and Wood, 2008): Eine gute Einführung in die ikonische Bildverarbeitung
  - (Bovik, 2009): Ein umfassendes Handbuch zur ikonischen Bildverarbeitung.
  - (Cyganek and Siebert, 2009): Eine Einführung in die Algorithmen zur geometrischen Szenenrekonstruktion.
  - Grundlagen der Bildverarbeitung: (?), (Sonka et al., 1999): Eine gute Übersicht zu Bildverarbeitung und Bildverstehen.
  - (Davies, 1997): Grundlagen des Bildverstehens, besonders gut auf die Verbindung von Computergrafik und Bildverarbeitung abgestimmt.
  - (Horn, 1986): Eine ausgezeichnete Darstellung der klassischen Algorithmen zur Szenenrekonstruktion.
  - (Duda et al., 2001): Eine ausgezeichnete Darstellung der klassischen Algorithmen zur Mustererkennung und Bildvorverarbeitung.

Spieleprogrammierung: (Davison, 2005)

IVC Interactive Visual Computing

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 2:

Geometrische Grundlagen: Koordinaten und Transformationen

Universität Hamburg
Fachbereich Informatik

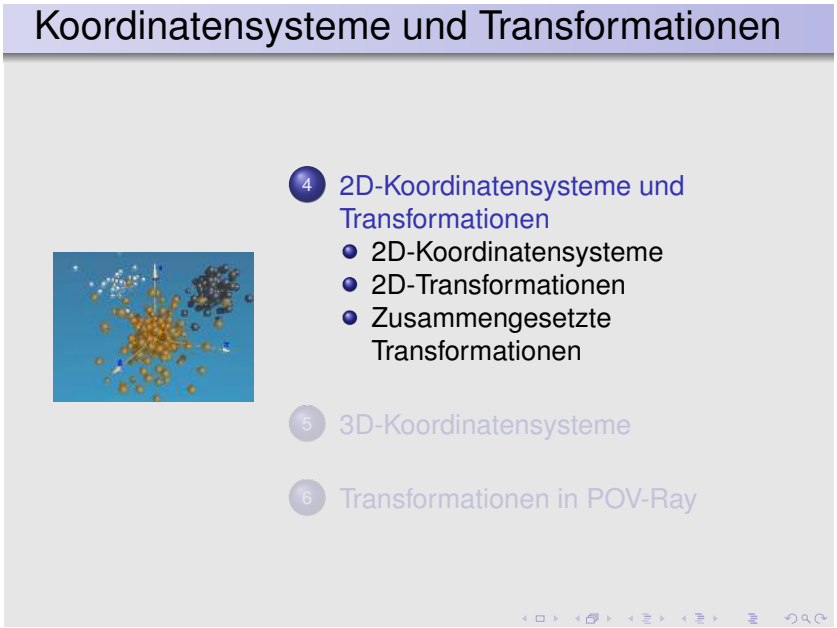
WS 2014/2015

Teil II

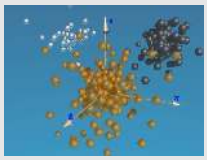
Geometrische Grundlagen: Koordinaten und Transformationen

4 2D-Koordinatensysteme und Transformationen

4.1 2D-Koordinatensysteme



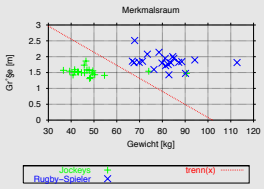
Anwendungen von Koordinatensystemen:



Ein 3D-Koordinaten-system

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

2D-Koordinatensysteme



Ein 2D-Koordinatensystem

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

Gebräuchlich sind:

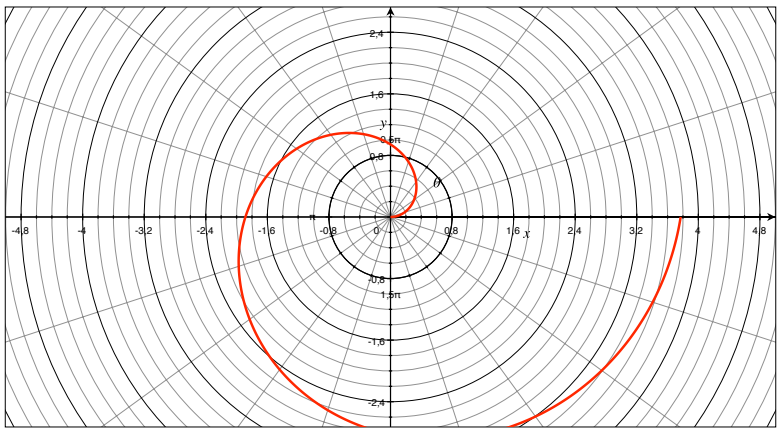
Kartesische Koordinaten: Ursprung links unten, x-Achse nach rechts, y-Achse nach oben.

Polarkoordinaten: (für mathematische Anwendungen).

☞ Beachte: In einigen Bildverarbeitungsanwendungen werden anstelle der kartesischen Koordinaten Zeilen- und Spaltennummern verwendet. Der Ursprung ist dann links oben, die Zeilennummern wachsen nach unten, die Spaltennummern von links nach rechts. Beim Umrechnen von Zeilennummern in y-Koordinaten sind also der Vorzeichenwechsel und die unterschiedliche Lage des Ursprungs zu beachten.

Beispiel:

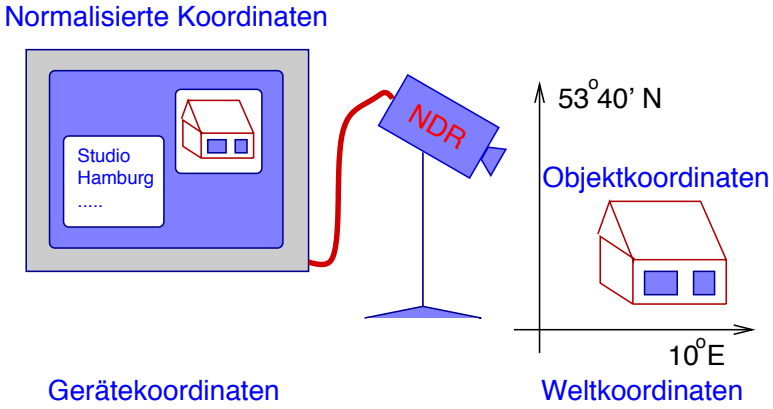
Eine Spirale in Polarkoordinaten



Spezifische Koordinatensysteme

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

Hierarchie der Koordinatensysteme



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.

4.2 2D-Transformationen

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).

4.2.1 2D-Translation

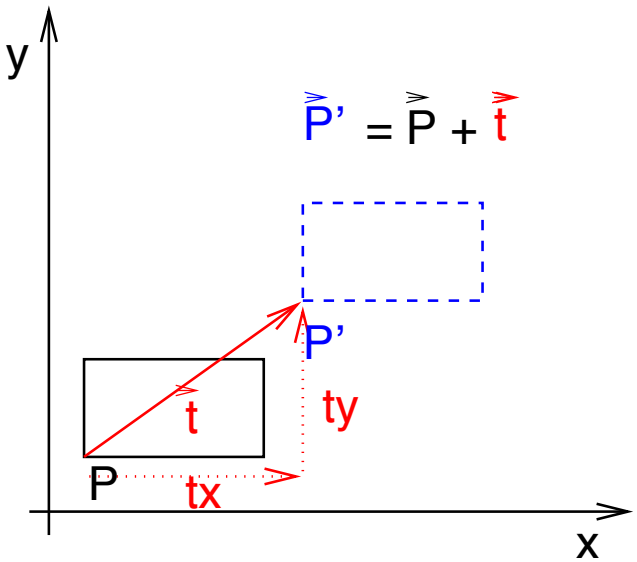
Translation

Definition: 3 (***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 $\vec{t} = \vec{0}$.

4.2.2 2D-Skalierung

Skalierung

Definition: 4 (*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: $s_x = s_y$.

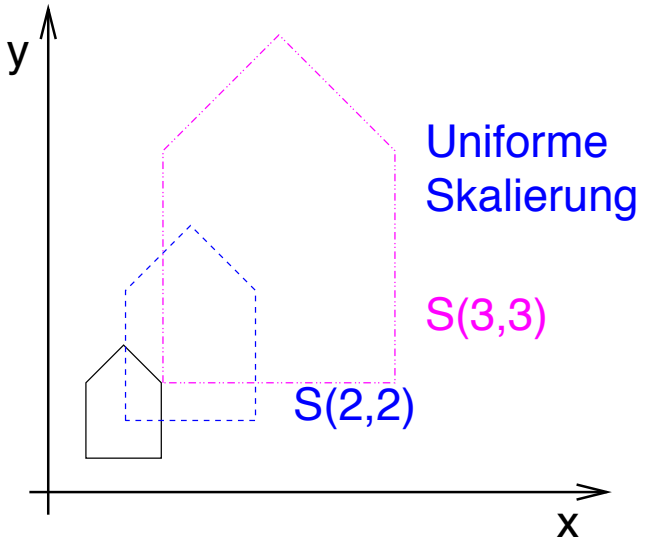
Beliebiger Fixpunkt: c_x, c_y

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

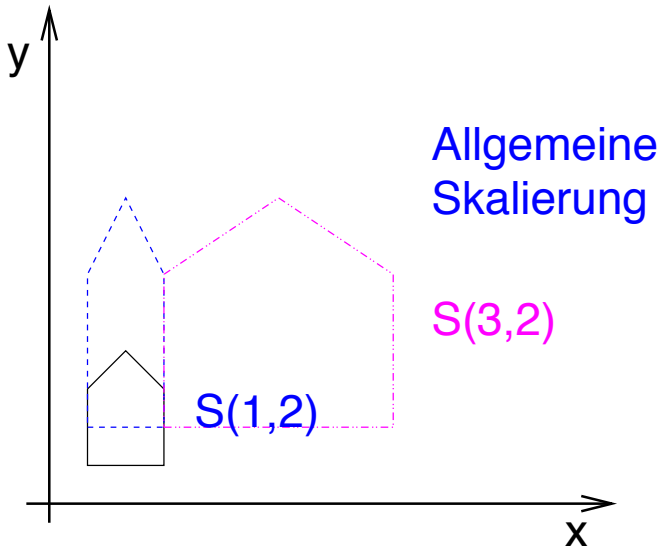
Skalierung um den Fixpunkt: $s(s_x, s_y)$,

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

Beispiel: Uniforme Skalierung



Allgemeine Skalierung



4.2.3 2D-Rotation

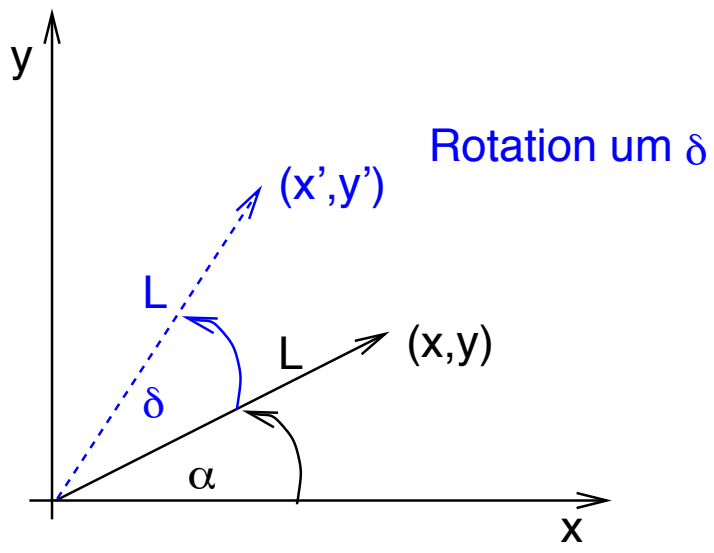
Rotation

Definition: 5 (***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:

$$
\begin{aligned}
\sin(\alpha + \delta) &= \frac{y'}{L} = \cos(\delta)\sin(\alpha) + \sin(\delta)\cos(\alpha) \\
y' &= L \cdot \cos(\delta) \cdot \frac{y}{L} + L\sin(\delta)\frac{x}{L} \\
y' &= y \cdot \cos(\delta) + x \cdot \sin(\delta) \\
\cos(\alpha + \delta) &= \frac{x'}{L} = \cos(\delta)\cos(\alpha) - \sin(\delta)\sin(\alpha) \\
x' &= L \cdot \cos(\delta) \cdot \frac{x}{L} - L\sin(\delta)\frac{y}{L} \\
x' &= x \cdot \cos(\delta) - y \cdot \sin(\delta)
\end{aligned}
$$

☞ Beachte: Die rotierten Koordinaten (x', y') lassen sich allein durch Angabe von (x, y) und δ berechnen, α braucht nicht angegeben zu werden.

Die drei folgenden Grafiken wurden mit gnuplot gezeichnet. Sie zeigen, wie 2D-Transformationen genutzt werden können, um grafische Muster zu erzeugen.

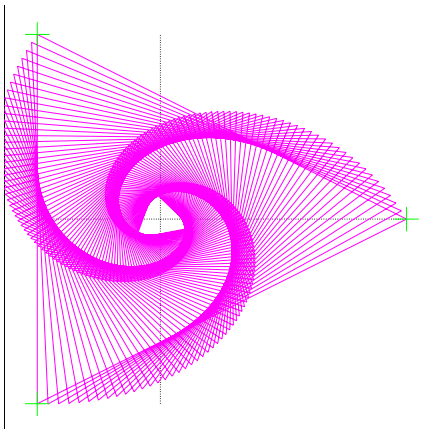
Das erste Bild enthält 100 Dreiecke, wobei jedes Dreieck gegenüber dem vorherigen um den Faktor 0.98 (im Ursprung) skaliert und um $\frac{\pi}{80}$ Grad gedreht wurde.

Das zweite Bild ist genauso entstanden, nur wurde als Ausgangsfigur ein Quadrat gewählt.

Für das dritte Bild wurde das Dreieck des ersten Bildes sechsmal um jeweils 60 Grad rotiert nebeneinandergesetzt. Zusätzlich wurden nur zwei der drei Dreiecksseiten gezeichnet.

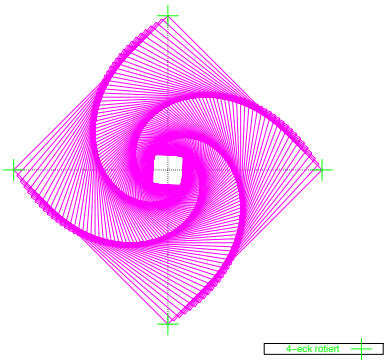
Beispiel:

Rotation und Skalierung von Dreiecken



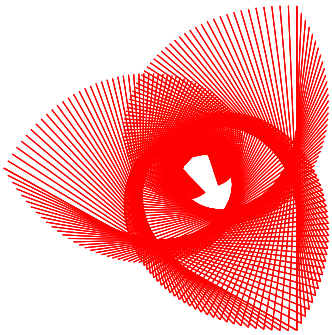
Beispiel:

Rotation und Skalierung von Quadraten



Beispiel:

Sechs rotierte Kopien der rotierten Dreiecke



4.3 Zusammengesetzte Transformationen

Zusammengesetzte Transformationen

Bei der Abbildung von Weltkoordinaten auf Gerätekoordinaten 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

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

Rotation

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

$$\begin{aligned} y' &= y \cdot \cos(\delta) + x \cdot \sin(\delta) \\ x' &= x \cdot \cos(\delta) - y \cdot \sin(\delta) \end{aligned}$$

Rotation um δ :

$$\begin{aligned} [x'y'1] &= [xy1] \cdot R(\delta) \\ R(\delta) &= \begin{bmatrix} \cos(\delta) & \sin(\delta) & 0 \\ -\sin(\delta) & \cos(\delta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Aufbau der Rotationsmatrix:

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

- 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: 6 (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 $(A \cdot B \cdot C)$ für alle Punkte genutzt werden.

Satz: 7 (Kommutativgesetz)
Das Kommutativgesetz *gilt meistens nicht:*

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

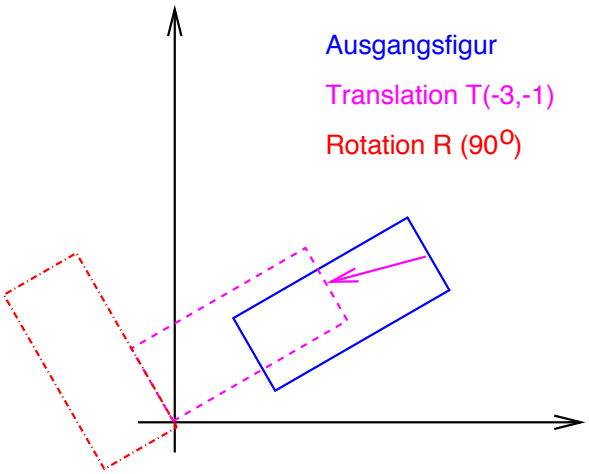
nur innerhalb derselben Gruppe von Transformationen (z.B. eine Folge von Rotationen oder Skalierungen) gilt:

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

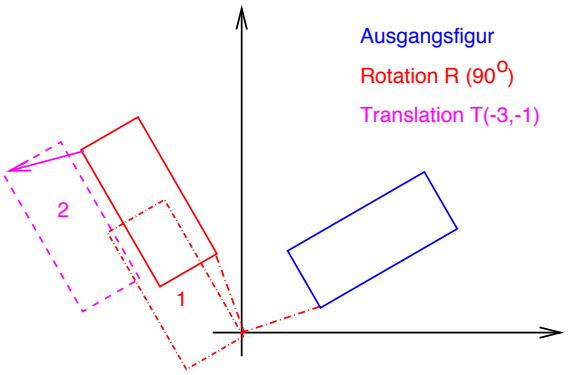
.

✎ **Anmerkung:** Die beiden folgenden Folien zeigen, daß die Hintereinanderausführung von Transformationen nicht unbedingt kommutativ ist. Im ersten Bild wird ein Rechteck zunächst mit der Translation $T(-3, -1)$ in den Ursprung verschoben und dann um 90° gedreht. Im zweiten Bild ist die Reihenfolge der Transformationen vertauscht: Zuerst erfolgt die Drehung und dann die Translation. Das Ergebnis ist unterschiedlich, wie man leicht sieht.

Erst Translation, dann Rotation



Erst Rotation, dann Rotation



Weitere Rechenregeln:

$$
\begin{aligned}
T(t_{x1}, t_{y1}) \cdot T(t_{x2}, t_{y2}) &= T(t_{x1} + t_{x2}, t_{y1} + t_{y2}) \\
S(s_{x1}, s_{y1}) \cdot S(s_{x2}, s_{y2}) &= S(s_{x1} \cdot s_{x2}, s_{y1} \cdot s_{y2}) \\
R(\delta_1) \cdot R(\delta_2) &= R(\delta_1 + \delta_2)
\end{aligned}
$$

Beliebige Fixpunkte:

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: 8 (*Affine Transformation*)
Eine Transformation heißt affin, wenn die Transformationsmatrix A die folgende Form hat:

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

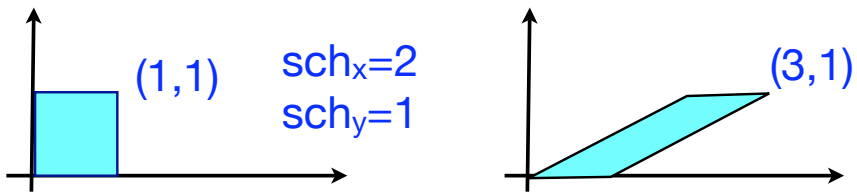
Alle bisher eingeführten Transformationen sind affine Transformationen. Weitere affine Transformationen:

- Scherung, Spiegelung, Zentralprojektion (3D).

Scherung

Scherung in x und y: $\text{sch}_x, \text{sch}_y$:

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



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.

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

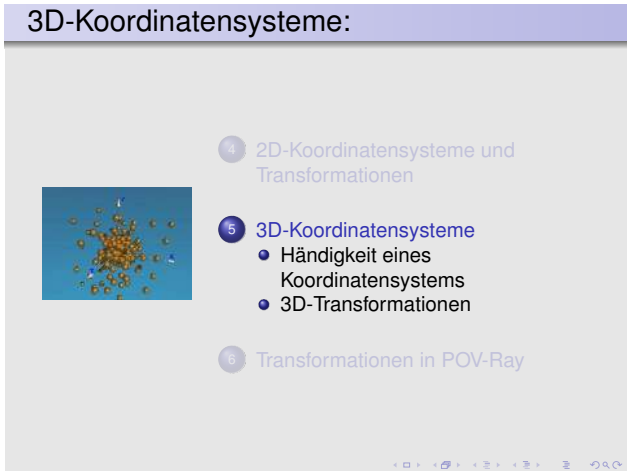
- Der Rechenaufwand für die Anzeige ist konstant und unabhängig von der Zahl der Transformationen.

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.

5 3D-Koordinatensysteme

5.1 Händigkeit eines Koordinatensystems



Händigkeit eines Koordinatensystems

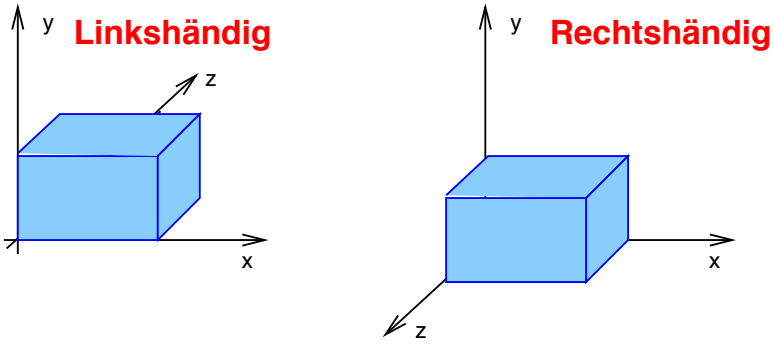
Es gibt zwei Möglichkeiten, die kartesische XY-Ebene durch eine Z-Koordinate (senkrecht auf der XY-Ebene) zu einem dreidimensionalen Koordinatensystem zu erweitern.

☞ **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



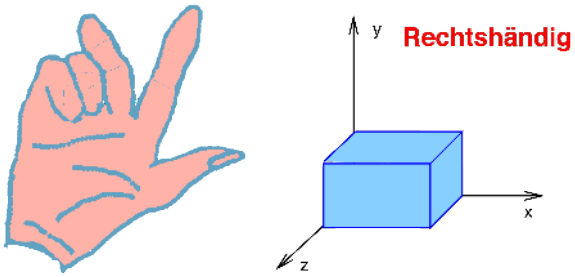
✎ **Anmerkung:** In der amerikanischen Computergrafikliteratur wird meist das linkshändige System verwendet, in der deutschen das rechtshändige.

Es gibt mehrere Eselsbrücken, wie man sich die Händigkeit eines Koordinatensystems merken kann. Die einfachste Regel besteht darin, die Finger einer Hand in der Reihenfolge Daumen (X), Zeigerfinger (Y), Mittelfinger (Z) mit den Achsen des Koordinatensystems gleichzusetzen. Wenn man jetzt die Finger so spreizt, daß ein rechtwinkliges Dreibein entsteht, dann bilden die Finger der linken Hand ein linkshändiges Koordinatensystem, die Finger der rechten Hand ein rechtshändiges.

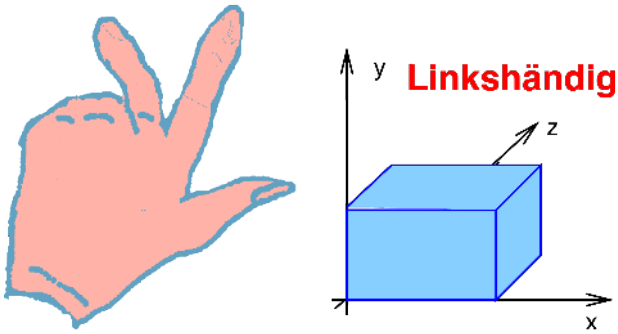
Eine andere Eselsbrücke besteht darin, sich vorzustellen, daß man mit den Fingern einer Hand die Z-Achse umfaßt, so daß der Daumen in die positive Z-Richtung zeigt. Dann ruhen die Fingerwurzeln auf der X-Achse und die Fingerspitzen zeigen zur Y-Achse. Je nachdem, ob man dieses Gedankenexperiment mit der linken oder rechten Hand durchführt, erhält man ein linkshändiges oder ein rechtshändiges Koordinatensystem.

Das POV-Ray-System, mit dem die Beispielen auf diesen Folien gerechnet wurden, verwendet normalerweise ein linkshändiges Koordinatensystem, aber die Händigkeit ist ein einstellbarer Parameter. Die hier angegeben Skripte beziehen sich auf das linkshändige Koordinatensystem.

Eselsbrücke: Rechtshändig



Eselsbrücke: Linkshändig



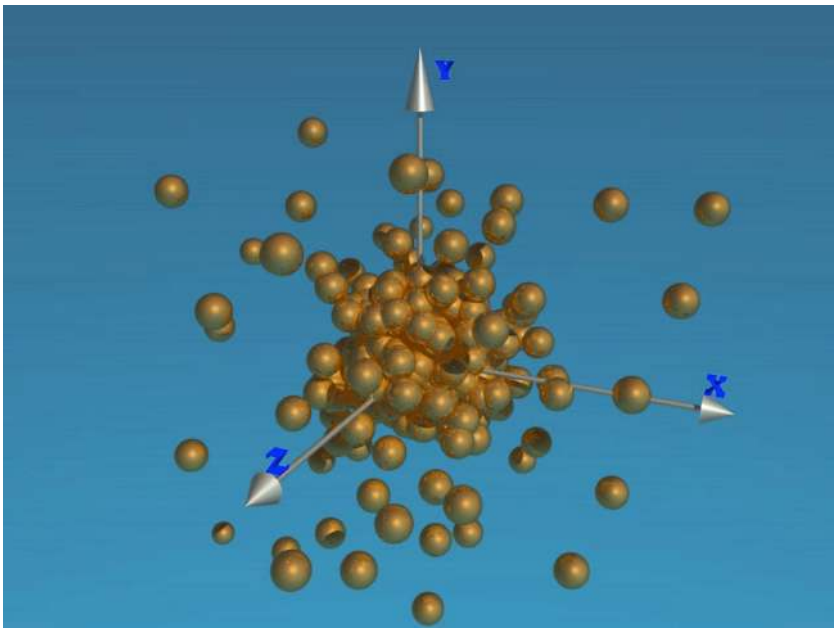
5.2 3D-Transformationen

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



✎ **Anmerkung:** Dieses Bild wurde durch ein POV-Ray-Skript erzeugt, bei dem ein Objektmodell "stardot" definiert wurde — eine Kugel im Ursprung mit dem Radius 0.7 und der Textur `T_Gold_1B` aus dem Standardkatalog `textures.inc` (gold-metallic).

Diese Kugel wurde an hundert verschiedenen Stellen wiederverwendet, wobei als einzige Transformation die Translation verwendet wurde. Translationen können durch das Schlüsselwort `translate`, gefolgt vom Translationsvektor angegeben werden. Vektoren können durch ihre Komponenten bezeichnet, durch einen definierten Namen bezeichnet oder durch Ausdrücke angegeben werden.

5.2.1 Transformationen in POV-Ray

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: 9
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>}
 ...
```

✎ **Anmerkung:** Es folgt der Rest des POV-Ray-Skripts, mit dem das Bild erzeugt wurde. Die Koordinatenachsen wurden als Zylinder modelliert, denen kleine Kegel als Spitzen aufgesetzt wurden. Die weiteren Details werden wir später behandeln, wenn wir die dazu nötigen Grundlagen in der Vorlesung besprochen haben.

```
//
// POV-Ray(tm) 3.0 Ein normalverteiltes Cluster
// Copyright 1998 by Leonie Dreschler-Fischer
//

#include "colors.inc"
```

```
#include "golds.inc"
#include "metals.inc"
#include "textures.inc"

global_settings { assumed_gamma 2.2 }

camera {
  location <15, 15, -33>
  look_at <0, 2, 0>
  angle 50
}

background { color SkyBlue }

sky_sphere {
  pigment {
    gradient y
    color_map {
      [0 colour SkyBlue]
      [1 colour MidnightBlue]
    }
    scale 2
    translate <-1, -1, -1>
  }
}
light_source { <100, 100, -100> color rgb 1 }

#declare stardot = sphere { 0, 0.7
  texture { T_Gold_1B }
}
#include "trivar1000.pov" /* Die Kugeln */

// Axis
#declare Alength = 12.0;
#declare Awidth = Alength*0.05;
cylinder { // x-axis
 <0.0, 0.0, 0.0>,
 Alength * x
  Alength / 100.0
  texture {T_Silver_1B}
}
```

```
cone {// tip
 Alength * x, Awidth,
 Alength*1.1 * x, 0.0
  texture {T_Silver_1B}
}
text {ttf "timrom.ttf" "X" 0.5, 0
 pigment {Blue}
   finish {
      ambient 0.5
      diffuse 0.8
   }
 translate <Alength,Awidth, 0.0> }

cylinder { // y-axis
 <0.0, 0.0, 0.0>,
 Alength * y,
  Alength / 100.0
  texture {T_Silver_1B}
}
cone {// tip
 Alength* y *0.9, Awidth,
 Alength* 1.1 * y, 0.0
  texture {T_Silver_1B}
}
text {ttf "timrom.ttf" "Y" 0.5, 0
 pigment {Blue}
   finish {
      ambient 0.5
      diffuse 0.8
   }
 translate <Awidth,Alength, 0.0> }
cylinder { // z-axis
 <0.0, 0.0, 0.0>,
 -Alength * z
  Alength / 100.0
  texture {T_Silver_1B}
}
cone {// tip
 -Alength * z, Awidth,
 -Alength*1.1 * z, 0.0
  texture {T_Silver_1B}
```

```
}
text {ttf "timrom.ttf" "Z" 0.5, 0
 pigment {Blue}
   finish {
      ambient 0.5
      diffuse 0.8
   }
 translate <0.0,Awidth, -Alength> }
```

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.

5.2.2 3D-Translation

3D-Translation in homogenen Koordinaten

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

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

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

5.2.3 3D-Skalierung

3D-Skalierung in homogenen Koordinaten

Skalierung mit s_x, s_y, s_z :

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

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 :

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

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

5.2.4 3D-Rotation

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

$$\begin{aligned} x' &= x \cdot \cos(\psi) - y \cdot \sin(\psi) \\ y' &= x \cdot \sin(\psi) + y \cdot \cos(\psi) \\ z' &= z \\ R_z(\psi) &= \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 & 0 \\ -\sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

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:

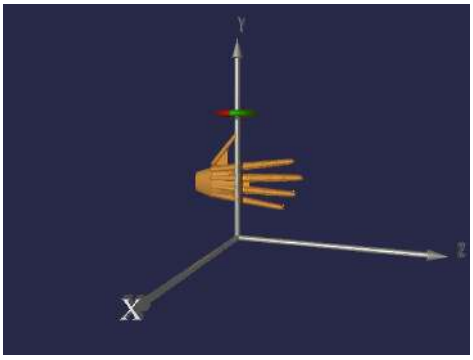
Linkshändiges System: Ein *positiver* Drehwinkel bewirkt eine Drehung *im* Uhrzeigersinn ↻.

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

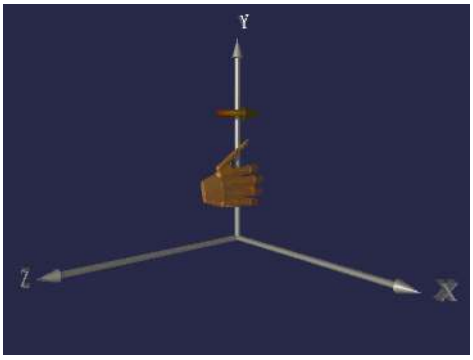
Wenn wir die Finger um die Drehachse legen und dabei den Daumen in die positive Richtung der Drehachse zeigen lassen, dann zeigen die Finger in Richtung der positiven Drehwinkel (linke Hand für ein linkshändiges System, rechte Hand für ein rechtshändiges).

Eselsbrücke: 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



rechtshändig

Zwei kleine Filme:

Linkshändige Rotation: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/linkshaendigRota.html

Rechtshändige Rotation: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/linkshaendigRota.html

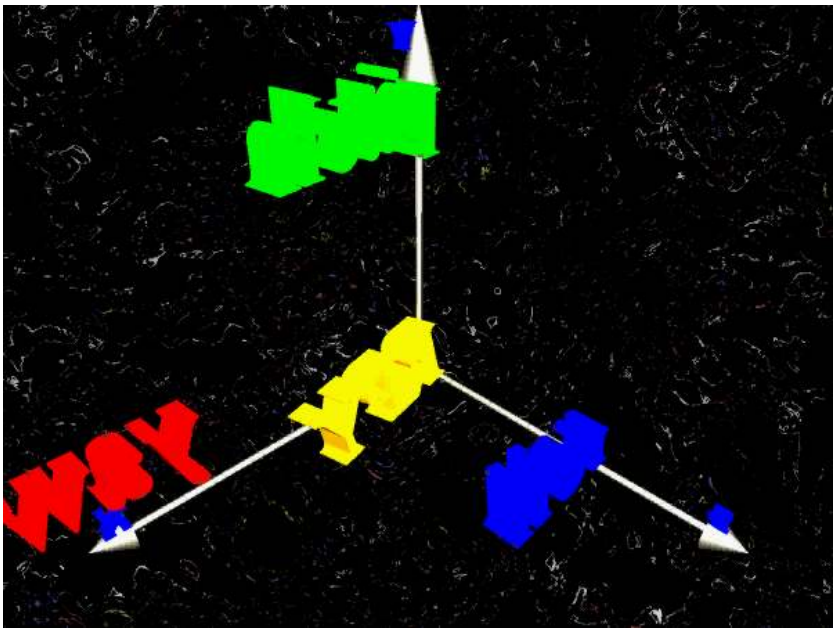
3D-Rotation in POV-Ray

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: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/rpy.html

Im Finale dieses Films von Peer Stelldinger ist auch

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

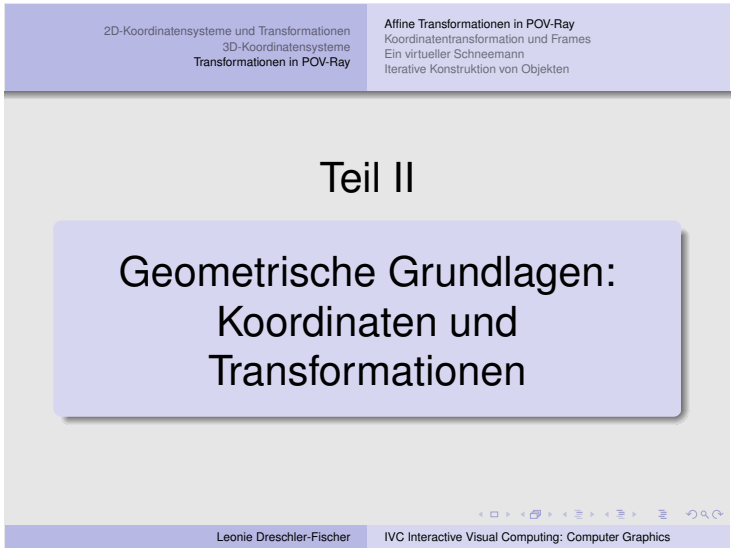
Drehungen um *beliebige Achsen* werden aus elementaren Transformationen zusammengesetzt:

- 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.

Die Herleitung finden Sie bei (Fellner, 2004), aber versuchen Sie es zunächst selbst zur Übung!

6 Transformationen in POV-Ray

6.1 Affine Transformationen in POV-Ray



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 >
```

6.1.1 3D-Scherung

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 > }
```

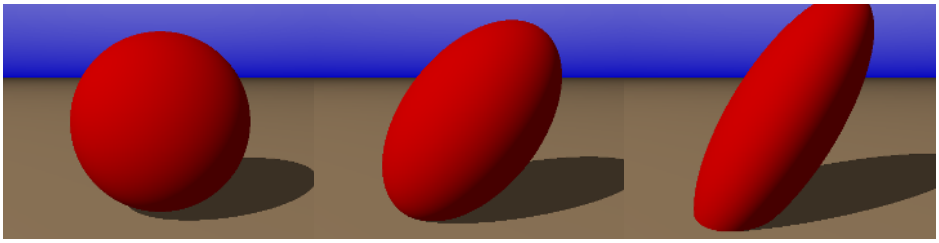
Scherung einer Kugel:

Animation

Beispiel: 10
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 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
```

Diese Matrix transformiert einen Einheitswürfel so, daß die Zeilenvektoren der Matrix in Richtung der neuen Kanten des transformierten Würfels zeigen.

Wenn die Zeilenvektoren der Matrix keine Einheitsvektoren sind, kann bei der Transformation auch gleichzeitig eine Skalierung auftreten.

Ein Film einer Scherung entlang der X-Achse: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movscherung.html

```
#include "colors.inc"

global_settings
{
  assumed_gamma 1.0
}

// ------------------------------------------------
camera
{
  location   <0.0, 0.5, -4.0>
  direction  1.5*z
  right      4/3*x
  look_at    <0.0, 0.0,  0.0>
}

sky_sphere
{
  pigment
  {
    gradient y
    color_map { [0.0 color blue 0.6] [1.0 color rgb 1] }
  }
}

light_source
{
  0*x // light's position (translated below)
  color red 1.0  green 1.0  blue 1.0  // light's color
  translate <-30, 30, -30>
}

// ------------------------------------------------
plane { y, -1 pigment {color rgb <0.7,0.5,0.3>}}

sphere {0, 1
   texture {pigment {radial frequency 8}
```

```
            finish{specular 1}}
matrix < 1, clock, 0,
             0, 1, 0,
             0, 0, 1,
             0, 0, 0 >
}
```

6.2 Koordinatentransformation und Frames

Koordinatentransformation

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

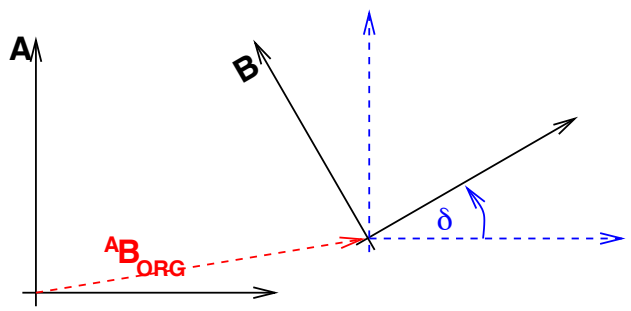
- Wir verschieben den Ursprung des Koordinatensystems in den Drehpunkt.
- Wir drehen die z-Achse parallel zur Rotationsachse.
- Im transformierten Koordinatensystem ist die Rotation dann trivial.
- Es ist wichtig, die Übersicht zu behalten, in welchen Koordinatensystemen die jeweiligen Koordinaten und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

Zur Notation:

- Koordinatensysteme werden mit großen Buchstaben bezeichnet: $A, B, \ldots$
- 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.

✎ **Anmerkung:** Wenn die Transformation zwischen zwei Koordinatensystemen nur aus der Translation und Rotation besteht, dann kann die Transformationsmatrix sehr einfach konstruiert werden. Sie wird aus der Rotationsmatrix, die die Orientierung der Achsen beschreibt, und der Translation des Ursprungs zusammengesetzt. Die Zeilen der Rotationsmatrix geben die Lage der Koordinatenachsen von B in den Koordinaten von A an.

Transformation zwischen Koordinatensystemen



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

Beispiel

$$\begin{aligned}
{}^{A}_{B}T &= \left[\begin{array}{rrr|r} 0.866 & 0.500 & 0.000 & 0.000 \\ -.0500 & 0.866 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 \\ \hline 10.000 & 0.000 & 5.000 & 1.000 \end{array}\right] \\
{}^{B}P &= [\ 3.0 \quad 3.0 \quad 7.0 \mid 1.0\ ] \\
{}^{A}P &= {}^{B}P \cdot {}^{A}_{B}T = [\ 11.098 \quad 4.098 \quad 12.0 \mid 1.0\ ]
\end{aligned}$$

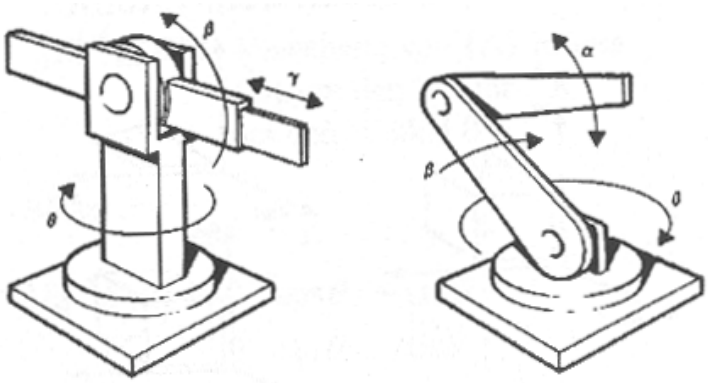
Orientierungen (Frames)

- 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

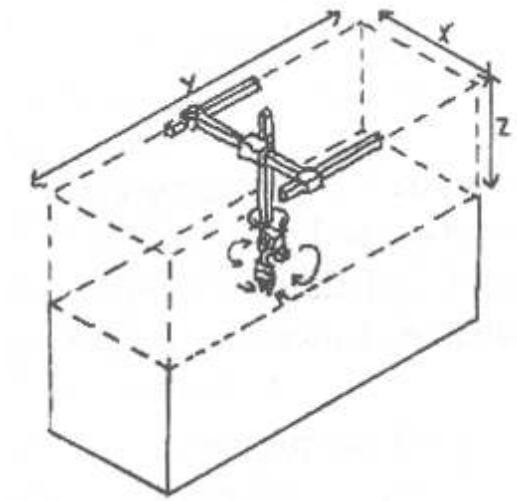
- Auch in der Robotik spielen Koordinatentransformationen eine große Rolle, da Handhabungsgeräte aus gegeneinander beweglichen Gliedern aufgebaut sind (Translationsgelenke, Rotationsgelenke).
- In der Robotik heißt ein Paar aus Translation und Rotation *frame* und wird zur Beschreibung der Lage der Glieder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

Bauformen von Robotern

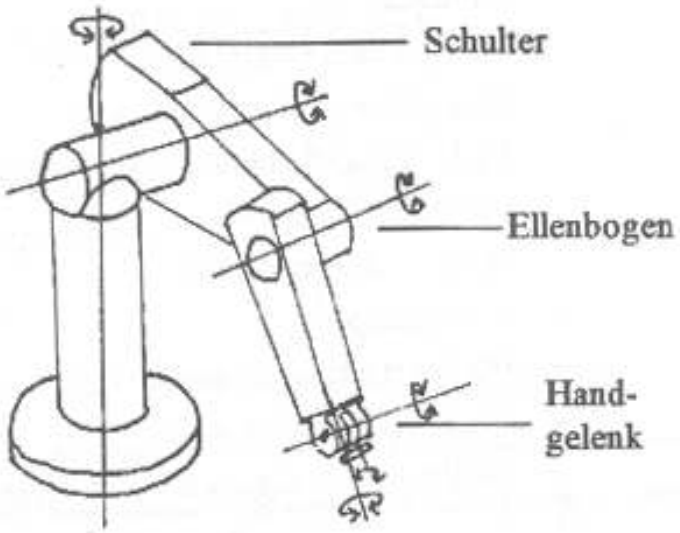


Translatationsgelenke und Rotationsgelenke

Ein Roboter mit Translationsgelenken:



Ein Roboter mit Rotationsgelenken:



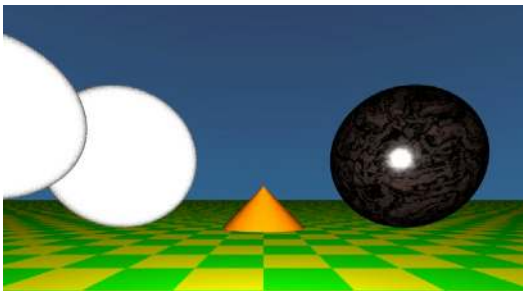
✎ **Anmerkung:** Die folgenden Folien zeigen ein Beispiel für die Modellierung einer Szene mittels einer Folge von Transformationen. Aus weißen und schwarzen Kugel und einem orangen Kegel wird mittels Skalierung, Translation und Rotation ein Schneemann aufgebaut.

6.3 Ein virtueller Schneemann

Ein virtueller Schneemann

Beispiel: 11
Wir konstruieren einen Schneemann aus gegebenen Grundobjekten, die wir durch geeignete Transformationen zusammenfügen.

Das Baumaterial: *Weiße und schwarze Kugeln, ein oranger Kegel*



```
#version 3.0 /*snowman.inc */
global_settings { assumed_gamma 2.2 }

#include "colors.inc"
#include "stones.inc"
#include "textures.inc"
camera {
   location  <0, 20, -200>
   direction <0,  0,    0.7>
   up        <0,  1,    0>
   right    <4/3,  0,    0>
}

background { color SkyBlue }
sky_sphere {
  pigment {
    gradient y
    color_map {
      [0 colour SkyBlue]
      [1 colour MidnightBlue]
    }
    scale 2
```

```
    translate <-1, -1, -1>
  }
}

/* Put down the beloved famous raytrace
   green/yellow checkered floor */
plane { y, 0
   pigment {
      checker colour Yellow colour Green
      scale 20
   }
   finish {
      ambient 0.2
      diffuse 0.8
   }
}

light_source {<0, 20, -200> colour White}
light_source {<0, 20, -200> colour White}
light_source {<0, 20, -200> colour White}


#declare snowball = sphere { 0,  1
   pigment { White }
   finish {
      ambient 0.9
      diffuse 0.9
      phong 1.0
      phong_size 20
      crand 0.5
   }
}

#declare coal = sphere { 0,  1
   texture { T_Grnt15 }
   finish {
      ambient 0.6
      diffuse 0.6
      phong 0.5
      phong_size 20
      crand 0.5
```

```
}}

#declare carrot = cone { <0, 0, 0>, 1.0, <0, 1, 0> 0.0
   pigment { Orange }
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20
      crand 0.1
   }
}
```

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:

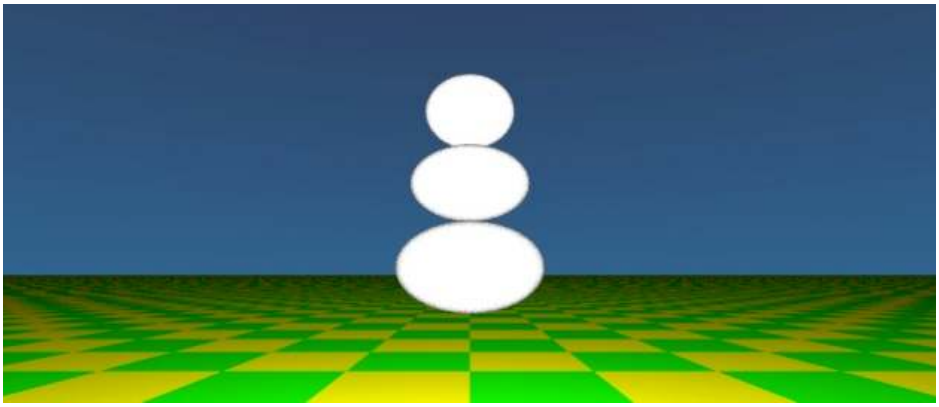
Der Körper: Der Körper wird aus passend skalierten weißen Kugeln zusammengesetzt, die durch Translation übereinandergestapelt werden.

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.

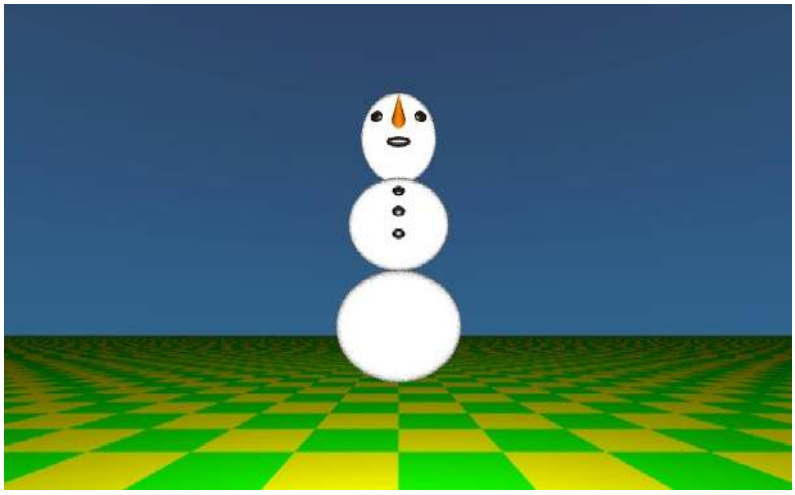
Kopf, Brust und Bauch aus skalierten Kugeln



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



```
#version 3.0
#include "snowman-1.pov" /* die bisherigen Definitionen */

#declare eyescalex = 0.15*headscale;
#declare eyescaley = 0.7*eyescalex;
#declare eyescalez = 0.4*eyescalex;
#declare middleoffacey
        = <0,2*bellyscaley+2*breastscaley+headscale,0>;
#declare middleofface = middleoffacey-z*headscale;
#declare middleofbreasty = <0,2*bellyscaley+breastscaley,0>;

object {coal /* left eye */
 scale <eyescalex, eyescaley, eyescalez>
 translate middleofface
 rotate <-5, -20.0, 0>}

object {coal /* right eye */
 scale <eyescalex, eyescaley, eyescalez>
 translate middleofface
 rotate <-5, 20, 0>}

object {carrot scale
<3, 20, 3>
```

```
rotate <-90, 0, 0>
translate middleofface
}
object {coal /*mouth middle*/
 scale <2*eyescalex, eyescaley, eyescalez>
 translate -headscale*z
 rotate <-25,0,0>
 translate middleoffacey}

object {coal /* top button */
 scale <eyescalex, eyescaley, eyescalez>
 translate -breastscalez*z
 rotate <20,0, 0>
 translate middleofbreasty*y
}
object {coal /* middle button */
 scale <eyescalex, eyescaley, eyescalez>
 translate -breastscalez*z
 translate middleofbreasty*y
}
object {coal /* lower button */
 scale <eyescalex, eyescaley, eyescalez>
 translate -breastscalez*z
 rotate <-20,0, 0>
 translate middleofbreasty*y
}
```

Mit Hut und Schal

```
#include "snowman-2.pov"

// declare rainbow's colours

#declare r_violet1
        = colour red 1.0 green 0.5 blue 1.0 filter 1.0;
#declare r_violet2
        = colour red 1.0 green 0.5 blue 1.0 filter 0.8;
#declare r_indigo
        = colour red 0.5 green 0.5 blue 1.0 filter 0.8;
#declare r_blue
        = colour red 0.2 green 0.2 blue 1.0 filter 0.8;
#declare r_cyan
        = colour red 0.2 green 1.0 blue 1.0 filter 0.8;
#declare r_green
        = colour red 0.2 green 1.0 blue 0.2 filter 0.8;
#declare r_yellow
        = colour red 1.0 green 1.0 blue 0.2 filter 0.8;
#declare r_orange
        = colour red 1.0 green 0.5 blue 0.2 filter 0.8;
#declare r_red1
        = colour red 1.0 green 0.2 blue 0.2 filter 0.8;
#declare r_red2
```

```
        = colour red 1.0 green 0.2 blue 0.2 filter 1.0;

// create the rainbow

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]
  }
}

rainbow {
  angle 37
  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]
  }
}
```

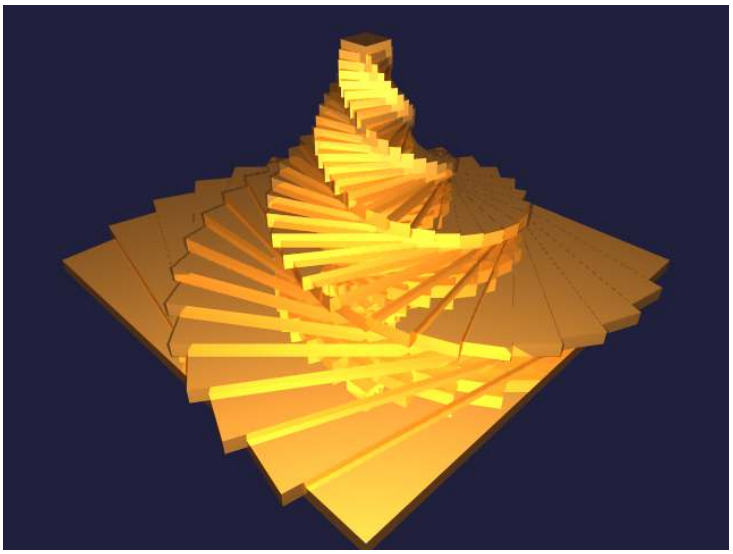
```
sky_sphere {
  pigment {
    gradient y
    color_map {
      [0 colour SkyBlue]
      [1 colour MidnightBlue]
    }
    scale 2
    translate <-1, -1, -1>
  }
}

#declare cylinderheight = 1.2*headscale;
#declare cylinderradius = 0.9*headscale;

cylinder { /* hat */
 <0, 0, 0> <0, cylinderheight, 0> cylinderradius
 rotate <0, 0, 5>
 translate  <0,2*bellyscaley+2*breastscaley+1.8*headscale,0>
 pigment {Black}
}

cylinder { /* hat border */
 <0, 0, 0> <0, 1, 0> 1.3*cylinderradius
 rotate <0, 0, 5>
 translate
        <0,2*bellyscaley+2*breastscaley+1.8*headscale,0>
 pigment {Black}
}

torus { /* scarf */
 0.9*headscale, 0.1*cylinderradius
 scale <1, 3, 1>
 translate
 <0,2*bellyscaley+2*breastscaley,0>
 pigment {checker Red, Blue}
}
box {<-bellyscalex, -1, -bellyscalex> ,
     <0, 1, 0.05*bellyscalex>
     rotate <0, 0, 20>
```

```
      translate
          <0,2*bellyscaley+2*breastscaley,0.05*bellyscalex>
 pigment {checker Red, Blue}
}
```

Der Film: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/Ein_wandelnder_Schneemann.html

6.4 Iterative Konstruktion von Objekten

Linear rekursive Objekte



Eine Pyramide, konstruiert mittels 3-D-Transformationen

Beispiel: 12
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.*

Der Film: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/SE3-Movies/Pages/Pyramide.html

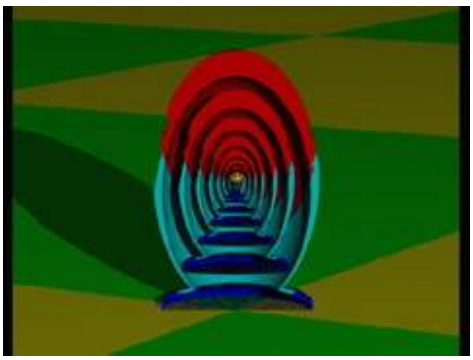
```
// POV-Ray(tm) 3.0 A golden pyramide
// Copyright 1999 by Leonie Dreschler-Fischer
//
// ==== Standard POV-Ray Includes ====
#include "colors.inc"    // Standard Color definitions
#include "textures.inc" // Standard Texture definitions

// POV-Ray(tm) 3.0, Animation of a golden pyramid
// Copyright 1998 by Leonie Dreschler-Fischer
```

```
//

#include "colors.inc"
#include "stars.inc"
#include "golds.inc"
#include "metals.inc"
#include "textures.inc"

global_settings { assumed_gamma 2.2 }

background { color MidnightBlue }


#declare nsteps=32;
#declare baselength=15;
#declare step=baselength/nsteps;
#declare scalefac=sqrt(sqrt(sqrt(sqrt(sqrt(baselength)))));

camera {
  location baselength*1.7*(x+y+z)
  look_at <0,-2.2*step,0>
  angle 60
}

light_source {  baselength*1.5*(x+y+z) color rgb 1 shadowless}
light_source {  baselength*1.5*(x+y+z) color rgb 1 }
light_source {  baselength*1.5*(x+y+z) color rgb 1 }

#declare basebox=box{ 1, -1
scale baselength*x+baselength*z+step*y
texture {T_Gold_2A}
}

#declare thisstep=clock*nsteps;
#declare i=0;
#declare iscale=1;
object {basebox}
#while (i<thisstep)
#declare i=i+1;
#declare iscale=iscale/scalefac;
object {basebox
```

```
   scale <iscale,1,iscale>
   translate step*y*i
   rotate y*360*i/nsteps}
#end
```

✎ **Anmerkung:** Die folgenden Animationen wurden, genau wie der Schneemann, erzeugt, indem geeignete Basisbausteine (Kugeln, Zylinder, Kegel) passend skaliert, rotiert und verschoben wurden. Allen diesen Sequenzen ist gemeinsam, daß die erzeugten Objekte linear rekursiv sind.

Eine Schachtelpuppe, aufgeschnitten



Der Film zur Skalierung: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/SE3-Movies/Pages/Schachtelpuppe_1.html

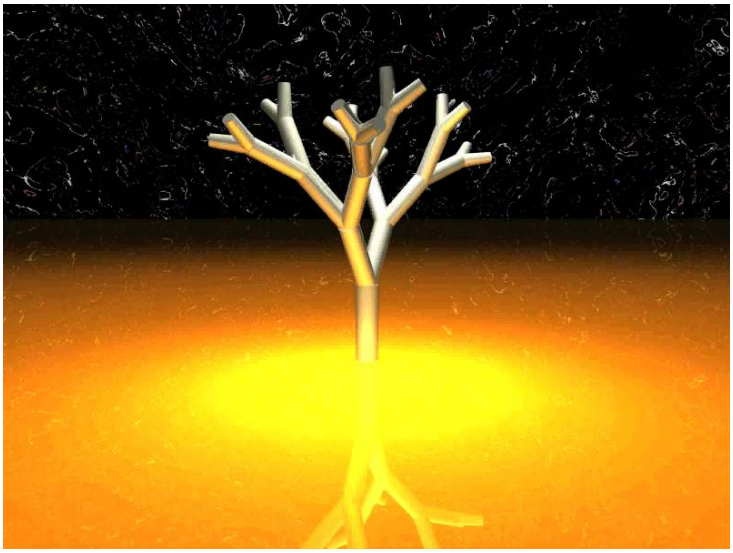
Variante 2 mit Translation: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/SE3-Movies/Pages/Schachtelpuppe_2.html

Eine Schachtelpuppe, rekursiv geöffnet

Ein Ammonit

Der Ammonit: Skalierung, Rotation und Translation: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/SE3-Movies/Pages/Ammonit.html

Eine baumrekursive Struktur



Ein rekursiver Blütenbaum: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/SE3-Movies/Pages/BinbaumBlueten.html

IVC Interactive Visual Computing

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 3:

Geometrische Grundlagen: Geometrisches Modellieren

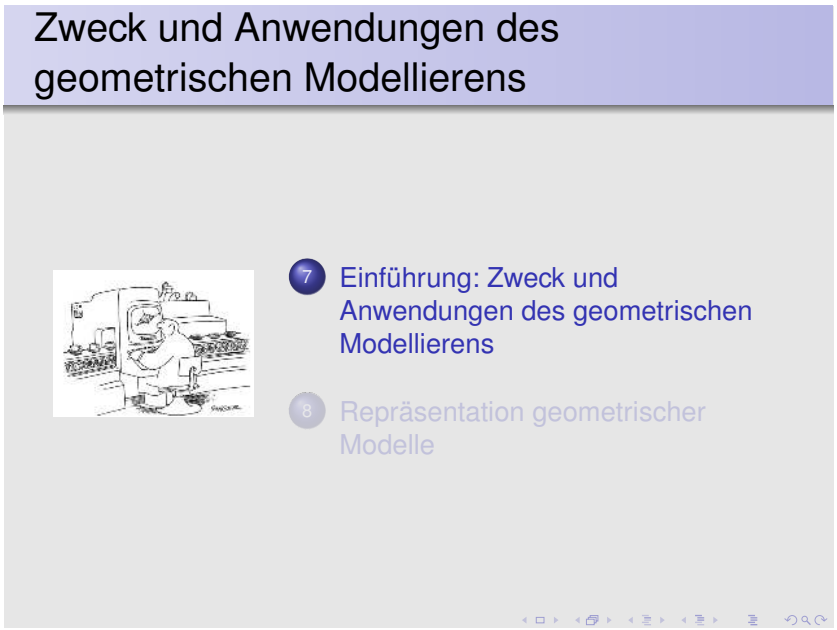
Universität Hamburg
Fachbereich Informatik

WS 2014/2015

Teil III

Geometrische Grundlagen: Geometrisches Modellieren

7 Einführung: Zweck und Anwendungen des geometrischen Modellierens



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.

8 Repräsentation geometrischer Modelle

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.

8.1 Volumenbasierte Repräsentationsformen

8.1.1 Octtree

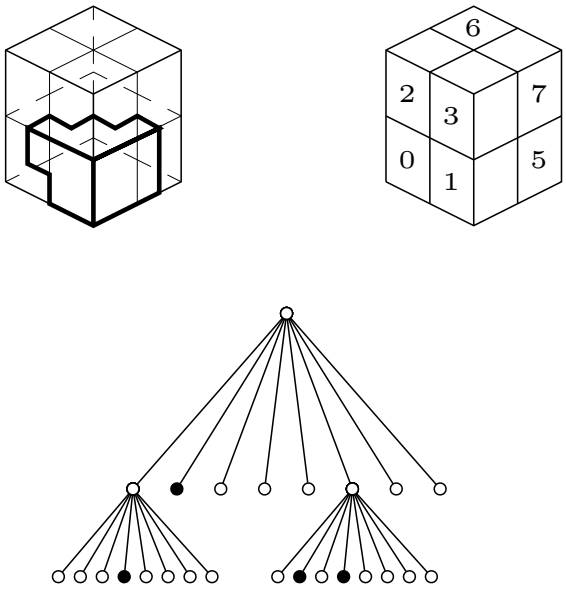
Volumenbasierte Repräsentationsformen

Octree

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

Anwendungsgebiet: volumenorientierte Objektrekonstruktion (z.B. Tomographie)

Octtree, Beispiel



8.1.2 Constructive Solid Geometry (CSG)

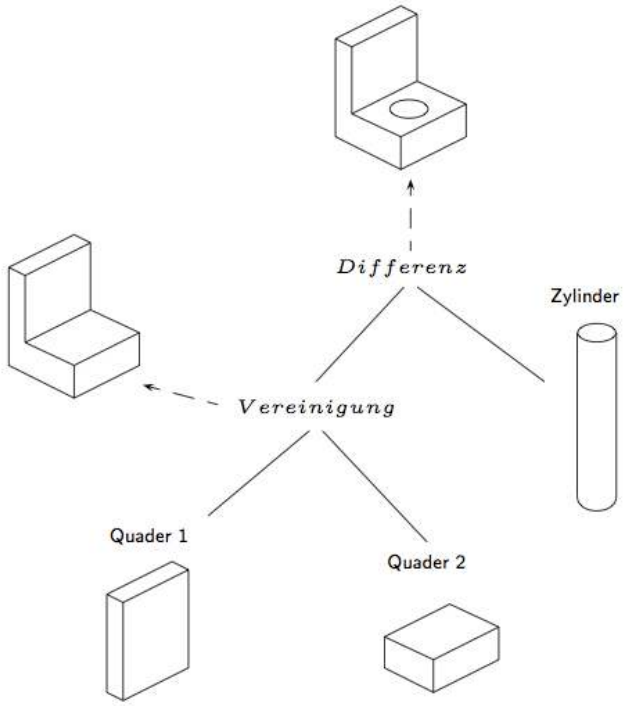
Constructive Solid Geometry (CSG)

Constructive Solid Geometry (CSG)

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

Anwendungsgebiet: volumenorientierte Objektrekonstruktion (z.B. Tomographie)

CSG, Beispiel



-
-
-

-
-
-

-
-
-

IVC Interactive Visual Computing

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 4:

Geometrische Grundlagen: Abbildung des Raumes

Universität Hamburg
Fachbereich Informatik

WS 2014/2015

Teil IV

Geometrische Grundlagen: Abbildung des Raumes

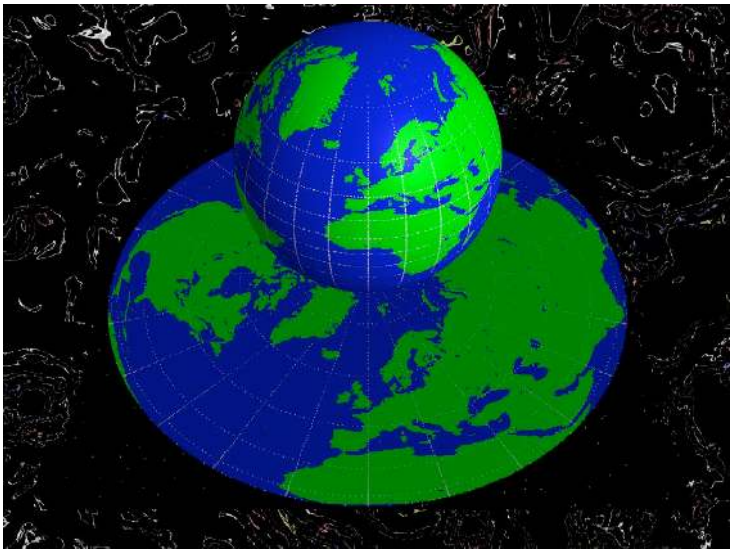
9 Projektionen

9.1 Grundbegriffe

Definition: 13 (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 $m < n$.

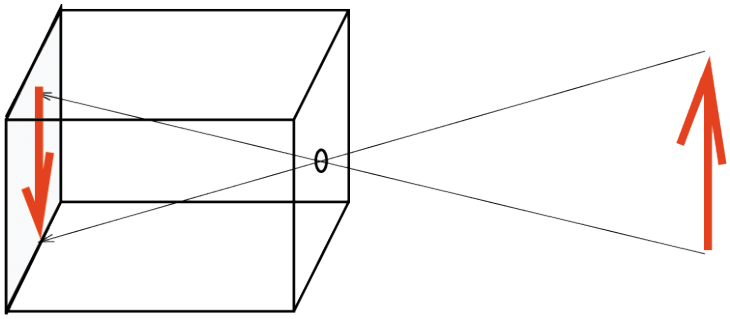
- Wir werden hier nur Abbildungen des 3D-Raumes auf eine Ebene betrachten, d.h. nur *planare* Projektionen mit $n = 3, m = 2$.
- Nicht-planare Projektionen werden häufig in der Kartographie eingesetzt: Zylinderprojektion, Kegelprojektion usw.

Planare und nicht-planare Projektionen



9.2 Lochkamera

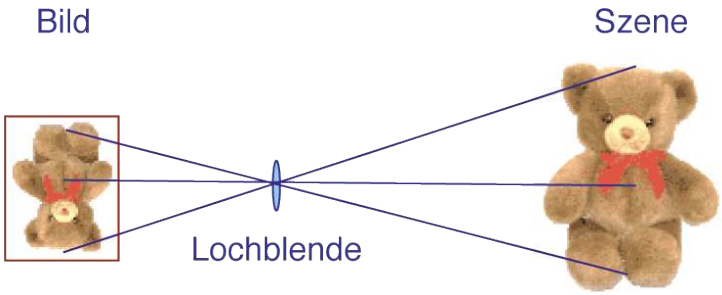
Die Abbildung mittels einer Lochkamera



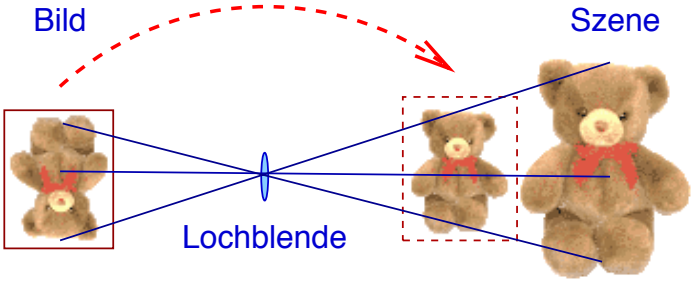
- Das Bild entsteht im Schnittpunkt der Projektionsstrahlen mit der Bildebene.
- Das Bild steht auf dem Kopf.

Abstraktion vom Gehäuse

Nur ein Loch und eine Bildebene



Gleichwertiges Modell: Zentralprojektion



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

Definition: 14 (Zentralprojektion)
Eine Zentralprojektion wird festgelegt durch die Angabe

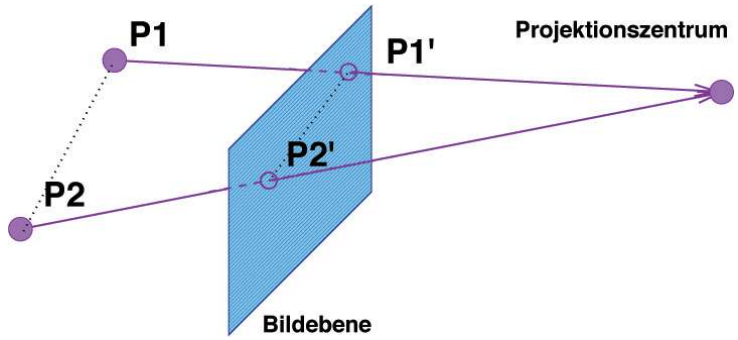
- *eines Projektionszentrums (Augpunkt)*
- *und einer Projektionsebene.*

Projektionsstrahlen sind Geraden durch den Augpunkt und die abzubildenden Punkte.

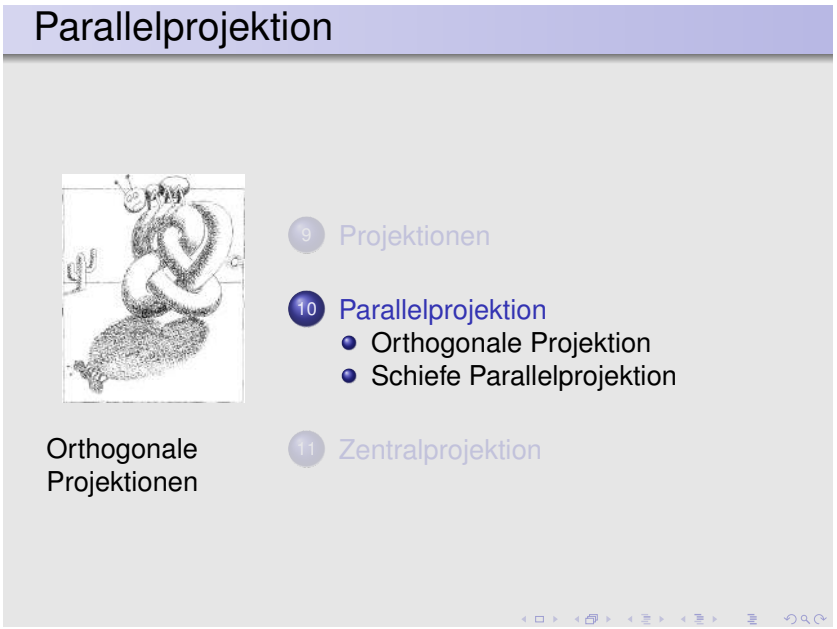
Das Abbild des Punktes ist der Schnittpunkt seines Projektionsstrahls mit der Bildebene (die Projektion).

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

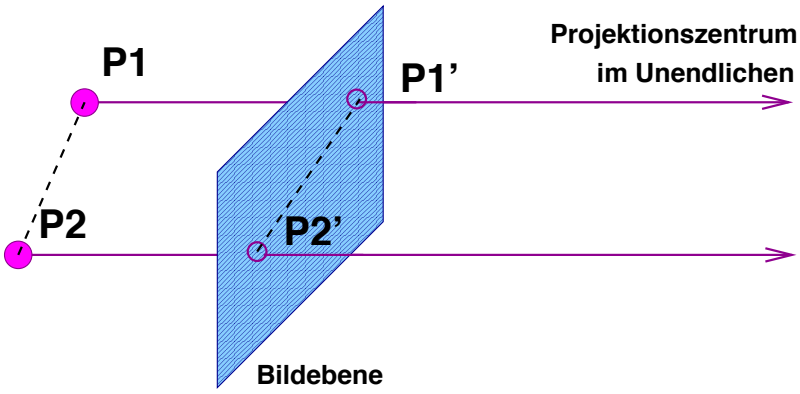
Zentralprojektion



10 Parallelprojektion



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 Abhängigkeit vom Winkel, unter dem die Projektionsstrahlen die Bildebene schneiden, in zwei Gruppen unterteilt:

- Orthogonale Projektion
- Schiefe Parallelprojektion

Definition: Orthogonale Projektion

Definition: 15 (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: 16 (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.

10.1 Orthogonale Projektion

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 Symmetrieachsen der abgebildeten Objekte sind, dann haben die Risse eine bestimmte Bedeutung und festgelegte Namen:

Grundriß: Projektion in die xz-Ebene.

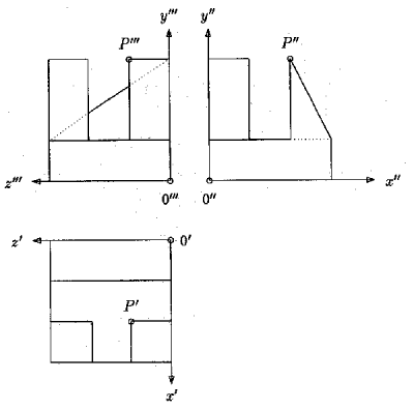
Aufriß: Projektion in die xy-Ebene.

Seitenriß, Kreuzriß: Projektion in die yz-Ebene.

Die Interpretation von Rissen

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

Aufriß, Seitenriß und Grundriß eines Objektes.

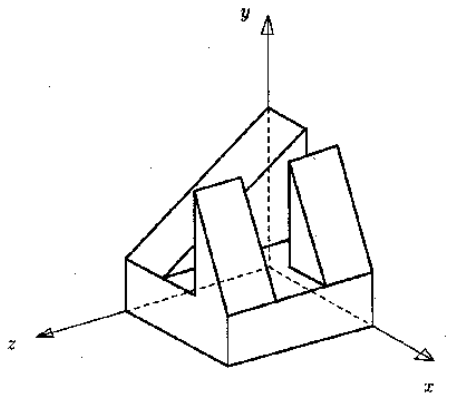


Definition: 17 (Normalaxonometrische Projektion)
Eine orthogonale Projektion heißt Normalaxonometrische Projektion, wenn keine Achse des Weltkoordinatensystems senkrecht zur Bildebene ist.

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

Normalaxonometrischer Projektion

Das Objekt von Folie 178 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.

10.1.1 Transformationsgleichungen für die orthogonale Projektion

Transformationsgleichungen für die orthogonale Projektion

Die Transformation erfolgt in zwei Schritten:

① Transformation des Weltkoordinatensystems:

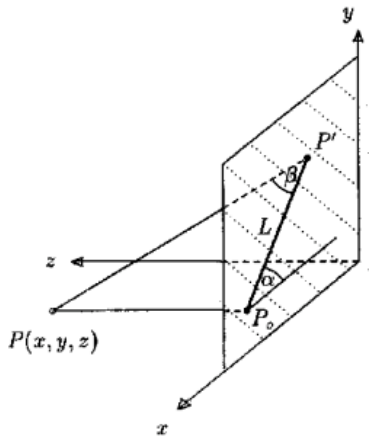
- Translation, so daß die Bildebene im Ursprung liegt.
- Rotation, so daß die Normale der Bildebene parallel zur z-Achse ist.

② Die Bildkoordinaten sind dann gleich den Koordinaten im transformierten System, abgesehen von der z-Koordinate, die gleich 0 ist.

10.2 Schiefe Parallelprojektion

Schiefe Parallelprojektion, Schrägriß

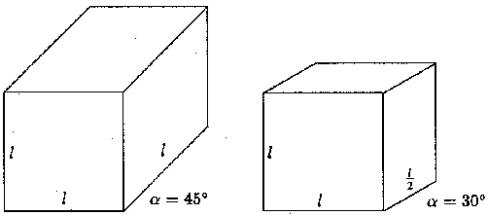
Die Projektionsstrahlen stehen *nicht senkrecht* auf der Bildebene, Angabe durch α und β.



Einige schiefe Parallelprojektionen:

Kavalierprojektion: $\tan\beta = 1$

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



10.2.1 Projektionsgleichungen für den Schrägriß

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

$$\begin{aligned}
\tan(\beta) &= \frac{z}{L}, \quad L = \overline{P_0P'} = \frac{z}{\tan(\beta)} \\
x' &= x - L \cdot \cos(\alpha) = x - z \cdot \frac{\cos(\alpha)}{\tan(\beta)} \\
y' &= y + L \cdot \sin(\alpha) = y + z \cdot \frac{\sin(\alpha)}{\tan(\beta)}
\end{aligned}$$

$$
\begin{aligned}
x' &= x - L \cdot \cos(\alpha) = x - z \cdot \frac{\cos(\alpha)}{\tan(\beta)} \\
y' &= y + L \cdot \sin(\alpha) = y + z \cdot \frac{\sin(\alpha)}{\tan(\beta)} \\
\begin{bmatrix} x' & y' & 0 & 1 \end{bmatrix} &= \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{\cos(\alpha)}{\tan(\beta)} & \frac{\sin(\alpha)}{\tan(\beta)} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{^{P}\mathsf{schief}_{xy}}
\end{aligned}
$$

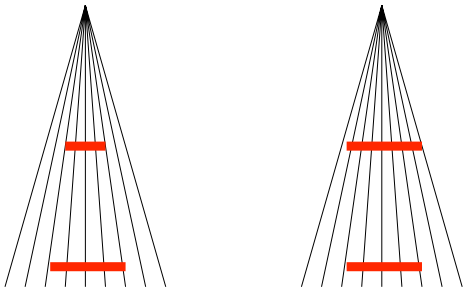
11 Zentralprojektion

11.1 Eigenschaften der Zentralprojektion

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.

✎ **Anmerkung:** Perspektivische Verkürzung:

Zwei Faktoren bestimmen wesentlich, wie ein Objekt abgebildet wird: Seine Orientierung im Raum relativ zur Projektions*richtung* und seine *Entfernung* vom Projektionszentrum.

Das vorherige Bild (Folie 114) mit den vielen Wegweisern (aufgenommen am Cap Reinga, Neuseeland) zeigt deutlich, wie die perspektivische Abbildung eines Objektes von seiner *Orientierung im Raum* abhängt. Wegweiser, die parallel zur Bildebene sind, werden nur wenig verkürzt, aber die Wegweiser, die nahezu senkrecht zur Bildebene sind, werden stark verkürzt abgebildet.

Die Entfernung vom Projektionszentrum bestimmt die zweite Art von perspektivischer Verkürzung. Diese ist für die Entstehung von Fluchtpunkten verantwortlich, da die Abstände zwischen parallelen Geraden mit zunehmender Entfernung vom Bild immer kleiner abgebildet werden[1] — das Bild des Abstands der Geraden im Unendlichen ist eine Strecke der Länge Null, eben der Fluchtpunkt dieser Geraden. Einen solchen Fluchtpunkt können Sie auf dem Bild des Deiches von Hallig Hooge oder am Bild der Avenida de 9. Julio (Buenos Aires) sehen.

[1] Sofern die Geraden nicht parallel zur Bildebene sind

Fluchtpunkte: Sommerdeich Hallig Hooge

Fluchtpunkte

Avenida de 9. Julio (Buenos Aires)

✎ **Anmerkung:** Wo liegt das Bild eines Punktes bei der Zentralprojektion? Die Projektionsgleichungen beschreiben, wohin ein Punkt aus dem n-dimensionalen Vektorraum im m-dimensionalen Raum abgebildet wird. Die Herleitung der Projektionsgleichungen ist im Falle der Zentralprojektion einfach, wenn das Koordinatensystem geschickt gewählt wird. Es gibt viele Möglichkeiten, das Koordinatensystem zu wählen – Sie werden in der Literatur ein buntes Durcheinander von Vorschlägen finden. Wichtig ist nur, daß eine Achse des Koordinatensystems senkrecht auf der Bildebene steht und daß das Projektionszentrum auf einer der Achsen liegt. Wir beziehen uns hier auf die Herleitung von Fellner, 1992, aber schauen Sie sich zum Vergleich auch einmal die Variante von Foley and van Dam, 1984 an (Ursprung im Projektionszentrum, linkshändiges Koordinatensystem).

Auch für den Abstand zwischen der Bildebene und dem Projektionszentrum gibt es verschiedene Bezeichnungen. Wir nennen diesen Abstand f (für „focal length“, Brennweite), aber gebräuchlich sind auch die Bezeichnungen a oder d.

11.2 Projektionsgleichungen

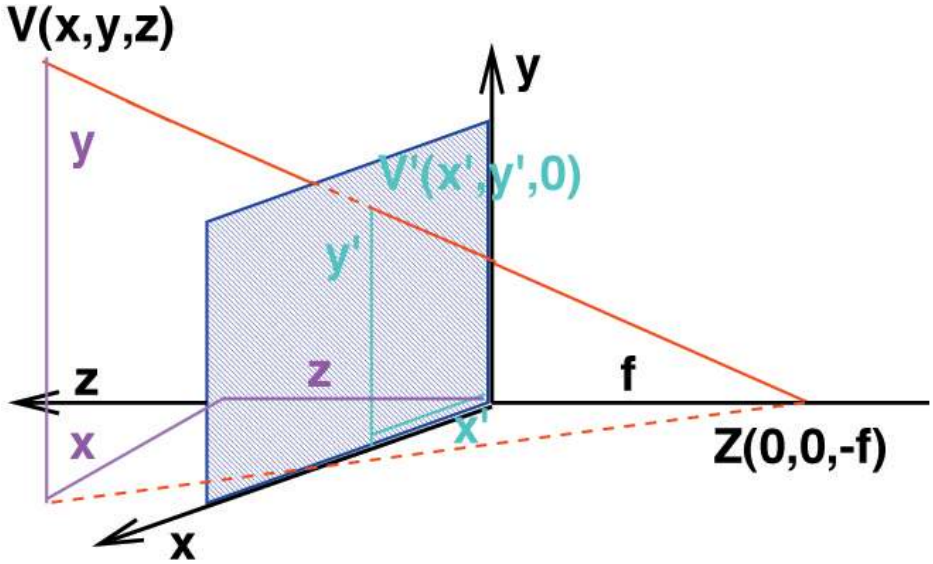
Projektionsgleichungen

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



✎ **Anmerkung:** Zur Herleitung der Projektionsgleichungen
Sei

$V = (x, y, z)$ der abzubildende Punkt, sei

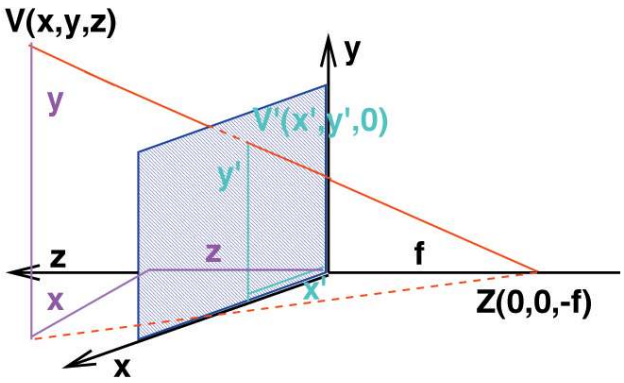
$V' = (x', y', 0)$ der zugehörige Bildpunkt und

$Z = (0, 0, -f)$ das Projektionszentrum.

Zur Herleitung der Projektionsgleichungen nutzen wir die Ähnlichkeit von zwei Dreiecken und die Parallelität von zwei Geraden (den Strahlensatz):
Das Dreieck mit den Eckpunkten $Z = (0, 0, -f), \quad V_{xx} = (x', 0, 0), \quad O = (0, 0, 0)$ ist ähnlich zum Dreieck $Z = (0, 0, -f), \quad V'_{xz}(x', 0, z), \quad V_z(0, 0, z)$.

1. Parallelität:
$$V'_{xz}O \parallel V_{xz}V_z$$

2. Streckenverhältnisse:
$$\frac{\overline{V'_{xz}O}}{\overline{V_{xz}V_z}} = \frac{\overline{OZ}}{\overline{V_zO}} = \frac{x'}{x} = \frac{f}{f+z}$$



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

$$\begin{aligned}
x' &= \frac{f \cdot x}{z+f} = \frac{x}{\frac{z}{f}+1} \\
\text{Analog: } y' &= \frac{f \cdot y}{z+f} = \frac{y}{\frac{z}{f}+1} \\
z' &= 0 \ \text{ in der Bildebene} \\
\begin{bmatrix} x' & y' & z' & 1 \end{bmatrix} &= \begin{bmatrix} x & y & 0 & \frac{z}{f}+1 \end{bmatrix} \\
&= \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}$$

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

- 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.

✎ **Anmerkung:** In homogenen Koordinaten sind die Bildkoordinaten x', y' zunächst gleich den ursprünglichen x, y-Koordinaten. Das Projektionsergebnis entsteht durch die Normierung der vierten Koordinate auf "1". Erst bei diesem Schritt erfolgt die Division mit $\frac{z}{f} + 1$.

11.2.1 Die inverse Zentralprojektion

Welcher Bildpunkt hat welches Urbild? Die Antwort sollten wir bekommen, wenn wir die Bildkoordinaten mit der Inversen der Projektionsmatrix rückprojizieren. Nur geht das leider nicht so einfach. Wir haben hier erzwungen, daß die z-Koordinate bei der Projektion ins Bild zu Null gesetzt wird. Die Projektionsmatrix ist nicht invertierbar, da eine Spalte Null ist (Rang 3).

Außerdem kann die Rückprojektion nicht eindeutig sein, da alle Punkte entlang eines Projektionsstrahls denselben Bildpunkt haben. Die Menge der Urbilder eines Bildpunktes ist eine Gerade. Die Rückprojektion sollte uns also zu einem Bildpunkt die Geradengleichung des zugehörigen Projektionsstahls geben.

Wenn uns das Ergebnis für die z-Koordinate gleichgültig ist, da wir ja wissen, daß der Bildpunkt die z-Koordinate “0” hat, können wir die Diagonale der Projektionsmatrix wie bei einer Einheitsmatrix mit “1” besetzen und erhalten eine invertierbare Matrix. Diese abgewandelte, invertierbare Projektionsmatrix nennen wir P_{IZ}. Weiterhin geben wir jedem Bildpunkt eine fiktive z-Koordinate z' verschieden von Null und können dann schreiben:

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

In diesem Fall ist z' eine freie Variable, die uns eine Gerade durch den Bildpunkt und das Projektionszentrum beschreibt.

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', $\longrightarrow$ (Duda and Hart, 1973, S. 384).

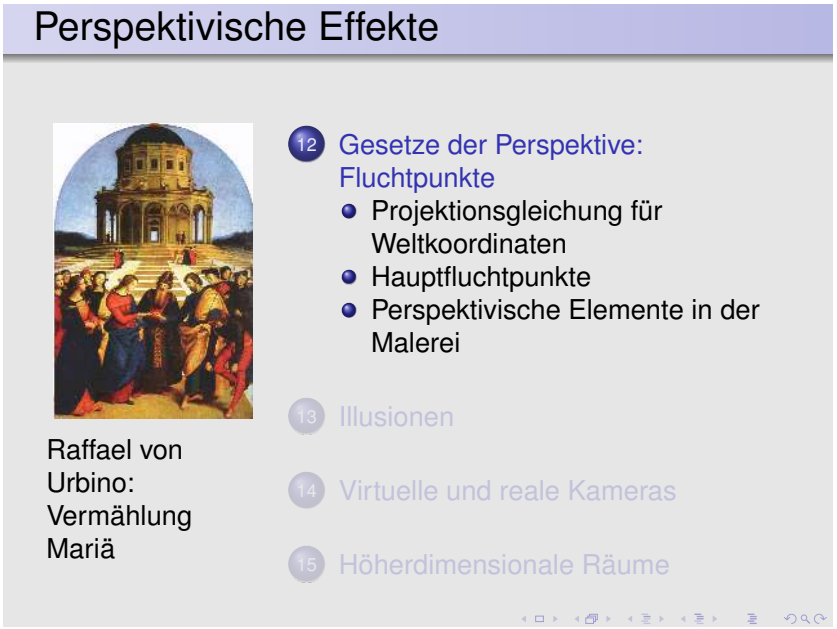
$$V = [x', y', z', 1] \cdot P_{IZ}^{-1}$$

Teil V

Perspektivische Effekte

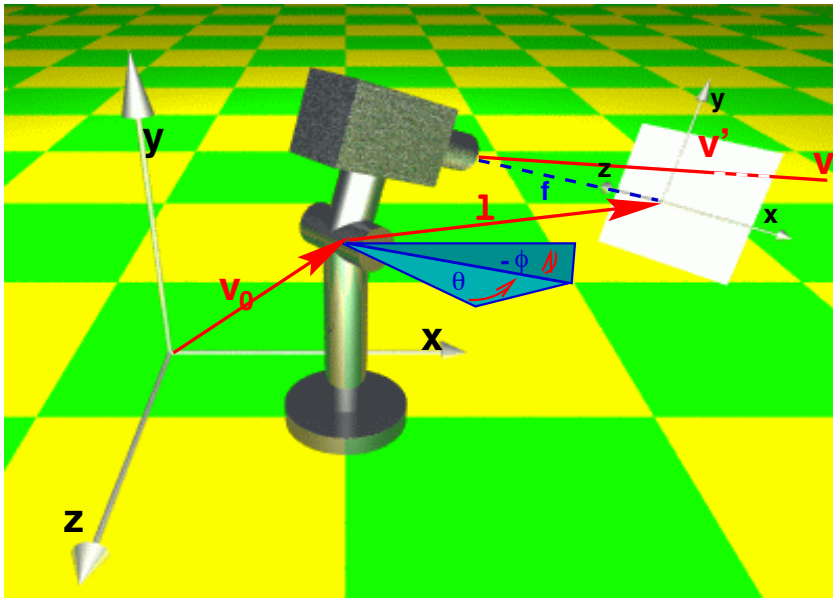
12 Gesetze der Perspektive: Fluchtpunkte

12.1 Projektionsgleichung für Weltkoordinaten



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 ${}^{K}_{G2}T = T(-{}^{G}\vec{l} - f \cdot {}^{G}\hat{z})$.

Zentralprojektion in die Bildebene: ${}^{K}_{K}P_Z$.

Projektionsgleichung in homogenen Koordinaten

In homogenen Koordinaten können wir die Transformationen als Matrizen zusammenfassen:

Der projizierte Punkt ${}^{K}V'$ des Punktes ${}^{W}V$ ist:

$$ {}^{K}V' = {}^{W}V \cdot {}^{G}_{W}T \cdot {}^{G2}_{G}R \cdot {}^{K}_{G2}T \cdot {}^{K}_{K}P_{Z}. $$

Und die inverse Zentralprojektion:

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

✎ **Anmerkung:** Die Herleitung der Transformationsmatrizen ist eine nette Übungsaufgabe für Sie. Deshalb werden wir sie hier nicht angeben. Bei (Duda and Hart, 1973, S. 388) finden Sie die Lösung für Spaltenvektoren im linkshändigen Koordinatensystem gerechnet.

12.2 Hauptfluchtpunkte

Als nächstes werden wir die Lage von Hauptfluchtpunkten herleiten — das sind Fluchtpunkte von Geraden, die parallel zu einer der Hauptachsen des Weltkoordinatensystems sind. Zur Herleitung werden wir ohne Verlust der Allgemeinheit die Parameter der äußeren Orientierung vereinfachen. Die Ergebnisse lassen sich leicht durch Transformationen auf allgemeinere Lagen übertragen.

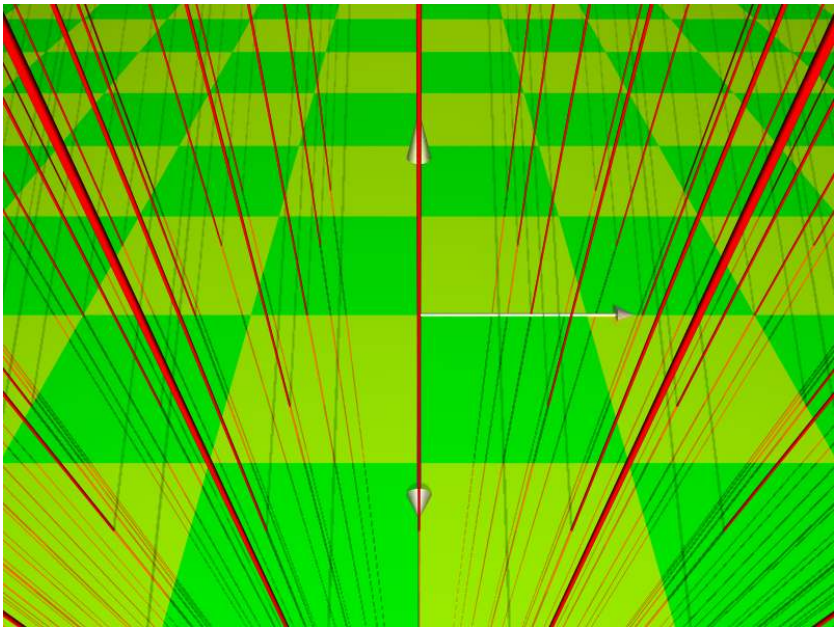
Definition: 18 (*Hauptfluchtpunkt*)
Ein Hauptfluchtpunkt ist der Fluchtpunkt von Geraden, die parallel zu einer der Achsen des Weltkoordinatensystems sind.

- 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.

$$\begin{aligned}
\text{Gimbal center } \vec{v}_0 &= \vec{0} \\
\text{Gimbal offset } \vec{l} &= \vec{0} \\
\text{Neigungswinkel } \varphi &= \text{beliebig} \\
\text{Brennweite } f &= \text{beliebig} \\
{}^K_W P &= R(\hat{x}, \varphi) \cdot P_Z
\end{aligned}$$

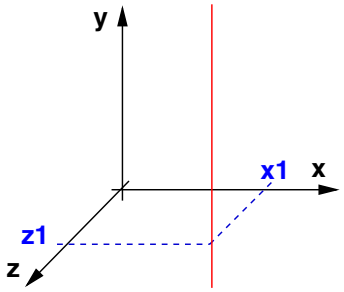
$$\begin{aligned}
{}^K_W P &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\varphi) & \sin(\varphi) & 0 \\ 0 & -\sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix}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

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

Division durch die homogene Koordinate w

$$\begin{aligned}
x' &= \frac{f \cdot x}{y \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f} \\
y' &= \frac{f \cdot y \cdot \cos(\varphi) - f \cdot z \cdot \sin(\varphi)}{y \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f}
\end{aligned}$$

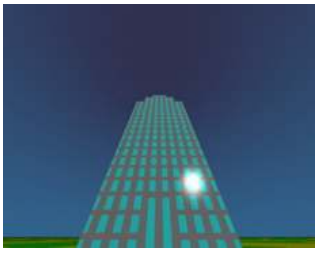
Eine vertikale Gerade durch den Punkt ${}^W(x_1, 0, z_1)$:



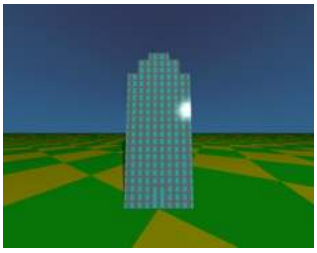
$$ {}^W\begin{bmatrix} x & y & z & 1 \end{bmatrix} = {}^W\begin{bmatrix} x_1 & y & z_1 & 1 \end{bmatrix} $$

$$
\begin{aligned}
x' &= \frac{f \cdot x_1}{y \cdot \sin(\varphi) + z_1 \cdot \cos(\varphi) + f} \\
y' &= \frac{f \cdot y \cdot \cos(\varphi) - f \cdot z_1 \cdot \sin(\varphi)}{y \cdot \sin(\varphi) + z_1 \cdot \cos(\varphi) + f} \\
\lim_{y \to \infty} x' &= 0 \\
\lim_{y \to \infty} y' &= f \cdot \frac{\cos(\varphi)}{\sin(\varphi)} = \frac{f}{\tan(\varphi)}
\end{aligned}
$$

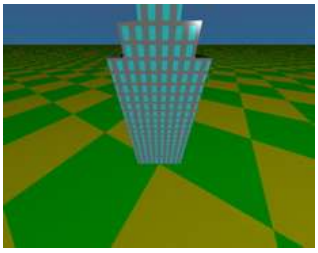
- Die Grenzwerte $\lim_{y\to\infty} x'$ und $\lim_{y\to\infty} y'$ sind unabhängig von x_1 und y_1 und damit für alle Vertikalen gleich!
- Der Punkt $(0, \frac{f}{\tan(\varphi)})$ ist der Hauptfluchtpunkt der Parallelen zur y-Achse!



$\varphi = -50^o$



$\varphi = 0^o$



$\varphi = 50^o$

Animation zum Fuchtpunkt: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/Fluchtpunkt.html

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*.

$$
\begin{aligned}
\text{Gimbal center } \vec{v}_0 &= (0, y_0, 0) \\
\text{Gimbal offset } \vec{l} &= \vec{0} \\
\text{Neigungswinkel } \varphi &= \text{beliebig} \\
\text{Brennweite } f &= \text{beliebig} \\
x' &= \frac{f \cdot x}{(y - y_0) \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f} \\
y' &= \frac{f \cdot (y - y_0) \cdot \cos(\varphi) - f \cdot z \cdot \sin(\varphi)}{(y - y_0) \cdot \sin(\varphi) + z \cdot \cos(\varphi) + f}
\end{aligned}
$$

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

$$
\begin{aligned}
{}^W\begin{bmatrix} x & y & z \end{bmatrix} &= {}^W\begin{bmatrix} x & 0 & m \cdot x + c \end{bmatrix} \\
x' &= \frac{f \cdot x}{-y_0 \cdot \sin(\varphi) + (m \cdot x + c) \cdot \cos(\varphi) + f} \\
y' &= \frac{f \cdot y_0 \cdot \cos(\varphi) - f \cdot (m \cdot x + c) \cdot \sin(\varphi)}{y_0 \cdot \sin(\varphi) + (m \cdot x + c) \cdot \cos(\varphi) + f} \\
\lim_{x \to \infty} x' &= \frac{f}{m \cdot \cos\varphi} \quad \text{abhängig von } m \\
\lim_{x \to \infty} y' &= \frac{-f \cdot m \cdot \sin(\varphi)}{m \cdot \cos(\varphi)} = -f \cdot \tan(\varphi)
\end{aligned}
$$

- 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 ($|\varphi| = \pi/2$).

✎ **Anmerkung:** Die Herleitung bezieht sich nur auf Geraden in der xz-Grundebene. Sie läßt sich aber leicht auf alle Geraden erweitern, die zur Grundebene parallel sind und dieselbe Steigung m haben.

Konjugierte Fluchtpunkte

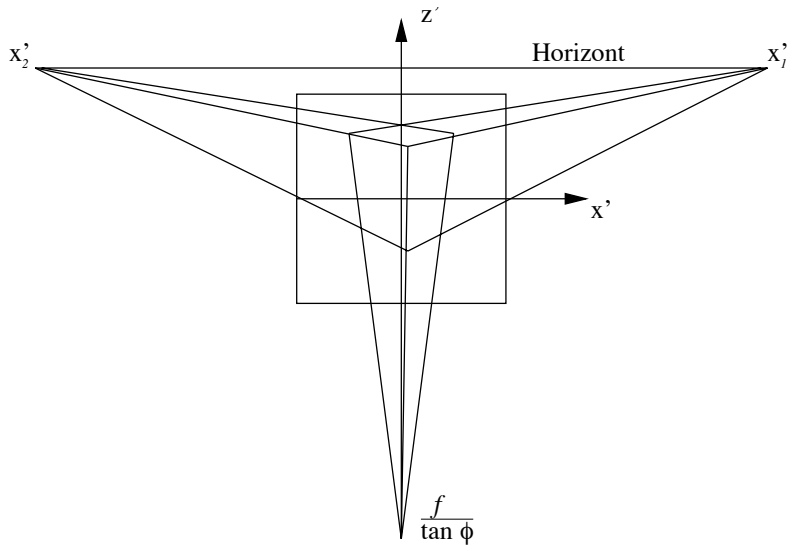
Satz: 19
Seien $g_1 \perp g_2$ Geraden in der Grundebene mit den Steigungen m_1 und $m_2 = -\frac{1}{m_1}$ und den Fluchtpunkten $(x'_1, -f \cdot \tan(\varphi)), (x'_2, -f \cdot \tan(\varphi))$. Dann gilt:

$$\begin{aligned} m_1 \cdot m_2 &= -1 \\ x'_1 \cdot x'_2 &= \frac{f}{m_1 \cdot \cos\varphi} \cdot \frac{f}{m_2 \cdot \cos\varphi} = -\frac{f^2}{\cos\varphi^2} \end{aligned}$$

Konjugierte Fluchtpunkte

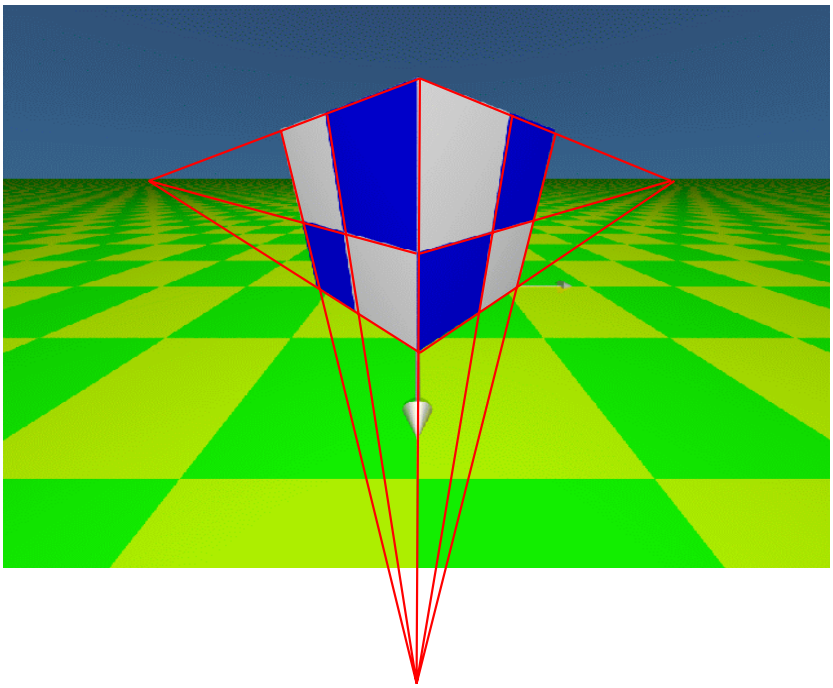
Definition: 20 (***Konjugierte Fluchtpunkte***)
Die Fluchtpunkte x_1 und x_2 der zueinander senkrechten Geraden $g_1 \perp g_2$ parallel zur Grundebene heißen konjugierte Fluchtpunkte.

Konjugierte Fluchtpunkte eines Quaders



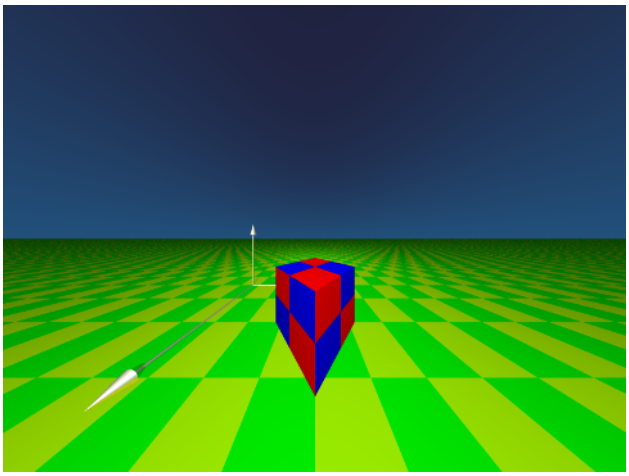
✎ **Anmerkung:** Hauptfluchtpunkte

Die Szene auf der folgenden Folie zeigt eine *Zweipunktprojektion*, auch wenn drei Fluchtpunkte eingezeichnet sind. Hauptfluchtpunkte liegen in diesem Beispiel auf der y-Achse und auf der z-Achse. Die Parallelen zur x-Achse haben keinen endlichen Fluchtpunkt, da die x-Achse parallel zur Bildebene ist. Bezogen auf die Hauptachsen des Würfels (Objektkoordinatensystem) dagegen handelt es sich um eine Dreipunktprojektion. Sie sehen daran, daß die Angabe der Projektionsart nur dann Sinn macht, wenn die Hauptachsen des Weltkoordinatensystems mit den Orientierungen der wichtigen Szenenkomponenten zusammenfällt.

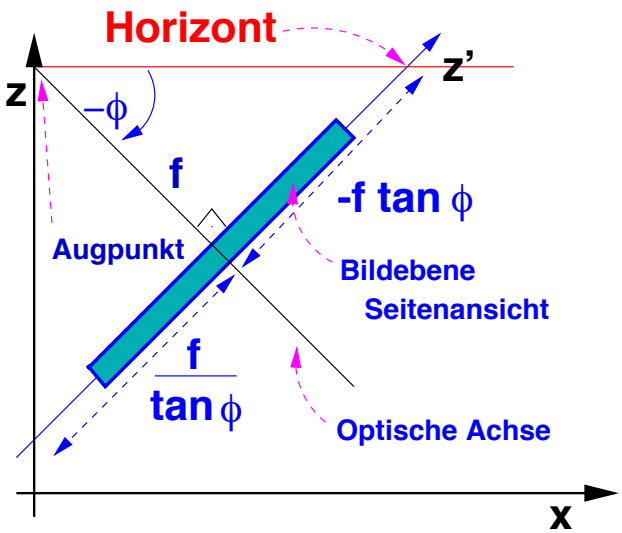


✎ **Anmerkung:** Auf der folgenden Folie sehen Sie eine *Einpunktprojektion*. Die z-Achse steht senkrecht auf der Bildebene. Nur die Parallelen zur z-Achse haben daher einen endlichen Fluchtpunkt.

Wäre dagegen das Weltkoordinatensystem auf den abgebildeten Kubus bezogen, dann hätten wir eine typische Zweipunktprojektion, wie sie gerne für Architekturaufgaben verwendet wird. Bei Abbildungen von Gebäuden wird es als unangenehm und unnatürlich empfunden, wenn vertikale Geraden einen Fluchtpunkt bilden (stürzende Linie). Solche stürzenden Linien werden vermieden, wenn die Kamera nicht gegen die Grundebene geneigt ist.



Der Horizont



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

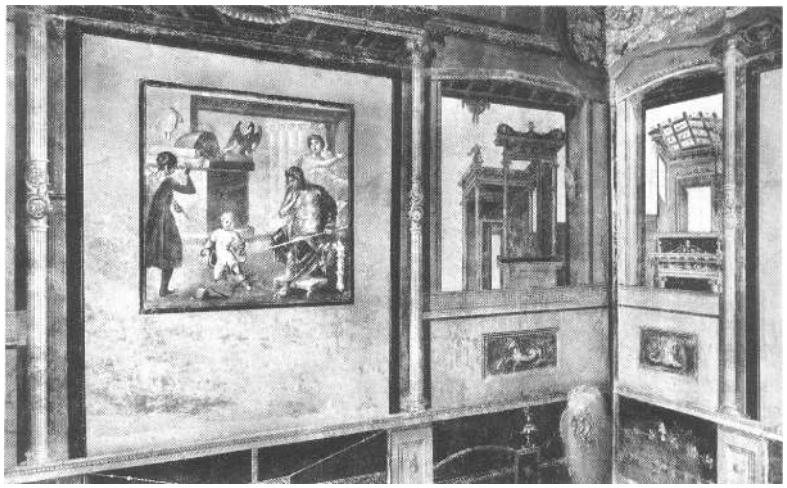
12.3 Perspektivische Elemente in der Malerei

Auf den folgenden Bildern sehen sie drei Beispiele dafür, wie in der Malerei die Perspektive berücksichtigt wurde. Das erste Beispiel ist eine römische Wandmalerei aus Pompeji (Casa dei Vettii), die um 70 n. Chr. entanden ist. Hier vergrößern perspektivisch richtig gezeichnete Bilder von Phantasiearchitekturen optisch den Raum.

Im Mittelalter wurde das Wissen über perspektivisches Zeichen im Abendland wieder vergessen und erst in der Renaissance wieder neu entdeckt. Das Bild von Duccio di Buonisegna „Die Versuchung Christi“ zeigt den Übergang von der flachen Byzantinischen Malerei (mit goldener Farbe gefüllter Hintergrund ohne Tiefe, flache, stereotype Figuren) zur perspektivischen Malerei der Renaissance (dekoriert mit perspektivisch wirkenden Gebäuden).

Das dritte Bild schließlich stammt vom Meister der Perspektive, Raffael von Urbino (1483–1520). Hier werden alle Register der Malkunst gezogen, um den Raum zu gliedern und die Figuren plastisch erscheinen zu lassen. Sie finden Texturgradienten, perfekte Schattierung und perspektivische Verkürzung (siehe Teil 9, „Räumliches Sehen“).

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



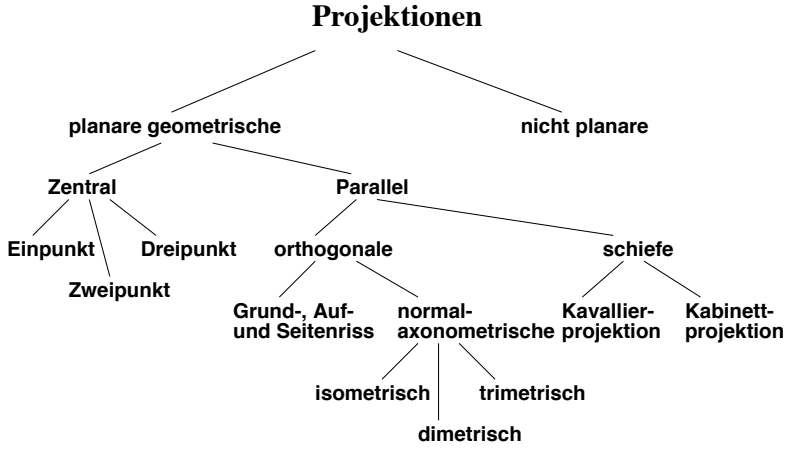
Duccio di Buonisegna

„Die Versuchung Christi“

Raffael von Urbino (1483–1520)

„Schule der Philosophen“

Übersicht: Projektionen

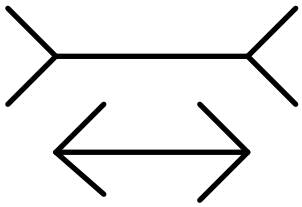


13 Illusionen

13.1 Interpretation von perspektivischen Effekten durch das visuelle System

Eine Reihe von optischen Täuschungen (Illusionen) kann mit der Annahme erklärt werden, daß das visuelle System des Menschen bei der Bewertung von Längenverhältnissen automatisch die perspektivische Verkürzung berücksichtigt. Diese Illusionen werden gerne von Zauberern genutzt, die vortäuschen wollen, daß ein Raum kleiner ist, als er in Wahrheit ist.

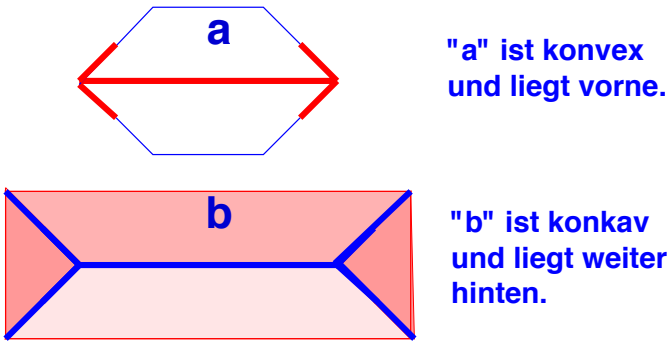
Welche Strecke ist länger? a oder b?



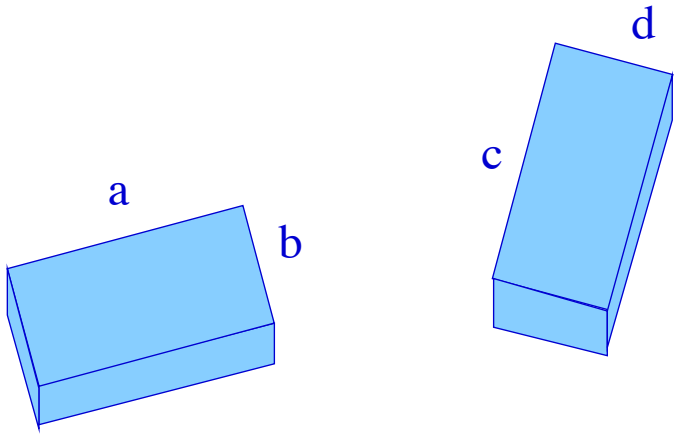
✎ **Anmerkung:** Beide Strecken a und b sind gleich lang. Strecke b erscheint aber länger. Warum? Unser Sehsystem vermittelt uns nicht die absoluten Längenmessungen auf der Retina sondern vorverarbeitete, interpretierte Maße. Es versucht Winkel perspektivisch zu interpretieren.

Eine Möglichkeit, die Illusion der unterschiedlichen Längen zu deuten, besteht darin, daß die beiden Linien wegen der angrenzenden Winkel als Kanten von Polyeders gedeutet werden. Strecke a scheint eine *Außenkante* eines konvexen Polyeders zu sein, Strecke b eine *Innenkante* eines konkaven Polyeders. Eine konkave, hinten liegende Kante ist vermutlich weiter entfernt als eine vorne liegende konvexe Kante. Da die Bilder auf der Netzhaut gleich groß erscheinen, die Entfernungen aber vermutlich unterschiedlich groß sind, sind auch wohl auch die absoluten Längen der Strecken unterschiedlich groß.

Erklärungsversuch: Konvexe und konkave Kanten



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



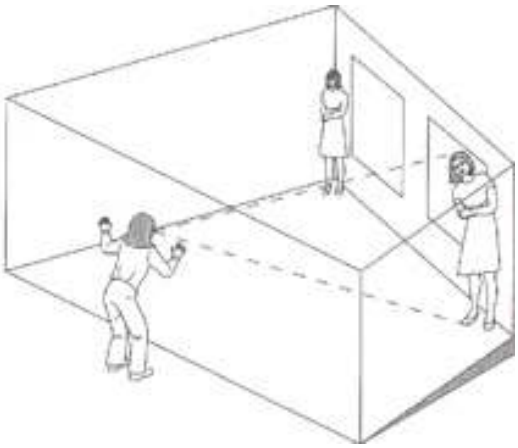
Streckenillusion: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movquaderBbasicQT.html

✎ **Anmerkung:** Die Rechtecke, die die Deckel der Quader bilden, sind völlig gleich, nur um 90° gegeneinander gedreht. Der eine Quader erscheint etwas schmaler und länger, weil vermutlich unser Sehsystem schon eine perspektivische Verkürzung berücksichtigt.

13.2 Scheinriesen und Scheinzwerge — der Ames-Raum

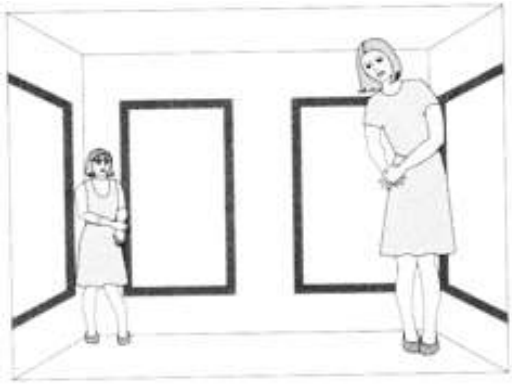
Beim Ames-Raum erscheint durch die vorgetäuschte Perspektive die rechts stehende Person gleich weit entfernt wie die linke, obwohl sie dichter beim Betrachter steht. Weil eine gleiche Entfernung angenommen wird, erscheint die rechte Person größer als die linke.

Der Ames-Raum: Blick von außen



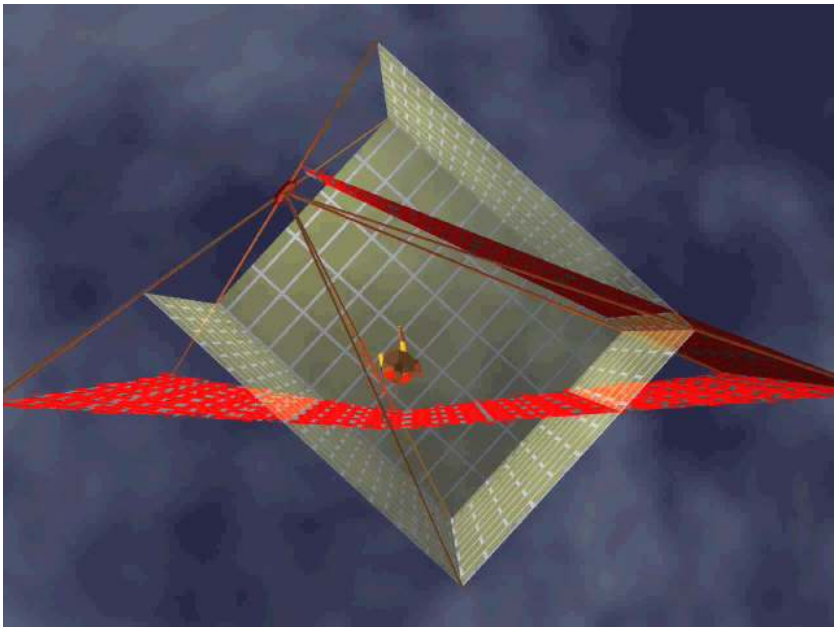
Der Ames-Raum:

Speziellen Betrachterstandort



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/amesPerspective.html

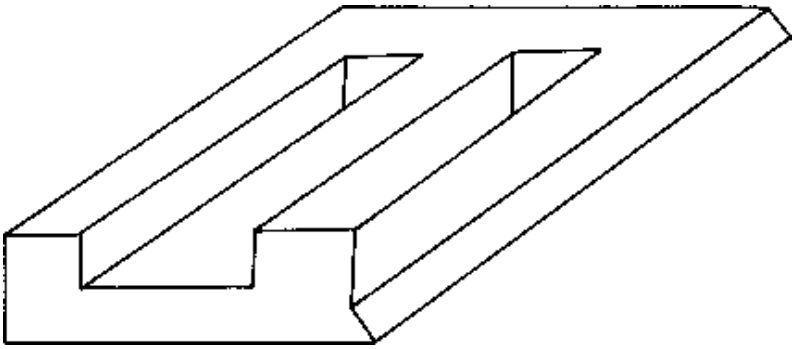
Der Ames-Raum: Wahre und scheinbare Wände



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/amesRevealed.html

13.3 Unmögliche Objekte

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

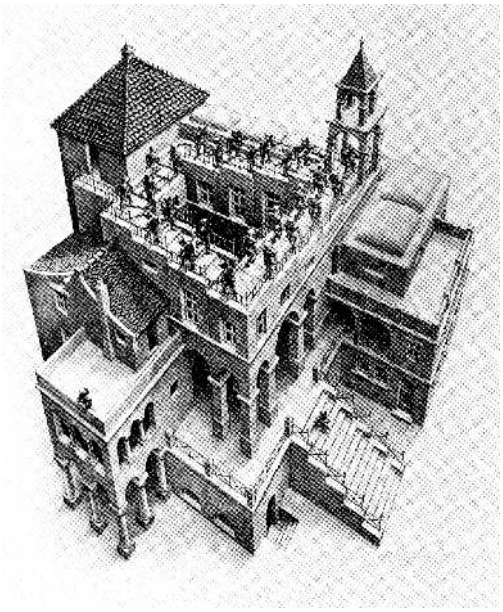
William Hogarth: Kupferstecher, 1697–1764

M.C. Escher: Grafiker, 1898–1972

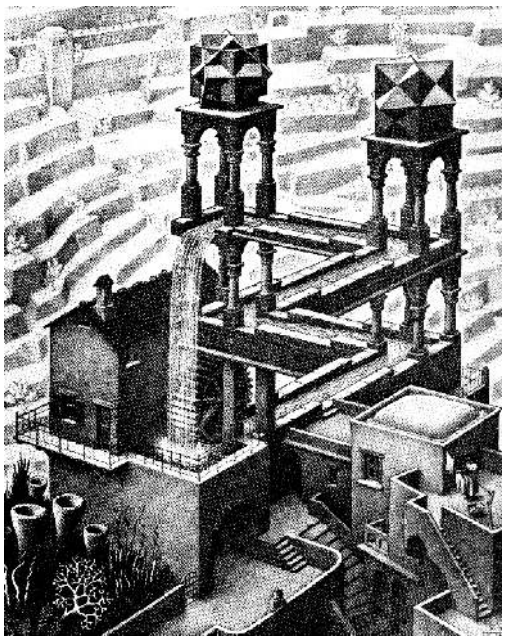
Der englische Kupferstecher William Hogarth (1697–1764) entwarf diesen Stich im Jahr 1794, um seinen Schülern die Perspektiven beim Zeichnen zu erklären. Er zeigt, welche Fehler unbegabte Künstler machen können und enthält somit viele unmögliche Objekte. So zündet z.B. im rechten, oberen Eck die aus dem Fenster lehnende Frau einem Mann die Pfeife an, obwohl dieser weit vom Haus entfernt auf einem Hügel steht.

Ein moderner Meister im Spiel mit der Perspektive ist der Grafiker M.C. Escher, von dem die beiden folgenden Bilder stammen (Wasserfall, 1961, Treppauf und treppab, 1960). Beide Bilder sind nach dem Prinzip des „Penrose Dreiecks“ konstruiert.

M.C. Escher



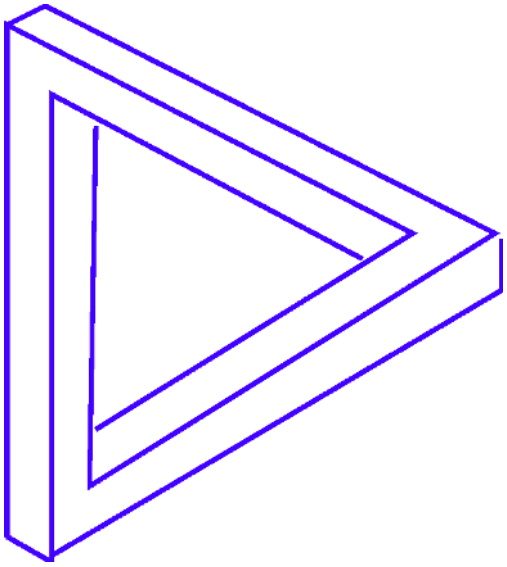
Treppauf und treppab, 1960



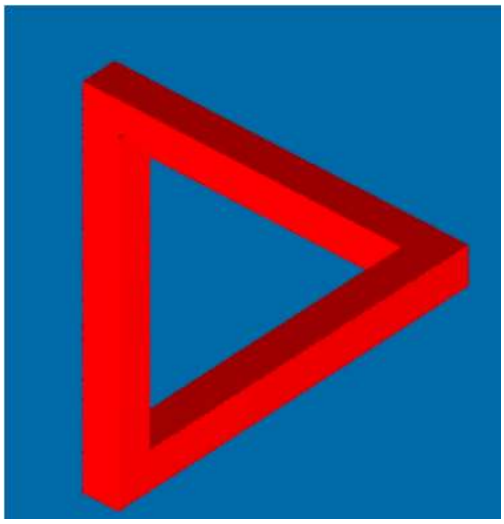
Wasserfall, 1961

Das „Penrose triangle“

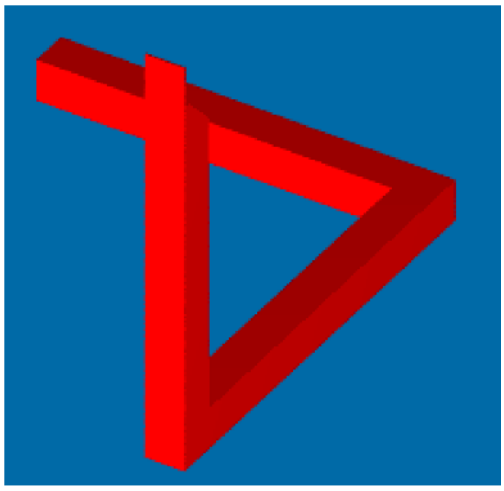
Ein unmögliches Objekt?



Dieses unmögliche Objekt gibt es wirklich.



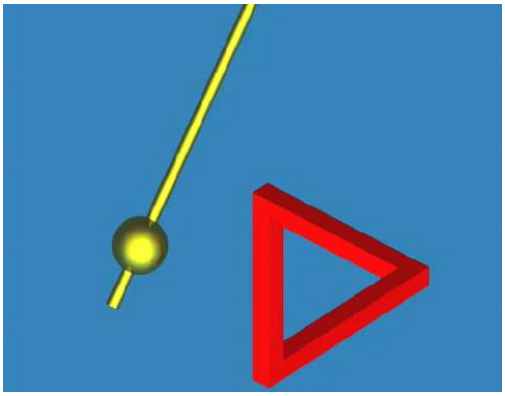
Aus einem anderen Blickwinkel betrachtet



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/unmoeglich.html

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/unmoeglich2.html

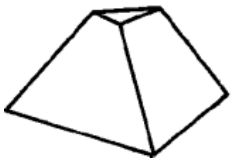
Das Pendel durchdringt das Objekt

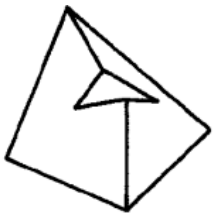


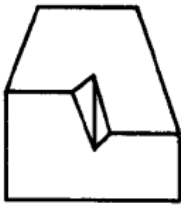
✎ **Anmerkung:** Unmögliche Polyeder Einige Linienzeichnungen stellen nur dann unmögliche Objekte dar, wenn alle Flächen, die mit Geraden begrenzt sind, als planare Polygone interpretiert werden.

Die folgenden Objekte sind keine unmöglichen Objekte, wenn die Flächen gekrümmt sein dürfen.

Linienzeichnungen, die keine 3D-Polyeder darsstellen:

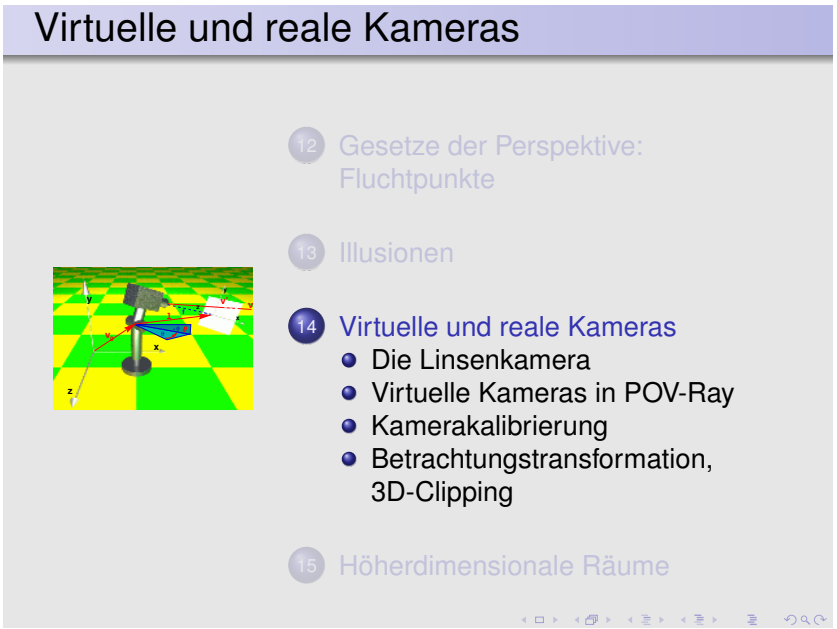






14 Virtuelle und reale Kameras

14.1 Die Linsenkamera



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:

$$
\begin{aligned}
s &= \text{Gegenstandsweite} \\
s' &= \text{Bildweite} \\
f &= \text{Brennweite} \\
\frac{1}{s} + \frac{1}{s'} &= \frac{1}{f}
\end{aligned}
$$

- 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.

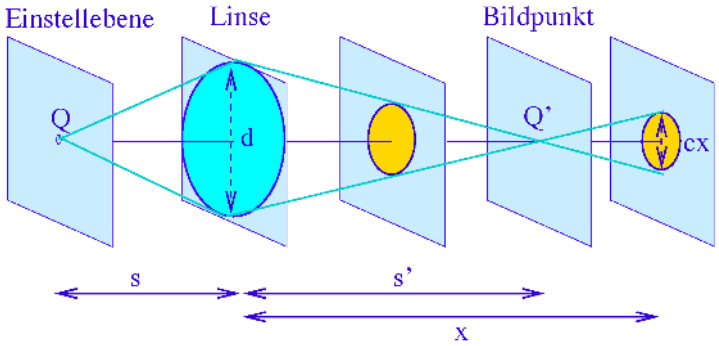
Nur die Blüte in der Einstellebene erscheint scharf.

✎ **Anmerkung:** Einstellebene

Wenn wir eine Szene mit einer Linsenkamera abbilden, müssen wir uns für eine Entfernung entscheiden, auf die wir fokussieren wollen. Das Objektiv wird dann so eingestellt, daß für diese Entfernung (Gegenstandsweite s) der Abstand der Linse von der Bildebene optimal ist. Die Abbildungsstrahlen konvergieren in einem Punkt[2]. Für Objektpunkte in anderer Entfernung von der Linse konvergieren die Strahlen vor oder hinter der Bildebene. Die Bilder solcher Punkte sind Kreise (Zerstreuungskreise genannt). Der Durchmesser $d(x)$ eines Zerstreuungskreises in der Entfernung x von der Linse ist proportional zum wirksamen Durchmesser d der Linse (Blendenöffnung, Apertur) und umgekehrt proportional zur Entfernung der Einstellebene und damit zum Abbildungsmaßstab. (Glassner, 1995)

[2]genaugenommen nicht in einem Punkt, sondern in einem Beugungsscheibchen, dessen Durchmesser umso größer ist, je kleiner die Apertur ist

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



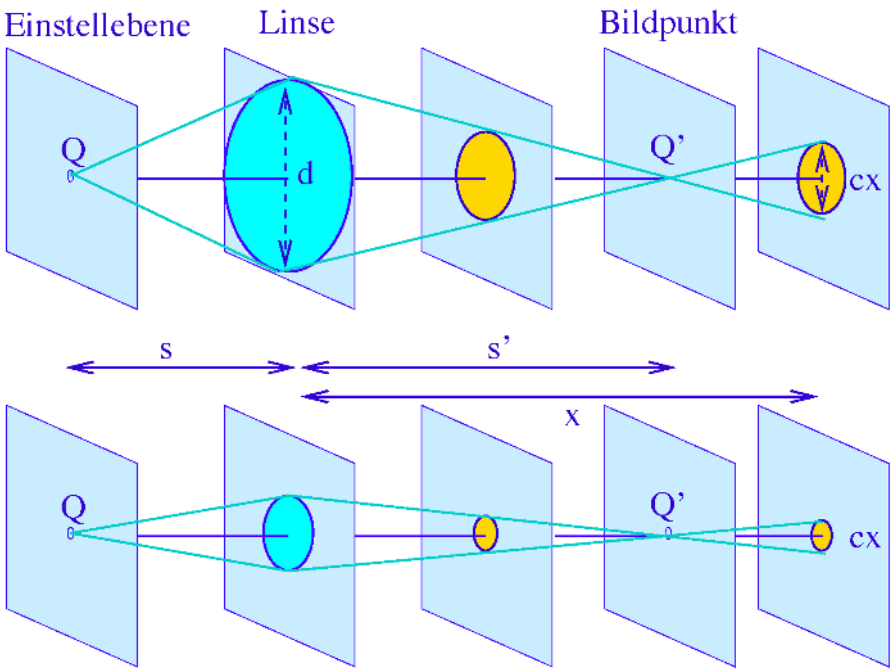
$$\frac{c(x)}{d} = \frac{x - s'}{s'} = \frac{\Delta_x}{s'}$$

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

Durchmesser des Zersteuungskreises

- Der Durchmesser des Zersteuungskreises 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, Zersteuungskreise



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: 21 (***Nah- und Fernpunkt***)
Die Ebenen, die den Bereich begrenzen, in dem bei einer gegeben 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: 22 (***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 Portraits

Landschaftsbild mit großer Tiefenschärfe

Gravitationslinsen

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

14.2 Virtuelle Kameras in POV-Ray

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: 23 (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 }
```

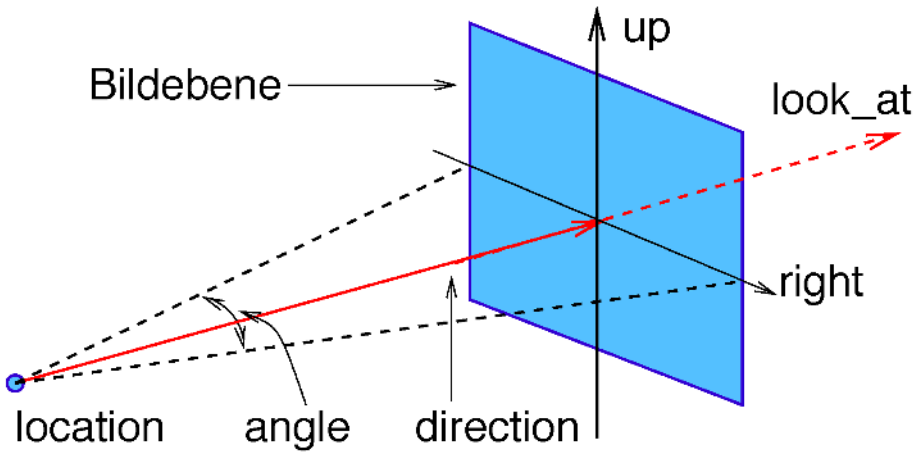
✎ **Anmerkung:** Die Parameter der Kamera: Der erste Parameter bestimmt die Art der Projektion: Zentralprojektion, Paralleprojektion usw. Der `location`-Parameter ist ein Vektor, der angibt, wo sich das Projektionszentrum befindet. Der `look_at`-Vektor legt die Richtung der Kamera fest. Er zeigt auf einen Punkt, der auf der optischen Achse liegt. Der `angle` legt indirekt die Brennweite fest. Es ist der Öffnungswinkel des Bildfeldes. *aperture* ist die simulierte Apertur und `focal_point` ein Punkt auf der Einstellebene.

Die vollständige Liste der Parameter: Für die Erklärung schauen Sie bitte ins POV-Ray-Handbuch.

```
 camera {
[ perspective | othographic | fisheye |
  ultra_wide_angle | omnimax | panoramic |
  cylinder FLOAT ]
```

```
location <vector>
look_at <vector>
up <vector>
right <vector>
direction <vector>
sky <vektor>
angle FLOAT
blur_samples
aperture FLOAT
focal_point
normal {NORMAL} }
```

Parameter der Kamera in POV-Ray



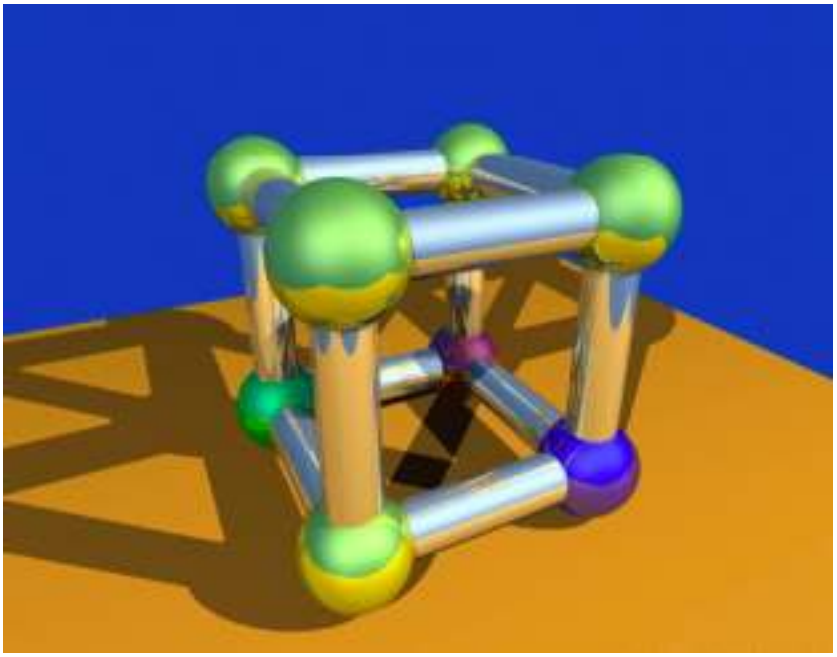
Die folgenden Beispiele stammen aus den POV-Ray-Demos, die bei der Suse-linux-Installation mitgeliefert wurden. Unter linux sollten sie im Verzeichnis `/usr/lib/povray3/pov3demo/camera` installiert sein.

Zentralprojektion

Beispiel: 24
Eine Kamera, Zentralprojektion, Öffnungswinkel 70°

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

Das erzeugte Bild (Beisp. 24)

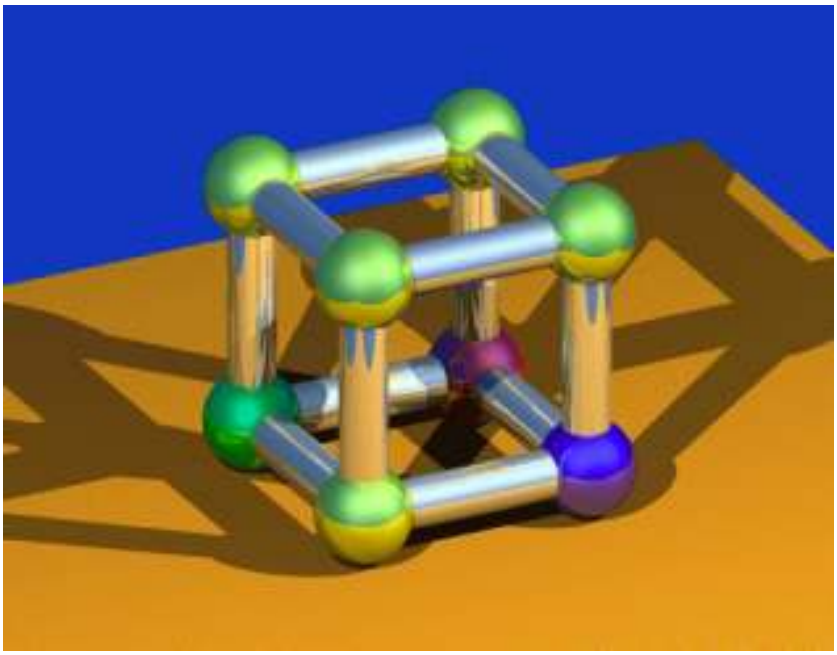


Orthogonale Parallelprojektion

Beispiel: 25
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)

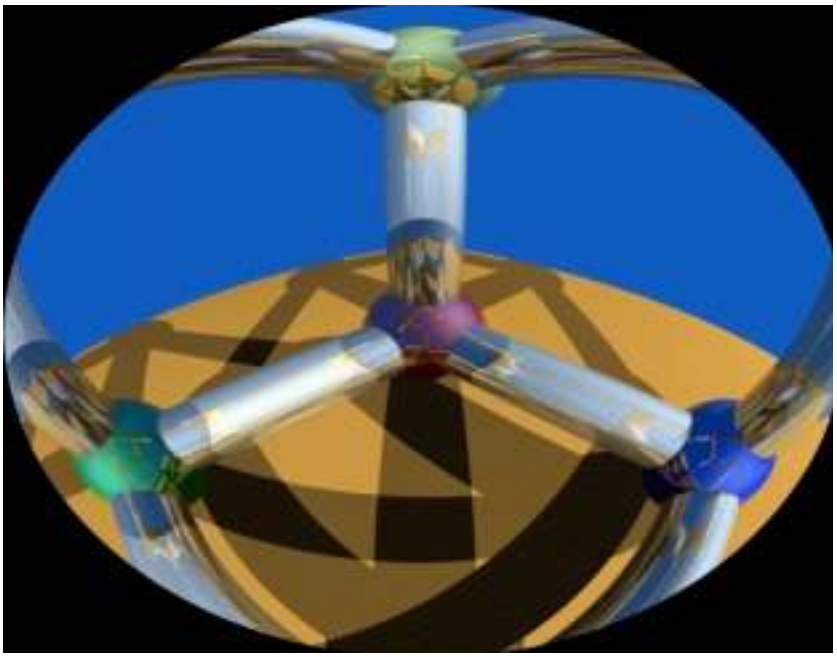


Fischauge mit 180°

Beispiel: 26
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)

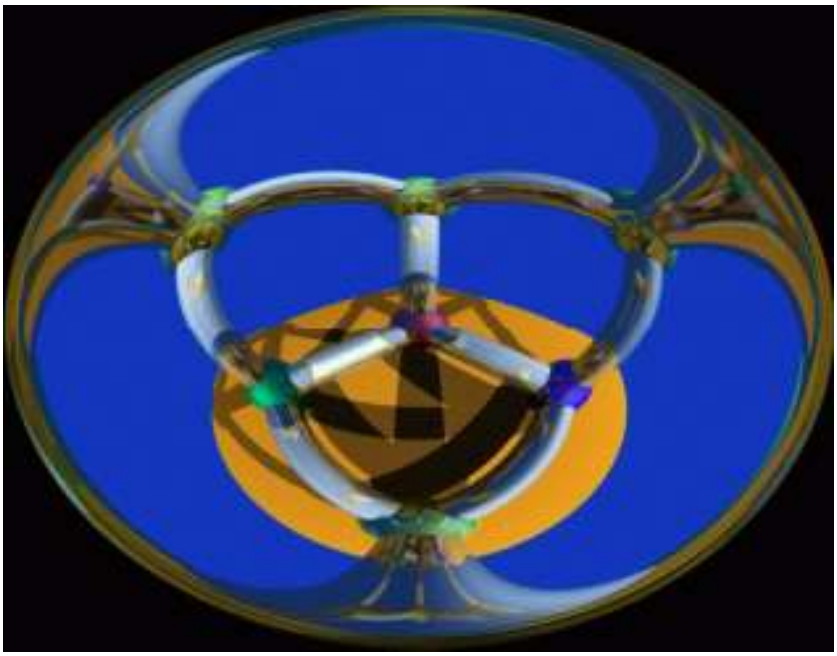


Fischauge mit 360°

Beispiel: 27
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°

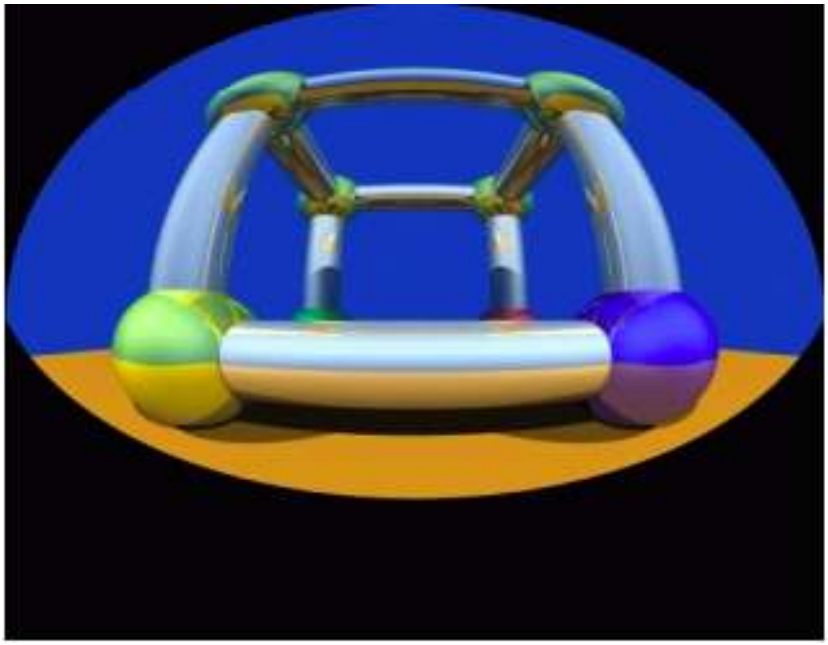


Omnimax-Projektion 180°

Beispiel: 28
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)

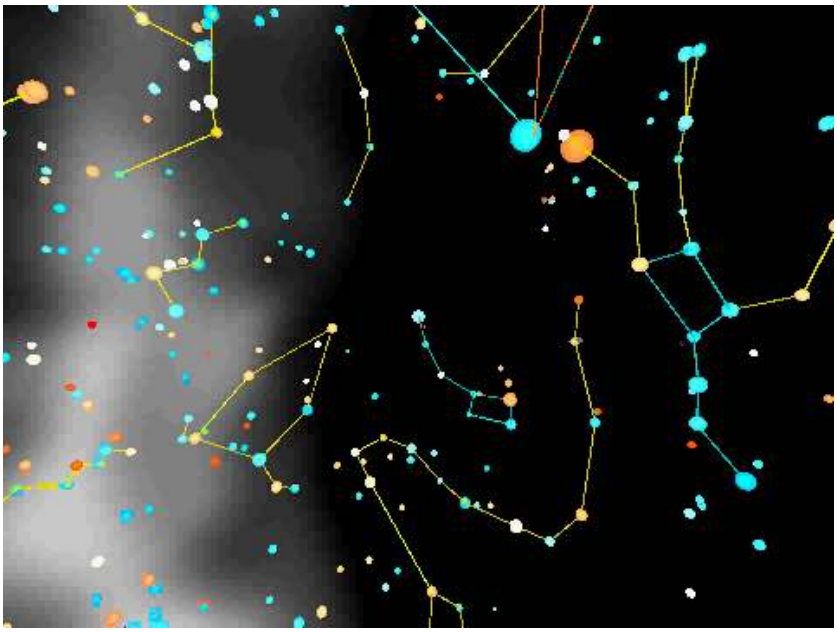


Ein Film mit bewegter Kamera: Tux in der Achterbahn http://kogs-www.informatik.uni-hamburg.de/~dreschle/informatik/Abschlussfilme_CGB/Pages/2001_Rollercoasters.html

14.2.1 Verzerrung der Zentralprojektion

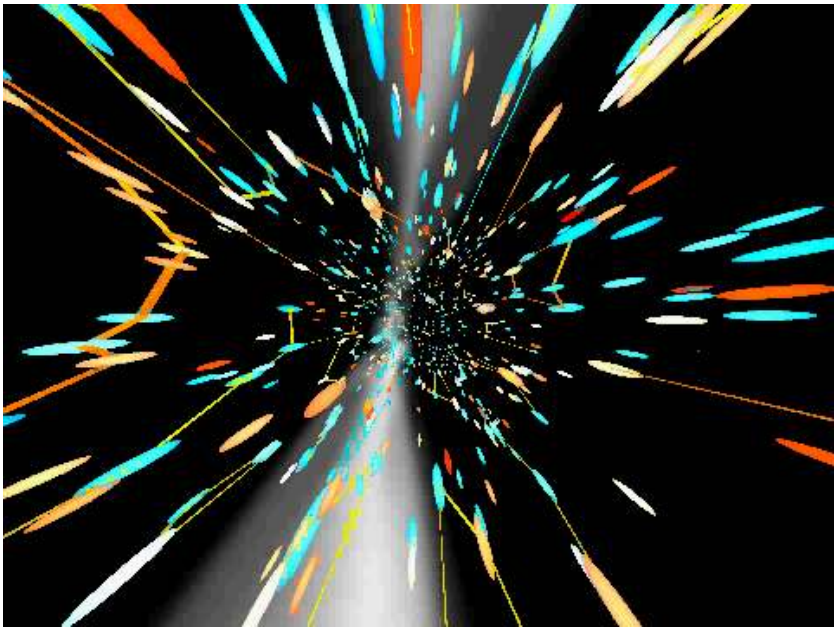
Verzerrung: Zentralprojektion

Zentralprojektion, Öffnungswinkel 90°, leichte elliptische Verzerrung



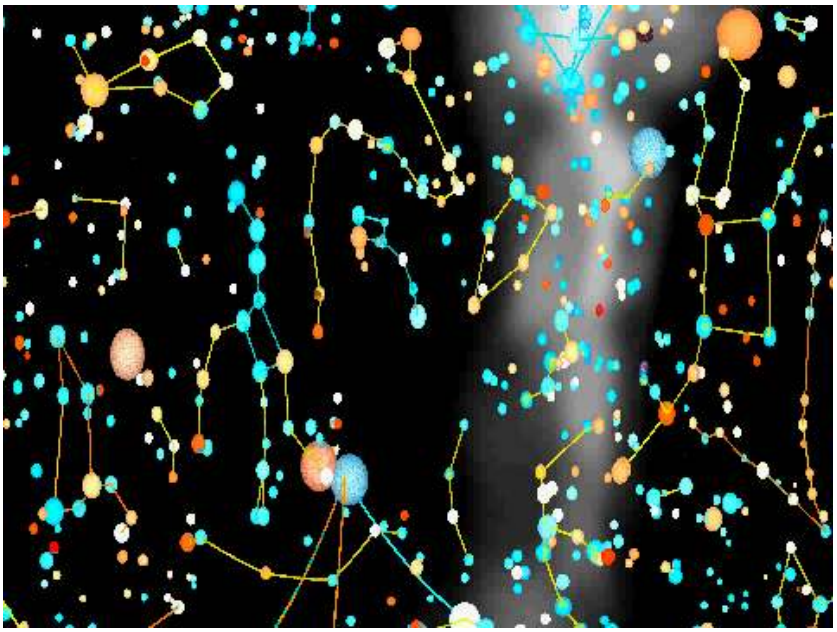
Verzerrung: Zentralprojektion

Zentralprojektion, Öffnungswinkel 170°, starke elliptische Verzerrung



Besser: Sphärische Projektion

Sphärische Projektion, Öffnungswinkel 170°, nur leichte elliptische Verzerrung



14.2.2 Fokussierungsunschärfe

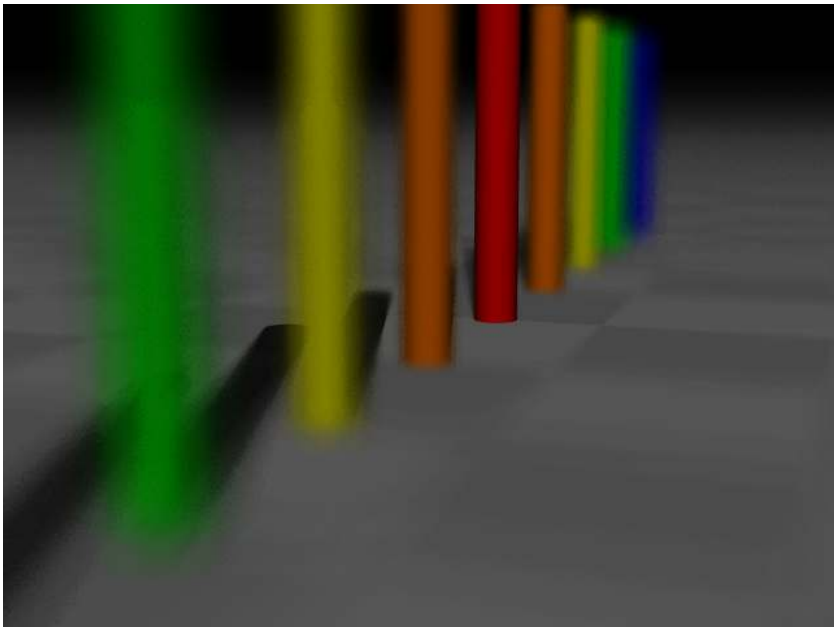
Fokussierungsunschärfe in POV-Ray

Beispiel: 29
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>
}
```

✎ **Anmerkung:** Der Parameter **focal_point** definiert einen Punkt der Einstellebene; je größer die Apertur ist, desto kleiner ist der Tiefenschärfebereich.

Das erzeugte Bild (Beisp. 29)



14.3 Kamerakalibrierung

In der Computergrafik benutzen wir virtuelle Kameras, deren Abbildungsparameter wir selbst bestimmen und daher sehr gut kennen. In der Bildverarbeitung dagegen, insbesondere in der Photogrammetrie, müssen die Parameter einer Kamera bestimmt werden, bevor die Bilddaten ausgewertet werden können. Dieser Vorgang heißt Kalibrierung der Kamera und ist nicht trivial, da sehr viele Parameter bestimmt werden müssen. Bei realen Kameras sind ja auch noch die Abbildungseigenschaften der Linse (Verzeichnung usw.) zu berücksichtigen, sowie die Koordinaten der Sensoren in der Bildebene.

Zur Vereinfachung der Kalibrierung unterscheidet man zwei Gruppen von Parametern: Innere und äußere Orientierung.

Innere Orientierung: Alle Parameter, die sich durch eine starre Bewegung der Kamera nicht verändern, beispielsweise die Brennweite und radiale Verzeichnung der Linse, Lage der CCD-Sensorfläche oder der Filmebene relativ zur optischen Achse usw.

Äußere Orientierung: Die Parameter der Kamera, die durch eine Bewegung der Kamera verändert werden, also die Lage der Kamera relativ zum Weltkoordinatensystem.

Die innere Orientierung kann an einem geeigneten Meßplatz ermittelt werden, bei dem die äußere Orientierung wohldefiniert ist. Beim Einsatz der Kamera an verschiedenen Einsatzorten bleibt die innere Orientierung erhalten und es muß dann nur noch jeweils die äußere Orientierung bestimmt werden.

Eine Kamera mit wohldefinierter innerer Orientierung für Vermessungsaufgaben heißt *Meßkammer*. Kameras mit Wechselobjektiven oder Zoom-Objektiven sind als Meßkammern für photogrammetrische Anwendungen ungeeignet, da sich bei solchen Optiken ständig die innere Orientierung ändert und die Kalibrierung unnötig erschwert wird.

Verfahren zur Kamerakalibrierung werden im Modul Bildverarbeitung 2 vorgestellt.

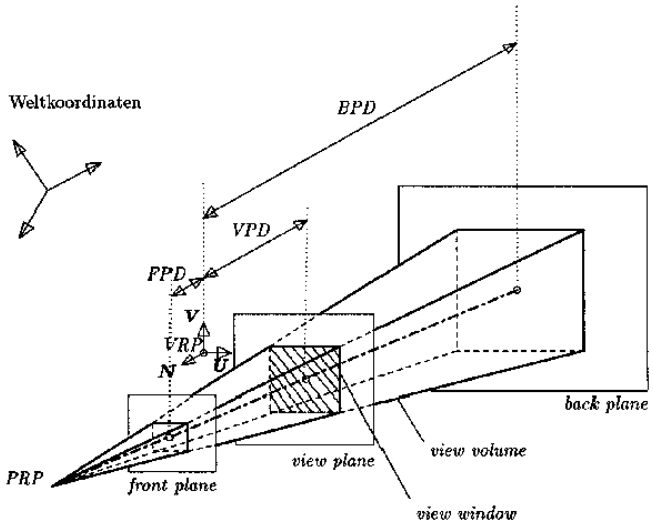
14.4 Betrachtungstransformation, 3D-Clipping

Projektionen lassen sich sehr elegant auf eine Koordinatentransformation zurückführen. Hierzu wird der Objektraum passend affin transformiert, so daß im transformierten Raum die xy-Bildkkoordinaten den xy-Objektkoordinaten entsprechen.

(siehe Skriptteil von W. Hansmann)

Betrachtungstransformation, 3D-Clipping

Der Bildraum: View-Reference Coordinate System



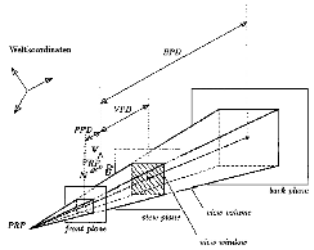
Festlegung des view coordinate systems durch view reference point (VRP), view plane normale (N), view plane distance (D) und view up vector (VUV).

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: EineEbene 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

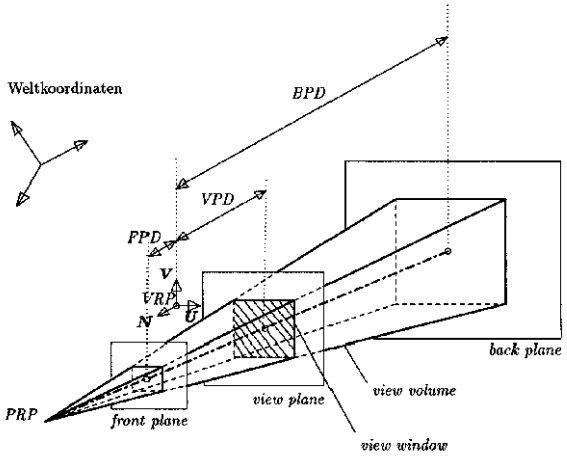
Läßt sich leicht durch Rotation und Translation aus dem Weltkoordinatensystem herleiten.

View Window und View Volume

Festlegung des abzubildenden Ausschnitts der Szene

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

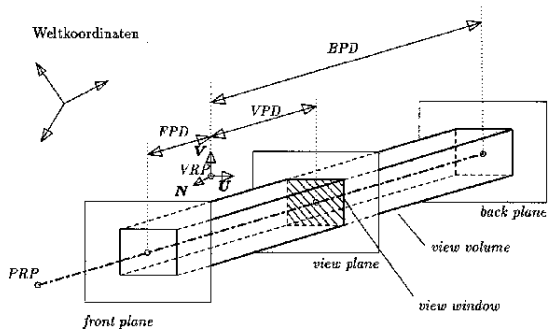
Durch das view window definierter Bildraum bei der Zentralprojektion



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)

Duch 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.

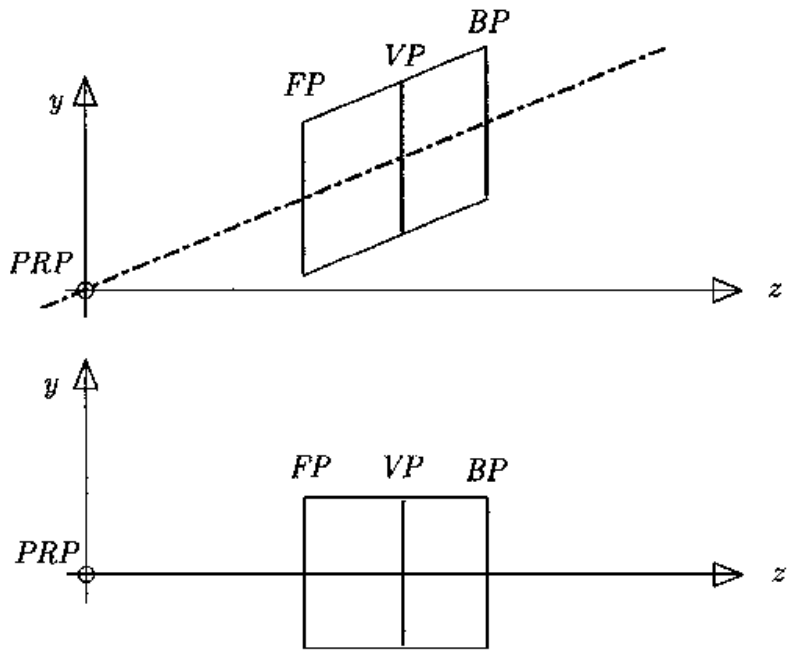
Betrachtungstransformation Parallelprojektion

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

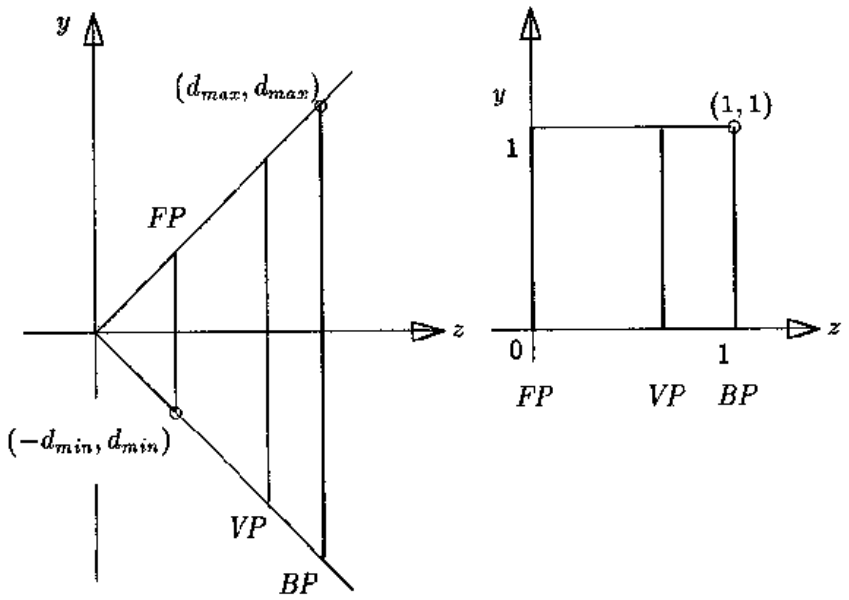
$\Longrightarrow$ Transformation des Bildraumes achsenparallel. (Fellner, 1992, S. 263)

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



15 Höherdimensionale Räume

15.1 Reduktion der Anzahl der Dimensionen

_ **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** $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ und ihre

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

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:

$$
\begin{aligned}
J_{ee} &= \frac{1}{\sum_{i<j} \delta_{ij}^2} \sum_{i<j} (d_{ij} - \delta_{ij})^2 \\
J_{ff} &= \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2 \\
J_{ef} &= \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}
\end{aligned}
$$

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

Optimierung durch Gradientensuche:

$$
\begin{aligned}
d_{ij} &= ||\vec{y}_i - \vec{y}_j|| \\
\vec{\nabla}_{\vec{y_k}} J_{ee} &= \frac{1}{\sum_{i<j} \delta_{ij}^2} \sum_{i \neq k} (d_{kj} - \delta_{kj}) \frac{\vec{y}_k - \vec{y}_j}{d_{kj}} \\
\vec{\nabla}_{\vec{y_k}} J_{ff} &= 2 \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \frac{\vec{y}_k - \vec{y}_j}{d_{kj}} \\
\vec{\nabla}_{\vec{y_k}} J_{ef} &= \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \frac{\vec{y}_k - \vec{y}_j}{d_{kj}}
\end{aligned}
$$

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

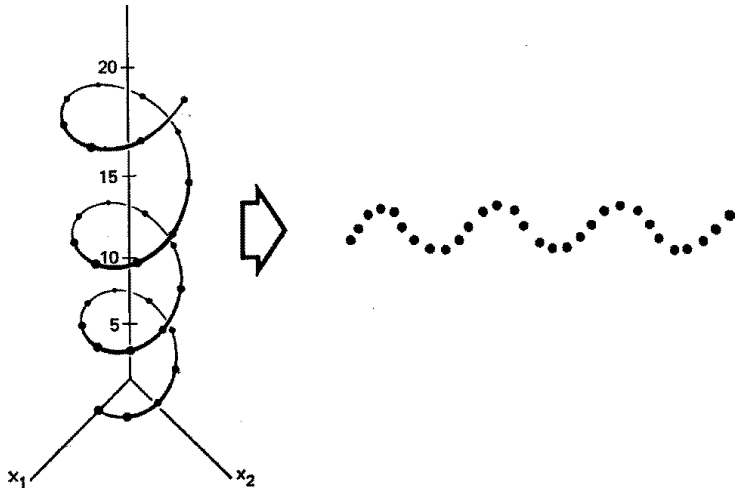
Beispiel: 30 Punkte einer Helix

Beispiel: 30

$$
\begin{aligned}
x_1(k) &= \cos x_3(k) \\
x_2(k) &= \sin x_3(k) \\
x_3(k) &= \frac{k}{\sqrt{2}}, \quad k = 0, 1, \ldots, 29.
\end{aligned}
$$

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

Reduktion der Dimensionalität von 3D auf 2D



15.2 Abbildung der räumlichen Dimensionen auf andere Parameter

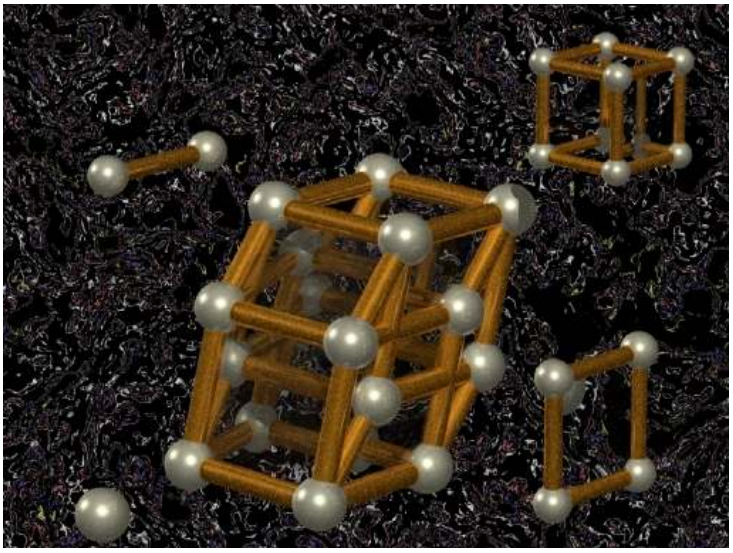
Visualisierung von Objekten mit mehr als 3 Dimensionen

- 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.

15.3 Projektion in Unterräume

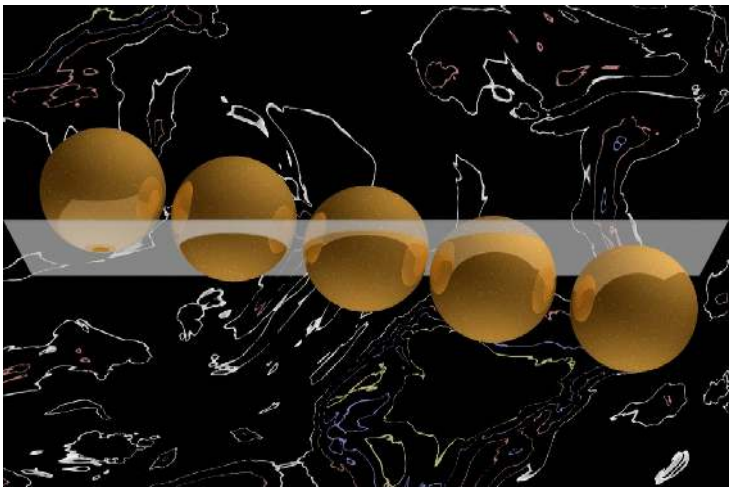
Von Objekten mit mehr als drei Dimensionen können wir mit unserem Gesichtssinn nur drei räumliche Dimensionen erfassen. Wir sehen nur ihre Projektion in die uns vertrauten drei Dimensionen. was sehen wir von einem vierdimensionalen Würfel?

Würfel in 1, 2, 3 und 4 Dimensionen



Ein Gedankenexperiment

Eine dreidimensionale Kugel durchdringt eine Ebene.



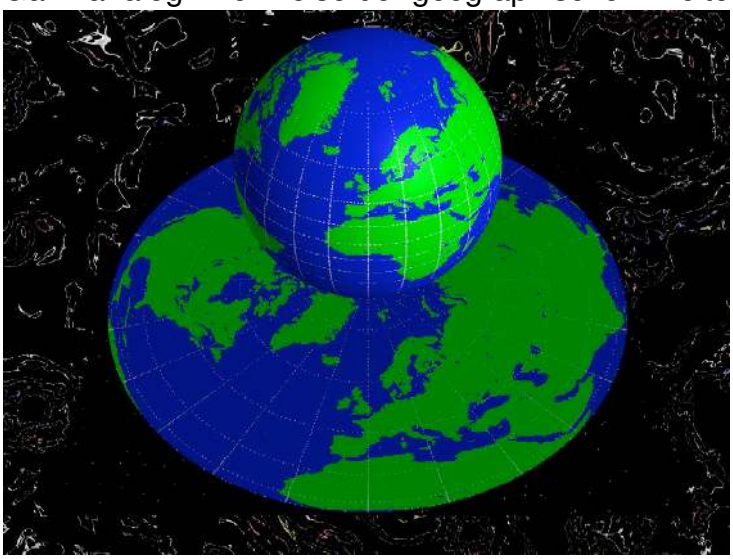
Ein Gedankenexperiment:

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



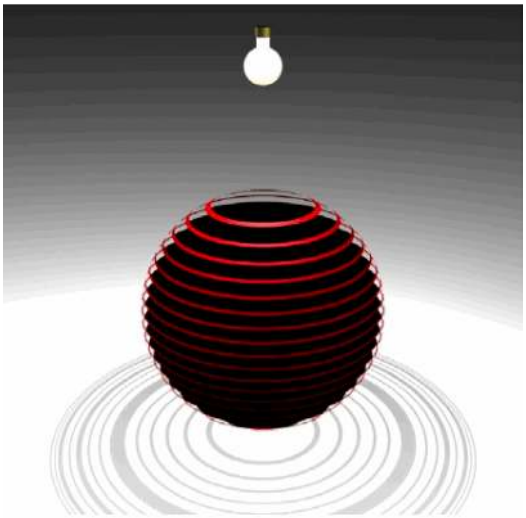
Ein zweidimensionales Abbild der Kugeloberfläche:

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.

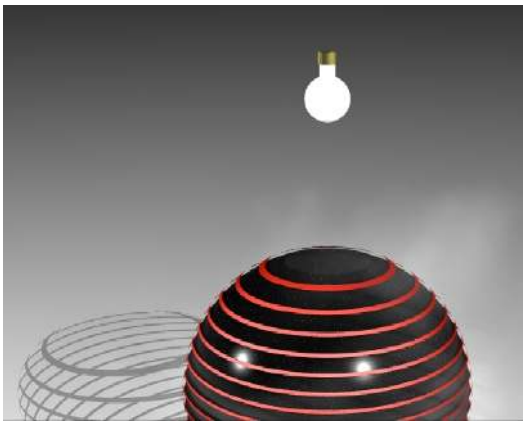
Projektion senkrecht zu den Schnittebenen

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.



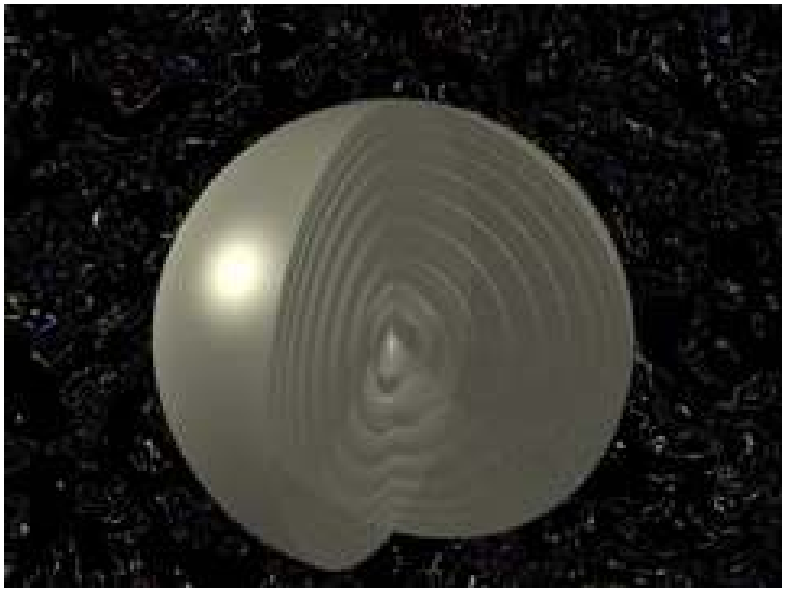
Eine 4D-Einheitskugel im Ursprung:

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

- 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.

✎ **Anmerkung:** Auf der folgenden Folie sieht man die Projektion einer 4D-Kugel in den dreidimensionalen x_1, x_2, x_3-Unterraum entlang der x_4-Achse. Die dargestellten Kugeln sind die Projektionen von Schnitten der 4D-Kugeloberfläche mit Hyperebenen, deren Normale die x_4-Achse ist. Diese Schnitte bilden also konzentrische Kugeln. Wenn die Schnittebenen nicht orthogonal zur x_4-Achse sind, dann sind die Projektionen der Schnitte mit der Kugeloberläche in den x_1, x_2, x_3-Unterraum Ellipsoide, analog zu den Ellipsen, die sich bei schräger Projektion von Kreisen auf der Oberfläche einer 3D-Kugel ergeben. Damit man in das Innere der Kugeln und Ellipsoide hineinsehen kann, wurde der x-z-Sektor herausgeschnitten.

Eine 4D-Einheitskugel im Ursprung:



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movwanderkanimQT.html

Eine 4D-Einheitskugel im Ursprung

geschnitten mit einer Hyperebene in beliebiger Lage

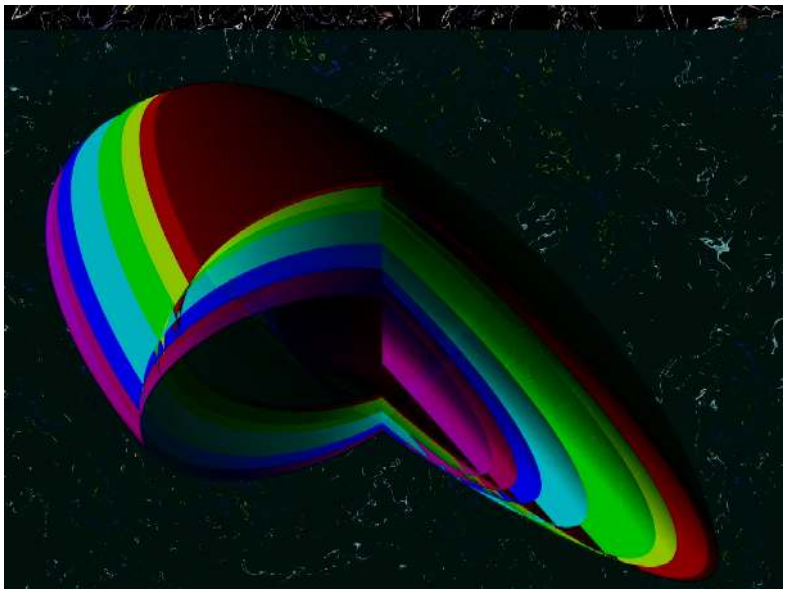
$$
\begin{array}{lrcl}
\text{Einheitskugel:} & x_1^2 + x_2^2 + x_3^2 & = & 1 - x_4^2 \\
\text{Hyperebene:} & Ax_1 + Bx_2 + Cx_3 + x_4 + \delta_i & = & 0 \\
\text{einsetzen für } x_4^2\text{:} & (Ax_1 + Bx_2 + Cx_3 + \delta_i)^2 & = & x_4^2 \\
 & (1 + A^2)x_1^2 + (1 + B^2)x_2^2 + (1 + C^2)x_3^2 + \delta_i^2 & & \\
 & +2ABx_1x_2 + 2ABx_1x_3 + 2ABx_2x_3 & & \\
 & +2A\delta_i x_1 + 2B\delta_i x_2 + 2C\delta_i x_3 & = & 1
\end{array}
$$

Die Projektionen der Schnitte sind also gescherte Ellipsoide.

✎ **Anmerkung:** Wir sehen, daß die Schnitte der Oberfläche einer Hyperkugel mit parallelen Hyperebenen in beliebiger Lage sich als Ellipsoide in den x_1, x_2, x_3 Unterraum projizieren (Flächen zweiten Grades). Die Ellipsoide sind nicht konzentrisch, sondern in Abhängigkeit von δ_i leicht geschert und durchdringen sich gegenseitig.

Folie 177 zeigt diese Ellipsoide, gerechnet mit den folgenden Parametern für die Hyperebenen: $A = 1, B = 1, C = 1, \delta_i \in \{0.05, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Jedes Ellipsoid ist in einer anderen Farbe dargestellt, damit man das Durchdringen besser erkennen kann.

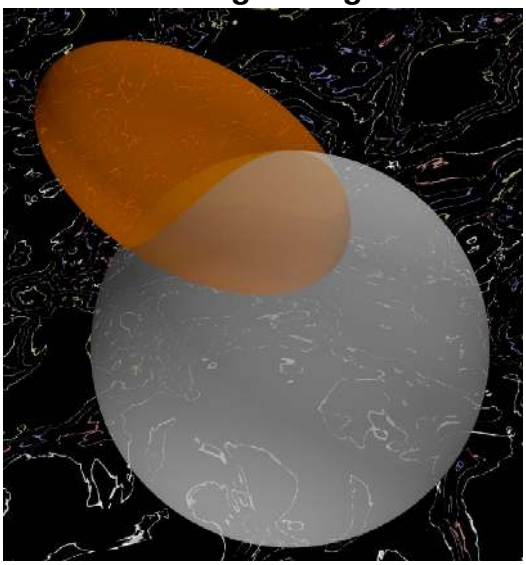
Ellipsoide auf der Oberfläche einer 4D-Kugel



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movwanderellipsanimQT.html

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movwanderellipsufo1QT.html

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 gekrümmt 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.

✎ **Anmerkung:** Nach demselben Prinzip können wir Teile aus der Oberfläche einer Hyperkugel ausschneiden, deren Projektionen in den dreidimensionalen Unterraum Flächen vierten Grades sind. Ein Beispiel dafür läßt sich über die Hopf-Kreise (benannt nach dem Mathematiker Heinz Hopf) konstruieren.(Peterson, 1988)

Um Hopf-Kreise zu finden, projizieren wir die Punkte (X_1, X_2, X_3, X_4) der Oberfläche einer vierdimensionalen Einheitskugel auf die Oberfläche einer dreidimensionale Kugel nach den Gleichungen (Hopf-Karte):

$$\begin{aligned} x_1 &= 2(X_1X_2 + X_3X_4) \\ x_2 &= 2(X_1X_4 - X_2X_3) \\ x_3 &= 2(X_1^2 + X_3^3) - (X_2^2 + X_4^2) \end{aligned}$$

Alle Punkte (X_1, X_2, X_3, X_4) der 4D-Kugel, die auf denselben Punkt (x_1, x_2, x_3) der Hopf-Karte abgebildet werden, definieren einen Hopf-Kreis auf der Oberfläche der vierdimensionalen Kugel.

Ein Kreis auf der Hopfkarte steht damit für eine unendliche Menge von Hopf-Kreisen auf der 4D-Kugel. Diese Menge von Hopf-Kreisen schneidet einen Teil aus der Oberfläche der 4D-Kugel aus, dessen Projektion eine Fläche 4. Ordnung ist, nämlich ein Torus.

Wenn wir alle durch Kreise auf der Hopfkarte definierten Teile der Oberfläche einer Hyperkugel in den dreidimensionalen Teilraum projizieren, dann erhalten wir eine unendliche Menge von ineinandergeschachtelten Tori.

Projizierte Hopf-Kreise

IVC Interactive Visual Computing

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 5:

Photometrische Grundlagen

Universität Hamburg
Fachbereich Informatik

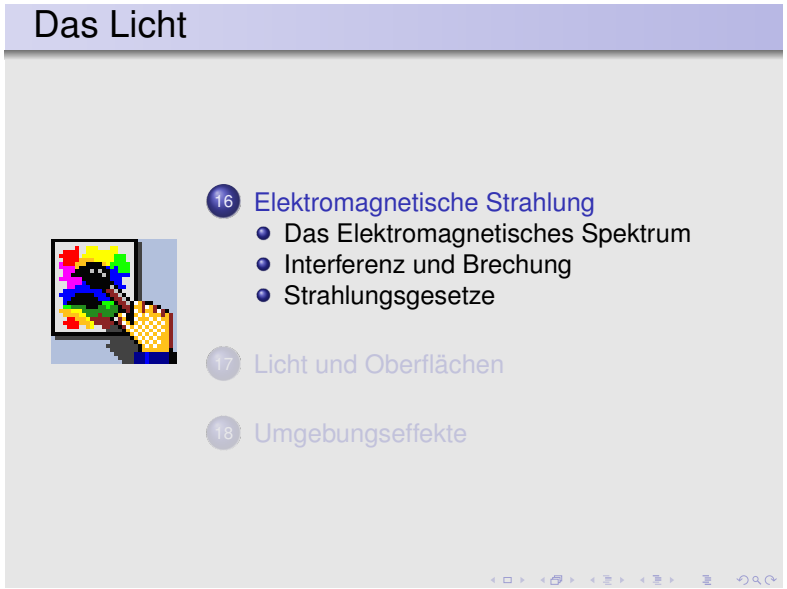
WS 2014/2015

Teil VI

Das Licht

16 Elektromagnetische Strahlung

16.1 Das Elektromagnetisches Spektrum



✎ **Anmerkung:** Licht: Licht ist eine *sichtbare* elektromagnetische Wellenstrahlung mit Wellenlängen von etwa 4000 bis 8000 Å. Im übertragenen Sinn wird auch längerwellige und kürzerwellige elektromagnetische Strahlung als Licht bezeichnet (infrarotes und ultraviolettes Licht), obwohl solche Strahlung nicht sichtbar ist.

Normalerweise sendet eine Lichtquelle Licht sehr vieler verschiedener Wellenlängen aus. Wird diese Strahlung nach Wellenlängen geordnet nebeneinander gelegt, so erhält man das Spektrum der Strahlung.

Die Ausbreitungsgeschwindigkeit des Lichtes, die Lichtgeschwindigkeit, beträgt etwa $300000\frac{\text{km}}{\text{s}}$, genauer $299790\frac{\text{km}}{\text{s}}$ im Vakuum. Die Lichtgeschwindigkeit ist beim Durchgang des Lichtes durch Materie im allg. kleiner als im Vakuum.

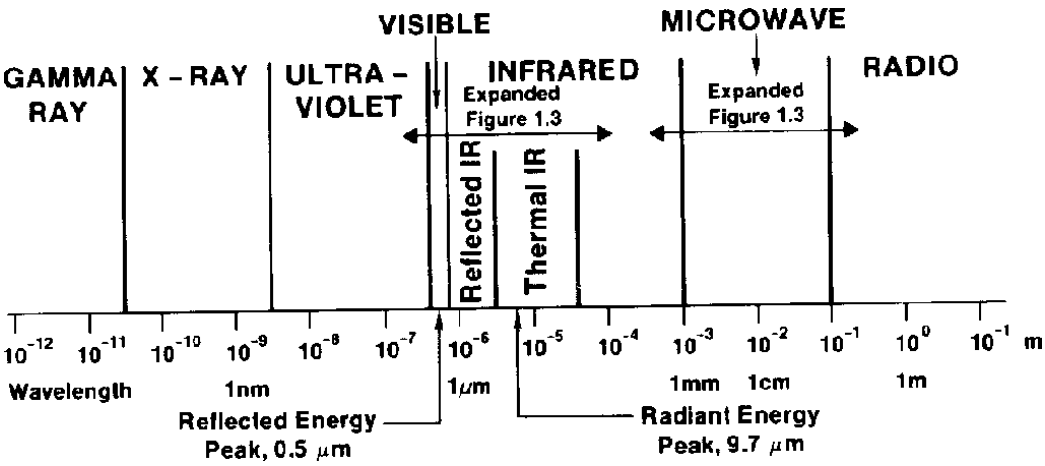
Normalerweise erfolgen im Licht die die Schwingungen in allen zur Fortpflanzungsrichtung senkrechten Richtungen. Erfolgen die Schwingungen jedoch bevorzugt in *einer* Richtung (senkrecht zur Fortpflanzungsrichtung), so nennt man das Licht *polarisiert*.

Mit der *Intensität* des Lichtes ist die auf die Flächeneinheit bezogene Strahlungsleistung gemeint, also z.B. die in Joule gemessene Energie, die pro Sekunde auf eine $1m^2$ große Fläche fällt, welche *senkrecht* zur Fortpflanzungsrichtung steht. Die Messung der Helligkeit ist die Aufgabe der Photometrie (Lichtmessung). (Weigert and Zimmermann, 1971)

Die folgenden Folien zeigen die wichtigsten Bereiche des elektromagnetischen Spektrums und dazu die ungefähre Tiefe, bis zu der die senkrecht auf die Erdoberfläche fallende Strahlung verschiedener Wellenlängen in die Erdatmosphäre eindringt. Bei der durch die Kurve angegeben Höhe über dem Erdboden ist die Intensität der Strahlung auf 37% gesunken.

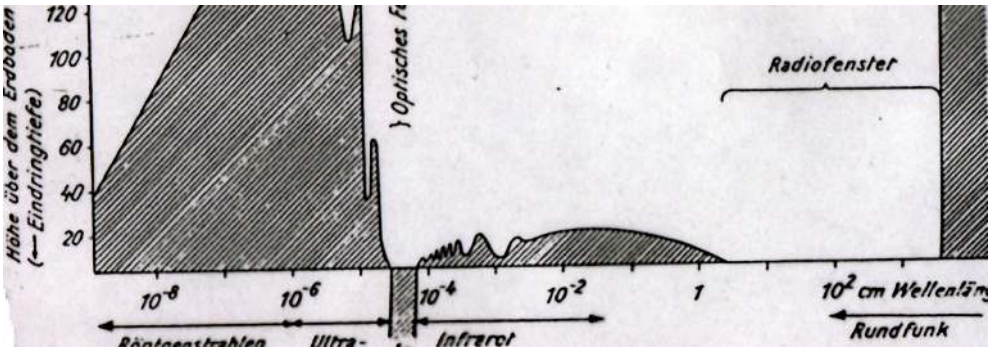
Beachten Sie die beiden „Fenster“, das optische Fenster und das Radiofenster, für die unsere Atmosphäre besonders gut durchlässig ist.

Das elektromagnetische Spektrum



Das elektromagnetische Spektrum

Die Eindringtiefe in die Atmosphäre, optisches Fenster und Radarfenster



Spektrale Bänder

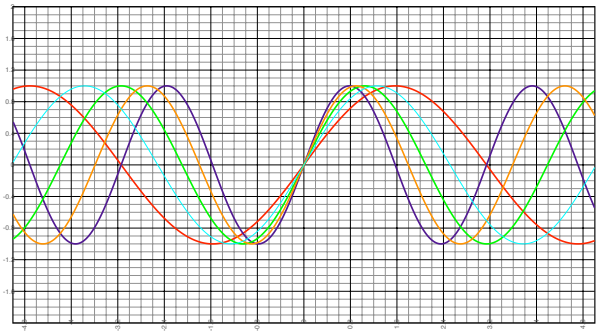
Gammastrahlen	< 0.03 nm
Röntgenstrahlen	$0.03 - 3.0$ nm
UV-Strahlen	$0.03 - 0.4\mu$m
Photographisches UV-Band	$0.3 - 0.4\mu$m
Sichtbares Licht	$0.4 - 0.7\mu$m
Infrarot	$0.7 - 100\mu$m
Photographisches IR-Band	$0.7 - 3\mu$m
Thermisches IR-Band	$3 - 5\mu$m, $8 - 14\mu$m
Mikrowellen	$0.1 - 30$ cm
Radar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

Eine Animation der Feldstärken: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV2_Movies/Pages/movPlanareEMWelleGroQT.html

Eine Animation der Vektoren der elektrischen und magnetischen Feldstärken bei einer planaren elektromagnetischen Welle: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV2_Movies/Pages/movEmagnWelleFelderQT.html

Das elektrische Feld und das Magnetfeld schwingen mit gleicher Phase. Die Welle breitet sich senkrecht zu beiden Feldern aus.

Licht als elektromagnetische Welle



Weißes Licht ist ein Gemisch elektromagnetischer Wellen unterschiedlicher Wellenlängen.

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movlightwaveWeissAllesQT.html

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.

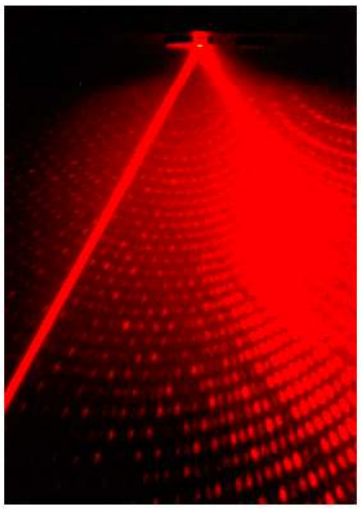
Lichtquanten (Photonen): Wenn Licht als Strahlung von Teilchen unterschiedlicher Energie modelliert wird, lassen sich Phänomene der Wechselwirkung zwischen Materie und Licht (Absorption, Streuung, Lumineszenz usw.) erklären.

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: $c = 299790$km/s.
- Ein Lichtquant hat eine Energie E
$$E = h \cdot \nu = \frac{h \cdot c}{\lambda}, \quad \text{wobei } h = 6,6260693 \cdot 10^{-34}\text{J s}$$

16.2 Interferenz und Brechung

Interferenz



✎ **Anmerkung:** Licht und Materie:

Wenn Licht auf Materie trifft, so wird ein Teil wieder zurückgeworfen (*Reflexion*), ein Teil absorbiert. Bei der Absorption wird die Energie des Lichts in andere Energieformen gewandelt, beispielsweise Wärmeenergie.

Die unregelmäßige Ablenkung, die das Licht durch kleine Teilchen erfahren kann, bezeichnet man als *Streuung*.

Im Kapitel über „Photorealismus“ werden wir uns genauer mit Modellen für Reflexion, Streuung und Absorption befassen. Eine möglichst genaue Modellierung der Interaktion von Licht und Materie ist in der Computergrafik für eine photorealistische Darstellung von virtuellen Szenen wichtig, und in der Bildverarbeitung für die Klassifikation von Materialien anhand ihrer spektralen Signatur. Für beide Zwecke muß die Beeinflussung eines Lichtstrahls auf dem gesamten Weg von der Lichtquelle durch die Atmospäre zur reflektierenden Objektoberfläche und bis zum bildgebenden System berücksichtigt werden.

✎ **Anmerkung:** Lichtbrechung: Durch die unterschiedliche Geschwindigkeit des Lichtes durch zwei verschiedene Stoffe tritt an deren Grenzfläche eine *Brechung* des Lichtes auf. Sei c_0 die Lichtgeschwindigkeit im Vakuum und c die Lichtgeschwindigkeit in einem bestimmten Stoff. Das Verhältnis beider Geschwindigkeiten $n = \frac{c_0}{c}$ wird die *Brechzahl* oder der Brechungsindex genannt und ist für jeden Stoff eine charakterische Konstante.[3]

[3] Zur Notation: Die Brechzahl wird in der deutschen Literatur meist mit n bezeichnet, in der amerikanischen mit η und heißt dann *index of refraction.*

Die Größe der Richtungsänderung wird durch das Brechungsgesetz gegeben:

Der Sinus der Winkel α und β, die ein Lichstrahl vor und nach der Brechung mit dem Einfallslot, d.h. das im Auftreffpunkt errichtete Lot auf der Grenzfläche, einschließt, verhalten sich wie die Lichtgeschwindigkeiten c_1 und c_2 in den beiden Stoffen:

$$\frac{\sin\alpha}{\sin\beta} = \frac{c_1}{c_2}$$

Die Lichtgeschwindigkeit in Materie ist eine Funktion der Wellenlänge λ, so daß auch die Brechzahl von der Wellenlänge abhängt. Konsequenzen dieser Wellenlängenabhängigkeit der Brechzahl sind beispielsweise die *Dispersion* des Lichts beim Durchgang durch ein Prisma oder die chromatische Aberration, der Abbildungsfehler von Sammellinsen, der darin besteht, daß die Linse für jede Wellenlänge einen anderen Brennpunkt hat. Ein analoges Phänomen tritt für Zerstreuungslinsen auf.

Moderne Linsen für Photoapparate werden daher nicht nur aus einem einzigen Glas hergestellt, sondern verschiedene Gläser werden so kombiniert, daß die chromatische Aberration weitgehend kompensiert wird. Das ist besonders bei Zoom-Objektiven sehr schwierig.

Lichtbrechung beim Übergang Luft / Wasser

Hier funkeln Diamanten (mit konstantem Brechungsindex ohne Dispersion gerechnet): http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movdiamondbQT.html

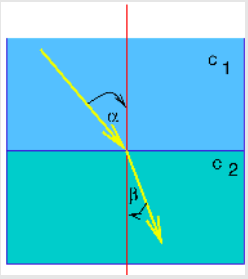
Die Brechzahl (der Brechungsindex) n

α: Austrittswinkel im Medium 1

α: Eintrittswinkel im Medium 2

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

n: Brechzahl



Satz (Brechungsgesetz)

$$\frac{\sin\alpha}{\sin\beta} = \frac{c_1}{c_2}$$

$$n = \frac{c_0}{c} \qquad (1)$$

Lichtbrechung in POV-Ray

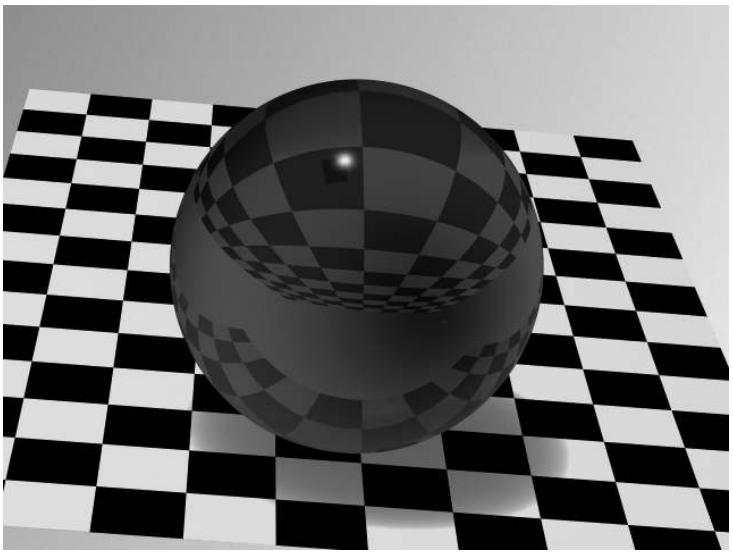
- Der Brechungsindex n von transparenten Objekten wird seit der POV-Ray-Version 3.5 in einem **interior**-statement spezifiziert.
- Vordefinierte Brechungsindizes und weitere Konstanten finden Sie im file consts.inc.
- Vordefinierte transparente Materialien finden Sie in glass.inc.
- Beispiel: Eine Kugel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

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

Der **filter** -Parameter sagt, wie transparent das Pigment des Objektes ist. Mit einem Filter-Wert von eins wäre es vollständig transparent.

Die Angabe von „**hollow**“ spezifiziert, daß das Objekt hohl ist. Auch wenn unsere Diamantkugel gar nicht hohl sein soll, ist die „**hollow**“-Spezifikation notwendig, wenn die „**interior**“-Spezifikation wirksam sein soll.

Eine Diamantkugel



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

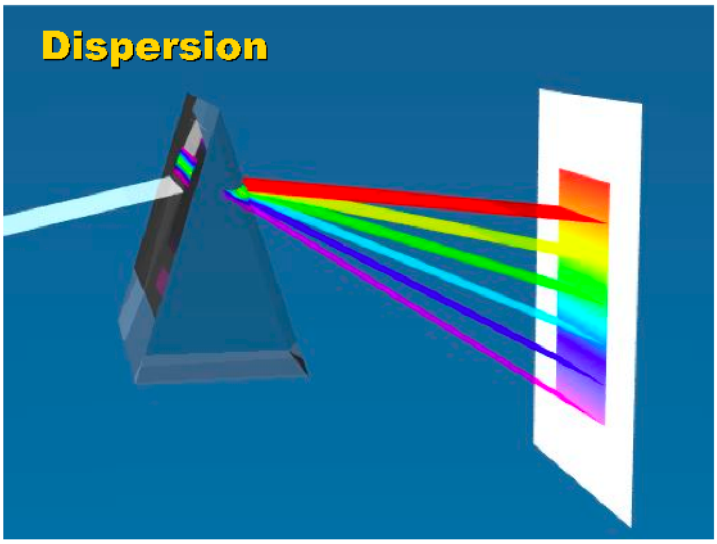
Beachte:

- 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.

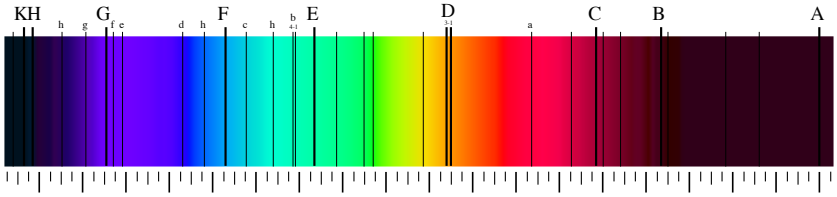
Dispersion von weißem Licht beim Durchgang durch ein Prisma



Definition: 31 (***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**:** $\omega = \frac{n_F - n_C}{n_D - 1}$.

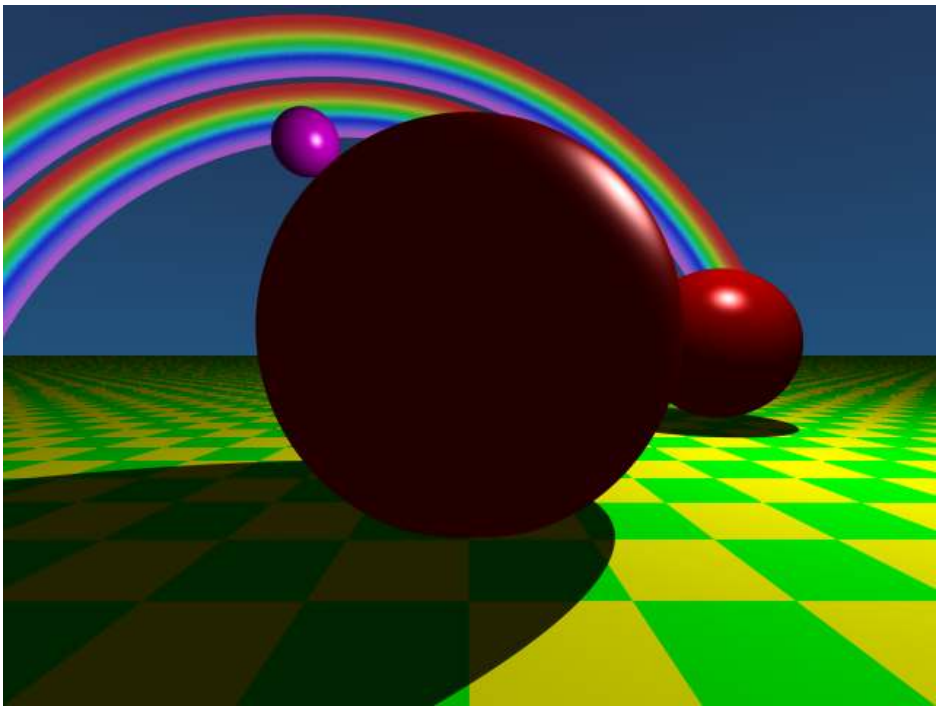
Abbesche Zahl ν**:** $\nu = \frac{1}{\omega}$.

Abbildung aus: http://de.wikipedia.org/wiki/Fraunhoferlinie

Ein Regenbogen

Hast-Paß, Neuseeland

Ein Regenbogen in POV-Ray (Beispiel: 32)



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: 32

```
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]  }}

// Persistence Of Vision raytracer version 3.5 sample file.
// File by Dieter Bayer.
// updated for compatibility with POV-Ray v3.5
// by Ken Tyler on 08-26-01
//
// -w320 -h240
// -w800 -h600 +a0.3

global_settings { assumed_gamma 2.2 }
```

```
#include "colors.inc"

camera {
   location  <0, 20, -100>
   direction <0,  0,    0.7>
   up        <0,  1,    0>
   right    <4/3,  0,    0>
}

background { color SkyBlue }

// declare rainbow's colours

#declare r_violet1
        = colour red 1.0 green 0.5 blue 1.0 filter 1.0;
#declare r_violet2
        = colour red 1.0 green 0.5 blue 1.0 filter 0.8;
#declare r_indigo
        = colour red 0.5 green 0.5 blue 1.0 filter 0.8;
#declare r_blue
        = colour red 0.2 green 0.2 blue 1.0 filter 0.8;
#declare r_cyan
        = colour red 0.2 green 1.0 blue 1.0 filter 0.8;
#declare r_green
        = colour red 0.2 green 1.0 blue 0.2 filter 0.8;
#declare r_yellow
        = colour red 1.0 green 1.0 blue 0.2 filter 0.8;
#declare r_orange
        = colour red 1.0 green 0.5 blue 0.2 filter 0.8;
#declare r_red1
        = colour red 1.0 green 0.2 blue 0.2 filter 0.8;
#declare r_red2
        = colour red 1.0 green 0.2 blue 0.2 filter 1.0;

// create the rainbow

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]
    }}

  rainbow {
    angle 37
    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]
    }}

  sky_sphere {
    pigment {
      gradient y
      color_map {
        [0 colour SkyBlue]
        [1 colour MidnightBlue]
      }
      scale 2
      translate <-1, -1, -1>
```

```
  }}

/* Put down the beloved famous raytrace
   green/yellow checkered floor */
plane { y, -10
   pigment {
      checker colour Yellow colour Green
      scale 20
   }
   finish {
      ambient 0.2
      diffuse 0.8 }}

sphere { <0, 25, 0>, 40

   pigment {Red}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20
   }}

sphere { <-100, 150, 200>,  20
   pigment {Magenta}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20
   }}

sphere { <100, 25, 100>, 30
   pigment {Red}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20
   }}
```

```
light_source {<100, 120, 40> colour White}
```

Brechzahlen für transparente Materialien

Brechzahl n

Fraunhoferlinie λ[Å]	C 6563	D 5890	F 4862	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

Zur Erzeugung fotorealistischer Bilder, die transparente Materialien wie Glas, Wasser, Ölfilme usw. abbilden, muß die Dispersion für alle sichtbaren Wellenlängen berücksichtigt werden. Daher sind die folgenden Näherungsformeln wichtig, mit deren Hilfe sich die Brechzahl in Abhängigkeit von der Wellenlänge berechnen läßt. Die erste Formel (von Sellmeier) gilt für einen weiten Bereich des elektromagnetischen Spektrums und ist recht genau, ist aber aufwendiger zu berechen als Cauchys Näherung, die nur im sichtbaren Bereich gilt und etwas ungenauer ist. Für einfache Animationen ist Cauchys Formel ausreichend.

Sellmeier Formel: Sellmeiers Modell der Dispersion geht davon aus, daß jedes Material eine oder mehrere Resonanzfrequenzen besitzt, bei denen die Moleküle mit dem Licht in Resonanz geraten und Interferenzen hervorrufen.
Bei nur einer Resonanzfrequenz $\nu_0 = \frac{c}{\lambda_0}$ gilt die Formel:

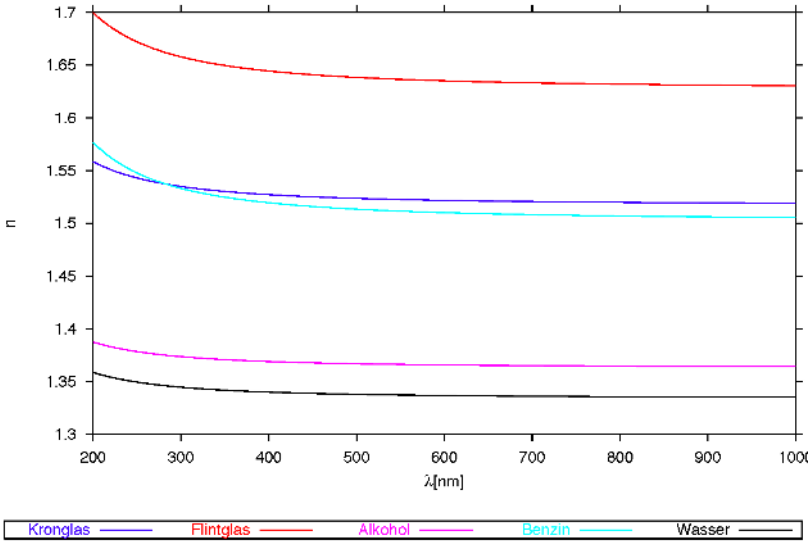
$$n_\lambda^2 = 1 + \frac{A\lambda^2}{\lambda^2 - {\lambda_0}^2}$$

A ist eine materialspezifische Konstante. A und λ_0 können aus zwei bekannten Brechzahlen abgeleitet werden (Glassner, 1995) . [4]

[4] Die Herleitung bei Glassner enthält einen Flüchtigkeitsfehler. Das Aufspüren sei zur Übung empfohlen!

✎ **Anmerkung:** Die Cauchy Formel: Die Cauchyformel beruht auf einer Reihenentwicklung für n_λ. Es wird angenommen, daß $\lambda \gg \lambda_0$ und die Terme höherer Ordnung dann wegen $\frac{\lambda_0}{\lambda} \to 0$ vernachlässigt werden können. (Glassner, 1995, S. 570) Die Näherung ist daher nur für den Bereich des sichtbaren Lichts anzuwenden und auch dort nicht sehr genau. Es wird aber die Berechnung von Wurzeln vermieden, so daß sie gut für aufwendige ray tracing Anwendungen zu verwenden ist.

Sellmeierformel: Die Brechzahl in Abhängigkeit von der Wellenlänge



Dispersion in POV-Ray

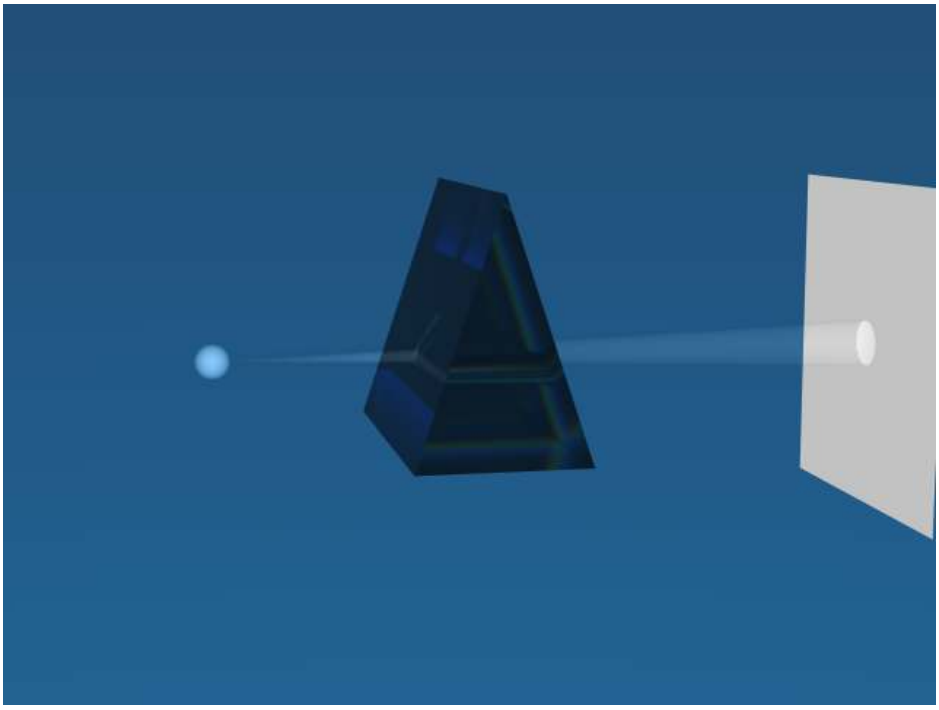
- In der **interior**-Spezifikation kann neben dem Brechungsindex auch die Dispersion spezifiziert werden.
  - dispersion: Das Verhältnis der Brechungsindizes von Violett to Rot.
  - dispersion_samples: Zahl der Farbstufen, mit denen das Spektrum gerechnet wird,
  - caustic: Simulierte Kaustiken (Lichtbündelingseffekte)
  - **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: 33
Brechung mit Dispersion: Eine **interior**-*Spezifikation, um auch die Dispersion sichtbar zu machen.*

```
interior { ior Diamond_Ior
           dispersion 2 //Diamond Dispersion = 1.035
           fade_distance 1
         fade_power 2
         caustics 1.0
           dispersion_samples 100 }
```

Der Lichweg durch ein Prisma (Beispiel: 33)



Beachte: Die Dispersion wird nur angenähert simuliert. Der Lichtstrahl durch das Prisma, der die Leinwand beleuchtet, wird nicht aufgespalten.

Das POV-Ray-Skript für das Prisma mit Punktlichquelle und Nebel, um den Lichtstrahl sichbar zu machen.

```
//Persistence of Vision Ray Tracer Scene Description File
#version unofficial MegaPov 1.21;
global_settings { assumed_gamma 2.2 }
#include "colors.inc"
#include "textures.inc"
#include "finish.inc"
#include "consts.inc"

camera {
   location  <-4, 3, -20>
   look_at <0,  0, 0>
   angle 30}

sky_sphere {
  pigment {
    gradient y
    color_map {
      [0 colour SkyBlue]
      [1 colour MidnightBlue]     }
    scale 2
    translate <-1, -1, -1> }}


#declare screenpos = <5.01,0,0>;
#declare lamppos = < -3, 0, 0>;

#declare lampshape =
                  cone {
        lamppos+0.02*x, 0.1,  screenpos+0.1*x, .3}

light_source {
        lamppos
        rgb <1.000000, 1.000000, 1.000000>
        spotlight
        point_at screenpos
        radius 2
        falloff 2
        tightness 2
```

```
        media_attenuation
}
#declare prisma=prism{linear_sweep -2, 2,
          4, // number of points
          <0,1>, <2,1>, <1,-2>, <0,1>
      hollow
        pigment {White filter 0.95 }
        finish {
          specular 0.7
         roughness 0.001
         ambient 0
         diffuse 0
        reflection {
         0.2, 1.0
      fresnel on }
          conserve_energy
  }
          interior { ior Diamond_Ior
                     dispersion 2 //Diamond Dispersion = 1.035
                        fade_distance 1
                     fade_power 2
                     caustics 1.0
                     dispersion_samples 100 }
         rotate 90*x
         translate -x}

object{prisma}

#declare screenWidth= <0,0,2>;
#declare screenHeight= <0,2,0>;
 polygon {//screen
 5, screenpos+screenHeight-screenWidth, //<5.01, 2,-2>,
 screenpos+screenHeight+screenWidth, //<5.01,2,2>,
  screenpos-screenHeight+screenWidth, //<5.01,-3,2>,
  screenpos-screenHeight-screenWidth, //   <5.01,-3,-2>,
  screenpos+screenHeight-screenWidth //<5.01,2,-2>
    pigment{White}
   finish {ambient .8 diffuse .6}
 }

 sphere // transparent sphere containing media
```

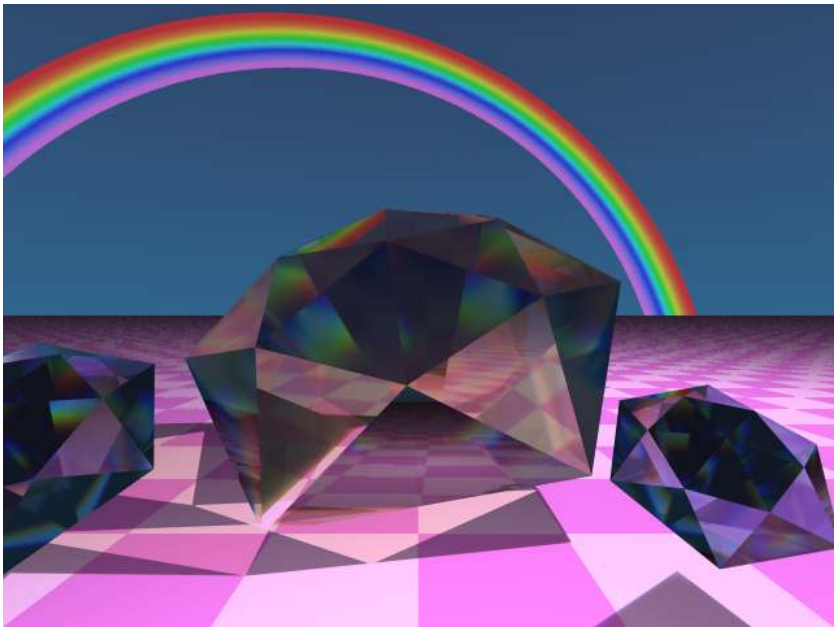
```
 { lamppos-0.02*x, 0.2 pigment { rgbt 1 } hollow
   interior
   { media
     { emission 1
       density
       { spherical density_map
         { [0 rgb 1]
           [0.4 rgb <1,0,0>]
           [0.8 rgb <1,1,0>]
           [1 rgb 0]}
         scale 0.2} }} }

object{lampshape
 pigment { rgbt 1 } hollow
   interior
   {media {
        scattering {
                1,       //mie_hazy_scattering
                rgb <1.000000, 1.000000, 1.000000>
                extinction 0.1} }}}
```

Diamanten mit Dispersion und Kaustiken

Beispiel: 34
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}
```

Diamanten mit Dispersion



Die Berechnung der Dispersion ist sehr aufwendig. Auf einem 2 GHz Intel Core Duo hat die Berechnung des Diamantenbildes mit Dispersion ca. 11 Minuten gedauert.

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movdiamondDispersion.html

```
//Persistence of Vision Ray Tracer Scene Description File

#include "colors.inc"
#include "textures.inc"
#include "metals.inc"
#include "finish.inc"
#include "consts.inc"

#version unofficial MegaPov 1.21;

camera {
    location   <0, 20, -100>
    direction  <0,  0,    0.7>
    up         <0,  1,    0>
    right    <4/3,  0,    0>
}
```

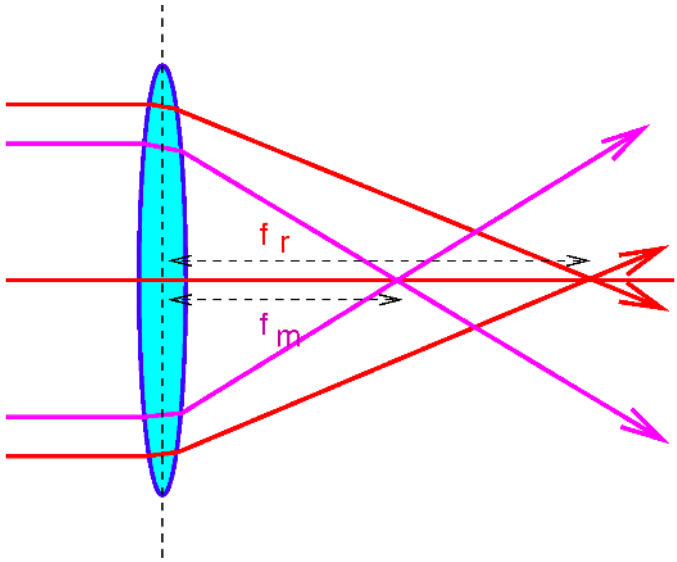
```
background { color SkyBlue }

// declare rainbow's colours

#declare r_violet1
  = colour red 1.0 green 0.5 blue 1.0 filter 1.0;
#declare r_violet2
  = colour red 1.0 green 0.5 blue 1.0 filter 0.8;
#declare r_indigo
  = colour red 0.5 green 0.5 blue 1.0 filter 0.8;
#declare r_blue
  = colour red 0.2 green 0.2 blue 1.0 filter 0.8;
#declare r_cyan
  = colour red 0.2 green 1.0 blue 1.0 filter 0.8;
#declare r_green
  = colour red 0.2 green 1.0 blue 0.2 filter 0.8;
#declare r_yellow
  = colour red 1.0 green 1.0 blue 0.2 filter 0.8;
#declare r_orange
  = colour red 1.0 green 0.5 blue 0.2 filter 0.8;
#declare r_red1
  = colour red 1.0 green 0.2 blue 0.2 filter 0.8;
#declare r_red2
  = colour red 1.0 green 0.2 blue 0.2 filter 1.0;

// create the rainbow

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]
  }}

sky_sphere {
  pigment {
    gradient y
    color_map {
      [0 colour SkyBlue]
      [1 colour MidnightBlue]
    }
    scale 2
    translate <-1, -1, -1> }}

/* Put down the beloved famous raytrace  checkered floor */
plane { y, -10
   pigment {
      checker colour Orchid colour Plum //colors changed!
      scale 20    }
   finish {
      ambient 0.2
      diffuse 0.8    }}

light_source {<100, 120, 40> colour White}
light_source {<0,20, -100> colour White}

#declare diamondInterior = interior {
         ior Diamond_Ior
         dispersion 2 //Diamond Dispersion = 1.035
          fade_distance 1
      fade_power 2
       caustics 1.0
          dispersion_samples 100}
#declare diamondFinish =  finish {
           specular 0.7
          roughness 0.001
          ambient 0
         diffuse 0
         reflection {
          0.2, 1.0
```

```
      fresnel on }
      conserve_energy}

#declare Radius = 10.0;
#declare P1 = <0, 1.5*Radius, 0>;
#declare P11 = vaxis_rotate (<0, 0, Radius> , x, 15.0) + P1;
#declare P12 = vaxis_rotate (P11, y, 60.0) ;
#declare P13 = vaxis_rotate (P11, y, 120.0) ;
#declare P14 = vaxis_rotate (P11, y, 180.0) ;
#declare P15 = vaxis_rotate (P11, y, 240.0) ;
#declare P16 = vaxis_rotate (P11, y, 300.0) ;

#declare P21 = vaxis_rotate (<0, 0, Radius> , x, 15.0)
                  * cos (radians (30.0));
#declare P21 = P21 + vaxis_rotate (P21, x, 15);
#declare P21 = vaxis_rotate (P21, y, 30.0) + P1;
#declare P22 = vaxis_rotate (P21 , y, 60.0);
#declare P23 = vaxis_rotate (P21 , y, 120.0);
#declare P24 = vaxis_rotate (P21 , y, 180.0);
#declare P25 = vaxis_rotate (P21 , y, 240.0);
#declare P26 = vaxis_rotate (P21 , y, 300.0);
#declare P6 = <0.0, -Radius, 0.0>;

#declare Diamond = mesh { //cap
 triangle { P1, P11, P12 }
 triangle { P1, P12, P13 }
 triangle { P1, P13, P14 }
 triangle { P1, P14, P15 }
 triangle { P1, P15, P16 }
 triangle { P1, P16, P11 }
                  // outer facettes
   triangle { P12, P21, P22 }
    triangle { P13, P22, P23 }
    triangle { P14, P23, P24 }
    triangle { P15, P24, P25 }
    triangle { P16, P25, P26 }
    triangle { P11, P26, P21 }
                  // outer facettes
    triangle { P11, P26, P21 }
    triangle { P12, P21, P22 }
    triangle { P13, P22, P23 }
```

```
    triangle { P14, P23, P24 }
    triangle { P15, P24, P25 }
    triangle { P16, P25, P26 }
                    // the stam
    triangle { P21, P6, P22 }
    triangle { P22, P6, P23 }
    triangle { P23, P6, P24 }
    triangle { P24, P6, P25 }
    triangle { P25, P6, P26 }
    triangle { P26, P6, P21 }
hollow
 pigment {White filter 0.9}
        finish { diamondFinish }
   interior{diamondInterior}}

object {Diamond
pigment {color <1.0 1.0 0.7> filter 0.9}
 rotate -x*(10+sin(clock*90))
       rotate clock*360*y
scale <3,3,3>
rotate <-30.0, 0.0, 10.0>
translate 10*z}

object {Diamond
 rotate -x*(10+sin(clock*90))
       rotate -clock*720*y
pigment {color <0.7 0.8 1.0> filter 0.9}
  scale <0.7, 0.7, 0.7>
rotate <-30.0, 0.0, 10.0>
  translate <32, 0.0, -50.0>}

object {Diamond
 rotate -x*(10+sin(clock*90))
       rotate -clock*720*y
pigment {color <0.8 0.9 1.0> filter 0.9}
  scale <1.0, 1.0, 1.0>
rotate <-30.0, 0.0, -10.0>
  translate <-45, 0.0, -48.0>}
```

Chromatische Aberration bei einer Sammellinse



Chromatische Aberration von Linsen

Definition: 35 (***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)

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

16.3 Strahlungsgesetze

Lichtquellen und Strahlungsgesetze

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

Definition: 36 (*Schwarze Strahlung*)
Idealisierter Grenzfall der Strahlung eines Körpers, wobei die Energieverteilung durch das Plancksche Strahlungsgesetz beschrieben wird.

Definition: 37 (*Schwarzer Körper*)
Ein Körper, der eine schwarze Strahlung aussendet.

✎ **Anmerkung:** Plancksches Strahlungsgesetz:
Die pro Wellenlänge ausgesendete Energie B_λ eines schwarzen Körpers der Temperatur T ist

$$B_\lambda = \frac{2hc^2}{\lambda^5} \cdot \frac{1}{e^{\frac{c_2}{\lambda T}} - 1} [\text{erg cm}^{-2}\text{sec}^{-1}\text{sterad}^{-1}\text{cm}^{-1}]$$

$$c_2 = \frac{hc}{k}, \text{ Strahlungskonstante } = 1.44\text{cm grad}$$

$$k = \text{Boltzmannkonstante } = 1.410^{-16}\text{erg / grad}$$

✎ **Anmerkung:** Stephan-Boltzmann-Gesetz: Die Energie E, die ein schwarzer Körper integriert über alle Richtungen und Frequenzen ausstrahlt, ist proportional zur 4. ten Potenz seiner Temperatur T.

$$E = \sigma T^4$$

$$\sigma = 5.6710^{-5}\text{erg cm}^{-2}\text{sec}^{-1}\text{grad}^{-4}$$

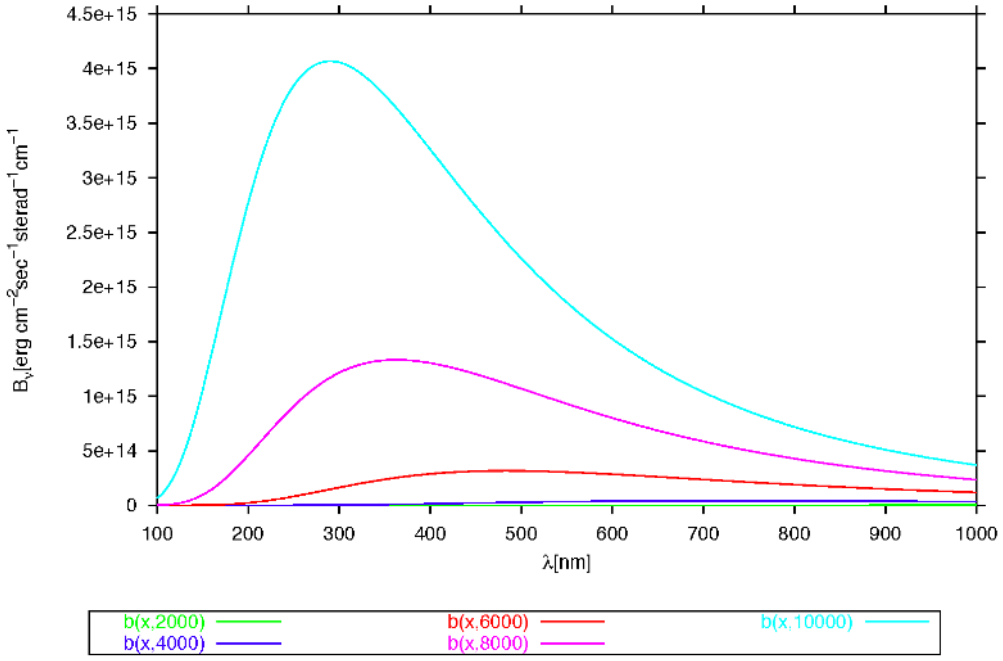
Wiensches Verschiebungsgesetz:

Satz: 38 (Wiensches Verschiebungsgesetz:)
Je heißer ein schwarzer Körper ist, desto mehr verschiebt sich das Maximum λ_{max} der Strahlung zu den kürzeren Wellenlängen:

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

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV2_Movies/Pages/movPlanckStrahlung.html

Energiedichte der Strahlung



Temperatur und Strahlungsleistung von schwarzen Körpern

Strahlungsleistung von schwarzen Körpern

Temperatur [K]	λ_{max} [Å]	Farbe	Strahlungs-leistung [$\frac{\text{Watt}}{\text{cm}^2}$]
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	Röntgen-strahlung	$5.8 \cdot 10^{12}$

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: 39 (*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 ähnlichste Farbtemperatur zugeordnet. (DIN 5033, Bl.1, 1953)

Definition: 40 (*Farbe*)
ist eine Empfindung, die durch sichtbare Strahlung über das Sinnesorgan „Auge“ ausgelöst wird.

Lichtquelle und Farbtemperatur

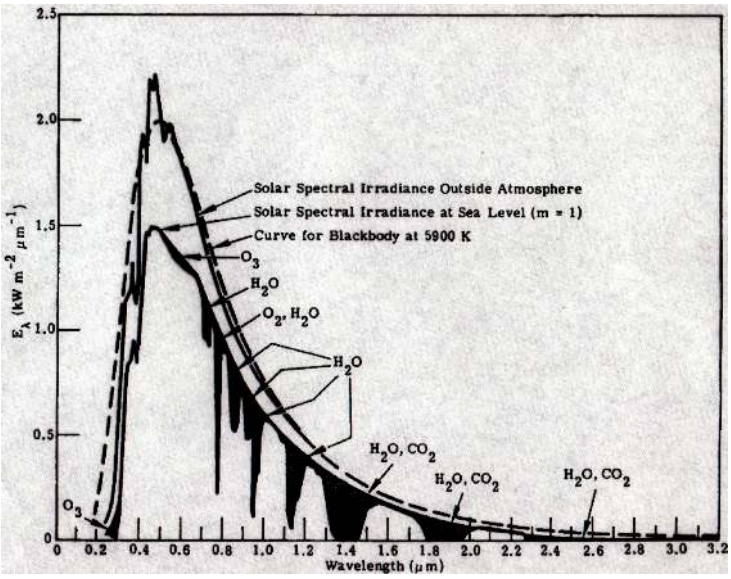
Lichtquelle	Temperatur
Kohlefadenlampe	2 100 °C
Goebel, Edison (1879)	
Wolframglühlampe	2 400 °C
Wolframlampe (Argon, Stickstoff)	2 750 °C
nach Langmuir	
Bogenlampe	4 000 – 5 000 °C
Sonne	≈ 5600 °C
Wega, Sirius	≈ 10 000 °C
Spica	≈ 37 000 °C

✎ **Anmerkung:** Die folgende Folie zeigt das Spektrum der Sonne, wie es außerhalb der Erdatmosphäre gemessen wird und was davon an der Erdoberfläche übrigbleibt. Zum Vergleich ist die spektrale Verteilung der Strahlung eines schwarzen Körpers der Temperatur 5800 K eingetragen.

Man sieht den Verlust an Sonnenenergie durch Absorption und Streuung in der Atmosphäre. Die Absorption erfolgt nicht gleichmäßig im Spektrum, sondern bestimmte Stoffe erzeugen charakteristische Absorptionsbänder, beispielsweise das Ozon, der Wasserdampf oder das Kohlendioxyd.

Unser Auge ist für diejenigen Teile des Spektrums empfindlich, bei denen das Maximum der Sonnenstrahlung liegt.

Die Sonne als schwarzer Strahler



Die Farbe der Sterne

Die Farben der Sterne sind ein Maß für deren Oberflächentemperatur:

17 Licht und Oberflächen

17.1 Die Farbe des Umgebungslichtes

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.

✎ **Anmerkung:** Weißabgleich: Wenn Sie gerne fotografieren, haben Sie sicher schon erlebt, daß Farbfotos nur natürlich aussehen, wenn der Film auf die Lichtverhältnisse abgestimmt ist: Wenn wir bei Lampenlicht fotografieren wollen, müssen wir einen speziellen Kunstlichtfilm oder Blitzlicht verwenden, das dem Tageslicht ähnlich ist, sonst erscheint das Bild gelbstichig. Aufnahmen im Hochgebirge oder im Schnee werden leicht blaustichig, wenn kein UV-Filter verwendet wird, und auf Bildern von einem Picknick im Wald haben die Teilnehmer meist recht grüne Gesichter, weil das von den Blättern reflektierte Licht einen hohen Grünanteil hat. Auch hier hilft das Blitzlicht.

Digitalkameras werden auf die Farbe des Umgebungslichtes eingestellt, indem ein *Weißabgleich* durchgeführt wird.

Für Unterwasserfotos müssen wir eine Lichtquelle mit Tageslicht mitnehmen, wenn wir die Unterwasserwelt in aller farbigen Pracht festhalten wollen, denn unter Wasser erscheint sonst alles blau.

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.

17.2 Pigmente und Licht

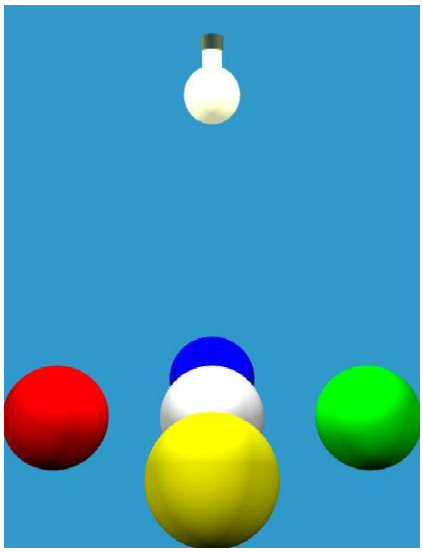
Auf den folgenden Folien sehen Sie mit dem POV-Ray System erzeugte Simulationen von unterschiedlichen Beleuchtungssituationen. Die Szene ist einfach: Sie enthält eine Kamera, eine Lichtquelle und fünf Kugeln mit unterschiedlichen Pigmenten: in der Mitte eine Kugel mit weißen Pigmenten, vorne eine mit gelben, links eine mit roten, hinten eine mit blauen und rechts eine Kugel mit grünen Pigmenten.

Die erste Folie zeigt diese Szene mit weißem Licht beleuchtet: Die Kugel mit weißen Pigmenten reflektiert alles einfallende Licht und erscheint weiß, die Kugel mit roten Pigmenten reflektiert alles rote Licht und absorbiert das Licht aller anderer Frequenzen, sie erscheint rot usw.

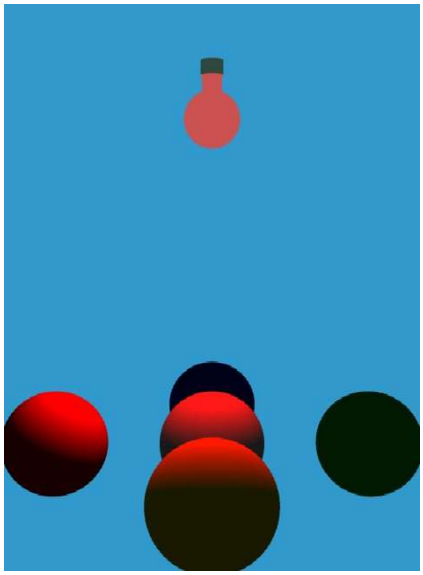
Die zweite Folie zeigt dieselbe Szene bei rotem Licht: Die Kugel mit weißen Pigmenten reflektiert wiederum alles einfallende Licht, aber da dieses nur rote Strahlung enthält, erscheint diese Kugel jetzt rot und nicht weiß. Die Kugel mit roten Pigmenten reflektiert alles rote Licht und absorbiert das Licht aller anderer Frequenzen, sie erscheint ebenfalls rot und ist von der Kugel mit den weißen Pigmenten nicht zu unterscheiden, ebenso die Kugel mit den gelben Pigmenten. Gelbes Licht kann als Mischung von rotem Licht und grünem Licht erzeugt werden. Die gelben Pigmente reflektieren daher auch rotes Licht. Die blauen und grünen Pigmente erscheinen bei Rotlicht schwarz.

Es folgt dieselbe Szene bei gelbem Licht und abschließend bei drei Lichtquellen, eine rot, eine grün und eine blau.

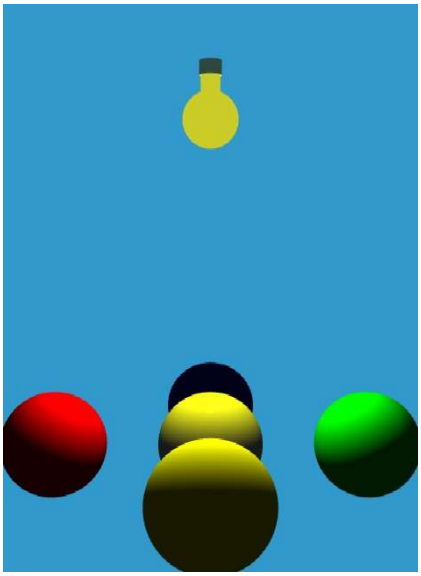
Beispiel: Vier Kugeln in weißem Licht



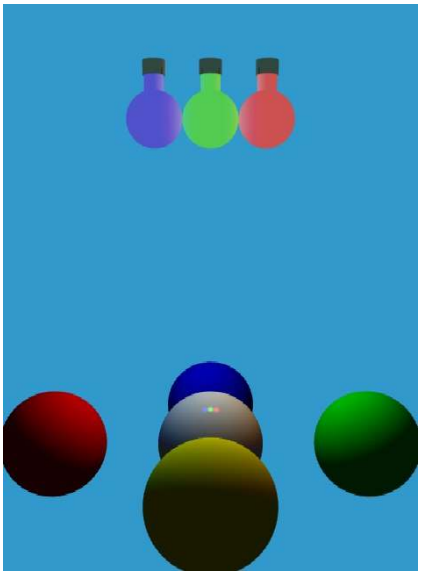
Beispiel: Vier Kugeln in rotem Licht



Beispiel: Vier Kugeln in gelbem Licht



Beispiel: Vier Kugeln in rot-grün-blau Licht



✎ **Anmerkung:** Uneindeutigkeit von Farben: Am letzten Beispiel ist zu sehen, daß wir den Eindruck von weißem Licht mit nur drei Grundfarben erzeugen können. Generell gilt, daß keinem Farbeindruck ein eindeutiges Spektrum zuzuordnen ist, sondern daß unterschiedliche Spektren zu gleichen Farbeindrücken führen können. Dieses ist eine Folge davon, wie unser visuelles System (das Auge und die Teile des Gehirns, die die optischen Reize verarbeiten) das im Auge eintreffende Licht verarbeitet.

Beispiel: 41
Farbige Kugeln in weißem Licht

```
global_settings { assumed_gamma 2.2 }

#include "colors.inc"
#include "textures.inc"
#include "finish.inc"
#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}
#declare sphradius = 2;
camera {
   location  <0, 4, -25>
   look_at <0, 0, 0>
}

background { color SkyBlue }

light_source { y*4*sphradius  color White
looks_like {Lightbulb}}
#light_source { <100, 100, -100> color White shadowless}
#light_source { <-200, 100, -100> color White shadowless}

sphere { <0,-2*sphradius,0>,  sphradius
   pigment { colour White }
```

```
    finish {Shiny}}
sphere { <-3*sphradius,-2*sphradius,0>,  sphradius
    pigment { colour Red }
}
sphere { <3*sphradius,-2*sphradius,0>,  sphradius
    pigment { colour Green }
}
sphere { <0,-2*sphradius,-3*sphradius>,  sphradius
    pigment { colour Yellow }
}
sphere { <0,-2*sphradius,3*sphradius>,  sphradius
    pigment { colour Blue }
}
```

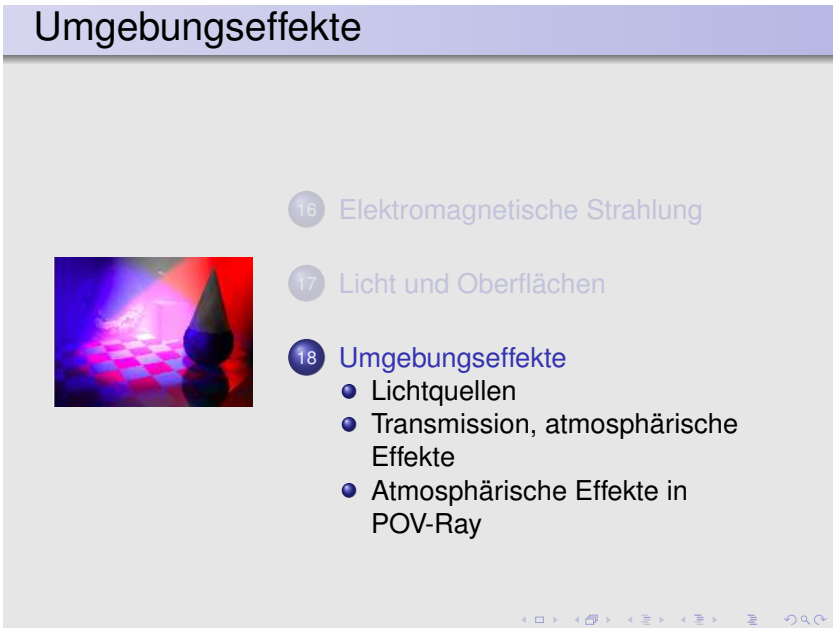
Dem Weißmacher
Newtonisch Weiß den Kindern vorzuzeigen,
Die pädagogischem Ernst sogleich sich neigen,
Trat einst ein Lehrer auf, mit Schwungrads Possen,
Auf selbem war ein Farbenkreis geschlossen.

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

18 Umgebungseffekte

18.1 Lichtquellen



Punktlichtquellen

Punktlichtquellen strahlen von einem Punkt aus isotrop in alle Richtungen.

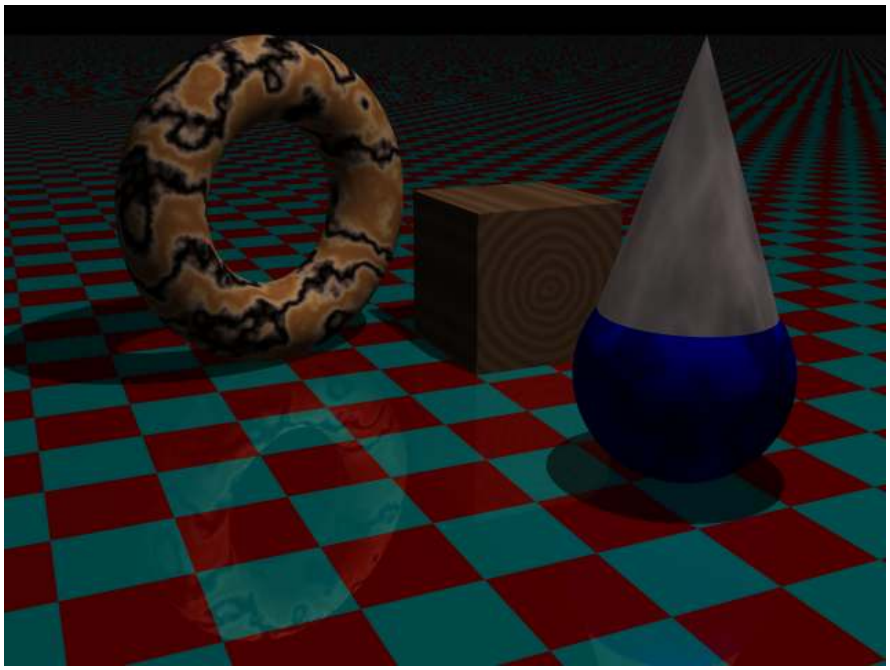
- Die Lichtstärke ist in allen Raumwinkeln gleich.
- Die Beleuchtungsstärke nimmt mit dem Quadrat der Entfernung von der Lichtquelle ab.
- Eine Punktlichtquelle erzeugt scharf begrenzte Schlagschatten.

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



```
#include "colors.inc"
#include "textures.inc"

camera {
        location <-4, 3, -9>
        look_at <0, 0, 0>
        angle 48}

plane {y, -1
       texture {
       pigment {
         checker
         color rgb<0.5, 0, 0>
```

```
        color rgb<0, 0.5, 0.5>
        }
        finish {
        diffuse 0.4
        ambient 0.2
        phong 1
        phong_size 100
        reflection 0.25 }}}

torus { 1.5, 0.5
      texture {Brown_Agate }
      rotate <90, 160, 0>
      translate <-1, 1, 3> }

box {<-1, -1, -1>, <1, 1, 1>
    texture {DMFLightOak }
    translate <2, 0, 2.3>}
cone { <0,1,0>, 0, <0,0,0>, 1
     texture {PinkAlabaster }
     scale <1, 3, 1>
     translate <1.5, 0, -2> }
sphere { <0,0,0>, 1
       texture {Sapphire_Agate }
       translate <1.5, 0, -2>}

light_source {
             <2, 10, -3>
             color White
             fade_distance 10
             fade_power 2
}

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

Das folgende Bild zeigt eine Platte mit einem versteinerten Panzerlurch (Naturhistorisches Museum Salzburg). Die Ränder der Platte sind weiter vom Licht entfernt als die Mitte der Platte. Deshalb ist dort die Beleuchtungsstärke E und als Folge davon auch die Leuchtdichte D kleiner als in der Mitte der Platte. Wir sehen sie am Rand also dunkler.

Beispiel: Punktlichtquelle

Punktlichtquellen mit abnehmende Beleuchtungsstärke

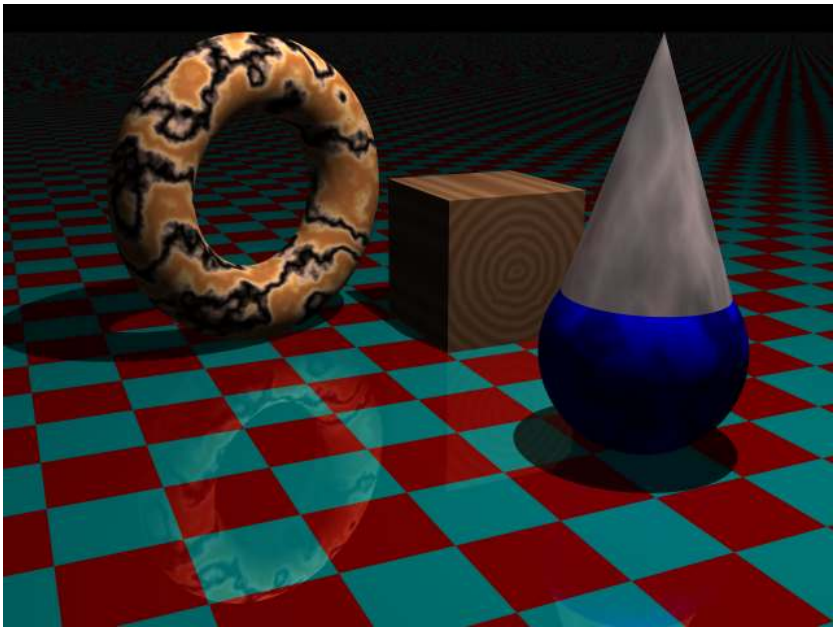
```
light_source {
        <2, 10, -3> /* Position */
        color White /* Farbe */
        fade_distance 10
        fade_power 2}
```

`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

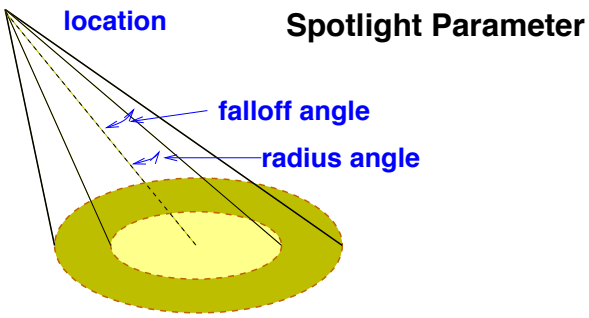
`fade_distance 10, fade_power 2`



✎ **Anmerkung:** In virtuellen Welten können wir also unsere eigenen physikalischen Gesetze schaffen: Die Beleuchtungsstärke kann mit jeder beliebigen Potenz der Entfernung von der Lichtquelle variieren. Auch das gute, alte Sprichwort: „Wo Licht ist, ist auch Schatten.“ gilt nicht unbedingt. Wenn uns der Schattenwurf stört, können wir schattenloses Licht schaffen. Dazu wird bei der Lichtquelle zusätzlich das Schlüsselwort `shadowless` angegeben. Solche Lichtquellen sind nützlich, um die Schattenbereiche auszuleuchten.

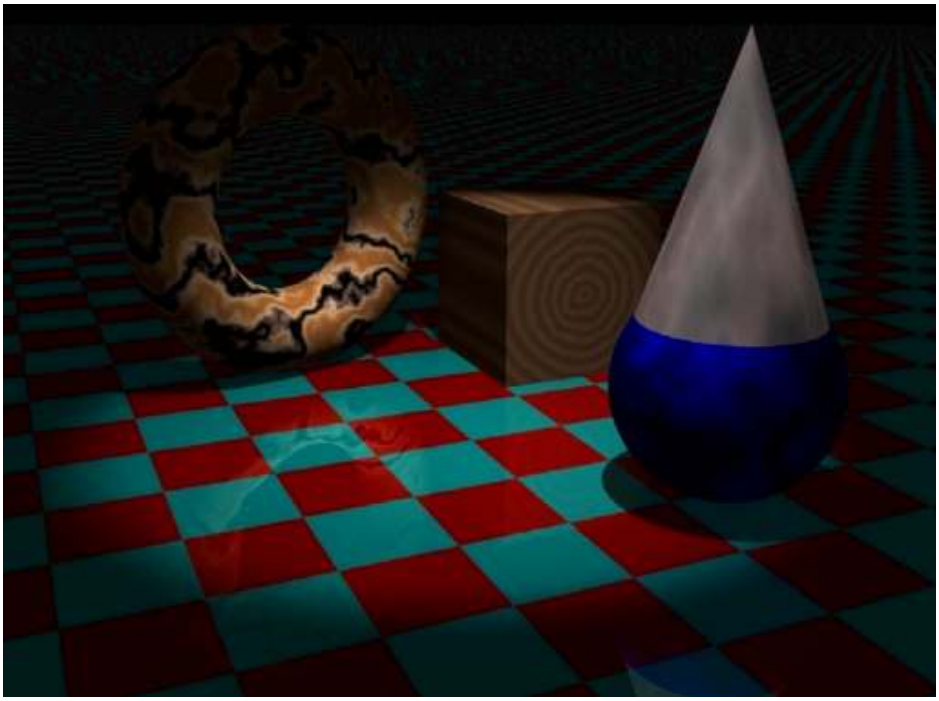
Scheinwerfer: spotlight

Scheinwerfen 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

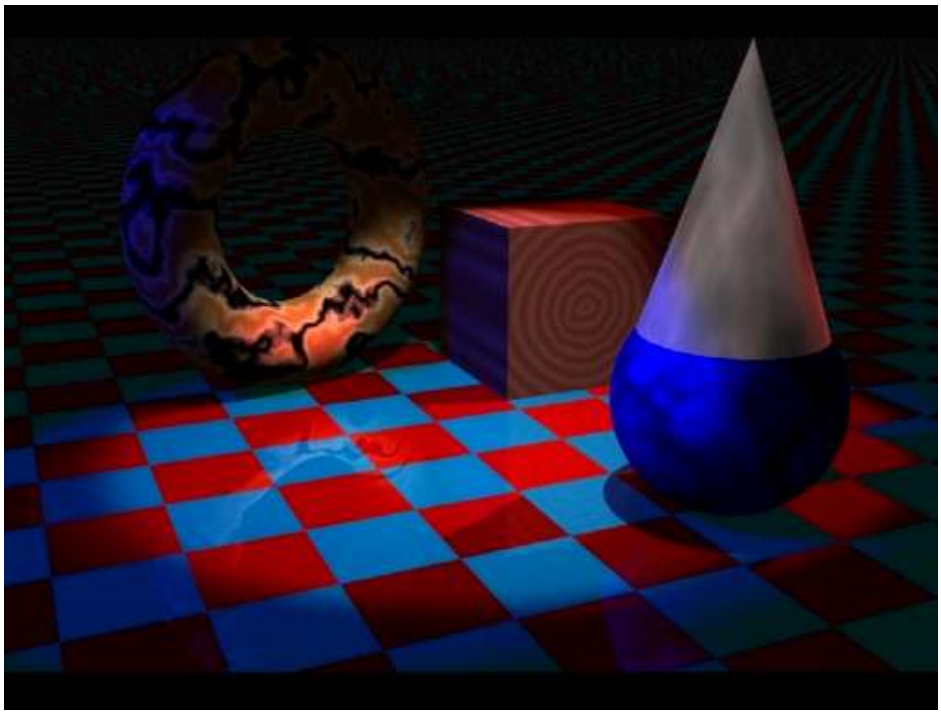


Zwei Scheinwerfer

```
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
```

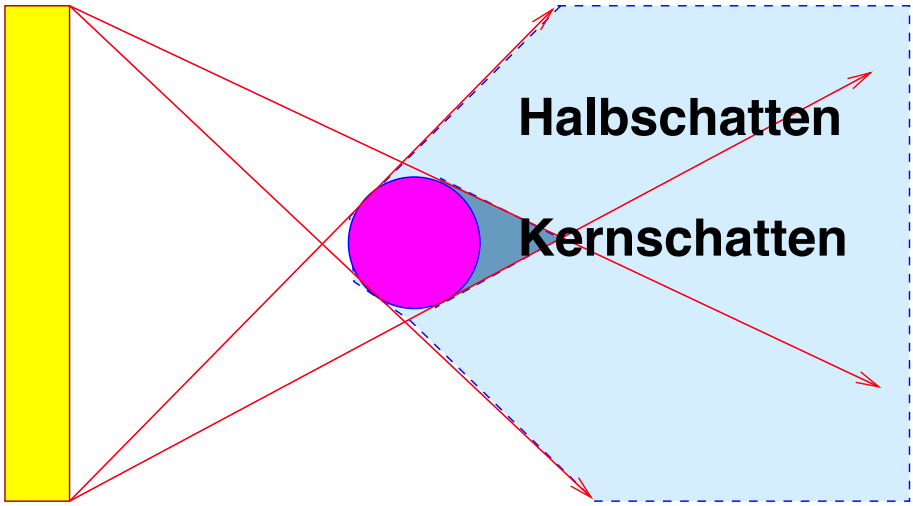
```
spotlight radius 12 falloff 14
tightness 10
point_at <-2,0,0>}
```

Drei Scheinwerfer: Weiß, rot, blau



Flächenlicht: Ausgedehnte Lichtquellen

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

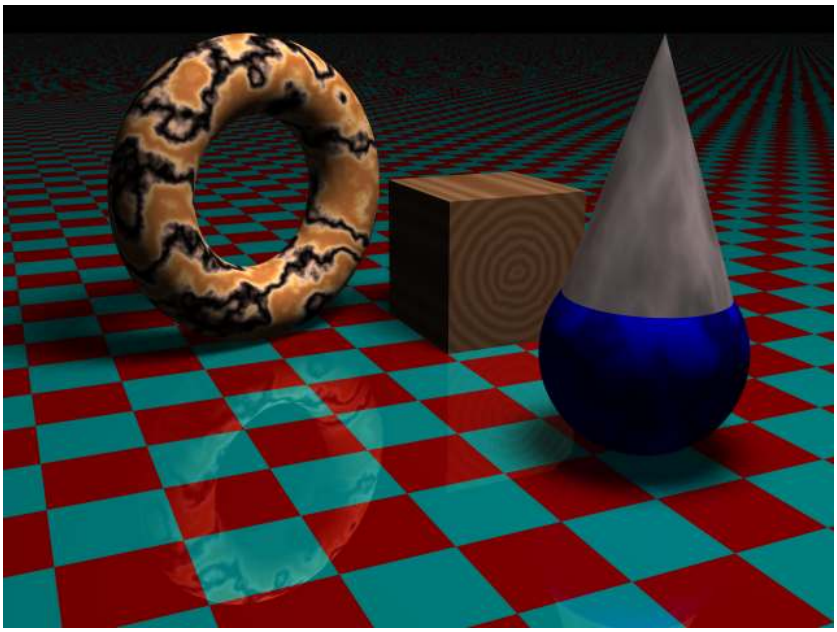


area_light

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)



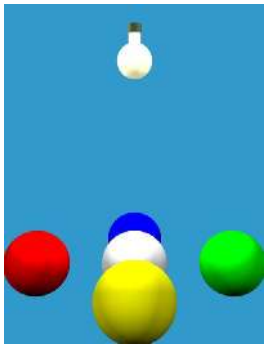
✎ **Anmerkung:** Der Rechenaufwand für ein `area_light` ist proportional zur Anzahl der Teillichtquellen. Die Berechnung des folgenden Bildes mit einem 5*5-Flächenlicht hat ca. 20 mal solange gedauert gedauert wie die Berechnung des Bildes mit einer Punktlichtquelle.

Die Parameter `jitter` und `adaptive` betreffen das rendering-Verfahren. Wir werden sie am Semesterende im Zusammenhang mit dem ray-tracing behandeln.

Die Lichtquellen werden von POV-Ray normalerweise nicht dargestellt — sie sind unsichtbar. Wenn wir die Lichtquellen zeigen wollen, können wir das mit dem `looks_like`-Schlüsselwort und einem Objekt, das das Aussehen der Lichtquelle beschreibt. Ein solches Objekt blockiert nicht das Licht der Lichtquelle. Auf diese Weise wurde die Glühbirne auf Seite 216 dargestellt. Die Glühbirne ist eine Zusammensetzung (merge) aus einer Glaskugel und einem Glaszylinder, mit einem metallischen Sockel.

Beispiel: Eine sichtbare Lichtquelle: looks_like

Beispiel: 42
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}}
```

18.2 Transmission, atmosphärische Effekte

18.2.1 Atmosphärische Trübung

In der Atmosphäre wird das Licht gestreut, gedämpft und verfärbt. Kurzwelliges Licht wird an den Molekülen stärker gestreut (auch Rayleigh-Streuung genannt) als langwelliges.

Als Folge davon erscheint der Himmel blau und der Sonnenuntergang rot, siehe beispielsweise das Bild vom blauen Himmel über Lhasa (Seite 229) Sonnenuntergang auf Tahiti (Seite 230). Außerdem ist die Rayleigh-Streuung stark gerichtet, so daß das Himmelslicht je nach Abstand von der Sonne einen anderen Farbton hat.

Streuung an Staub und Aerosolteilchen (Mie-Streuung) vermindert den Kontrast. Weiter entfernte Berge werden durch blau-weiße Dunstschleier verdeckt, wie auf den folgenden Bildern (der Marlborough-Sound, Neuseeland, die Atacama-Wüste, das Huangshan-Gebirge) zu sehen. Abhängig von der Größe der streuenden Teilchen (Staub, Wassertropfen, Eiskristalle, Moleküle) ist die Stärke der Streuung unterschiedlich stark gerichtet und unterschiedlich stark von der Wellenlänge abhängig. Daher entstehen sehr unterschiedliche Trübungseffekte.

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

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

18.3 Atmosphärische Effekte in POV-Ray

Im POV-ray gibt es zwei einfache Wege, um atmosphärische Effekte zu modellieren: Nebel (fog) und Bodennebel (ground fog). Für anspruchsvolle Modellierungen sollte auf media-Spezifikationen zurückgegriffen werden. Hiermit kann sehr genau das Streuverhalten der Partikel spezifiziert werden. Atmosphärische Effekte sind besonders eindrucksvoll, wenn sie mit Scheinwerferlicht kombiniert werden, da so der Lichtweg sichtbar gemacht werden kann.

18.3.1 Nebel und Bodennebel

Nebel: fog

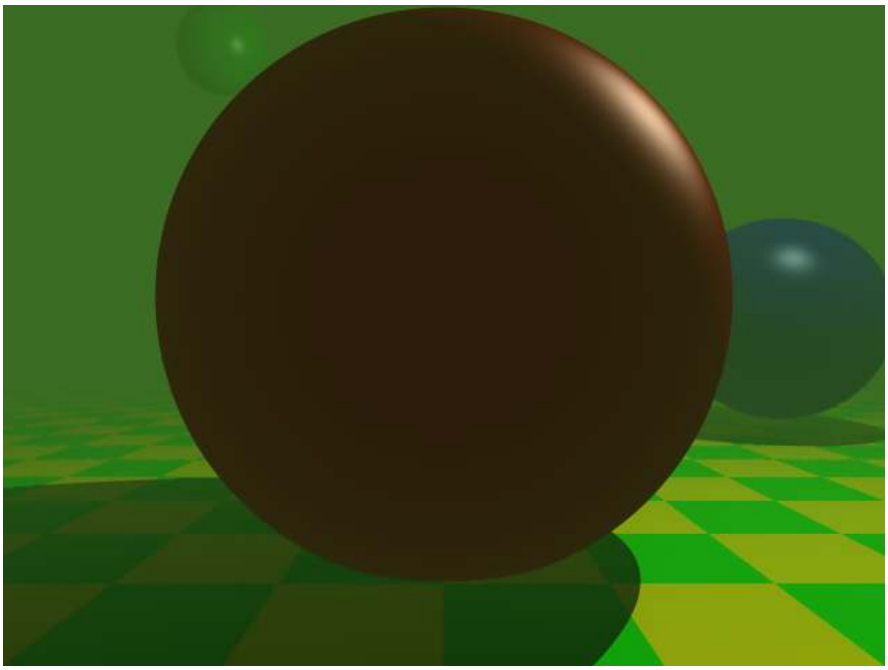
Beispiel: 43
Nebel Ein konstanter Nebel wird über ein ***fog****-statement in die Szene eingefügt. Parameter sind:*

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



POV-Ray Standarddemo fog1.pov

```
// Persistence Of Vision raytracer version 3.5 sample file.
// This scene shows fog without filter and transmittance.
//
// -w320 -h240
// -w800 -h600 +a0.3

global_settings { assumed_gamma 2.2 }

#include "colors.inc"

camera {
   location  <0, 20, -100>
   direction <0,  0,    1>
   up        <0,  1,    0>
   right    <4/3,  0,    0>
}

background { colour SkyBlue }

// Standard fog
fog {colour rgbft<0.3, 0.5, 0.2, 0.0, 0.0> distance 150}

// Put down the beloved famous raytrace
// green/yellow checkered floor
plane { y, -10
   pigment {
      checker colour Yellow colour Green
      scale 20}
   finish {
      ambient 0.2
      diffuse 0.8}}

sphere { <0, 25, 0>, 40
   pigment {Red}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
```

```
        phong_size 20}}

sphere { <-100, 150, 200>,  20
   pigment {Green}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20}}

sphere { <100, 25, 100>, 30
   pigment {Blue}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20}}

light_source {<100, 120, 40> colour White}
```

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.

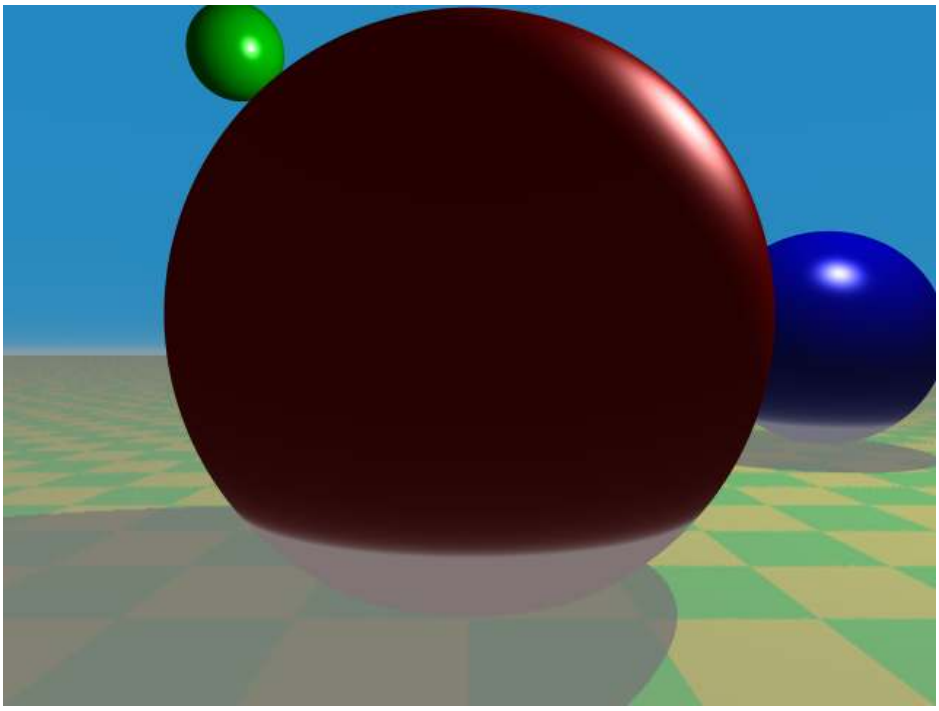
Bodennebel in POV-Ray

Beispiel: 44

```
  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



In diesem Beispiel hat die Farbe des Nebels zusätzlich eine transparente Komponente, so daß der Hintergrund bis zum Horizont sichtbar bleibt.

```
// Persistence Of Vision raytracer version 3.5 sample file.
// File by Dieter Bayer.
// This scene shows fog with transmittance used.
//
// -w320 -h240
// -w800 -h600 +a0.3

global_settings { assumed_gamma 2.2 }

#include "colors.inc"

camera {
    location  <0, 20, -100>
```

```
   direction <0,  0,    1>
   up        <0,  1,    0>
   right    <4/3,  0,    0> }

background { colour SkyBlue }

fog{
    color rgbt <.7,.7,.7,.25>
    fog_type 2
    fog_alt 0.5
    fog_offset 0
    distance 1.5
    turbulence <.15, .15, .15>
    omega 0.35
    lambda 1.25
    octaves 5
}

// Put down the beloved famous raytrace
// green/yellow checkered floor
plane { y, -10
   pigment {
      checker colour Yellow colour Green
      scale 20
   }
   finish {
      ambient 0.2
      diffuse 0.8
   }
}

sphere { <0, 25, 0>, 40
   pigment {Red}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20
   }
}
```

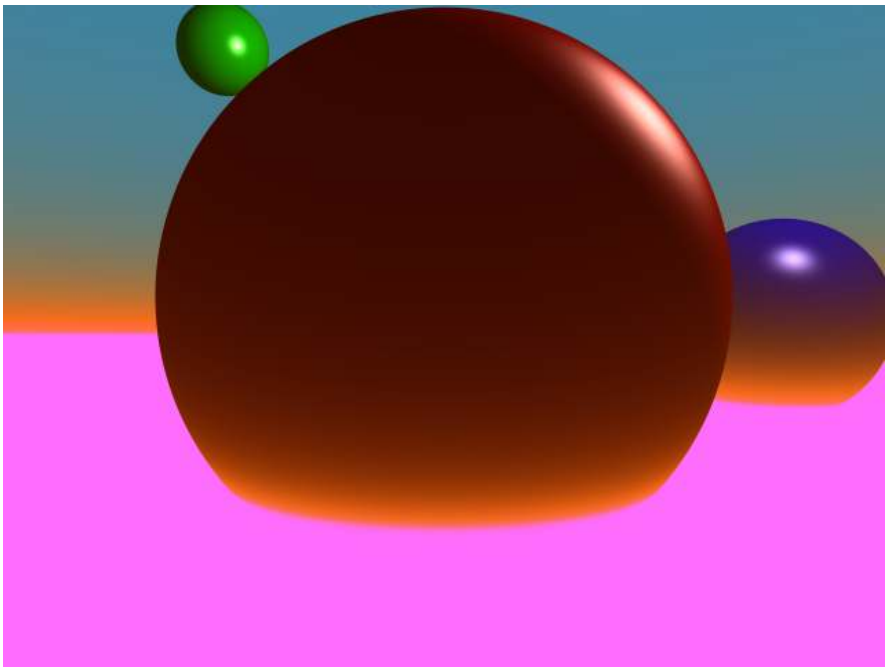
```
sphere { <-100, 150, 200>,  20
   pigment {Green}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20
   }
}

sphere { <100, 25, 100>, 30
   pigment {Blue}
   finish {
      ambient 0.2
      diffuse 0.6
      phong 1.0
      phong_size 20 }}

light_source {<100, 120, 40> colour White}
```

Geschichteter Nebel: layered fog

Beispiel: 45
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 <.2, .2, .2>}
```

Geschichteter Nebel (Beispiel: 45)



Scheinwerfer und Nebel

Die Modellierung der atmosphärische Effekte hat sich von POV-Ray-Version zu Version sehr stark geändert. In früheren Versionen gab es das atmosphere-statement, das inzwischen abgeschafft wurde.

18.3.2 Fortgeschrittene Transmissions-Effekte

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.

Die folgenden Beispiele stammen aus dem POV-Ray Media-Tutorial der Dokumentation zu POV-Ray 3.5.

Ein Beispiel: Ein emittierendes Medium

Beispiel: 46
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.

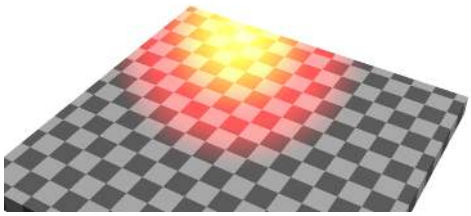
```
sphere // transparent sphere containing media
{ 0,1 pigment { rgbt 1 } hollow
  interior
  { media
    { emission 1
      density
      { spherical density_map
        { [0 rgb 0]
          [0.4 rgb <1,0,0>]
          [0.8 rgb <1,1,0>]
          [1 rgb 1]}}}}}
```

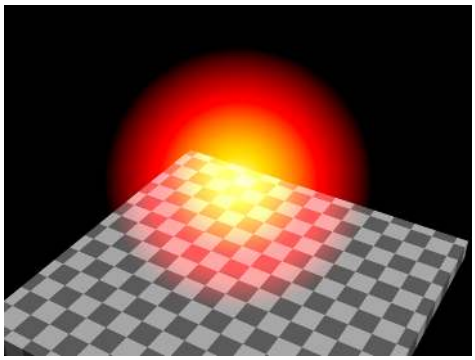
Der Eintrag mit dem Index 0 der **density_map** bestimmt der Dichte auf der Oberfläche einer Einheitskugel, der Index 1 die Dichte im Mittelpunkt der Kugel.

Die Farben in der **density_map** werden mit der Farbe im **emission**-statement multipliziert.

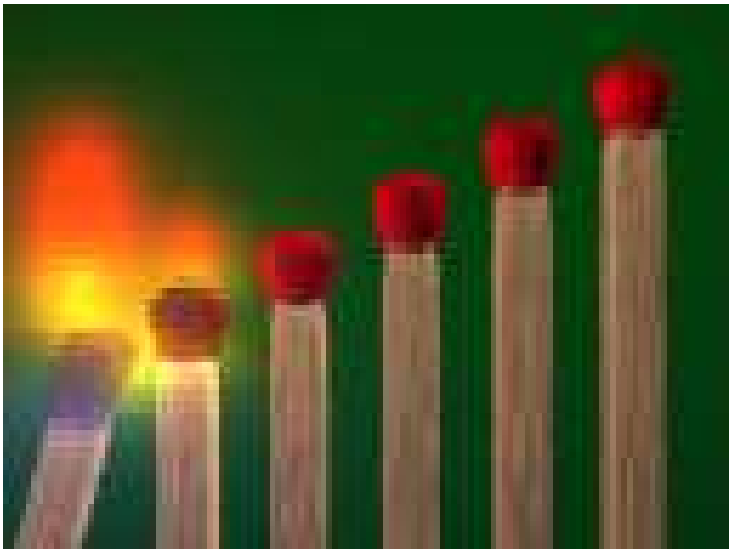
Die emittierende Kugel sieht wie eine Lichtquelle aus, wirkt aber in POV-Ray nicht als Lichtquelle. Sie beleuchtet keine Objekte und erzeugt auch keine Schatten.

Die leuchtende Kugel vor weißem und schwarzem Hintergrund





Ein komplexeres Beispiel für emittierende Pigmente



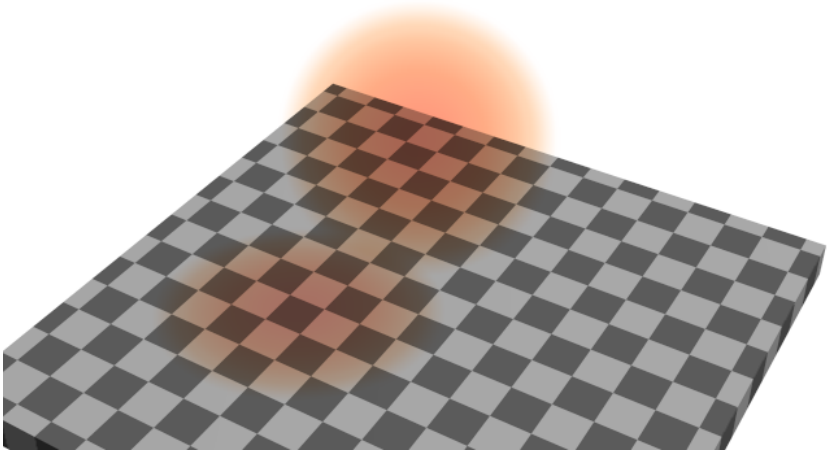
// By Dan Farmer 1996

Ein absorbierendes Medium

Beispiel: 47
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



Beachten Sie, daß die Kugel die Komplementärfarbe der in der Dichtetabelle angegeben Farben hat. Die Tabelle beschreibt, welche Farben aus dem weißen Hintergrundlicht absorbiert werden. Wir sehen die Kugel in den Farben, die übrig bleiben.

Die absorbierende Kugel wirft im Gegensatz zur emittierenden Kugel einen Schatten.

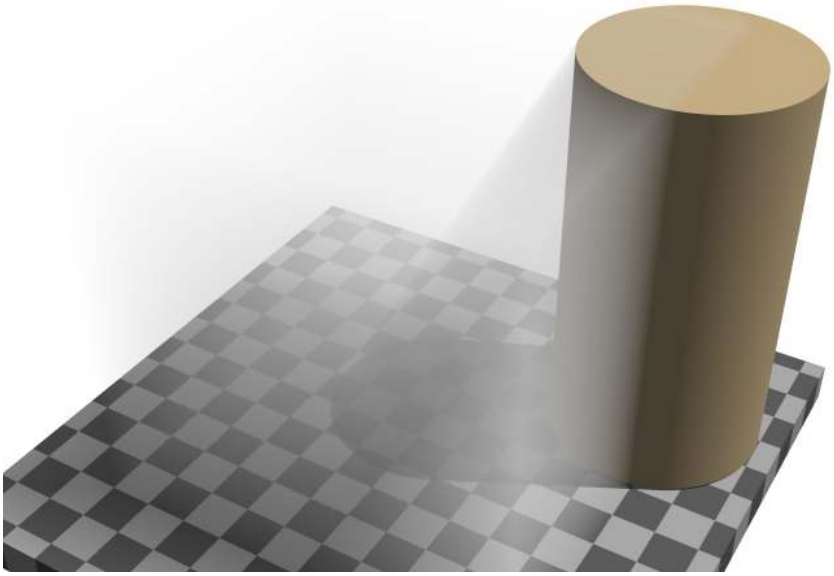
Ein isotrop streuendes Medium

Beispiel: 48
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



Beachten Sie, wie das streuende Medium den Kontrast zwischen dem Schatten und der Umgebung verringert. Der Schatten erscheint heller, als er ohne den atmosphärischen Effekt wäre.

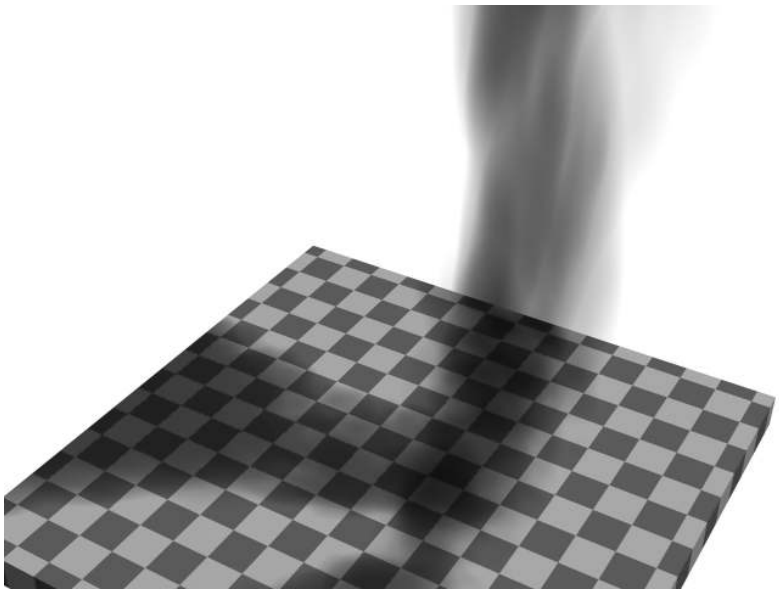
Typen der Streuung in POV-Ray

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

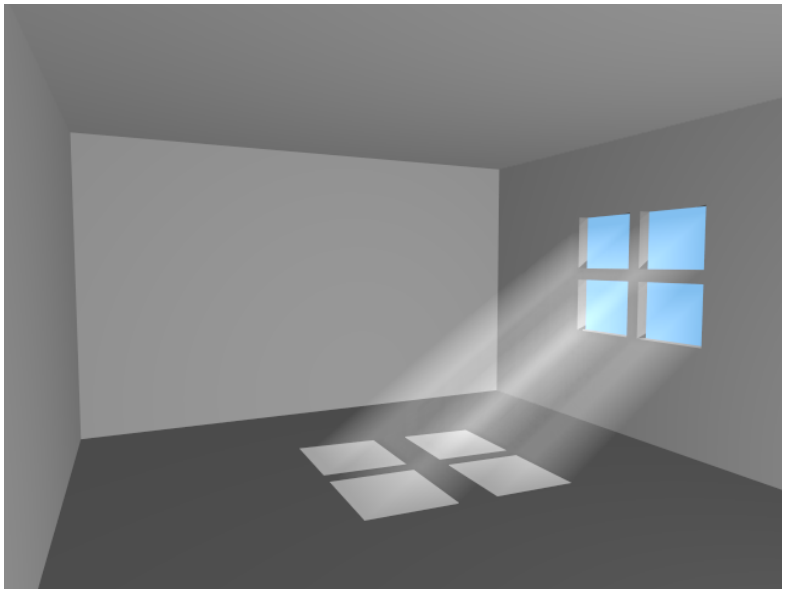
Eine Rauchsäule: Skalierung und Turbulenz

```
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

```
global_settings { assumed_gamma 1 }
 camera { location <14.9, 1, -8> look_at -z angle 70 }
 light_source { <10,100,150>, 1 }
 background { rgb <0.3, 0.6, 0.9> }

 // A dim light source inside the room which does not
 // interact with media so that we can see the room:
 light_source { <14, -5, 2>, 0.5 media_interaction off }

 // Room
 union
 { difference
   { box { <-11, -7, -11>, <16, 7, 10.5> }
     box { <-10, -6, -10>, <15, 6, 10> }
     box { <-4, -2, 9.9>, <2, 3, 10.6> }}
   box { <-1.25, -2, 10>, <-0.75, 3, 10.5> }
   box { <-4, 0.25, 10>, <2, 0.75, 10.5> }
   pigment { rgb 1 }}
 // Scattering media box:
 box
 { <-5, -6.5, -10.5>, <3, 6.5, 10.25>
   pigment { rgbt 1 } hollow
```

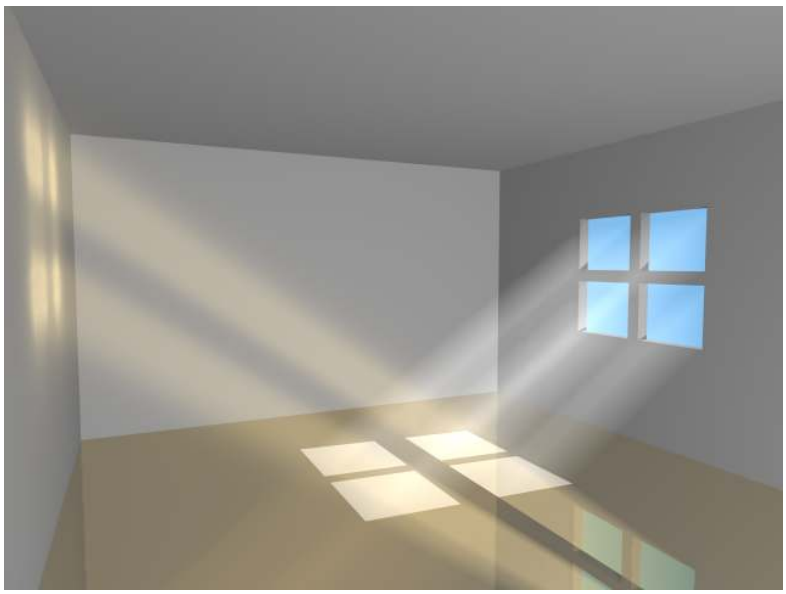
```
interior
{ media
  { scattering { 1, 0.07 extinction 0.01 }
    samples 30,100} }}
```

Normalerweise interagieren gebrochene und reflektierte Lichtstrahlen nicht mit den Medien. Soll auch das modelliert werden, muß in POV-Ray das Photonen-Rendering verwendet werden (siehe globale Beleuchtungsmodelle). Photonen-rendering ist extrem zeitaufwendig, kann aber bestechend photorealistische Effekte erzeugen.

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movHalley.html

Lichtstrahlen durch ein Fenster

mit Photonen gerechnet



Rechenzeit Intel: ca. 20 Minuten

The photon mapping technique can be used in POV-Ray for making stunningly beautiful images with light reflecting and refracting from objects. By default, however, reflected and refracted light does not affect media. Making photons interact with media can be turned on with the **media** keyword in the photons block inside **global_settings**.

To visualize this, let's make the floor of our room reflective so that it will reflect the beam of light coming from the window.

Firstly, due to how photons work, we need to specify:

photons { pass_through }

in our scattering media container box so that photons will pass through its surfaces.

Secondly, we will want to turn photons off for our fill-light since it's there only for us to see the interior of the room and not for the actual lighting effect. This can be done by specifying

photons { **reflection off** }

in that light source.

Thirdly, we need to set up the photons and add a reflective floor to the room. Let's make the reflection colored for extra effect:

```
global_settings { assumed_gamma 1 }

global_settings
 { photons
   { count 20000
     media 100}}

 camera { location <14.9, 1, -8> look_at -z angle 70 }
 light_source { <10,100,150>, 1 }
 background { rgb <0.3, 0.6, 0.9> }

 // A dim light source inside the room which does not
 // interact with media so that we can see the room:
 light_source { <14, -5, 2>, 0.5
 media_interaction off
 photons { reflection off }
 }

 // Room
 union
 { difference
   { box { <-11, -7, -11>, <16, 7, 10.5> }
     box { <-10, -6, -10>, <15, 6, 10> }
     box { <-4, -2, 9.9>, <2, 3, 10.6> }
   }
   box { <-1.25, -2, 10>, <-0.75, 3, 10.5> }
   box { <-4, 0.25, 10>, <2, 0.75, 10.5> }
   pigment { rgb 1 }
 }
```

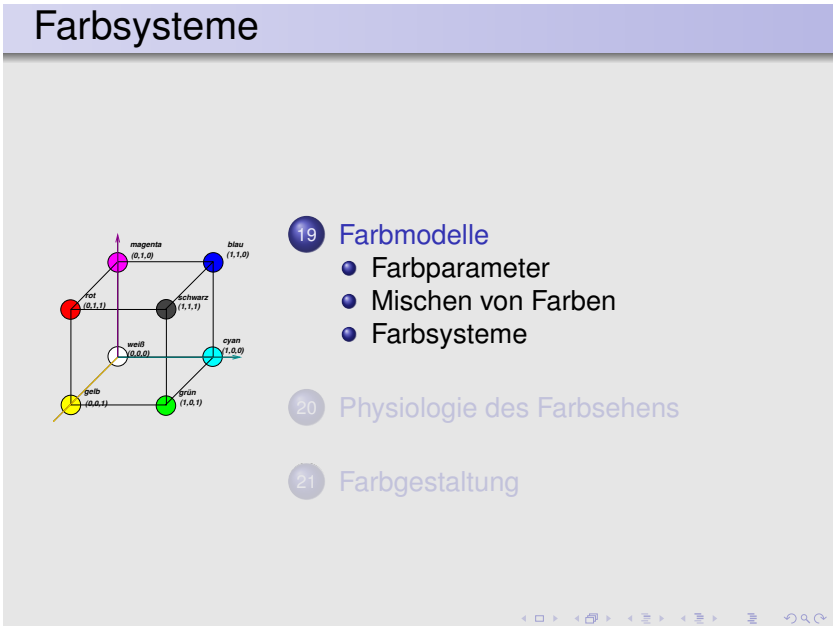
```
// Scattering media box:
box
{ <-5, -6.5, -10.5>, <3, 6.5, 10.25>
  pigment { rgbt 1 } hollow
  photons { pass_through }
  interior
  { media
    { scattering { 1, 0.07 extinction 0.01 }
      samples 30,100 } } }

// Reflective floor:
box
{ <-10, -5.99, -10>, <15, -6, 10>
  pigment { rgb 1 }
  finish { reflection <0.5, 0.4, 0.2> }
  photons { target reflection on }
}
```

Teil VII

Farbmodelle

19 Farbmodelle

19.1 Farbparameter



Farbparameter

Die Farbe einer Lichtquelle kann durch drei Parameter beschrieben werden:

- Farbton
- Sättigung (Reinheit)
- Helligkeit

Definition: 49 (*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.

siehe (Fellner, 2004).

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: 50 (*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.*

19.2 Mischen von Farben

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 verschiede 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.*

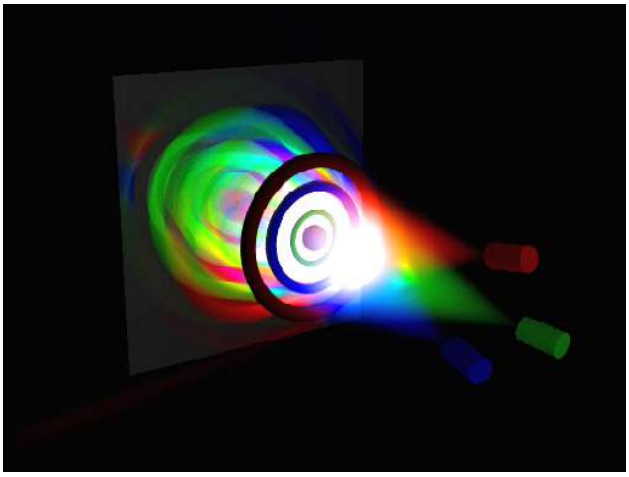
Trichromatische Theorie

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.

(Young, 1802) (von Helmholtz, 1852) (Maxwell, 1860)

Additives Farbmischen



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movaddsubQT.html

✎ **Anmerkung:** Farbdiagramm

Wenn wir drei Grundfarben auswählen und als Vektoren betrachten, dann lassen sich alle Farben, die aus diesen Grundfarben durch Mischung erzeugt werden können, durch eine Linearkombination dieser Vektoren beschreiben. Die Gewichtungsfaktoren geben dann an, wie stark eine einzelne Farbe der Mischfarbe beigemischt wird. Wie wir schon gesehen haben, ergibt die Mischung von rotem, grünen und blauen Licht zu gleichen Teilen ein weißes Licht, rot + grün ergibt gelb usw.

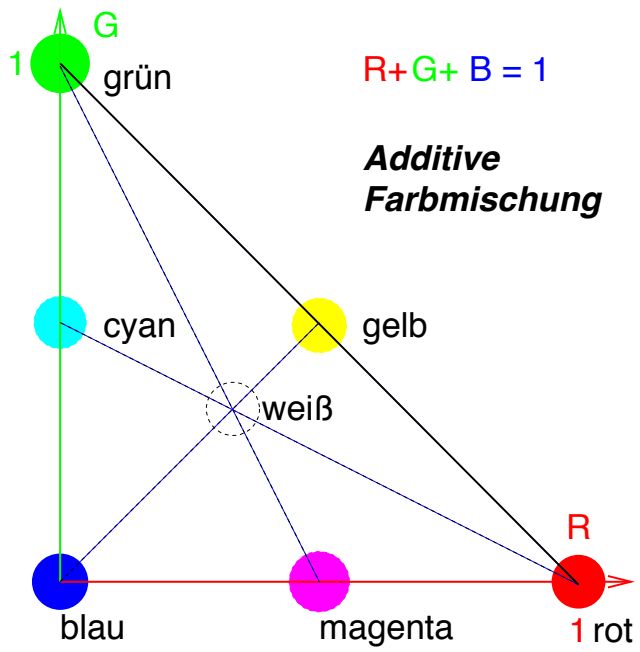
Wird die Länge eines Farbvektors auf „1“ normiert, beispielsweise

$$\text{rot} + \text{grün} + \text{blau} = 1,$$

dann reichen zwei Komponenten zur Spezifikation der Farbe aus, und wir können den Farbraum als zweidimensionales Diagramm darstellen. Die Grundfarben bilden die Eckpunkte eines Dreiecks; auf den Seitenlinien liegen die reinen Spektralfarben, im Schnittpunkt der Winkelhalbierenden liegt der Weißpunkt. Je ungesättigter eine Farbe ist, desto dichter liegt sie am Weißpunkt.

Ein solches Farbdiagramm gibt nur noch den Farbton und die Sättigung wieder, nicht mehr die Intensität der Farbe. Für die Farbe „schwarz“ müßte ja die Summe „rot+grün+blau=0“ ergeben, was wegen der Normung auf „1“ ausgeschlossen ist.

Additive Farbmischung

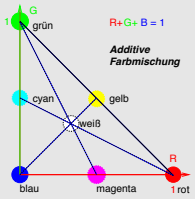


Komplementärfarben

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

19.3 Farbsysteme

19.3.1 Das CIE-System

Das CIE-System (Commission International de l'eclairage)

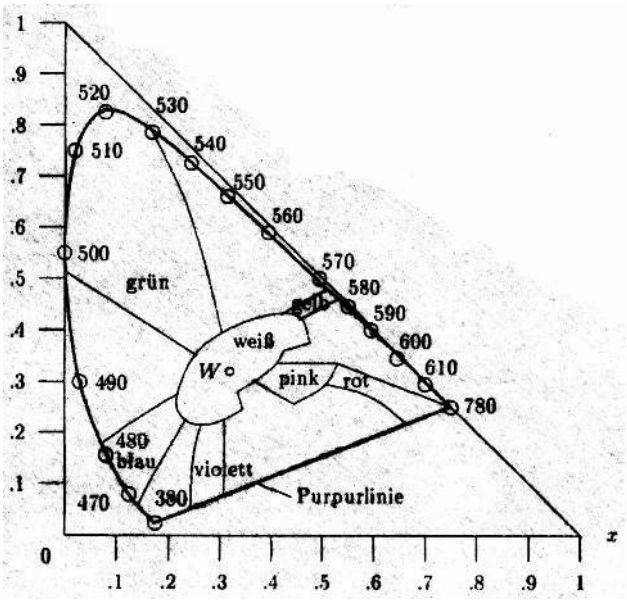
Das CIE-System ist seit 1931 ein internationaler Standard.

- 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

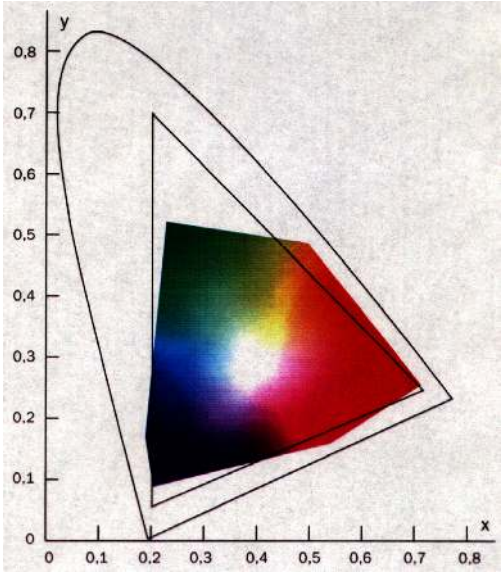


Das CIE-System

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



✎ **Anmerkung:** Darstellbare Farben: Die vorherige Folie zeigt die Teilmengen der darstellbaren Farben für unterschiedliche Medien. Die Farbfläche stellt diejenigen Farben dar, die auf gestrichenem Papier gedruckt werden können (die Farben sind nur angenähert genau). Das schwarze Dreieck enthält die Farben, die ein Bildschirm wiedergeben kann. Die äußere Linie enthält die Farben, die das Auge sehen kann.

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'.

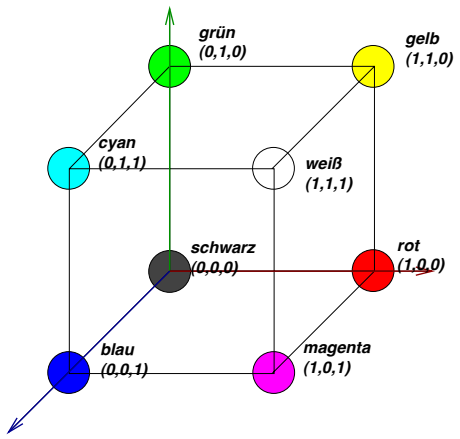
$$u' = \frac{2x}{6y - x + 1.5} \quad v' = \frac{4.5y}{6y - x + 1.5}$$

19.3.2 RGB-System und CMY-System

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 Schwarzpunkt 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.

✎ **Anmerkung:** Farbsysteme für die Fernsehübertragung

Für die Fernsehübertragung ist eine Repräsentation vorteilhaft, aus der sich sowohl ein Farbbild als auch ein Schwarz-Weiß-Bild ableiten lassen. Für die Fernsehübertragung werden daher zwar auch Farbvektoren mit drei Komponenten zur Beschreibung der Farbe verwendet, aber eine der Komponenten ist das Intensitätssignal, das so direkt in ein monochromes Bild umgesetzt werden kann. Die Systeme PAL und NTSC unterscheiden sich nur darin, wie die Farbe kodiert wird, nicht in der Codierung der Intensität.

YIQ

Farbsysteme für die Fernsehübertragung

- 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.

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

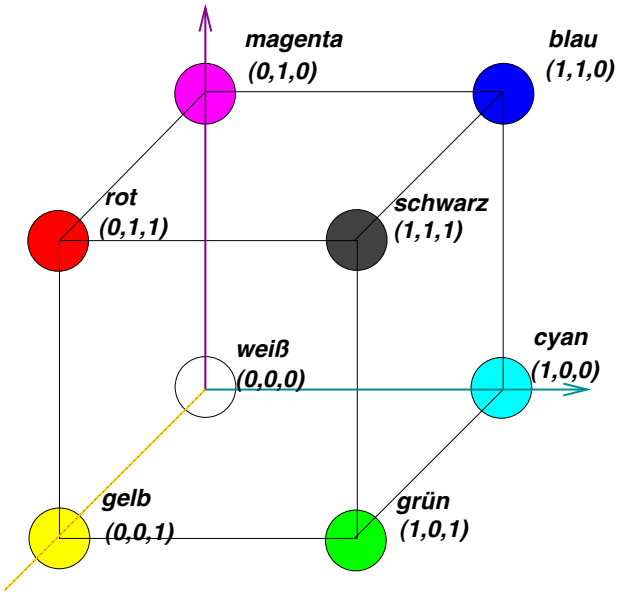
Das PAL-System verwendet die Komponenten: Y, R-Y, B-Y.

Das CMY-Modell

- Das CMY-Modell ist ein *subtraktives* Farbmodell für den Farbdruck. Es bescheibt, 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.

✎ **Anmerkung:** Druckfarben: Zwar lassen sich mit den drei Grundfarben Cyan, Magenta und Gelb alle Farben des RGB-Systems darstellen, aber für gute Druckresultate ist es notwendig, noch Schwarz dazuzumischen, um einen guten Kontrast zu erzielen.

Der CMY-Einheitswürfel



Umwandlung zwischen RGB-und CMY-Werten

Es werden die Farbvektoren der Komplementärfarben berechnet:

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

Die bisher vorgestellten Farbsysteme sind hervorragend zur technischen Erzeugung von Farben geeinet, aber nicht sehr angenehm zu benutzen, wenn man interaktiv Farben zusammenstellen will. Da ist es angenehmer, wenn man sich auf Farbsysteme beziehen kann, die den kognitiven Farbparametern näherkommen.

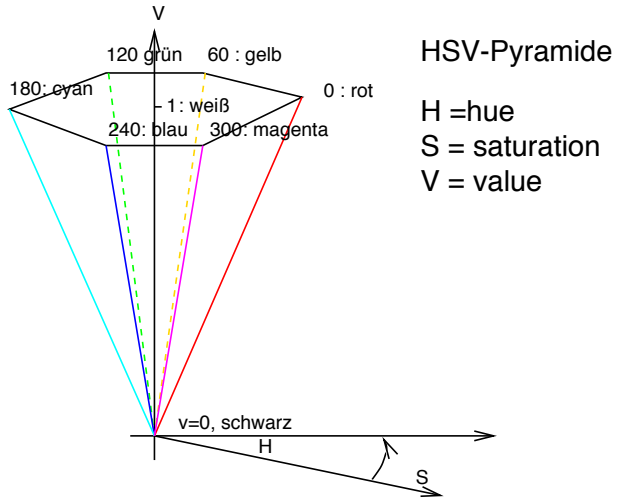
19.3.3 HSV-System

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.

✎ **Anmerkung:** : Es gibt eine Reihe weiterer firmeneigener Systeme, beispielsweise das HLS-System der Firma Tektronix. Hier ist der Farbkreis gegenüber dem HSV-System um 120°gedreht.

Das CNS-Modell (color naming system)

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

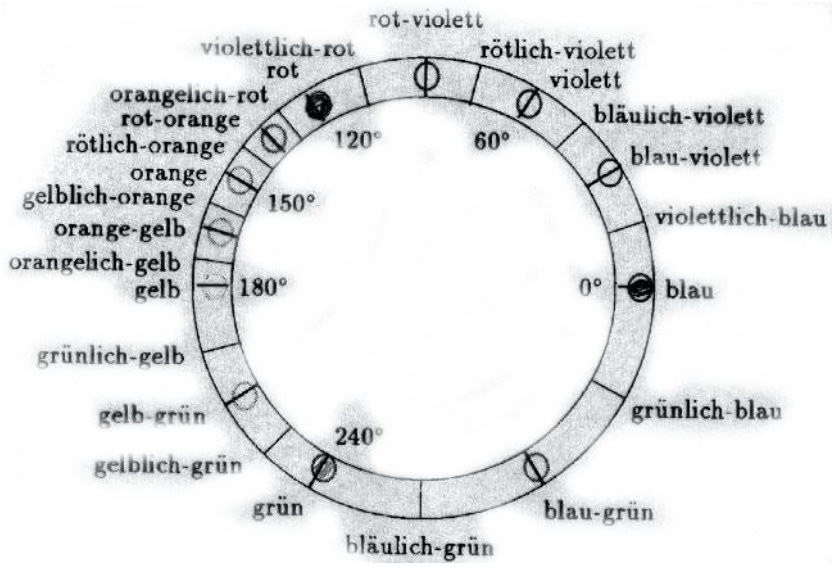
Parameter: Farbe, Sättigung, Intensität.

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, orangelich-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.

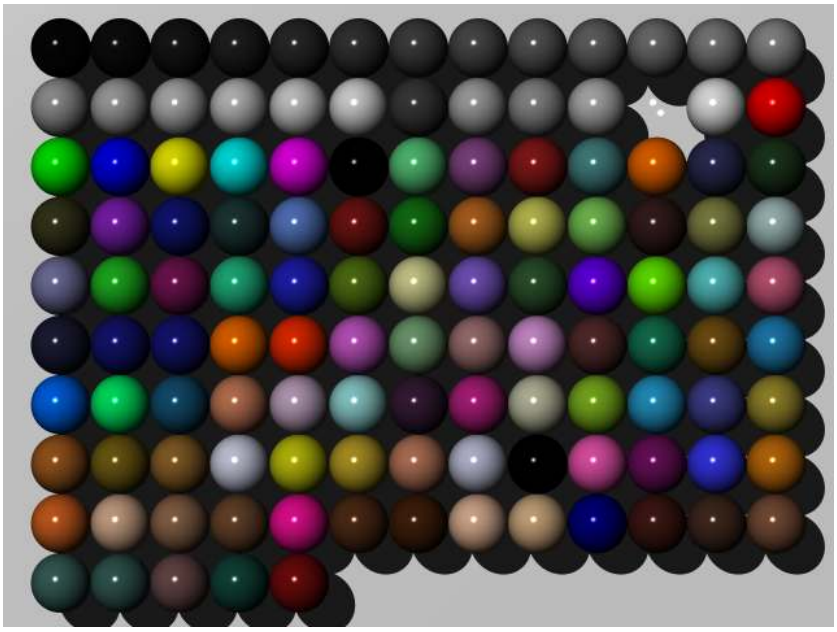
Farbvektoren: `color rgb` $< 1, 0.5, 0.5 >$, ein ungesättigtes Rot.

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

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 0.258824blue 0.435294
#declare HuntersGreen = color red 0.13
         green 0.37 blue 0.31
```

Die Farben aus colors.inc



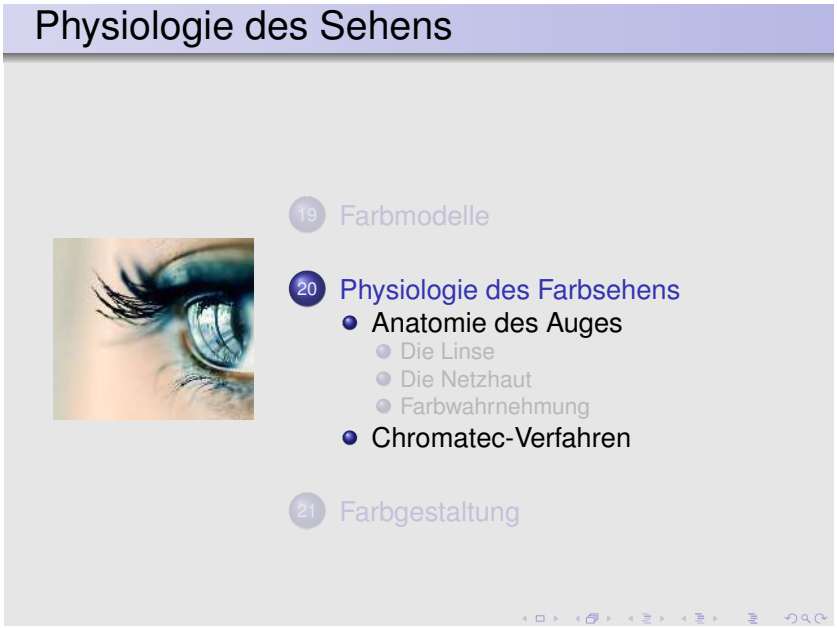
	0	1	2	3
1	Gray05	Gray10	Gray15	Gray20
2	Gray70	Gray75	Gray80	Gray85
3	Green	Blue	Yellow	Cyan
4	DarkOliveGreen	DarkOrchid	DarkSlateBlue	DarkSlateGrey
5	LightSteelBlue	LimeGreen	Maroon	MediumAquamarine
6	MidnightBlue	Navy	NavyBlue	Orange
7	SlateBlue	SpringGreen	SteelBlue	Tan
8	Copper	Bronze	Bronze2	Silver
9	MandarinOrange	LightWood	MediumWood	DarkWood
10	GreenCopper	DkGreenCopper	DustyRose	HuntersGreen

	4	5	6	7
1	Gray25	Gray30	Gray35	Gray40
2	Gray90	Gray95	DimGray	Grey
3	Magenta	Black	Aquamarine	BlueViolet
4	DarkTurquoise	Firebrick	ForestGreen	Gold
5	MediumBlue	MediumForestGreen	MediumGoldenrod	MediumOrchid
6	OrangeRed	Orchid	PaleGreen	Pink
7	Thistle	Turquoise	Violet	VioletRed
8	BrightGold	OldGold	Feldspar	Quartz
9	SpicyPink	SemiSweetChoc	BakersChoc	Flesh
10	Scarlet			

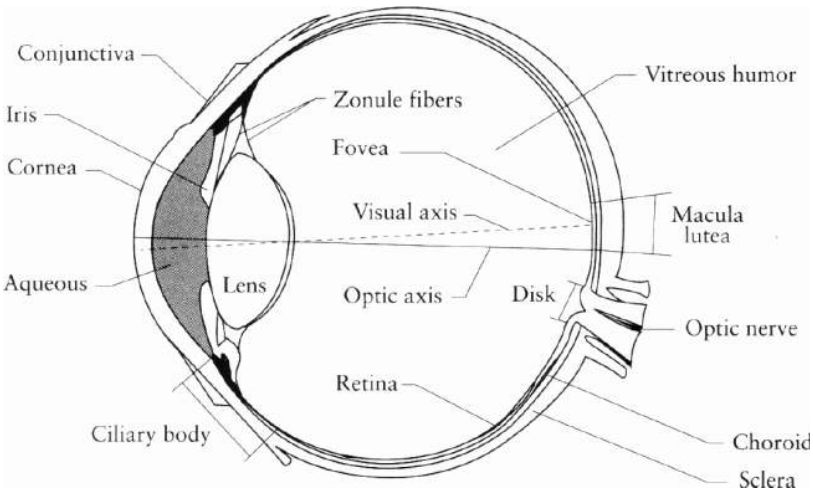
	8	9	10	11	12
1	Gray45	Gray50	Gray55	Gray60	Gray65
2	LightGrey	VLightGrey	Clear	White	Red
3	Brown	CadetBlue	Coral	CornflowerBlue	DarkGreen
4	Goldenrod	GreenYellow	IndianRed	Khaki	LightBlue
5	MediumSeaGreen	MediumSlateBlue	MediumSpringGreen	MediumTurquoise	MediumVioletRed
6	Plum	Salmon	SeaGreen	Sienna	SkyBlue
7	Wheat	YellowGreen	SummerSky	RichBlue	Brass
8	Mica	NeonPink	DarkPurple	NeonBlue	CoolCopper
9	NewTan	NewMidnightBlue	VeryDarkBrown	DarkBrown	DarkTan

20 Physiologie des Farbsehens

20.1 Anatomie des Auges



Ein Querschnitt durch das Auge



✎ **Anmerkung:** Der Aufbau des Auges Das Auge ist ein Organ, mit dem Abbildungen der Umwelt als Sinneseindruck an das Gehirn weitergegeben werden können. Wichtige Bestandteile des Auges sind:

Die Linse: Die Cornea und die Linse erzeugen ein Bild der Umwelt auf der Netzhaut. Die Cornea ist starr, aber die Linse ist eine elastische Sammellinse. Von der Form der Linse hängt es ab, ob nahe oder ferne Objekte auf der Netzhaut ein scharfes Bild ergeben. Im entspannten Zustand werden weit entfernte Gegenstände fokussiert, im angespannten Zustand nahe Gegenstände.

Die Iris: Die Iris ist eine kreisförmige Blende, mit der der Lichteinfall ins Auge geregelt werden kann.

Die Retina: Die Retina (Netzhaut) enthält Photorezeptoren (120 Mio. Stäbe (rods), 6 Mio. Zapfen (cones)), die auf Licht reagieren und Nervenreize erzeugen. Die Dichte der Rezeptoren ist am größten in der Fovea, dem Punkt schärfsten Sehens. Dort, wo der Sehnerv in die Retina mündet, gibt es einen „blinden Fleck“.

Der Sehnerv: Der Sehnerv hat ungefähr 1 Mio. Fasern. Die in den Signalen der Stäbchen und Zäpchen enthaltene Information wird also schon im Auge vorverarbeitet und reduziert, ehe sie an das Gehirn weitergeleitet wird.

Literaturhinweise: Zum Aufbau des Auges siehe (Glassner, 1995), zu den Theorien des Farbsehens (Erb and Fahle, 2006), zur Farbgestaltung (Fellner, 2004).

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.

20.1.1 Die Linse

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.

Die folgende Folie zeigt, wie sich die Durchlässigkeit der Linse mit zunehmendem Alter verändert. Dargestellt ist die Transmission in Abhängigkeit von der Wellenlänge für Linsen unterschiedlichen Alters.

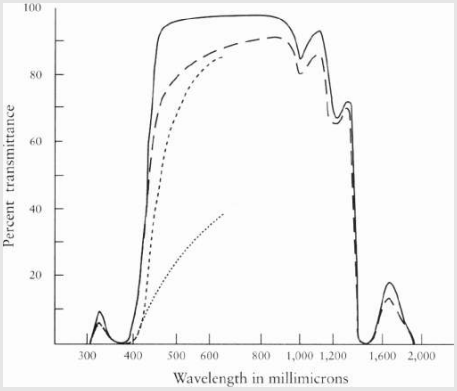
Durchgezogene Linie: Totale Transmission, für eine 4.5 Jahre alte Linse.

Lange Striche: Direkte Transmission, für eine 4.5 Jahre alte Linse.

Kurze Striche: Direkte Transmission, für eine 53 Jahre alte Linse.

Punkte: Direkte Transmission, für eine 75 Jahre alte Linse.

Altersbedingte Transmission der Linse



— — 4.5 Jahre
— 53 Jahre
. . . 75 Jahre

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.

Schon im 14. Jahrhundert wurden Linsen zur Verbesserung der Sehfähigkeit eingesetzt. Dieses Bild zeigt einen Chor aus dem 16. Jahrhundert bei einer Aufführung. Einer der Geistlichen trägt eine Brille, um die Noten besser lesen zu können.

20.1.2 Die Netzhaut

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. (Young, 1802) (von Helmholtz, 1852) (Maxwell, 1860)

Photometrische Auflösung des Auges

Das Auge kann ca.

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

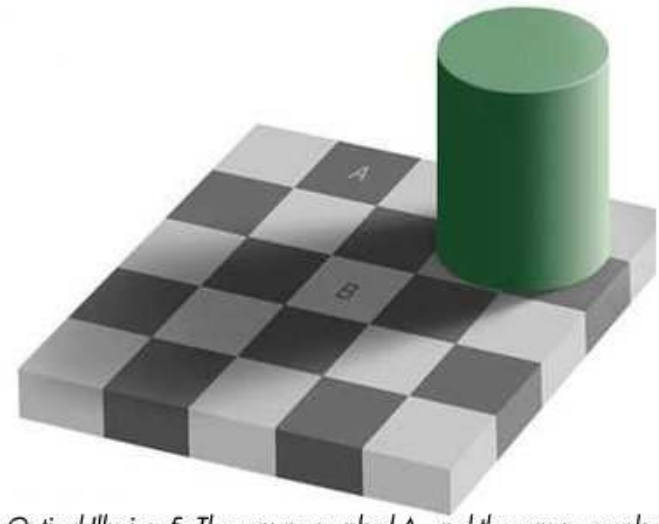
Beim achromatischen Sehen (nachts) können nur die ca. 500 Helligkeiten unterschieden werden, aber beim Farbsehen bis zu $500 \times 200 \times 26 = 2600000$ Farbstufen.

Nach (Erb and Fahle, 2006), (Fellner, 1992) dagegen sagt, daß ca. 350000 Farbstufen unterschieden werden können.

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.)

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)

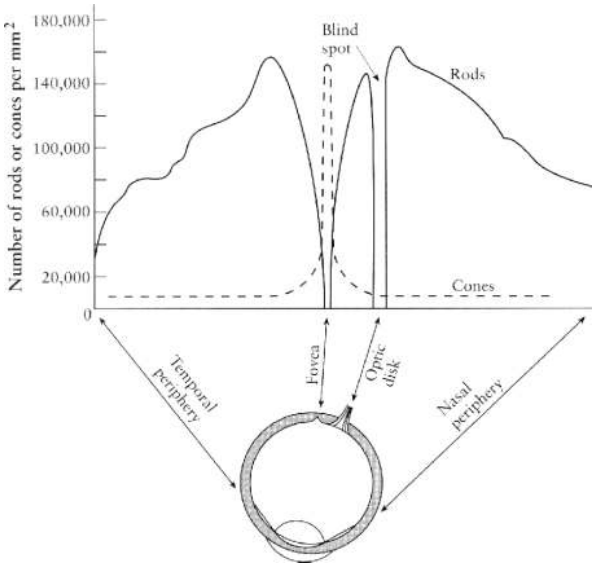
M-Zapfen: (530 nm)

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



Verteilung der Rezeptoren auf der Netzhaut: Auf der vorherigen Folie ist dargestellt, wie sich die Stäbe (rods) und Zapfen (cones) auf der Netzhaut verteilen. Die meisten Zapfen finden sich in der Fovea, wo also Farben besonders gut unterschieden werden können. Dagegen gibt es hier keine Stäbe, so daß Objekte mit geringer Helligkeit nicht gesehen werden können, wenn wir sie direkt fixieren. Wenn wir sie dagegen so betrachten, daß das Licht nicht in der Fovea sondern leicht daneben auf die Retina trifft (indirektes Sehen), dann werden sie plötzlich sichtbar. In den Randbereichen des Auges gibt es mehr Stäbe als Zapfen. Die Stäbe sind empfindlicher als die Zapfen und können auch noch bei schwachem Licht reagieren, aber keine Farbeindrücke vermitteln („Bei Nacht sind alle Katzen grau.“). Am Rand des Gesichtsfelds ist auch die Dichte der Stäbe gering. Daher ist die Auflösung in den Randbereichen des Auges deutlich schlechter als im Zentrum.

Ein Testbild Die Buchstaben auf der folgenden Folie sollten alle gleich gut zu erkennen sein, wenn Sie die Mitte des Bildes fixieren. Am Rand des Gesichtsfeldes müssen die Buchstaben sehr viel größer sein als im Zentrum des schärfsten Sehens.

Geometrische Auflösung der Netzhaut

Verteilung der Zapfen auf der Netzhaut:

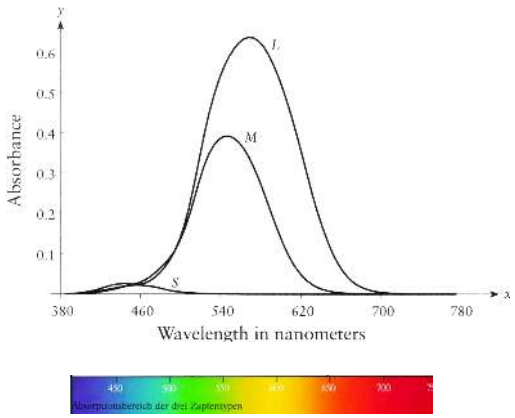
(S) **blau**: ca. 2% der Zapfen sind blau-empfindlich. Sie liegen überwiegend im äußeren Ring der Netzhaut. Kleine blaue Objekte können unsichtbar werden, wenn das Auge sich darauf konzentriert.

(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



20.1.3 Farbwahrnehmung

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 unter schiedlichen Spektren farblich identisch, wenn sie die Rezeptoren im gleichen Verhältnis erregen und daher dieselbe neuronale Reizantwort auslösen.

(Erb and Fahle, 2006)

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:

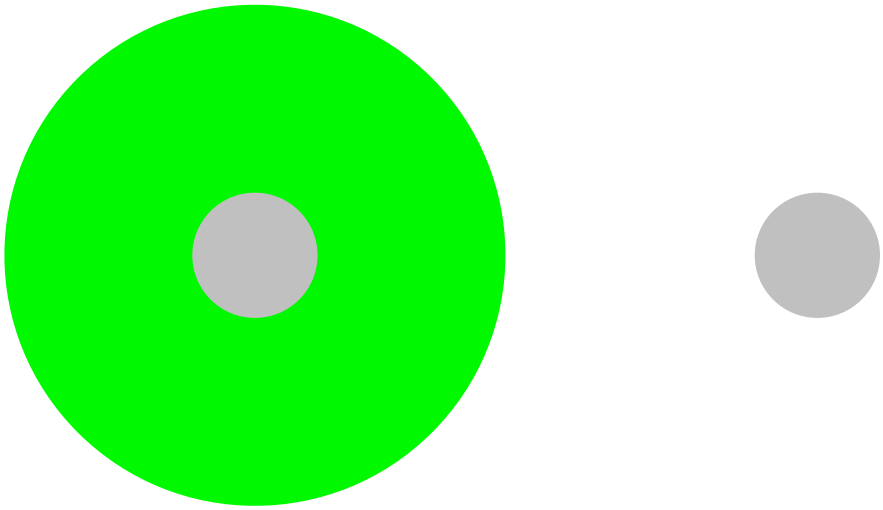
Satz: 52
Die Farben Rot und Grün so wie Blau und Gelb sind perzeptuell je zu einem Gegensatzpaar verbunden.

Beweis: 53
Durch Beobachtungen

Farbiger Sukzessivkontrast: *Farbige Nachbilder.*

Farbiger Simultankontrast: *Scheinfarben in der Nachbarschaft.*

Farbiger Simultankontrast



Beide grauen Kreise haben dieselbe Farbe.

Zonentheorie

Zonentheorie
Die Zonentheorie ist eine Synthese der Gegenfarbentheorie und der trichromatischen Theorie:

- Die trichromatischen Theorie gilt für den Bereich der Signalaufnahme in den Photorezeptoren;
- Die Gegenfarbentheorie (in neuer Form) gilt für die zeitliche und räumliche Signalverarbeitung in der Retina und dem afferenten visuellen System so wie dem Kortex.

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.

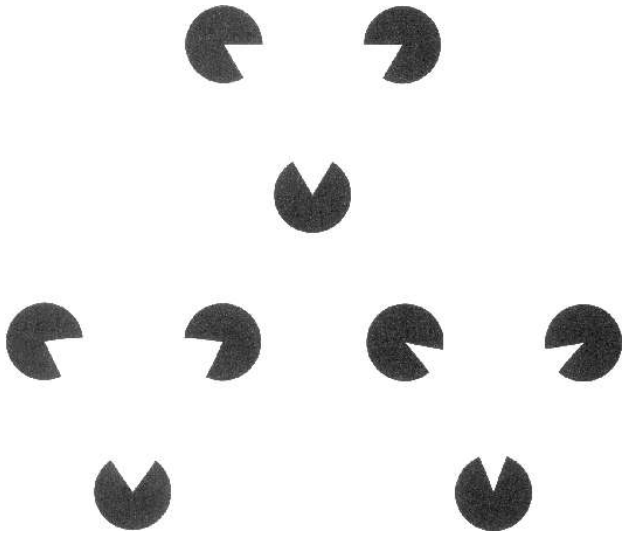
Beispiel: Drei Farbkomponenten eines Farbbildes

Zwei Formen der Fehlsichtigkeit

Deuteranopie Protanopie

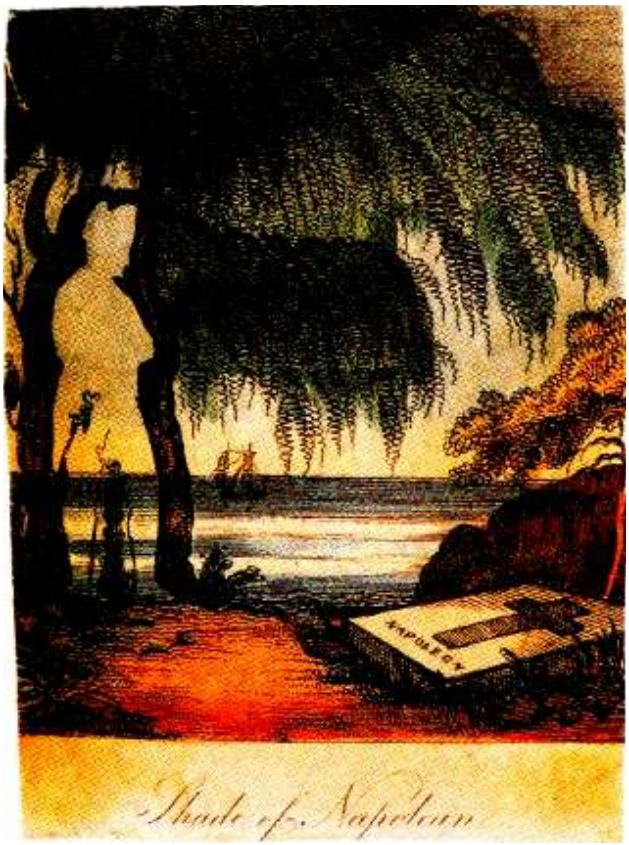
Verstehen visueller Signale Das menschliche Sehsystem besteht nicht nur aus dem Auge, sondern ein großer Teil des Gehirns ist mit der Interpretation der visuellen Signale beschäftigt. Ein Beleg dafür sind optische Täuschungen, die darauf zurückzuführen sind, daß wir bewußt gar nicht das optische Signal selbst wahrnehmen, sondern erst die vom Gehirn vorverarbeitete Information. So kann es dazu kommen, daß wir auch in einem Bild Strukturen sehen können, die im Bild gar nicht vorhanden sind. Ein Beispiel dafür ist das Bild mit den regelmäßig angeordneten Kreissegmenten auf der vorherigen Folie. Unser Sehsystem erkennt, daß die Aussparungen in den Kreisen sich zu Dreiecken ergänzen lassen und wir sehen in der Tat drei Dreiecke.

Gestaltswahrnehmung



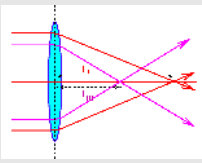
Lernfähigkeit des Sehsystems: Das Sehsystem ist lernfähig. Wenn wir einmal ein Bild betrachtet haben, das schwierig zu interpretieren ist, wie beispielsweise auf dem vorherigen Bild von Napoleons Geist an seinem Grab, und die Lösung gefunden haben, dann werden wir diese Lösung beim zweiten Betrachten sofort wieder sehen.

Napoleon an seinem Grab



20.2 Chromatec-Verfahren

Chromatec-Verfahren

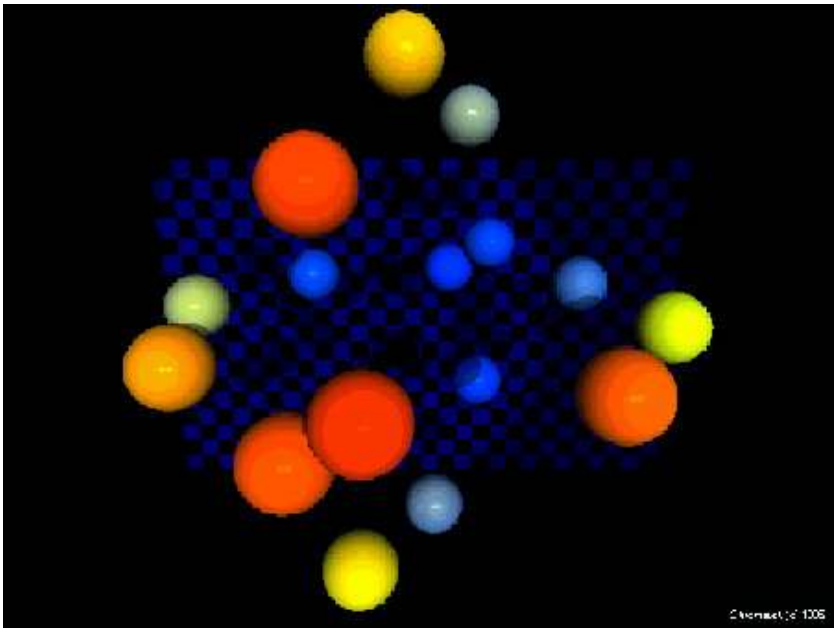


Chromatische Aberration

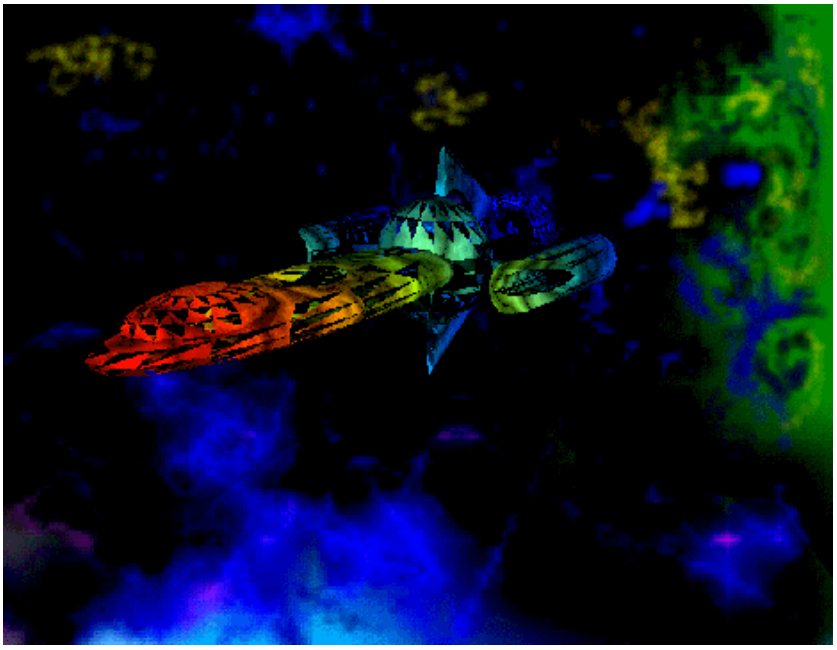
- Die Linse des Auges ist nicht gegen chromatische Aberration korrigiert.
- Rote Objekte erscheinen bei gleicher Entfernung näher als blaue Objekte.

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

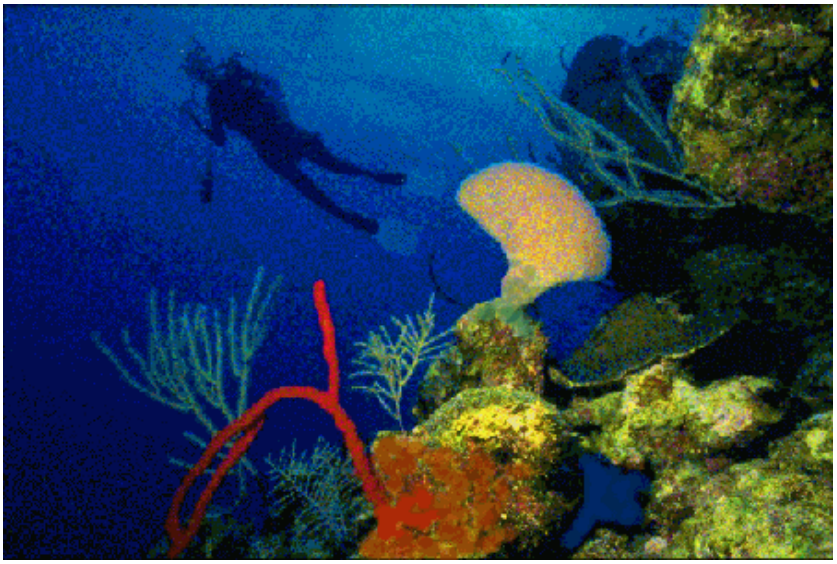
Beipiele:



Beipiele:

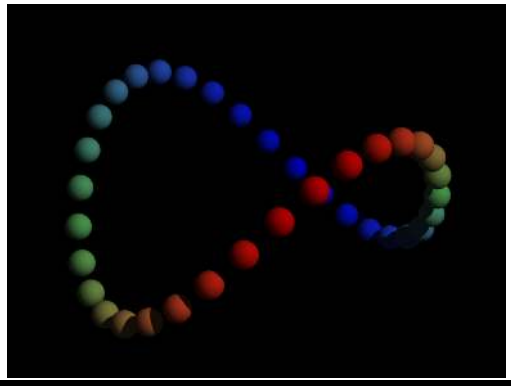


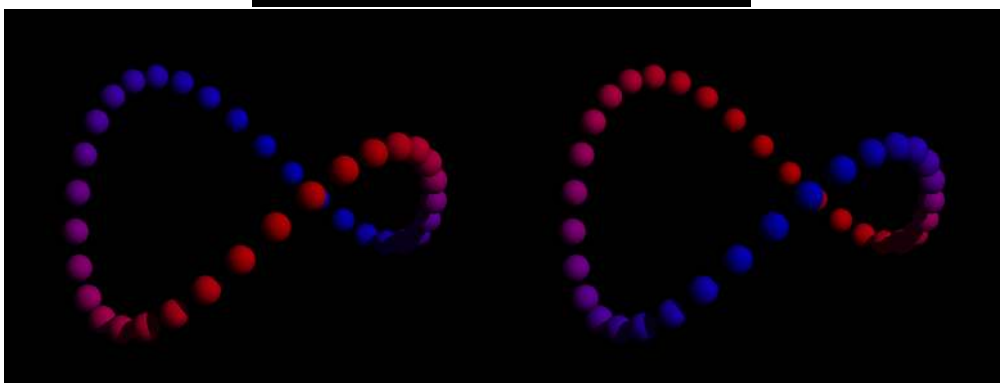
Beipiele:



Beipiele:

Welche Farbverteilung wirkt plastischer?





21 Farbgestaltung

nach (Fellner, 2004)

Farbgestaltung:

- Reines Blau sollte für Text, dünne Linien und kleine Objekte vermieden werden.
- Aneinandergrenzende 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.

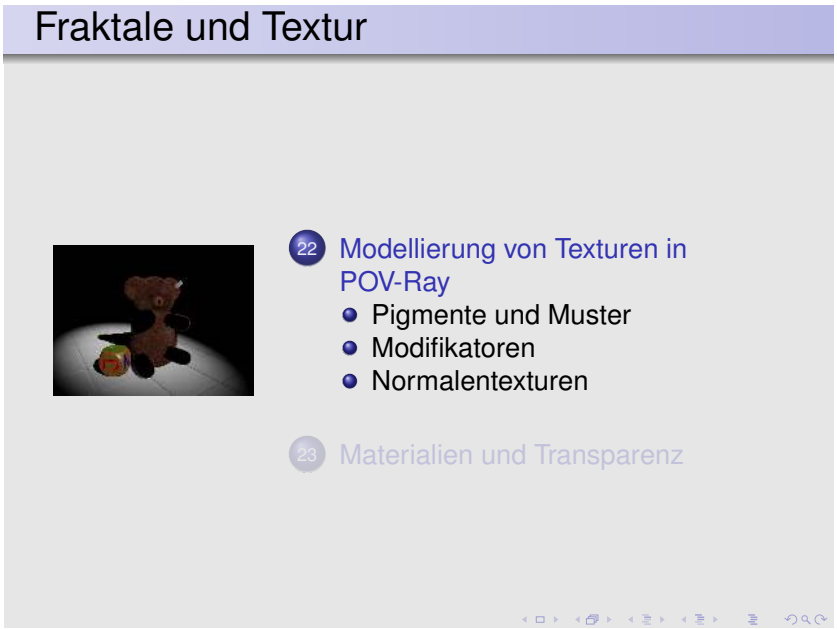
Farbgestaltung:

- 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



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 *Rauhigkeit der Oberfläche* oder beides zurück.
- Texturen sind wichtig für die Erzeugung photorealistischer Bilder.
- Texturgradienten sind wichtig für den räumlichen Eindruck.

Die folgenden Bilder zeigen typische, natürliche Texturen: Auf dem ersten Bild (Tahiti, Blick auf Morea) sind Wolkentexturen am Himmel, Wellentexturen auf dem Wasser und Texturen in der Vegetation zu sehen. Das zweite Bild (Pancake Rocks, New Zealand) zeigt die regelmäßigen Schichten von texturiertem Gestein.

Um solche Texturen synthetisch zu erzeugen, benötigen wir Algorithmen, die ein regelmäßiges Grundmuster zufällig verändern.

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 Rauhigkeit 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**.

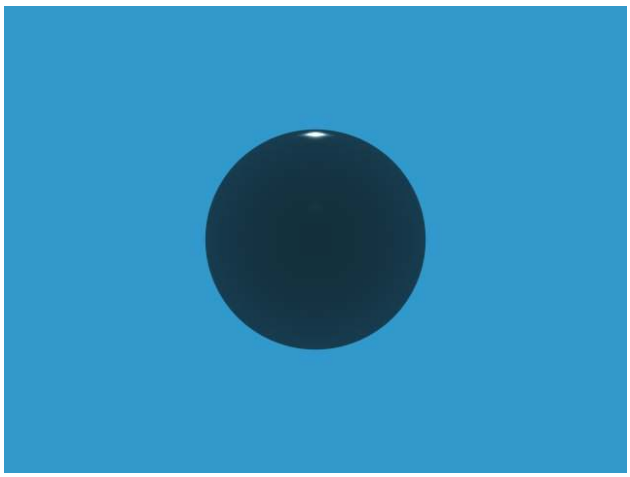
Für die Modellierung des Inneren siehe Kapitel 18.3.2.

Beipiel: Material

Beispiel: 54
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 Rauhigkeit der Oberfläche kann modelliert werden, indem die Orientierung der Oberflächenormalen lokal variiert wird. Weitere Parameter beziehen sich auf die Reflektanz: matt, spiegelnd usw.

Objekte sind nicht nur an der Oberfläche pigmentiert, sondern durchgefärbt. Das ist wichtig, wenn Objekte mit anderen geschnitten werden. An den Schnittflächen werden die Pigmente sichtbar.

Für Fortgeschrittene: In POV-Ray können Texturen geschichtet sein – es ist beispielsweise möglich, eine Holztextur mit einer dünnen, teilweise transparenten Schicht zu überziehen, um einen Klarlack zu simulieren.

Viele Beispiele finden Sie im file `textures.inc`, der im POV-Ray-include-Verzeichnis installiert sein sollte.

22.1 Pigmente und 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.

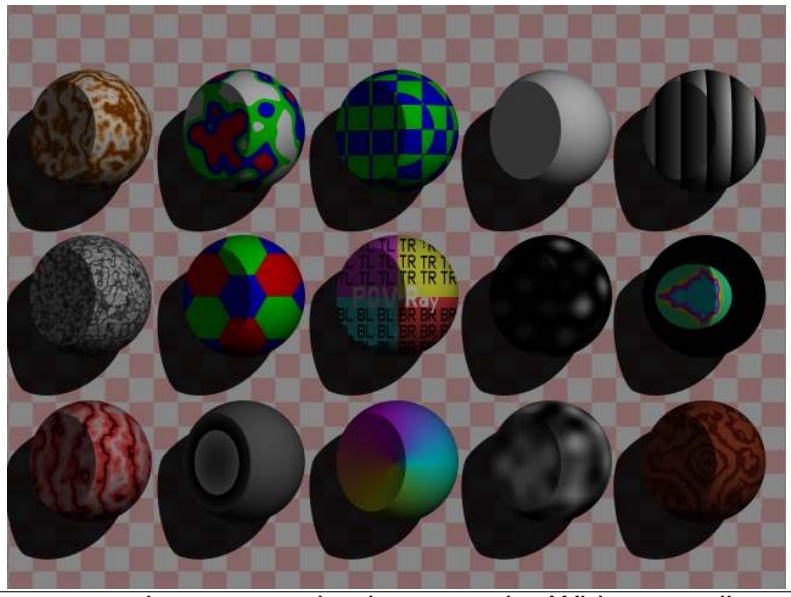
spiral1,2: Konzentrische Spiralen.

wood: Konzentrische Ringe, wie Holzmaserung.

mandel: Die Mandelbrot-Menge, ein fraktales Muster.

quilt: Linien, wie auf einer Stepdecke.

Der Musterkatalog

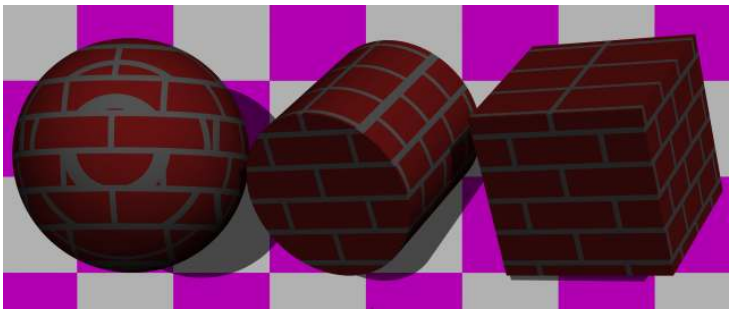


agate	bozo	checker	color White	gradient x
granite	hexagon	image_map	leopard	mandel
marble	onion	radial	spotted	wood

ALs Beispiel sehen wir auf den folgenden Folien eine Auswahl der Muster. Die Muster können entweder als Pigmentkarte eingesetzt werden – dann variiert die Farbverteilung – oder als Normalentextur – dann variiert die Oberflächennormale und damit die Orientierung zur Lichtquelle. Die Beispiele stammen aus den povray 3.6 Standard-Demos, im Verzeichnis `/POV-Ray 3.6/scenes/textures/pigments/`.

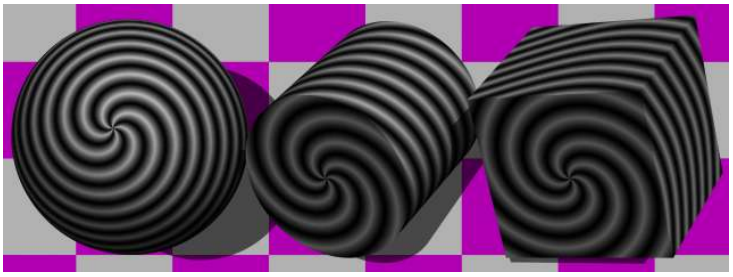
Beispiel: brick

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



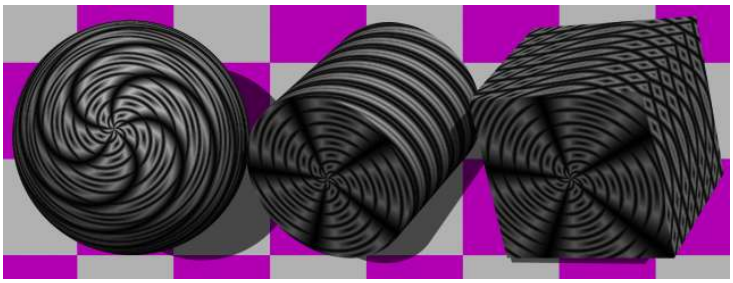
Beispiel: spiral1

```
#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}}
```



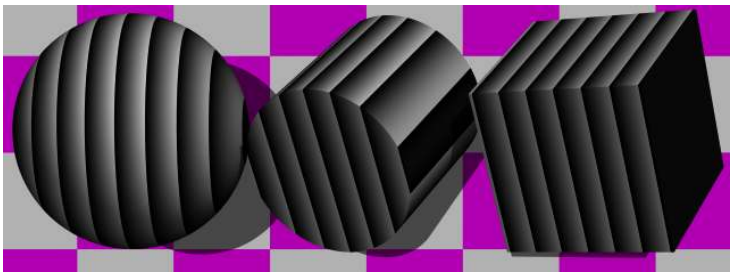
Beispiel: spiral2

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



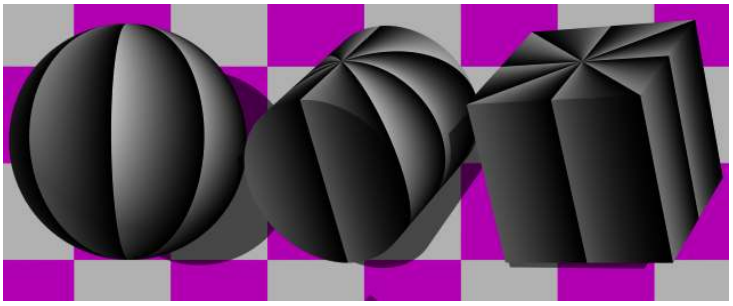
Beispiel: gradient

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



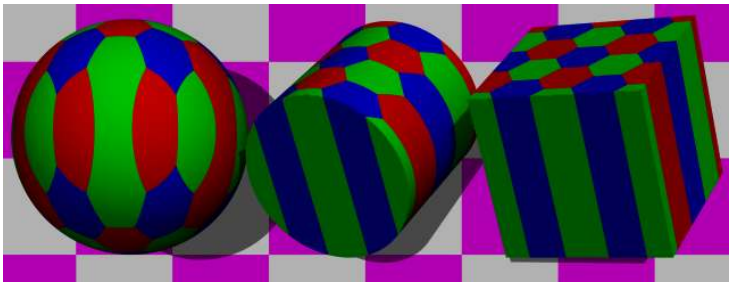
Beispiel: radial

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



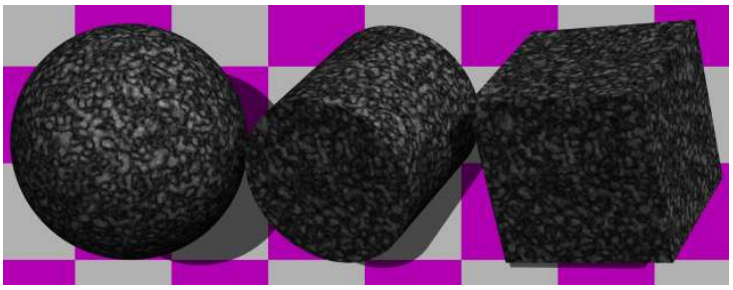
Beispiel: hexagon

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



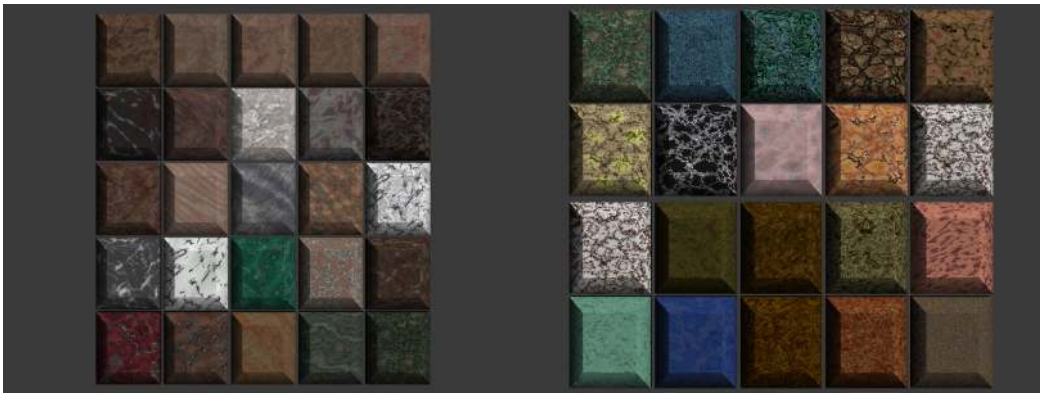
Beispiel: granite

```
#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



22.2 Modifikatoren

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.

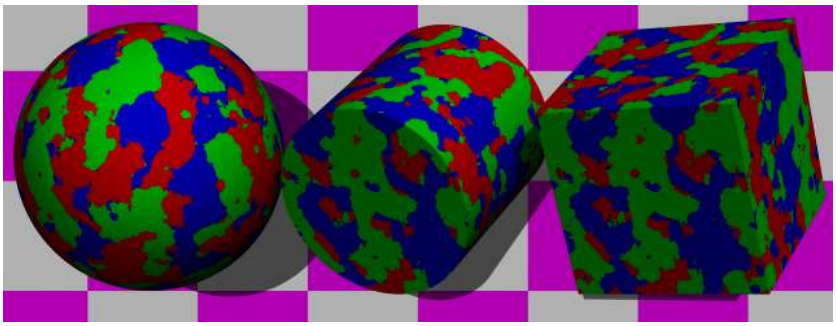
frequency, phase: Systematische Veränderung der Periode und Phasenlage.

turbulence: Zufällige Modifikationen:

```
turbulence <turbulence vector>
```

Beispiel: hexagon-Muster mit Turbulenz

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

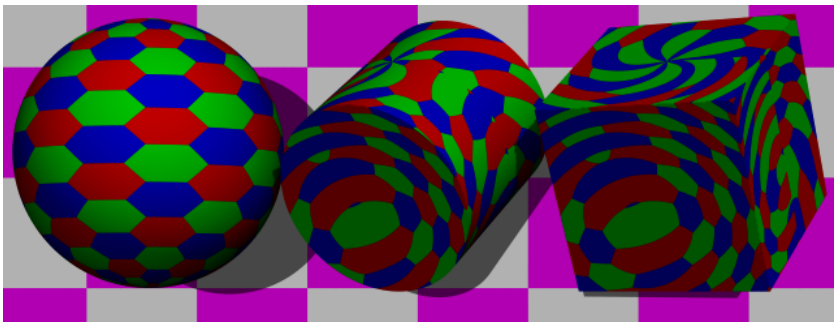


Beispiel: hexagon-Muster mit warp

Verdrehen um die Y-Achse:

```
#declare T1=
 texture{
   pigment{
     hexagon
     scale 0.048
```

```
warp {
  spherical
  orientation y
  dist_exp 1 }}}
```



Die Erdoberfläche als Texturkarte modelliert:
http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movTaeglBewegungWahreErde.html

Das Windsystem der Erde dynamisch mit Warp und Turbulence modelliert:
http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movWindsystem.html

22.3 Normalentexturen

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.

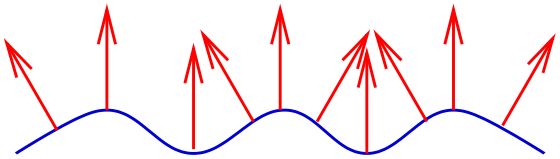
- ☞ Beispiele: Wellen, Dünen usw.

- ☞ Diese Texturen können in POV-Ray dadurch modelliert werden, daß die Richtung der Oberflächennormalen systematisch und/ oder zufällig variiert wird.

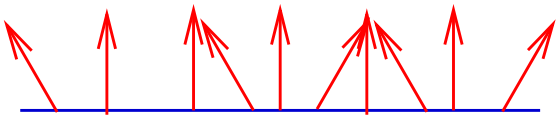
Variation der Oberflächennormalen

Variation der Oberflächennormalen

Eine rauhe Oberfläche, Normalenvektoren



Modellierung in POV-Ray



Über Normalen-Texturen lassen sich Wellen auf dem Wasser, Sand, rauhes Gestein und vieles mehr modellieren und mit wenig Aufwand berechnen.

Achtung! POV-Ray (und viele andere Systeme) variieren wirklich *nur* die *Orientierung* der Oberflächennormalen, mit der sie rechnen, nicht den *Verlauf* der Oberfläche. Physikalisch sind diese Flächen nicht möglich und meistens leicht daran zu erkennen, daß die Oberflächenstrukturen keine Schatten werfen und die Kontur der dargestellten Objekte glatt ist. Diese Schummelei läßt sich teilweise dadurch kaschieren, daß sie die Kamera am selben Ort aufstellen wie die Lichtquelle. Die Schatten wären dann ohnehin nicht sichtbar, da sie durch das Objekt verdeckt werden.

An den folgenden Beispielen aus dem Verzeichnis
`POV-Ray 3.6/scenes/textures/patterns/`
können Sie den Unterschied bei der Modellierung einer Textur als Normalen-Textur gegenüber einer Modellierung als Pigmentvariation sehen. Bei den Objekte in der oberen Zeile variiert jeweils die Verteilung der Pigmente, bei den Objekten in der unteren Zeile die die Orientierung der Oberflächennormale. Dargestellt sind Beispiele für die folgenden Texturen:

`granite`: Gesteinstextur.

`waves`: konzentrische Wellen.

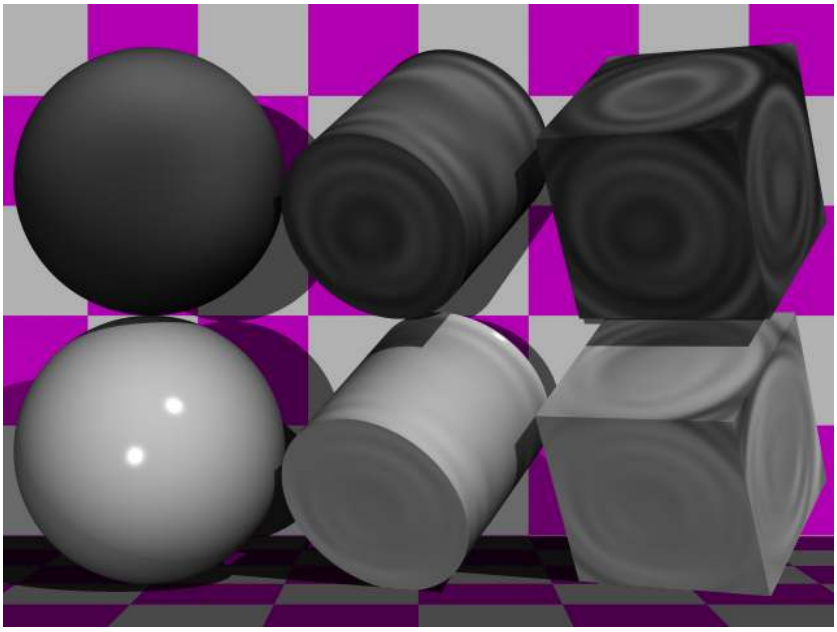
`quilted`: Rillen, wie in einer Steppdecke.

Auf alle diese Muster lassen sich die Modifikatoren anwenden, genauso wie auf die Pigmentvariation.

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}}
```

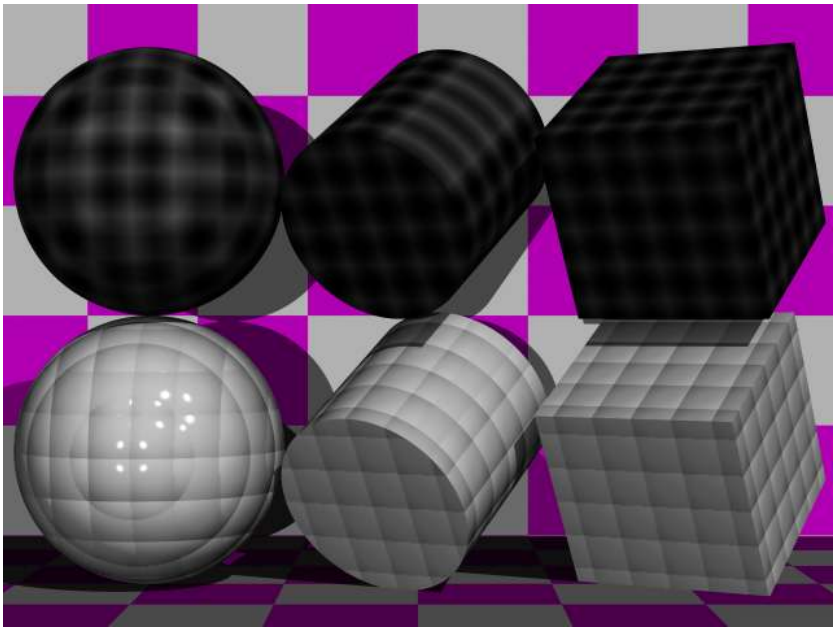
waves: als Pigment und als Normalentextur



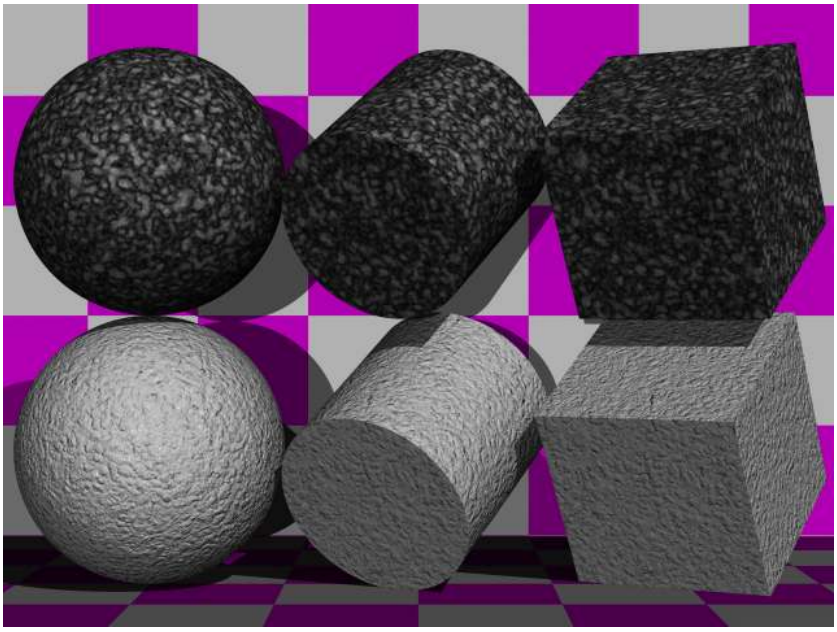
quilted: als Pigment und als Normalentextur

```
#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



Für die Texturen im folgenden Beispiel haben wir sowohl Normalen-Texturen als auch Pigmentvariation verwendet:

Wolkenhimmel: Der Himmel ist ein POV-Ray-Standardhimmel. Hier wird eine Pigmentvariation zwischen blau und weiß mit dem Modifikator `turbulence` statistisch verändert.

Sand: Der weiße Sand der Insel ist als Normalen-Textur modelliert. Das Pigment ist weiß und die Oberflächennormale simuliert regelmäßige, parallele Wellen mit viel `turbulence`.

Wasser: Die Wasseroberfläche ist eine transparente, cyan-farbene Ebene mit der Normalentexture `waves` (konzentrische Ringe) und viel `turbulence`. Etwas tiefer liegt eine zweite, cyan-farbene Ebene, die den Blick in die Tiefe versperrt.

Palmenblätter: Die Palmenblätter sind konstant grün pigmentiert und haben Blattadern als Normalentextur (`quilted`).

Palmenstamm: POV-Ray-Standardtextur `cork`, eine Mischung aus Pigmentvariation Normalen-Textur.

Zur Konstrution der Palmen und der Insel wurde die CSG verwendet: Die Palmenblätter sind aus jeweils vier Dreiecken konstruiert, der Stamm aus einer Folge von rotierten und verschobenen Kegeln. Die Insel ist eine Kugelkalotte (der Teil über Wasser), die auf einem Kegelstumpf ruht (der Teil unter Wasser).

Beispiel: 55
Abendstimmung auf einer virtuellen tropischen Insel

```
#version 3.0
// a tropical island
// by Leonie Dreschler-Fischer
global_settings { assumed_gamma 2.2 }

#include "colors.inc"
#include "stones.inc"
#include "textures.inc"
#include "finish.inc"
#include "skies.inc"

sky_sphere {S_Cloud1}

camera {
```

```
   location   <0, 20, -150>
   up         <0,  1,    0>
   right    <4/3,  0,    0>
}
#declare R1 = seed(0)

light_source {<0, 200, 0> colour White shadowless}
light_source {<0, 100, 200> colour Red}
light_source {<0, -100, 200> colour Red}
light_source {<0, 20, 200> colour Red}
light_source {<0, 20, -200> colour White shadowless}

// and the surface of the water

plane { y, 0
  texture {
    pigment { rgbf <0.6, 0.67, 0.72, 0.75> }
    normal { waves 2.0 frequency 8.0
             turbulence 0.2 scale <8.0,12,8> }
    finish { Shiny }
  }
}

plane { y, -4
  pigment {colour 0.8*Cyan}
  finish {ambient 0.9 diffuse 0.9}
}


// and the island, the cap of a sphere
union {
sphere {-9*y, 50
 texture {
  pigment { color <0.8, 0.8, 1.0> }
  finish {
   ambient 0.7
   diffuse 0.6
   crand 0.2}
 }}
// and the shore line
cone {0, 50, -30*y,150
```

```
 texture {
  pigment { color <0.6, 1.0, 1.0> }
  finish {
   ambient 0.95
   diffuse 0.9
   crand 0.2}}}
 texture {
  normal {
   ripples 4.0
   turbulence 0.25
   frequency 2}
  scale <1,30,1>}
  scale <1, 1/5, 1>
}

// the palm tree
#declare stem_segment = cone {<0,0,0>, 0.2, <0,1,0>, 0.5
 texture {Cork}
}
#declare palmtree_stem = union {
#declare grad=0;
#declare xscale=0.8;
#declare zscale=0.8;
#while (grad<85)
 object {stem_segment
         scale<4*xscale,1,zscale>
         translate x*10
         rotate z*grad}
#declare grad=grad+5;
#declare xscale=xscale*0.95;
#declare zscale=zscale*0.95;
#end
 translate -x*10
 scale <1/4, 1, 1>}

// construct a slightly curved leaf from a mesh of smooth triangles
#declare tp0=y;
#declare tp1=vaxis_rotate(tp0,z,-20);
#declare tp3=vaxis_rotate(tp0,z,20);
#declare tp4=vaxis_rotate(tp0,x,5);
#declare tp2=vaxis_rotate(tp0,x,-5);
```

```
#declare leaf=mesh {
smooth_triangle {tp0, tp0,
                 tp1, tp1,
                 tp2, tp2}
smooth_triangle {tp0, tp0,
                 tp2, tp2,
                 tp3, tp3}
smooth_triangle {tp0, tp0,
                 tp3, tp3,
                 tp4, tp4}
smooth_triangle {tp0, tp0,
                 tp4, tp4,
                 tp1, tp1}
texture { pigment{colour Green}
          finish {Shiny}
          normal { quilted 1.0 control0 1 control1 1}
//          rotate 90*z
          translate 0.5
          scale <1,0.1,1>}
translate -tp3
scale 5
}

#declare palmtree_crown = union {
#declare i=0;
#while (i<20)
 object {leaf
         scale 2
         rotate (rand(R1)-0.5)*60*z
         rotate y*i*18}
#declare i=i+1;
#end
}

#declare palm=union {
object {palmtree_stem}
object {palmtree_crown
rotate z*30
translate y*10-2.5*x
}}
```

```
object {palm
 translate y-z
 scale 5}
object {palm
 translate -22*z
 rotate 45*y
 scale 2.0}
object {palm
 translate -22*z+0.3*y
 rotate -45*y
 scale 1.5}
```

23 Materialien und Transparenz

23.1 Texturkarten:

Ein einfacher Weg, um komplizierte Pigmentvariationen oder Oberflächenverläufe zu modellieren, sind Texturkarten. Hierzu muß eine Datei mit einem Bild (gif, png usw) angegeben werden, sowie eine Projektionsart. Von der Projektionsart hängt ab, wie und wo das Muster auf die Oberfläche übertragen wird. Möglich sind Parallelprojektion, Zylinderprojektion usw.

Im folgenden Beispiel wurden Texturkarten verwendet,

- um die tropische Insel mit einem Blumenteppich zu überziehen (eine Wiese mit Löwenzahn, aufgenommen auf Hallig Hooge),
- die Palmenwedel etwas realistischer zu pigmentieren,
- das Wasser mit Haifischen zu füllen
- und zum Ausgleich für die grimmigen Haie einen freundlichen Teddy auf der Insel auszusetzen.

Für alle drei Anwendungen wurde die Parallelprojektion verwendet. Hierbei nimmt POV-Ray an, daß das Bild parallel zur z-Achse in die xy-Ebene projiziert wird und das Rechteck mit den Eckpunkten (0,0,0) und (1,1,0) ausfüllt. Das Muster muß daher vor der Projektion passend skaliert, rotiert und verschoben werden.

Das gif-Bild für die Haifische wurde beim Programm `xfishtank` ausgeborgt, der knuddelige Teddy beim Programm `xteddy`.

Beispiel: Haie im Wasser

Beispiel: 56
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)

✎ **Anmerkung:** Transparenz

```
pigment {
  image_map {
    gif "shark.gif"
    transmit 10 1.0 }} // cyan ist transparent
```

Um einzelne Farben transparent zu machen, muß man deren Position in der Farbtabelle eines gif- oder png-Bildes kennen. In diesem Beispiel ist die Farbe des Hintergrundes - cyan - als zehntes Element in der Farbtabelle eingetragen. In der transmit-Anweisung wird diese Farbe als vollständig transparent spezifiziert (1.0). Die Farbtabelle eines Bildes kann man sich beispielsweise vom Programm xv anzeigen lassen.

Einfacher ist es, png-Bilder als Texturkarten zu verwenden, die schon transparente Teile enthalten. Dann ist keine **transmit**-Spezifikation notwendig. Das ist bisher aber nur mit png-Bildern möglich.

```
// Tropical island with flowers, sharks, stingrays, and xteddy
//Leonie Dreschler-Fischer
#include "colors.inc"
#include "stones.inc"
#include "textures.inc"
#include "finish.inc"
#include "skies.inc"

sky_sphere {
S_Cloud1\}

camera {
   location  <0, 20, -150>
   up        <0,  1,    0>
   right   <4/3,  0,    0>
}

#declare R1 = seed(0)

light_source {
<0, 200, 0> colour White shadowless}

light_source {
<0, 100, 200> colour Red}

light_source {
```

```
<0, -100, 200> colour Red}

light_source {
<0, 20, 200> colour Red}

light_source {
<0, 20, -200> colour White shadowless}


// and the surface of the water

plane {
 y, 0
  texture {
     pigment {
 rgbf <0.6, 0.67, 0.72, 0.75> }
     normal {
 waves 2.0
 frequency 8.0
 turbulence 0.2
 scale <8.0,12,8> }
     finish {Shiny }}}

box {
0, <1,1, 0.001> // teddy
pigment {
  image_map {

     gif "xteddy_color.gif"
     transmit 25 1.0 }
}
// weiss (Farbe 35) ist transparent
     scale 25
     translate <0,3,-25>}

#declare stingray=
box {0, <1,1, 0.001> // stingray
pigment {
  image_map {
     gif "/u/leonie/fish/stingray.gif"
     transmit 8 1.0 }} // tuerkis ist transparent
```

```
    scale <1,0.25,1>
}
#declare shark=
box {
0, <1,1, 0.001> // shark
pigment {
  image_map {
    gif "shark.gif"
    transmit 10 1.0 }}
// tuerkis ist transparent
    scale <1,0.5,1>}

object {
shark
     scale 20
     translate -75*z-6*y}

object {shark
     scale 21
     translate -70*z-6*y
     rotate 30*y}

object {shark
     scale 19
     translate -70*z-6*y
     rotate -30*y}

object {shark
     scale 21
     translate 150*z-6*y
     rotate 30*y}

object {shark
     scale 19
     translate 150*z-6*y
     rotate -30*y}


object {stingray
     scale 25
     translate -85*z-10*y
```

```
      rotate -28*y}

object {stingray
      scale 25
      translate -85*z-10*y
      rotate 25*y}

// and the island, the cap of a sphere
union {

sphere {
-9*y, 50
 texture {
pigment {
  image_map {
     gif "butterblumen-sec.gif"
     map_type 0      }
     translate -0.5*(x+y)
     rotate x*90
     scale 40
}}}

// and the shore line
cone {
0, 50, -30*y,150
 texture {

  pigment {
 color <0.6, 1.0, 1.0> }

  finish {
   ambient 0.95
   diffuse 0.9
   crand 0.2}}}

 texture {
  normal {
   ripples 4.0
   turbulence 0.25
   frequency 2}
  scale <1,30,1>}
```

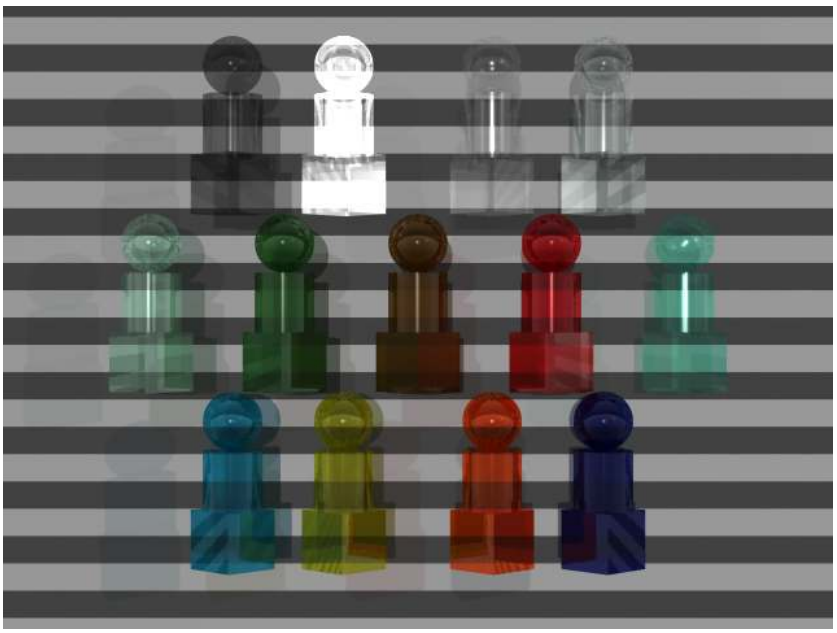
```
  scale <1, 1/5, 1>}


// the palm tree
#declare stem_segment = cone {
<0,0,0>, 0.2, <0,1,0>, 0.5
 texture {Cork}}

#declare palmtree_stem = union {
#declare grad=0;
#declare xscale=0.8;
#declare zscale=0.8;
#while (grad<85)
 object {stem_segment
         scale<4*xscale,1,zscale>
         translate x*10
         rotate z*grad}
#declare grad=grad+5
#declare xscale=xscale*0.95
#declare zscale=zscale*0.95
#end
 translate -x*10
 scale <1/4, 1, 1>}

// construct a slightly curved leaf from a mesh of smooth triangle
#declare tp0=y
#declare tp1=vaxis_rotate(tp0,z,-20);
#declare tp3=vaxis_rotate(tp0,z,20);
#declare tp4=vaxis_rotate(tp0,x,5);
#declare tp2=vaxis_rotate(tp0,x,-5);

#declare leaf=mesh {
smooth_triangle {
tp0, tp0,
                tp1, tp1,
                tp2, tp2}
smooth_triangle {
tp0, tp0,
                tp2, tp2,
                tp3, tp3}
smooth_triangle {
```

```
tp0, tp0,
                    tp3, tp3,
                    tp4, tp4}
smooth_triangle {
tp0, tp0,
                    tp4, tp4,
                    tp1, tp1}
texture {

    pigment{
 image_map {gif "wedel.gif"
    once
    transmit 0 1.0 }
// weiss ist transparent
    translate -0.5
    rotate 90*x
    scale 2*<vlength(tp1-tp3),1,vlength(tp4-tp2)>}

finish {Shiny} }
translate -tp3
scale 5}

#declare palmtree_crown = union {
#declare i=0;
#while (i<20)
 object {leaf
         scale 2
         rotate (rand(R1)-0.5)*60*z
         rotate y*i*18}
#declare i=i+1;
#end
}

#declare palm=union {
object {palmtree_stem}
object {palmtree_crown
rotate z*30
translate y*10-2.5*x }}

object {palm
 translate y-z
```

```
 scale 5}
object {palm
 translate -22*z
 rotate 45*y
 scale 2.0}

object {palm
 translate -22*z+0.3*y
 rotate -45*y
 scale 1.5}
```

23.2 Lichtdurchlässige Materialien

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:

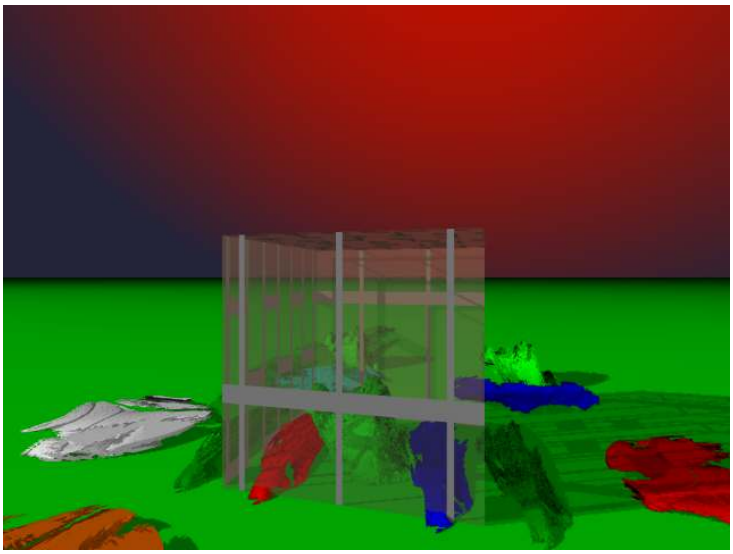
```
pigment {colour rot, grün, blau, Filter, Transparenz}
```

Ein virtuelles Treibhaus

Beispiel: 57
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

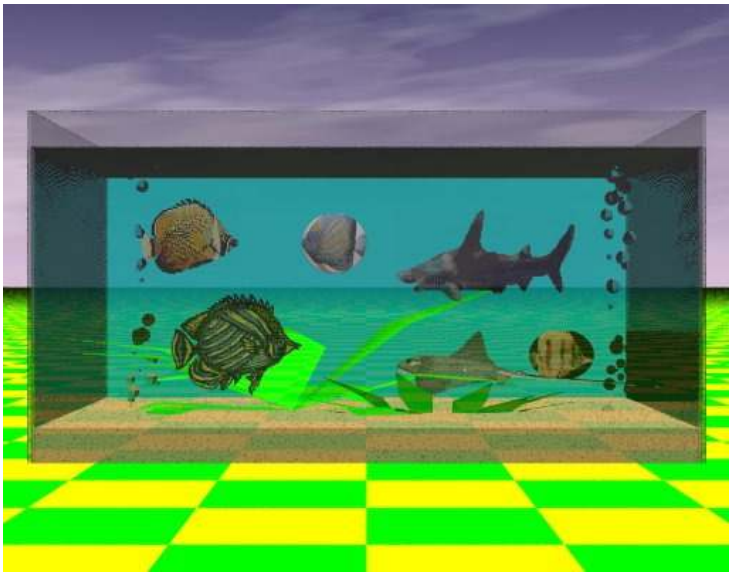


✎ **Anmerkung:** Modellierung der Blumenbeete: Die bunten Kleckse in diesem Bild sollen Blumenbeete darstellen und sind als Julia-Fraktale modelliert, wie wir sie im nächsten Kapitel kennenlernen werden.

Für das Aquarium im folgenden Bild wurden die Fische wieder mittels Texturkarten modelliert, als Algen wurden die Palmenwedel zweckentfremdet, die wir schon von der Inselszene kennen.

An dieser Szene können Sie gut ein Problem erkennen, das auftreten kann, wenn Sie sich überlagernde, transparente Materialien haben: Die Berechnung der Transparenz klappt nicht immer, wenn ein transparentes Objekt ein anderes transparentes Objekt verdeckt. Beim Fisch rechts unten, der durch die transparenten Teile des Stachelrochens verdeckt wird, sind die Konturen nicht ganz freigestellt worden.

Transparente Pigmente, Wasser und Glas



Eine Normalentexture „waves“ modelliert die Wellen auf den Ozeanen, das Wasser ist transparent: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movErdeMeereMitWellen.html

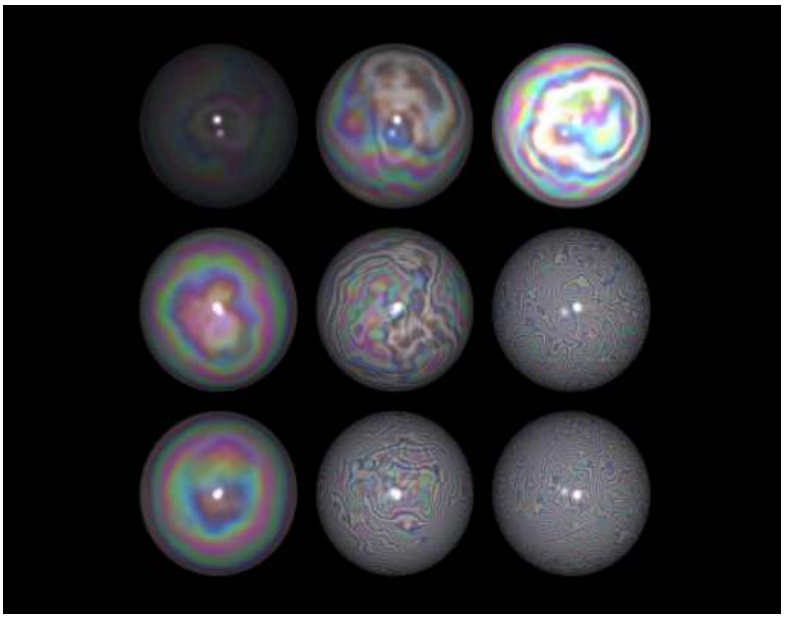
Hier folgen zusätzlich die Flutberge dem Mond um die Erde und das durch die Meere transmittierte Sonnenlicht ist als Widerschein auf dem Mond zu sehen: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movFlutberge.html

Ein Beispiel für die Verwendung einer Dichtefunktion: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movErdatmosphaere.html

Für die Interaktion von Licht und Materie und fotorealistische Modellierung von Pigmenten ist noch vieles mehr zu berücksichtigen, was wir wegen Zeitmangel in der Vorlesung leider nicht ausführlich ansprechen konnten, Streuung, Beugung, Interferenz, Emission usw. Viele dieser Effekte können Sie in POV-Ray modellieren: Schimmernde Seifenblasen, leuchtende Halos, Staubwolken und vieles mehr. Wenn Sie neugierig geworden sind und einen schnellen Rechner haben, dann sehen Sie sich einmal die Beispielszenen im Verzeichnis `POV-Ray 3.6/scenes/advanced/` an. Dort finden Sie viele spektakuläre Beispiele. Oder besuchen Sie die „Pov-Ray Hall of Fame“ auf der Pov-Ray Homepage.

Für die folgenden Beispiele wurde das irid-statement verwendet, um die Oberfläche der Objekte mit einer dünnen Haut zu überziehen.

Seifenblasen



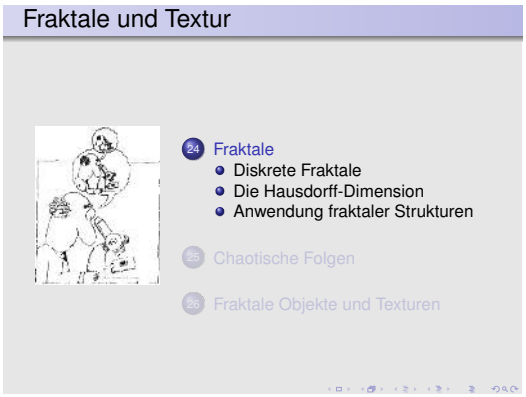
Ein dünner Ölfilm auf den Wogen

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

Teil IX

Fraktale

24 Fraktale



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.

Auf Mandelbrots ursprüngliche Frage nach der Küstenlänge von Großbritannien habe ich nie eine Antwort gefunden, aber Fellner hat sie für die Küstenlinie der U.S.A. beantwortet:

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
$\Longrightarrow$	∞

Länge ist bedeutungslos für fraktale Kurven.

24.1 Diskrete Fraktale

siehe (Fellner, 2004)

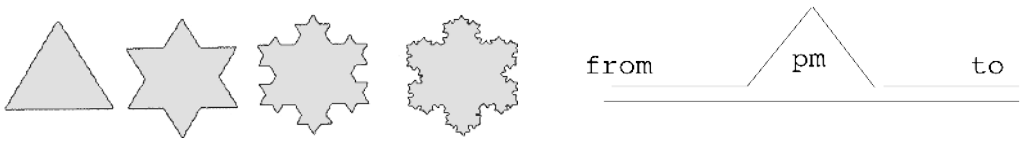
Erzeugung fraktaler Objekte

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.

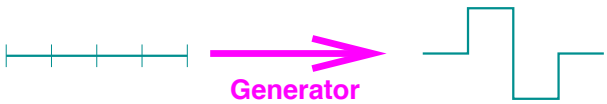
Tangenten: Es gibt nirgendwo eine Tangente, da es zwischen zwei beliebigen Punkten beliebig viele „zick-zack"-Kurven gibt.

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

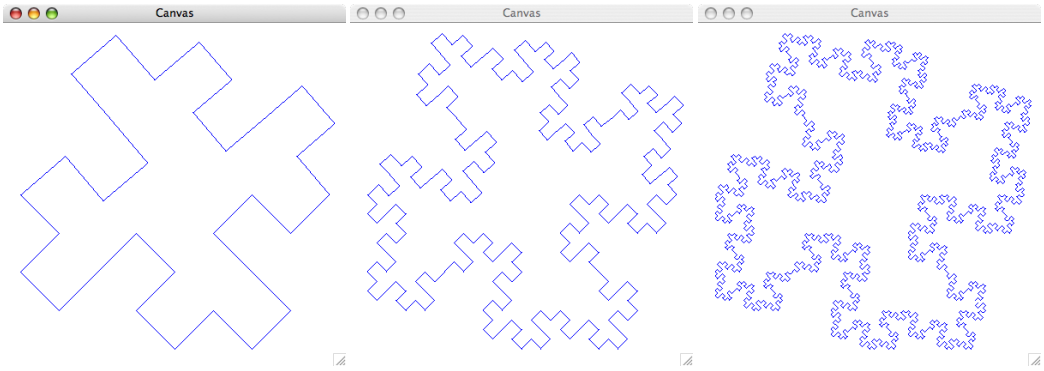
Ein fraktales Schneegestöber: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/SE3_Movies/Pages/Snowflake.html

Quadratische Koch-Kurve

Der Generator:

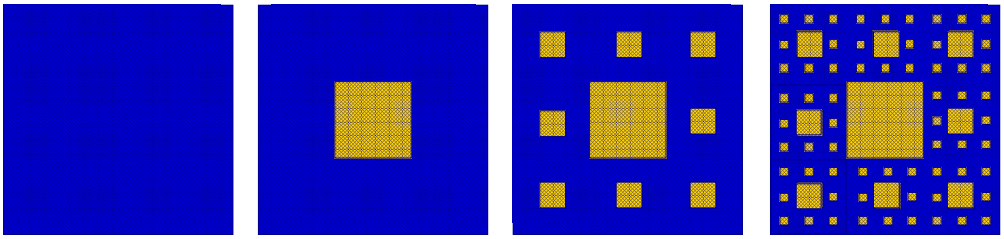


Die Länge der Kurve geht ebenfalls gegen ∞, da der Generator die Länge bei jeder Iteration verdoppelt.



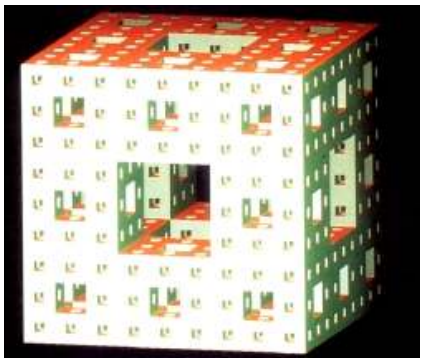
Iteration 1 Iteration 2 Iteration 3

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.

Eine Fahrt durch den Mengerschwamm: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Informatik/Abschlussfilme_CGB/2007_Menger_Schwamm.html

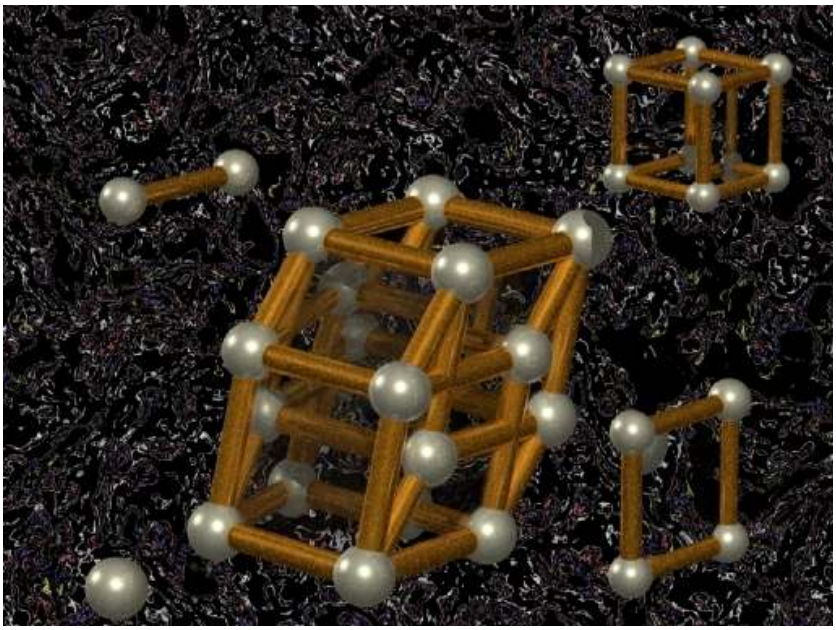
24.2 Die Hausdorff-Dimension

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: 58 (***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 $n(\varepsilon) \to \infty$ die Anzahl $n(\varepsilon)$ proportional zu ε^{-D} ist.*

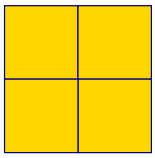
$$\begin{aligned} n(\varepsilon) &\sim \varepsilon^{-D} \\ \text{bzw. } n(\varepsilon) &= c \cdot \varepsilon^{-D} \\ D &= \frac{\log(n(\varepsilon)) - \log(c)}{-\log(\varepsilon)} \end{aligned}$$

Beispiele:

Gerade

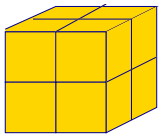
$$\begin{aligned} \varepsilon &= 0,5 \\ n &= 2 \\ D &= 1 \end{aligned} \qquad D = \frac{\log(n(\varepsilon)) - \log(c)}{-\log(\varepsilon)}$$

Quadrat



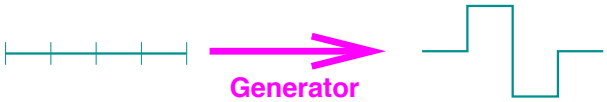
$$\begin{aligned} \varepsilon &= 0,5 \\ n &= 4 \\ D &= 2 \end{aligned}$$

Kubus



$$\begin{aligned} \varepsilon &= 0,5 \\ n &= 8 \\ D &= 3 \end{aligned}$$

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.

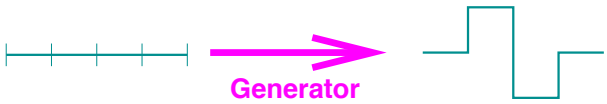
Sei S die Zahl der Kugeln, die den *Durchmesser* des Generators überdecken und N die Zahl der Kugeln, die den Generator vollständig überdecken.

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):



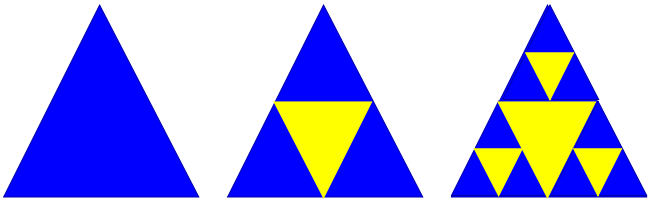
$$D = \frac{\log(N)}{\log(S)}$$

Für die Koch-Kurve: $N = 8, \quad S = 4$

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

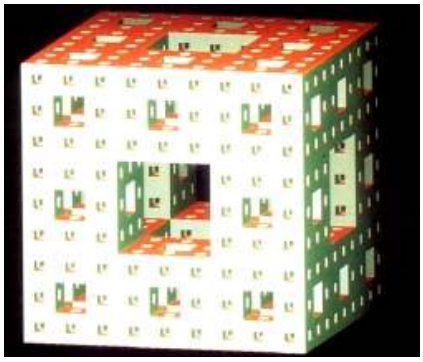
Beispiele für fraktale Dimensionen

Sierpinski-gasket



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

Die Dimension des Menger-Schwamms



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

24.3 Anwendung fraktaler Strukturen

Anwendung fraktaler Strukturen

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

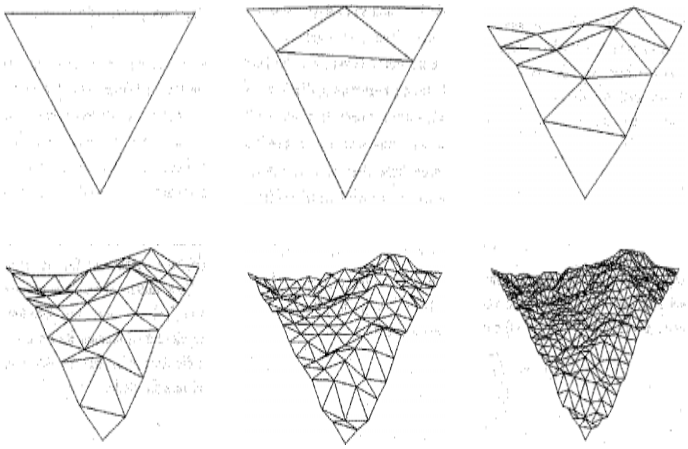
Anwendung fraktaler Strukturen

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

24.3.1 Modellierung von Oberflächen

Fraktale Oberflächen — beispielsweise Gebirgszüge — lassen sich gut modellieren, indem iterative Generatoren mit zufälligen Elementen kombiniert werden. Beispielsweise kann anhand einer Höhenkarte grob der Höhenverlauf eines Gebirgszugs festlegt werden. Anschließend werden die Flächen durch Triangulation weiter unterteilt, wobei die für Eckpunkte der neuen Dreiecke zufällig der Ort und die Höhe festgelegt wird. Indem lokal die Höhe auf einen bestimmten Wert fixiert wird, können auch Seen erzeugt werden.

Modellierung von Gebirgen



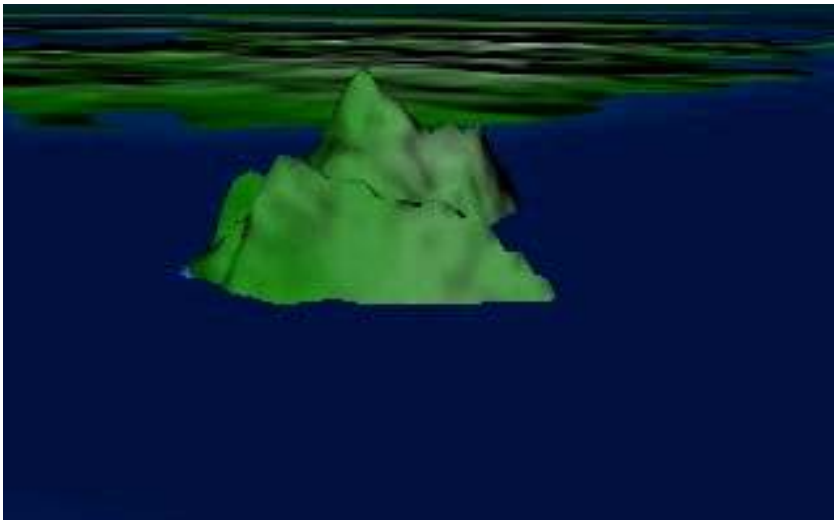
Erzeugung einer fraktalen Fläche durch sukzessive Teilung der Dreiecksflächen.

Erzeugung einer fraktalen Fläche durch sukzessive Teilung der Dreiecksflächen.

Fraktale Gebirgslandschaft

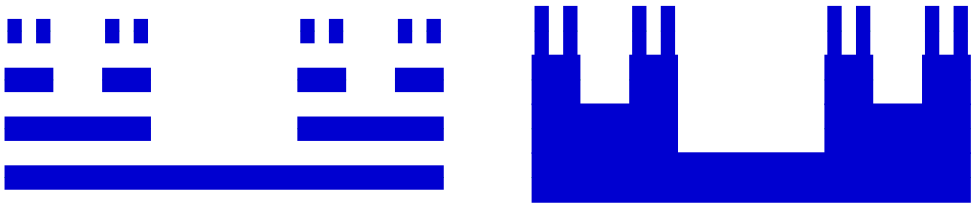
Fraktale Gebirgslandschaft, erzeugt durch sukzessive Dreiecksunterteilung, 8 Unterteilungs-Iterationen

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.

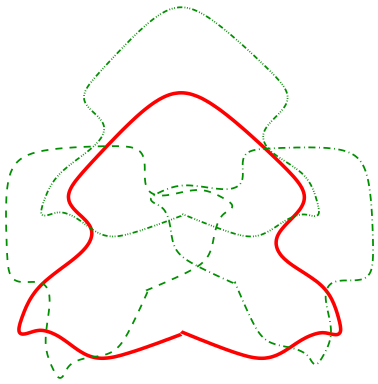
24.3.2 Modellierung vonPflanzen

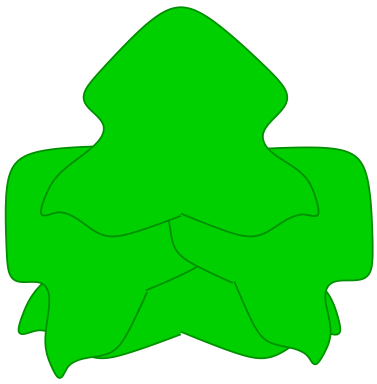
Komposition von höheren Strukturen:Kollagen

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

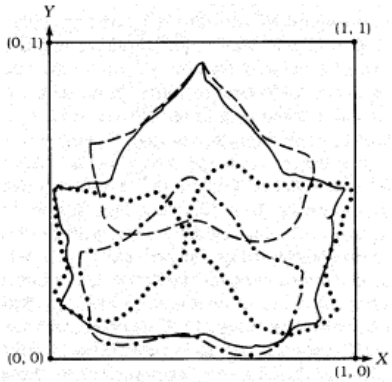
Ein fraktales Ahornblatt

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



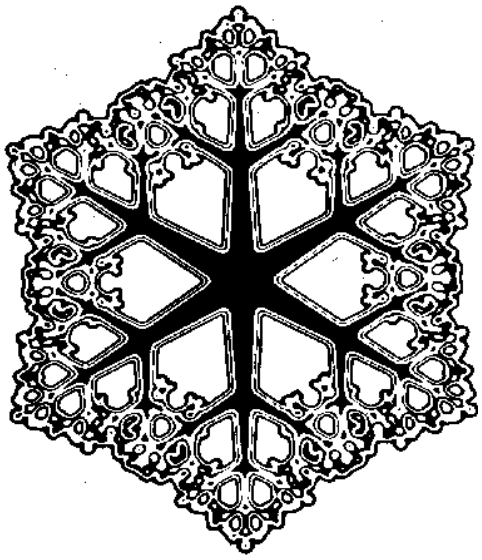


Beispiel: Kollagetechnik



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.

✎ **Anmerkung:** Das folgende Bild des Farns wurde mit nur vier Transformationen erzeugt. Ausgangspunkt sei ein Punkt der xy-Ebene. Der Punkt wird iterativ transformiert, indem zufällig (nach einer Wahrscheinlichkeitstabelle) eine der vier Transformationen ausgewählt wird. Die errechneten Punktkoordinaten werden durch einen schwarzen Punkt im Bild markiert. Auf die Koordinaten wird iterativ die nächste Transformation angewendet, solange bis das Bild genügend genau ist. Die vier Transformationen, zusammen mit den Wahrscheinlichkeiten, reichen also aus, um ein sehr detailreiches Bild zu repräsentieren.

Farben und Helligkeiten können eingeführt werden, indem gezählt wird, wie oft ein bestimmter Punkt besucht wird. (Peterson, 1988, S. 130)(Reiser and Wirth, 1994, S. 70)

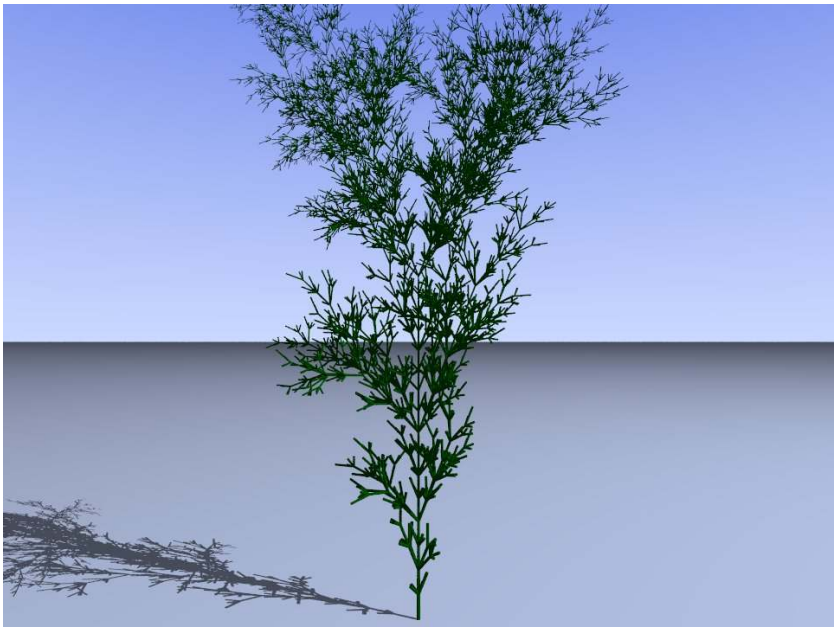
$$\begin{aligned} x_{n+1} &= r \cdot x_n \cdot \cos(A) - s \cdot y_n \cdot \sin(B) + h \\ y_{n+1} &= r \cdot x_n \cdot \sin(A) + s \cdot y_n \cdot \cos(B) + k \end{aligned}$$

Koeffizienten für den Farn							
	Translationen		Rotationen		Skalierungen		
#	h	k	A	B	r	s	P
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

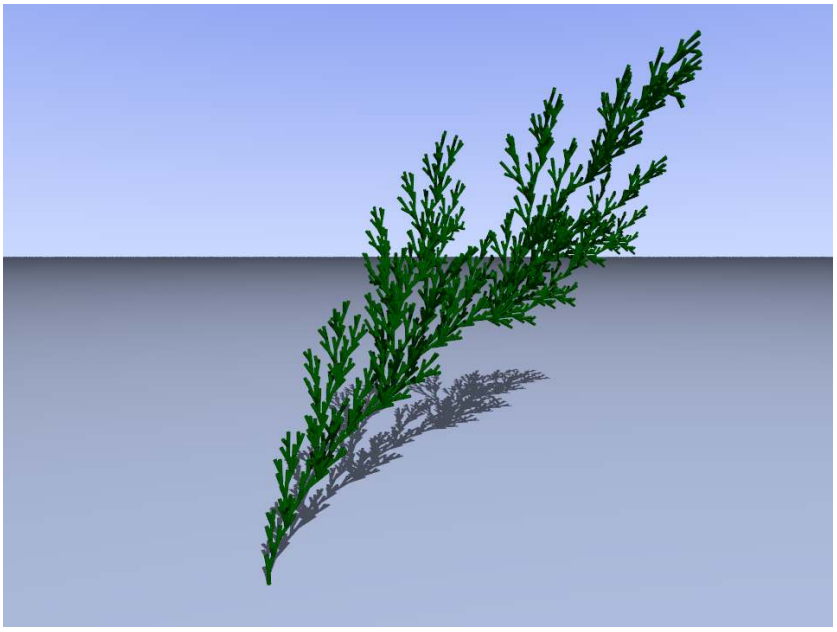
Ein fraktaler Farnwedel und eine Farnpflanze

Um die Folge der Transformationen zu beschreiben haben sich Erzeugendensysteme bewährt – beispielsweise Lindenmayer-Systeme (L-Systeme). Hier sind einige schöne Beispiele, die mir Max Brauer (Brauer, 2008) zur Verfügung gestellt hat.

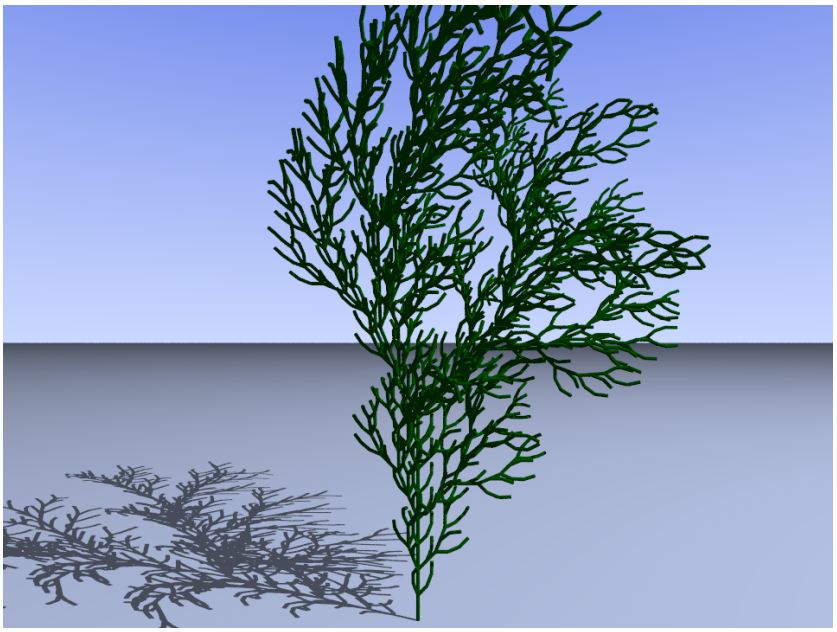
Erzeugung mit Lindenmayer-Systemen



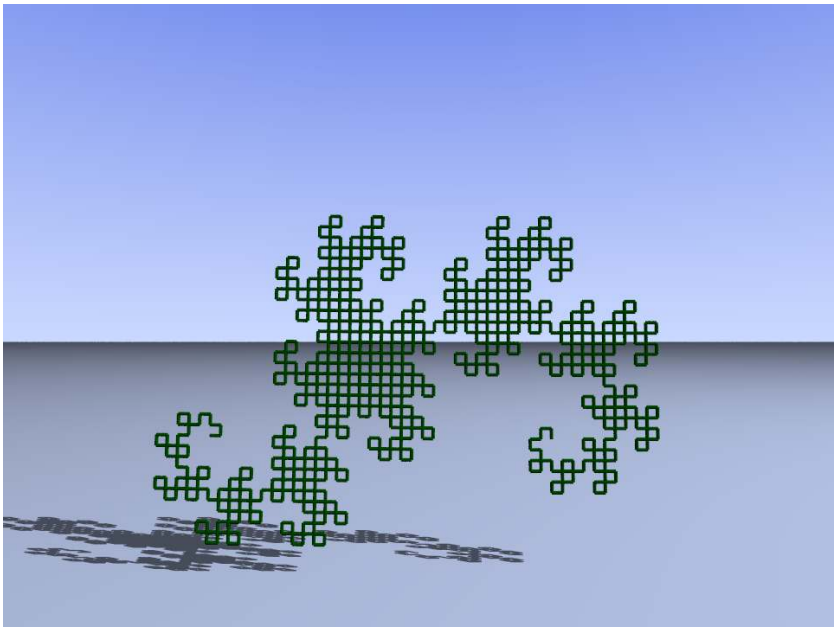
Erzeugung mit Lindenmayer-Systemen



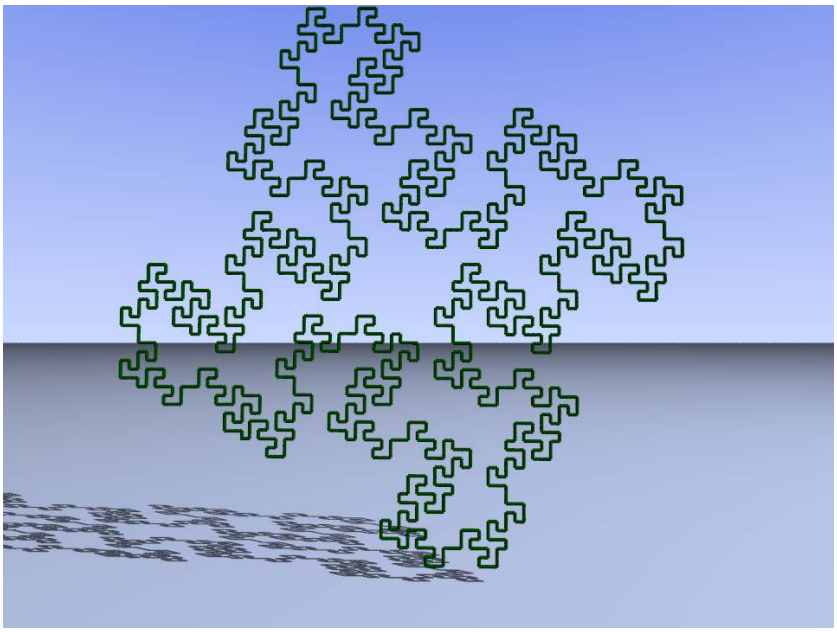
Erzeugung mit Lindenmayer-Systemen



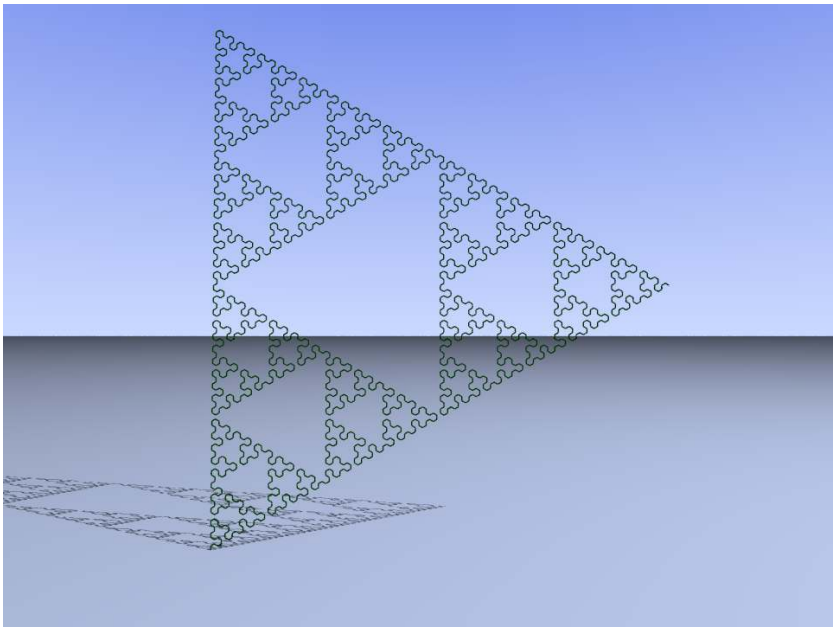
Erzeugung mit Lindenmayer-Systemen



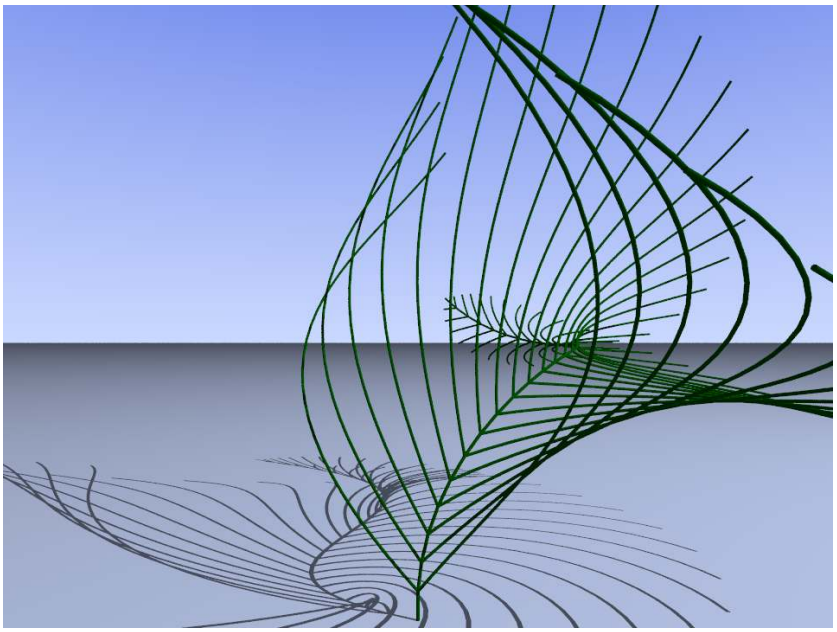
Erzeugung mit Lindenmayer-Systemen



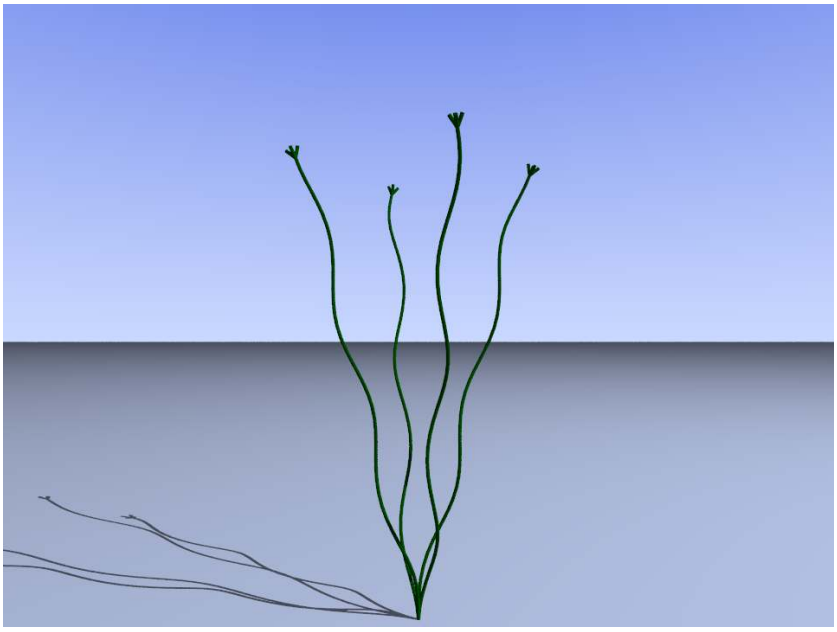
Erzeugung mit Lindenmayer-Systemen



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.

25 Chaotische Folgen

25.1 Topologische Dynamik

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

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

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.

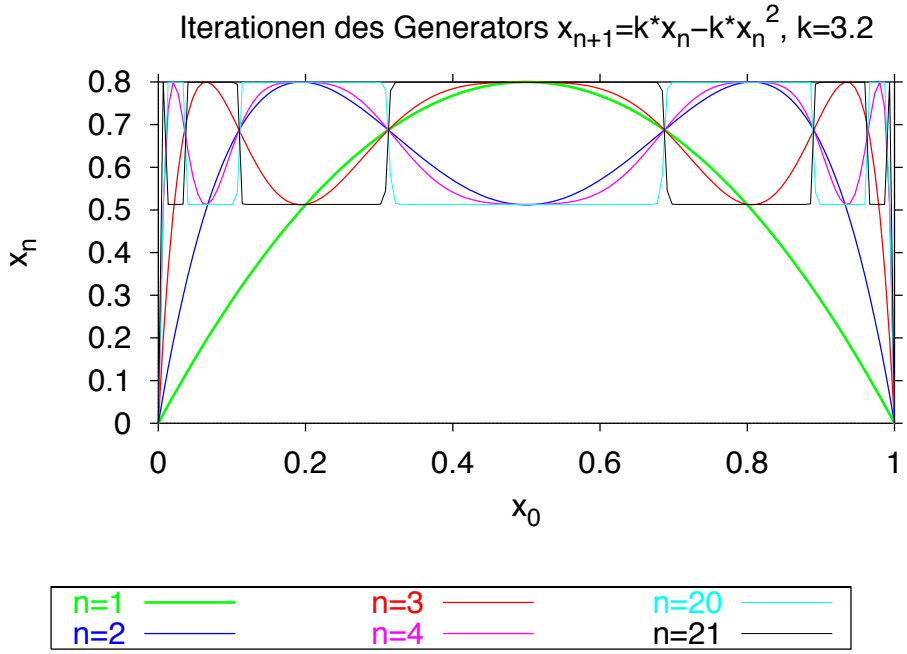
- Für *kleine* Werte von x dominiert der *lineare* Term:

  $k > 1.0$: 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.

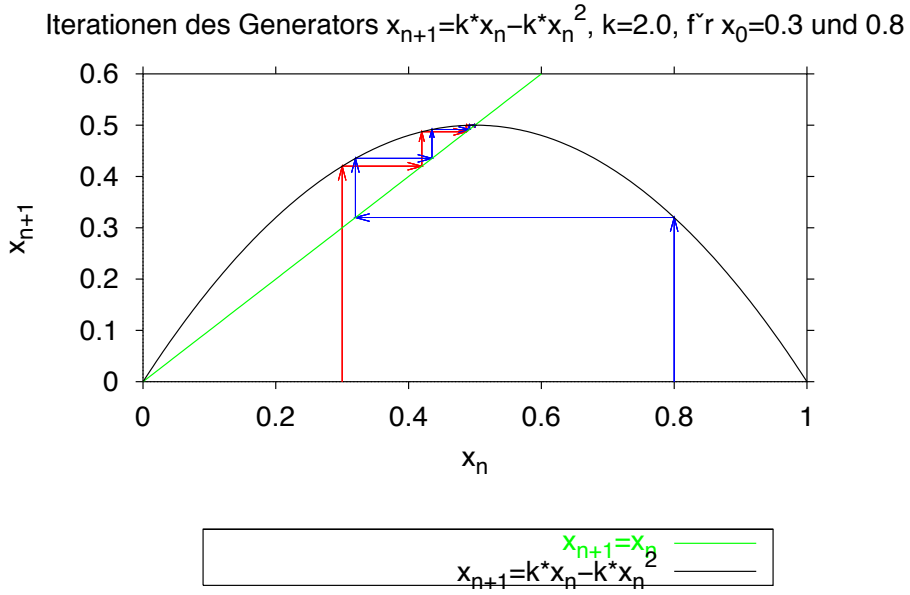
Iterationen des Generators



✎ **Anmerkung:** Die Folge der Werte kann man grafisch leicht visualisieren, indem man die Kurve des Generators (eine Parabel) und die Gerade $y = x$ zeichnet. Wir wählen einen Startwert x_0 auf der x-Achse und gehen nach oben (oder unten), bis wir die Parabel treffen. Der y-Wert dieses Punktes ist x_1. Für die Iteration gehen wir horizontal nach rechts oder links, bis wir die Gerade treffen. An diesem Punkt können wir wieder auf der Parabel den nächsten Iterationswert x_2 ablesen usw.

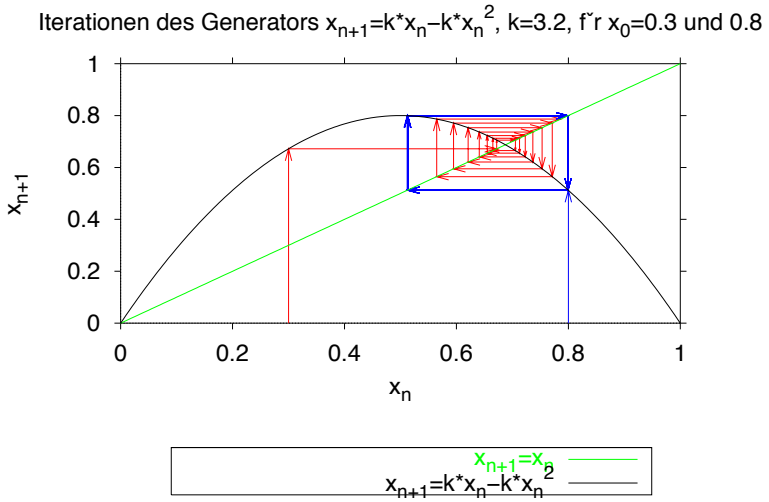
Für Werte von $k < 3.0$ konvergiert die Folge unabhängig vom Startwert x_0 gegen einen Fixpunkt, nämlich dem Schnittpunkt der Geraden mit der Parabel. Dieser Schnittpunkt hängt nur von k ab.

Iterationen des Generators für $k = 2$



Die folgende Folie zeigt die Iteration für einen anderen Wert von k, nämlich $k = 3.2$. Die Folge konvergiert nicht mehr, sondern oszilliert zwischen zwei Attraktoren (bei $x = 0.5$ und $x = 0.8$) hin und her. Für größere Werte von k erhöht sich die Zahl der Attraktoren. Dieses kann in einem Bifurkationsdiagramm dargestellt werden, indem die Fixpunkte gegen den Parameter k aufgetragen werden.

Iterationen des Generators für $k = 3.2$



Der Einfluß von k:

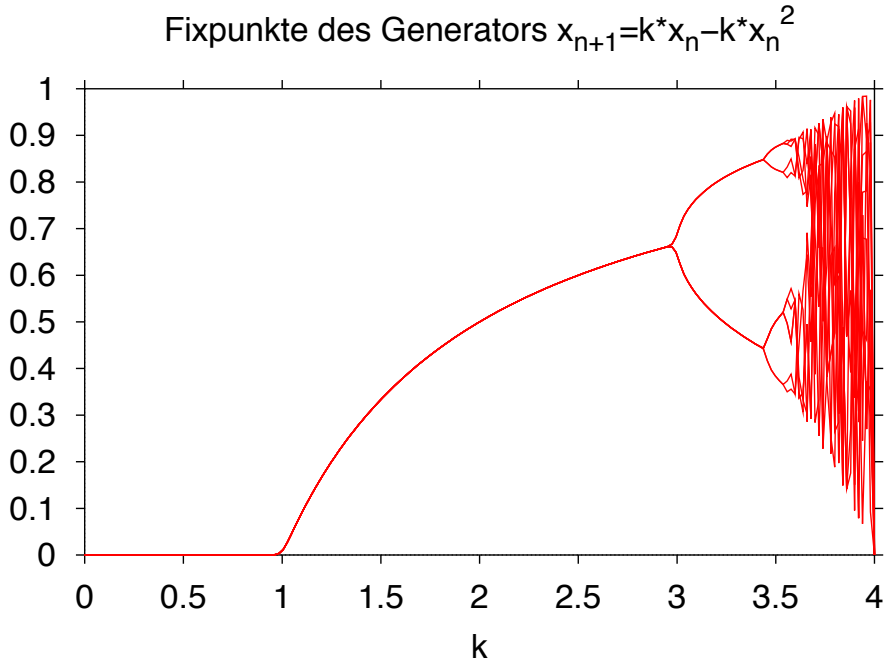
$0 <= k <= 3$: Die Folge $x_0, x_1, \ldots$ konvergiert gegen einen *Fixpunkt*, der nur von k, aber nicht von x_0 abhängt. Dieser Fixpunkt wird *Attraktor* genannt.

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

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

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

Bifurkationsdiagramm



✎ **Anmerkung:** Attraktoren können komplizierte Formen haben und Flächen formen, auf denen der Wert zum Fixpunkt gelenkt wird. Sie sind unproblematisch, solange nur ein Fixpunkt existiert. Systeme von nichtlinearen Diffenrentialgleichungen können aber unter bestimmtem Bedingungen Attraktoren haben, bei denen geringe Änderungen der Startwerte dazu führen, daß sehr unterschiedliche Endwerte erreicht werden. Solche Attraktoren heißen *chaotisch* oder *seltsam* (strange).

Im folgenden werden wir nicht-lineare Funktionen als Generatoren benutzen, um Mengen von Punkten zu definieren. Diese Punktmengen sind fraktale Linien, Flächen oder Volumina und dienen zur Modellierung von Texturen und Objektformen..

✎ **Anmerkung:** Literaturhinweise Zum Begriff der Hausdorff-Dimension finden Sie eine kurze Einführung bei Fellner, 2004, ebenfalls Definitionen der Mandelbrot-Menge und der Julia-Mengen. Eine gut verständliche populärwissenschaftliche Einführung gibt Peterson, 1988. In (Peitgen et al., 1992) finden Sie eine didaktisch sehr gut aufgebaute und ausführliche Einführung, die für den Schulunterricht gedacht ist. Die schönsten fraktalen Bilder und eine fundierte mathematische Darstellung finden Sie bei Peitgen and Richter, 1986, ebenso die Algorithmen, um Mandelbrot-Mengen und Julia-Mengen selbst zu berechnen.

25.2 Julia-Mengen

Folgen komplexer Zahlen

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

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

Sei $c = 0$. Die Folge $z_0, f(z_0), f(f(z_0)) \cdots$ ist dann

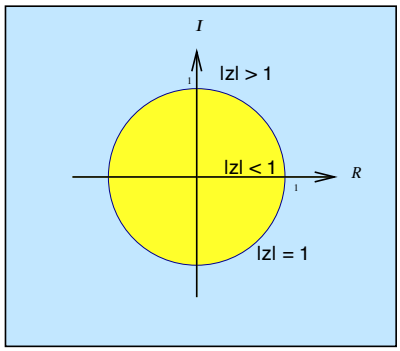
divergent für $|z_0| > 1$,

invariant gegen den Generator für $|z_0| = 1$,

konvergent für $|z_0| < 1$.

Folgen komplexer Zahlen: Beispiel für $c = 0$

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



Julia-Menge

Definition: 59 (*Julia-Menge*)
Julia-Menge: Die Menge aller $z \in \mathbb{C}$, die bezüglich des Generators $f(z) = z^2 + c \qquad c, z \in \mathbb{C}$ invariant ist.

Gaston Julia, am Anfang dieses Jahrhunderts.

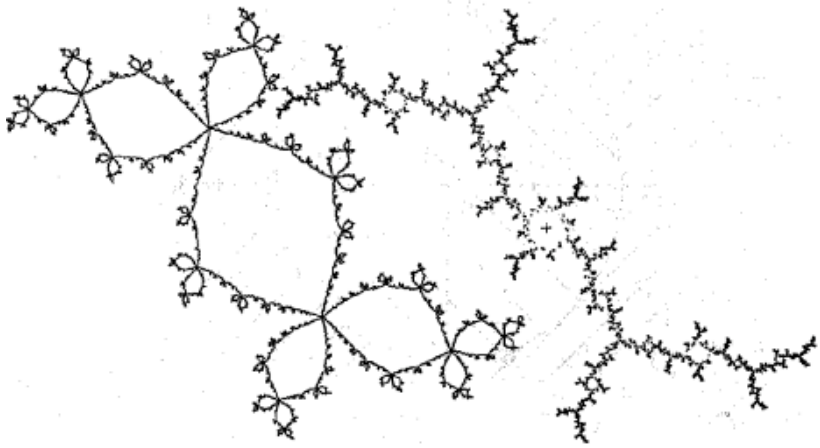
Julia-Kurven, auch Drachenkurven genannt, sind *zusammenhängende* Julia-Mengen.

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

Anmerkung: Im Zusammenhang mit Julia-Mengen wird Cantor-Staub auch Fatou-Staub genannt.

Die beiden folgenden Abbildungen stammen aus Fellner, 1992.

Julia-Kurven



Julia-Kurven für $f(z) = z^2 + c \qquad c, z \in \mathbb{C}$ mit
(a) $c = -0.12 + 0.74i$ und (b) $c = -0.15652 + 1.03225i$

Julia-Kurven

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

25.3 Die Mandelbrot-Menge

Definition nach Fellner, 1992:

Mandelbrot-Menge

Definition: 60 (***Mandelbrot-Menge***)
Die Mandelbrot-Menge für die Generatorfunktion $f(z) = z^2 + c$ ist die Menge aller Punkte $c \in C$ für die gilt:

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

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

Satz: 61
Julia-Kurven mit der Generatorfunktion $f(z) = z^2 + c$ sind genau dann zusammenhängend, wenn c in der Mandelbrot-Menge liegt.

Ausschnitte aus der Mandelbrot-Menge: Die folgenden Folien zeigen Ausschnitte aus der Mandelbrot-Menge für den Prozeß:

$$z \to z^2 + c$$

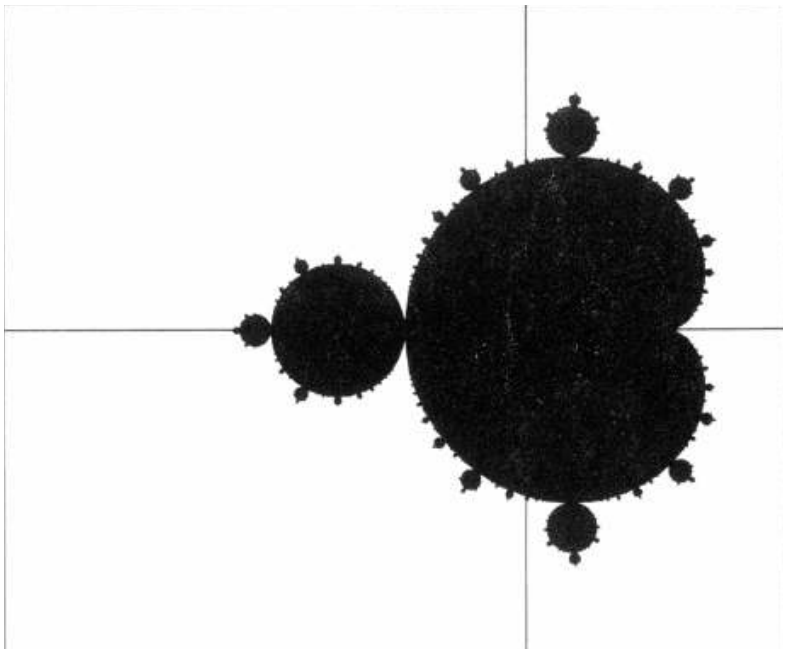
Die Ausschnitte sind:

- $-2.15 \leq c_{re} 0.75, -1.4 \leq c_{im} \leq 1.4$, 255 Iterationen,
- $-0.95 \leq c_{re} 0 - .99333, -0.23333 \leq c_{im} \leq 0.3$, 266 Iterationen,

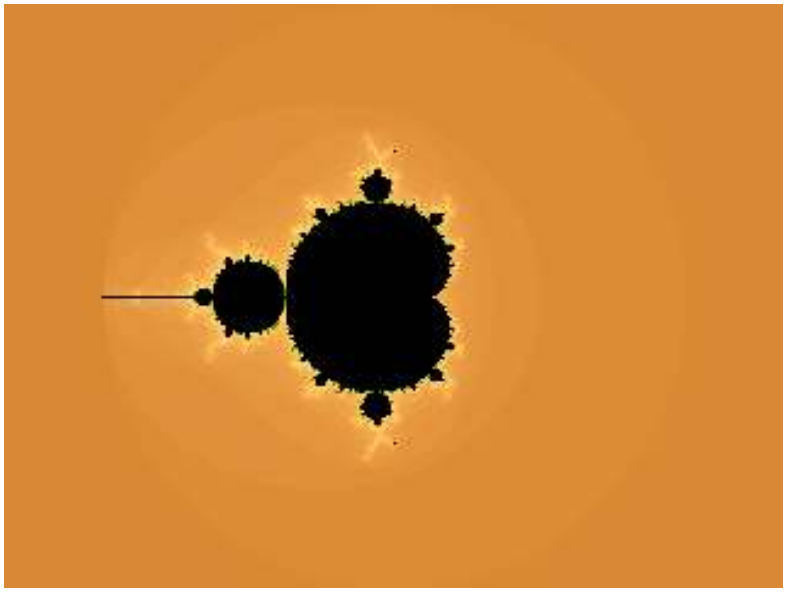
- $-0.1992 \leq c_{re} - 0.12954, 1.0148 \leq c_{im} \leq 1.06707,$ 264 Iterationen,
- $-0.74591 \leq c_{re} - 0.74448, 0.11196 \leq c_{im} \leq 0.11339,$ 296 Iterationen,

Alle Abbildungen stammen aus (Fellner, 1992).

Die Mandelbrot-Menge: Das Apfelmännchen

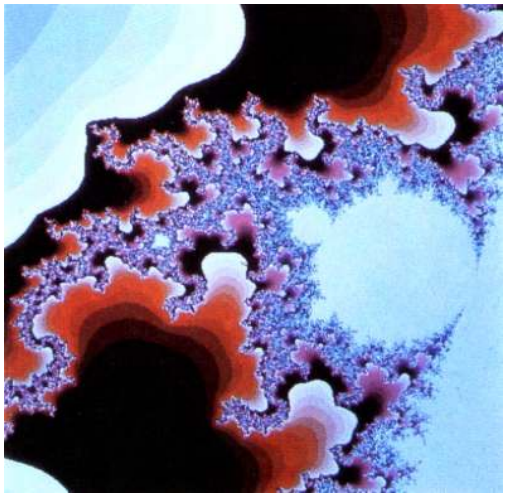


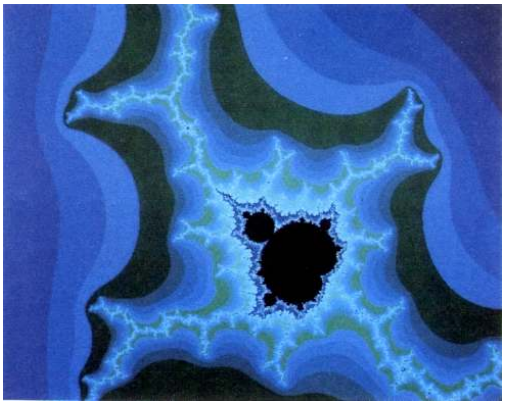
Die Mandelbrot-Menge: Das Apfelmännchen



Eingefärbt und gerechnet mit „Mandelbrot on Cocoa“

Die Mandelbrotmenge: Ausschnitte der Kontur 1





Die Mandelbrotmenge: Nachbarschaft

Genau wie die Drachenkurven der Julia-Mengen sind auch die Konturen der Mandelbrot-Menge selbstähnliche Strukturen. Wenn wir die Darstellung der Menge vergrößern, tauchen rekursiv immer wieder modifizierte verkleinerte Kopien des „Apfelmännchens“ auf.

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: 62 (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.

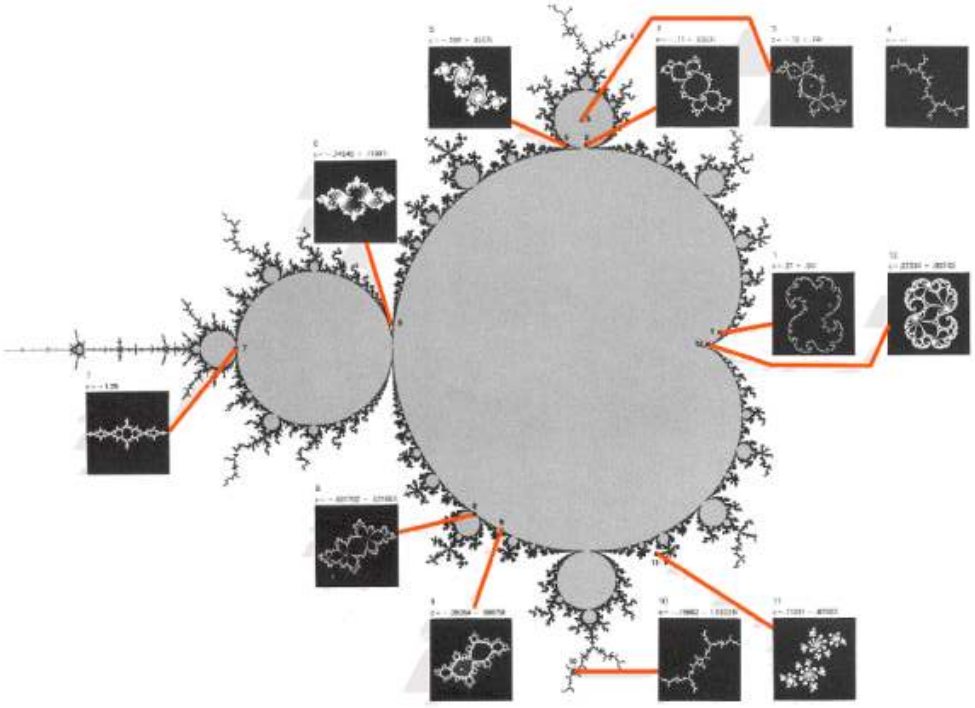
Zwei Probleme:

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 interessanten Julia-Mengen: Eine interessante Eigenschaft der Mandelbrot-Menge ist, daß die fraktalen Strukturen in der Nähe eines Punktes denjenigen Julia-Mengen ähnlich sehen, die sie erzeugen. Wenn wir die Koeffizienten für eine bestimmte Form suchen, hilft ein Blick auf die Mandelbrot-Menge. Die folgende Folie zeigt eine Übersicht: Für einige Pukte der Mandelbrot-Menge sind die entsprechenden Julia-Mengen angegeben.

Ein Wegweiser zu Juliamengen



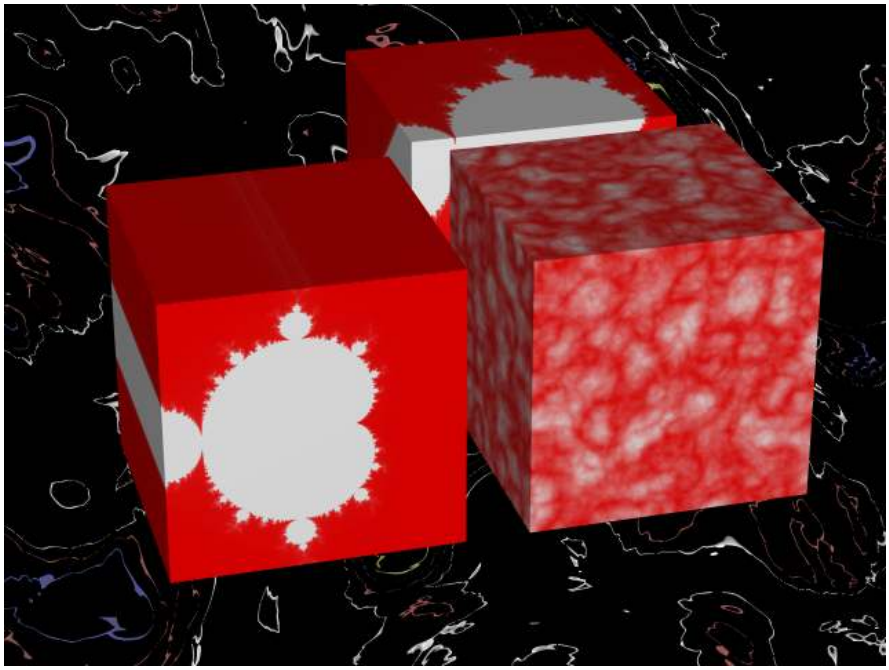
26 Fraktale Objekte und Texturen

Fraktale Texturen in POV-Ray

Beispiel: 63 (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 63



26.1 Julia-Mengen als fraktale Oberflächen

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:

✎ **Anmerkung:** Weder die Quaternionen noch die hyperkomplexen Zahlen sind perfekte Erweiterungen der komplexen Zahlen.

- Die Quaternionengruppe ist nicht kommutativ bezüglich der Multiplikation — die Quaternionen bilden also einen *Schiefkörper*.
- Die hyperkomplexen Zahlen sind zwar kommutativ, es existiert aber nicht für alle Zahlen ein *inverses Element* bezüglich der Multiplikation.

Beide Systeme enthalten die komplexen Zahlen als echte Untermenge.

Die Bezeichnung für die vier Basisvektoren des Vektorraums der Quaternionen wird in der Literatur nicht einheitlich gehandhabt. Fellner, 1992 bezeichnet die drei imaginären Komponenten mit i, j, k; in einigen Mathematikbüchern werden sie mit j, k, l bezeichnet.

Quaternionen

Definition: 64 (***Quaternionen***)
Ein Quaternion q *ist eine Linearkombination aus einem reellwertigen Basisvektor* 1 *und drei imaginären Basisvektoren* i, j, k.

$$\begin{aligned}
q &= q_0 + q_1 i + q_2 j + q_3 k \quad \textit{mit} \\
i^2 &= j^2 = k^2 = i \cdot j \cdot k = -1 \quad \textit{und} \\
i \cdot j &= k = -j \cdot i \quad \textit{und} \\
j \cdot k &= i = -k \cdot j \quad \textit{und} \\
i \cdot k &= -j, \quad k \cdot i = j
\end{aligned}$$

Hyperkomplexe Zahlen

Definition: 65 (*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:

$$\begin{aligned} h &= h_0 + h_1 i + h_2 j +_3 k \quad mit \\ i^2 &= j^2 = k^2 = -1, i \cdot j \cdot k = 1 \quad und \\ i \cdot j &= k = j \cdot i \quad und \\ j \cdot k &= -i = k \cdot j \quad und \\ i \cdot k &= -j = k \cdot i \end{aligned}$$

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.

✎ **Anmerkung:** Julia-Fraktale in POV-Ray

In POV-Ray können wir Julia-Fraktale sowohl mit Quaternionen als auch mit hyperkomplexen Zahlen berechnen. Es sind 18 verschiedene Transformationsfunktionen möglich. Zur Berechnung der Fraktale sind die folgenden Angaben nötig:

Julia-Parameter: Die additive Konstante in der Iterationsformel

Die Funktion: Die Transformationsfunktion, z.B. sqr, sin, exp ...

Zahl der Iteration: Wie oft soll maximal iteraiert werden?

Genauigkeit: Die Genauigkeit, mit der geprüft werden soll, ob ein Punkt innerhalb oder außerhalb des Objektes liegt.

Hyperebene: Die Hyperebene, mit der das Objekt zum Schnitt gebracht wird. Sie wird durch einen Normalenvektor und den Abstand vom Ursprung angegeben.

POV-Ray berechnet die Julia-Fraktale, indem iterativ für jeden Punkt geprüft wird, ob er innerhalb oder außerhalb des Objektes liegt.

Ein Punkt h_0 wird als innerhalb des Objekts betrachtet, wenn während der Iteration keiner der h_i weiter als ein vorgegebener Abstand vom Ursprung entfernt ist. Bei einer geringen Anzahl von Iterationen werden also mehr Punkte als zum Objekt gehörig betrachtet als bei einer hohen Iterationszahl.

Das folgende POV-Ray-Skript erzeugt ein fraktales Objekt mit der Generatorfunktion **sqr**, dem Startpunkt $-0.083, 0.0, -.83, -0.025$, mit maximal 20 Iterationen, einer Geanuigkeit von 20 und der Schnittebene mit dem Normalenvektor $(0, 0, 0, 1)$. Der Animationsmechanismus von POV-Ray wird genutzt, um eine Folge von Schnitten zu berechnen. Die Schnittebene wird abhängig von der simulierten Zeit parallel zur Normale verschoben. Nur so zum Spaß wird auch der Farbton des Objektes in Abhängigkeit von der Uhrzeit verändert. Beginnend bei Rot rotiert der Farbton einmal durch den ganzen Farbenkreis.

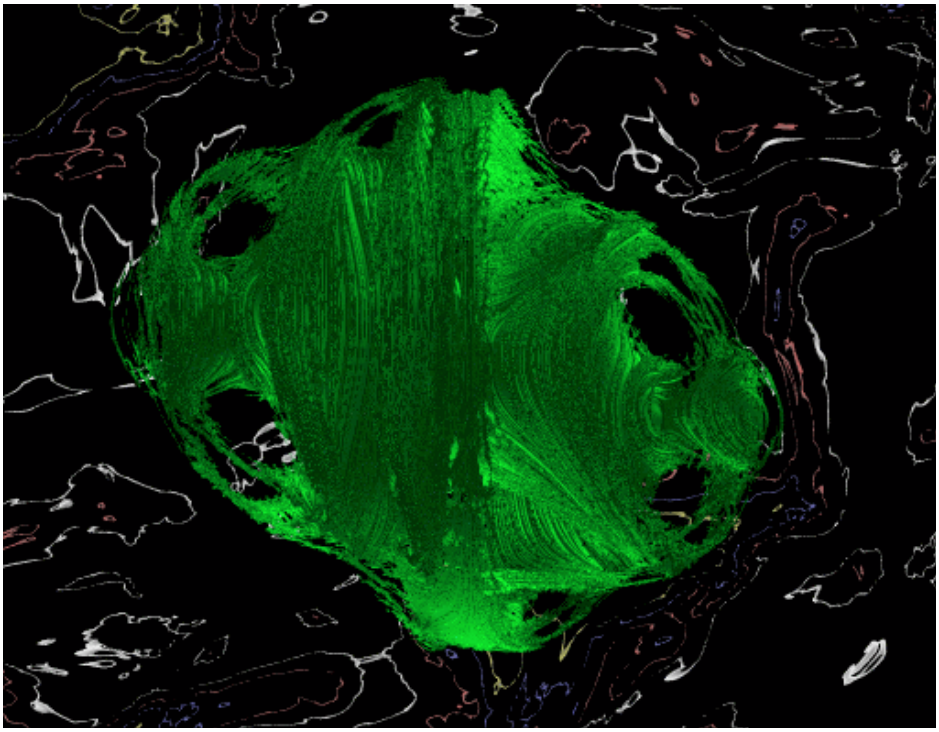
Alle erzeugten Volumen sind sehr lockere, löcherige Objekte, die etwas an Wollknäule oder lockeres Gebäck erinnen. Von der Lage der Hyperebene hängt es ab, ob die erzeugte Julia-Menge zusammenhängend ist oder nicht.

Ein Julia-Fraktal mit Quaternionen

Beispiel: 66 (Schnitte durch ein Quaternion:)

```
#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. 66)

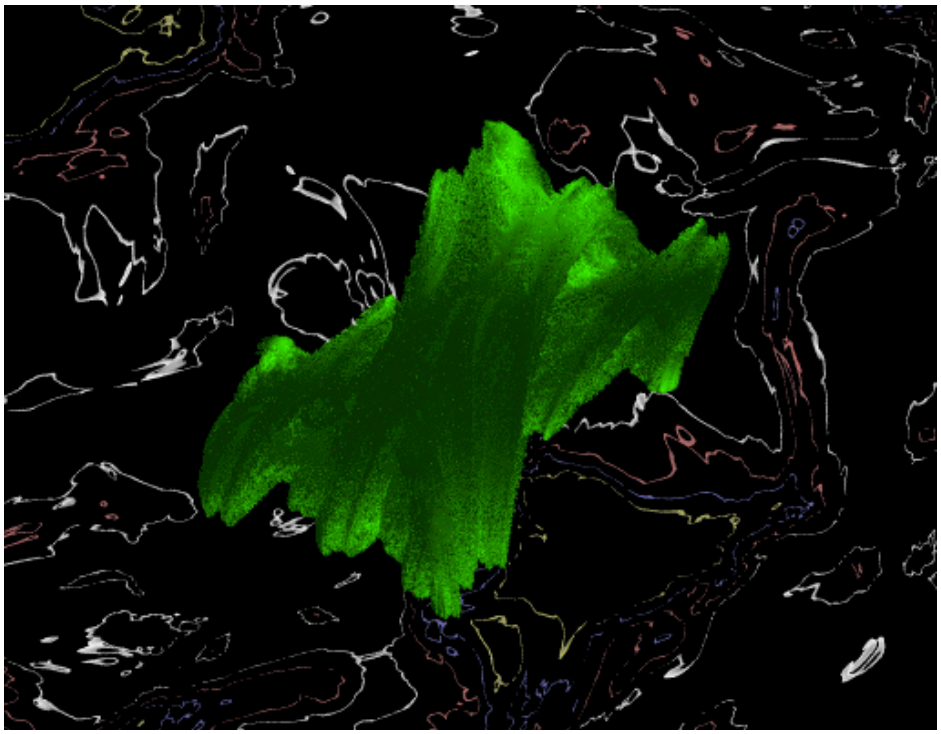


Ein Julia-Fraktal mit Quaternionen

Beispiel: 67 (Schnitte durch ein Quaternion:)
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. 67)



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/movjulia-fracSmall

IVC Interactive Visual Computing: Computer Vision

Leonie Dreschler-Fischer

Prüfungsunterlagen Band 2:

Computer Vision

Universität Hamburg
Fachbereich Informatik

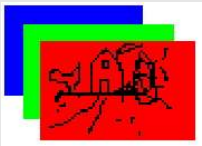
WS 2014/2015

Teil X

Bilder und Bildfunktionen

27 Analoge Bildfunktionen

Bildfunktionen



27 Analoge Bildfunktionen

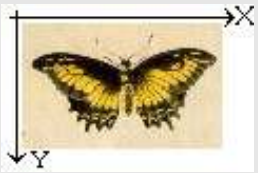
28 Digitale Bildfunktionen

29 Repräsentation von Bildern

Analoge Bildfunktionen

Definition (Bild:)

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



Bildfunktion

- Für Grautonbilder:

  $$g(x, y) : \Re \times \Re \Rightarrow \Re$$

- für Farbbilder:

  $$g(x, y) : \Re \times \Re \Rightarrow \Re \times \Re \times \Re$$

Es gibt zwei mögliche photometrische Interpretationen der Bildfunktion::

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

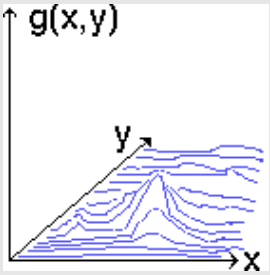
Häufig wie 3 Einzelfunktionen betrachtet...

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

Wenn nicht anders angegeben, wird von Intensitätsbildern ausgegangen.

RGB-Auszüge eines Farbbildes

Grauwertgebirge



Anschauliche Vorstellung:
Grauwertgebirge —

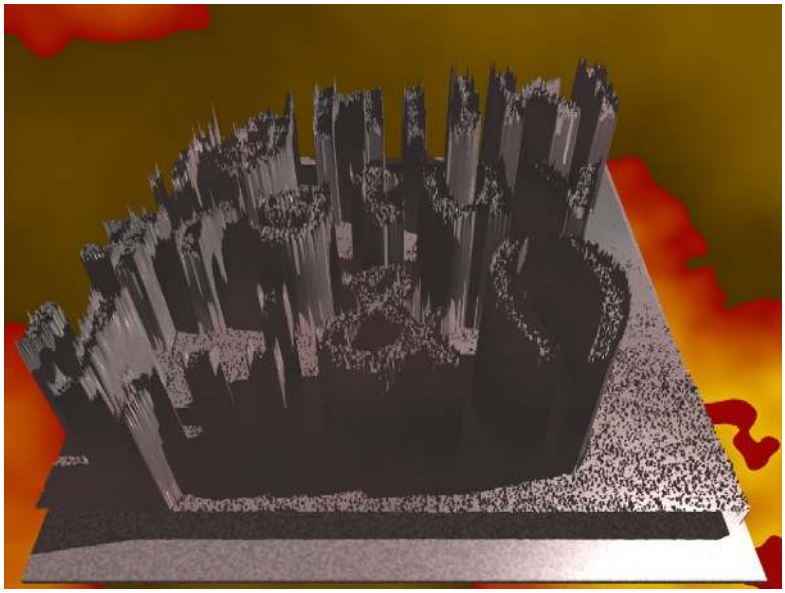
Annahme:

- $g(x, y)$ ist beschränkt

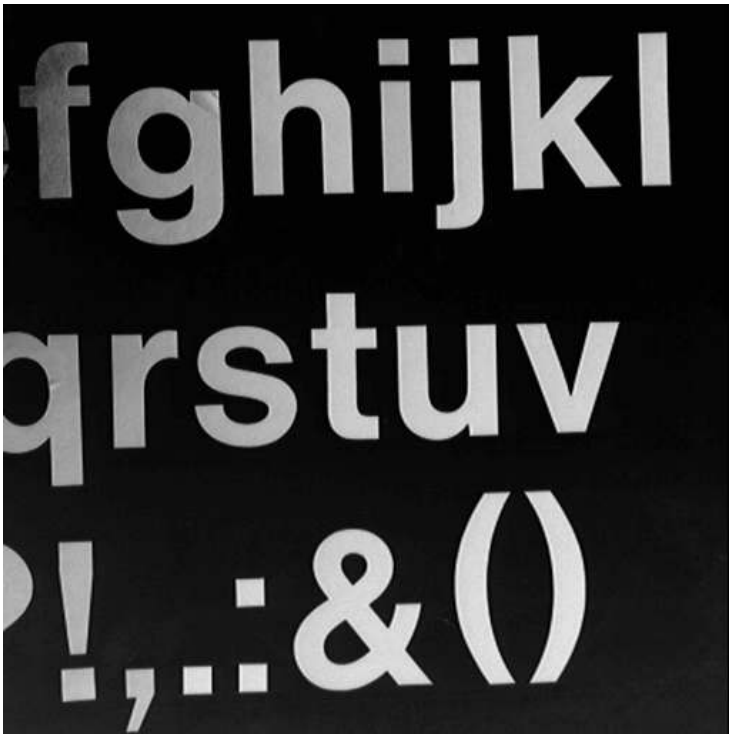
$$0 \leq g(x, y) \leq g_{\max}$$

- (und überhaupt mathematisch problemlos).

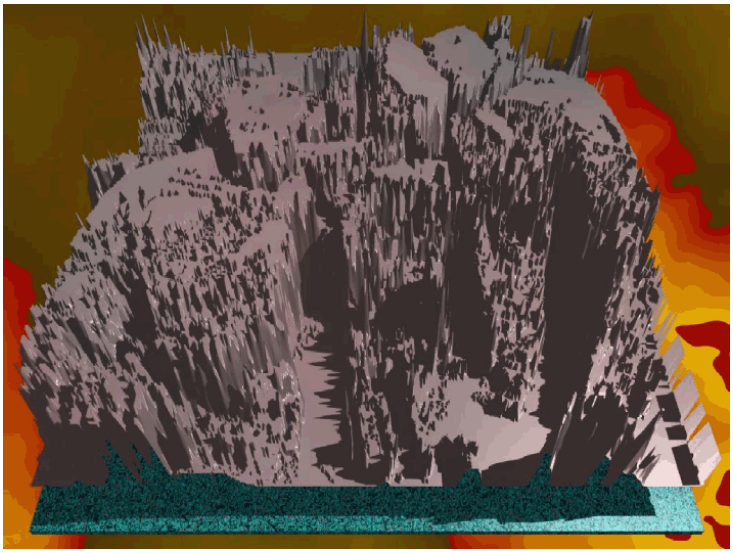
Das Grauwertgebirge für ein einfaches Bild



Das Bild zum Grauwertgebirge



Die Bildfunktion eines komplexen Grautonbildes



Das Bild zum Grauwertgebirge

Multispektralbilder

Definition: 68 (*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

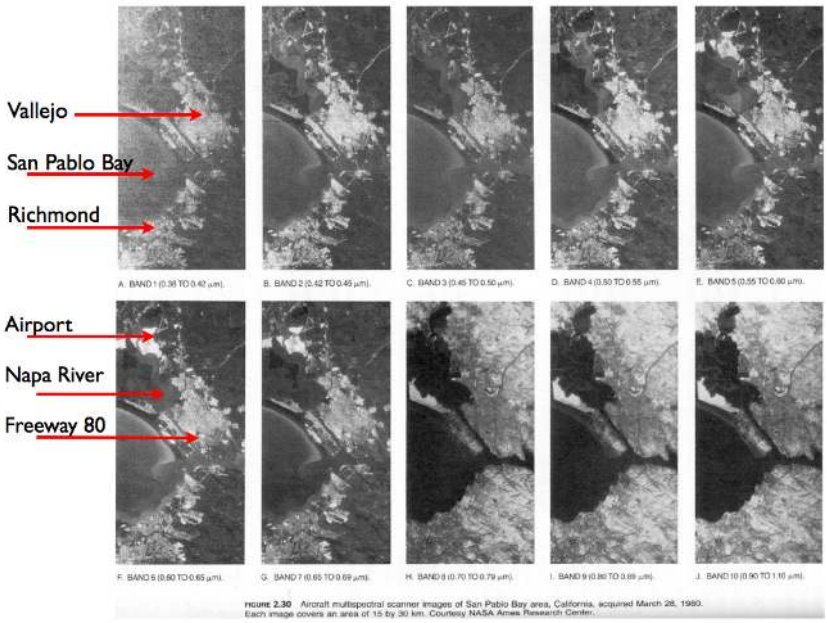
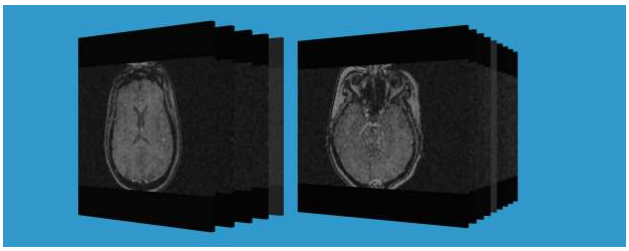


FIGURE 2.30 Aircraft multispectral scanner images of San Pablo Bay area, California, acquired March 28, 1980. Each image covers an area of 15 by 30 km. Courtesy NASA Ames Research Center.

Volumenbilder



Ein MRT

Definition: 69 (*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).*

Der Ortsraum

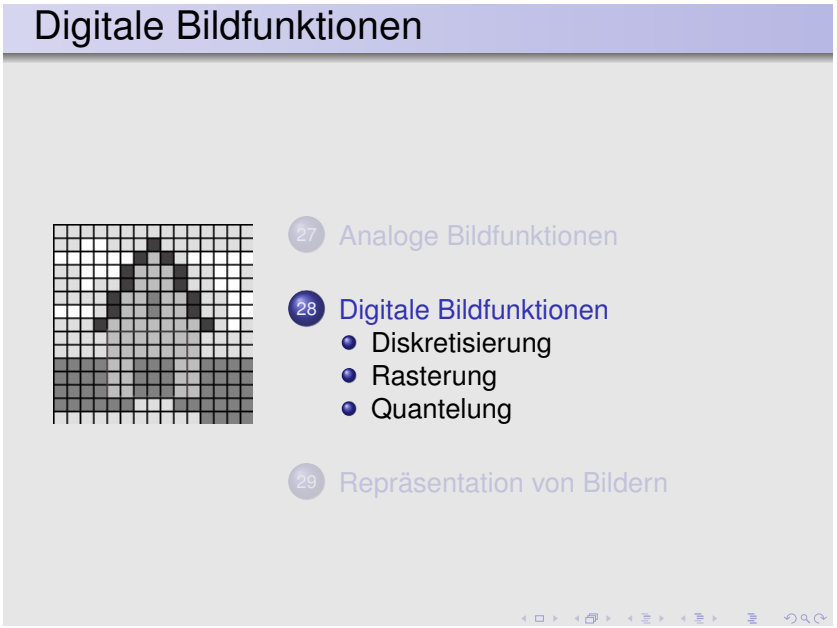
Definition: 70 (*Ortsraum*)
Die Bildebene $X \times Y$ eine zweidimensionalen Bildes oder das Volumen $X \times Y \times Z$ eines Volumenbildes heißen der Ortsraum.

- 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 *Bildfunktion: Frequenzraum*: Die Bildfunktionen werden durch ihr Fourier-Spektrum beschrieben.

zum Frequenzraum siehe Kapitel.:30: Repräsentation im Frequenzraum, Seite: 379

28 Digitale Bildfunktionen

28.1 Diskretisierung



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.

28.2 Rasterung

Beispiel: Rasterung eines Bildes

Grobes Raster und quantisierte Farbwerte

Feines Raster und quantisierte Farbwerte

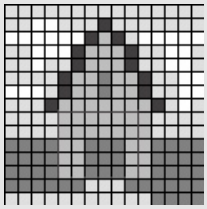
http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movPixellate.html

Bei der Diskretisierung repräsentieren wir ein Bild durch eine endliche Zahl von diskreten Abtaststellen, die wiederum nur eine endliche Zahl von Farben oder Helligkeiten haben. Das entspricht der Wiedergabe als ein Mosaik — auch hier wird das Bild aus einem endlichen Vorrat von Steinchen zusammengesetzt. Die folgende Folie zeigt ein Mosaik aus einer Kirche in Ravenna (Der wundersame Fischzug, Sant' Apollinare Nuovo, 6. Jahrhundert nach Chr).

Ein Mosaik aus Ravenna

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

Matrix-Rasterung

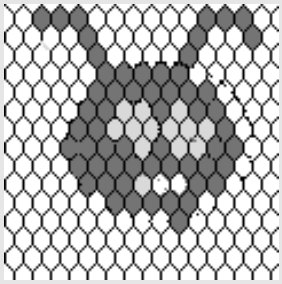


$$g_{ij}: \quad i = 0 \ldots I-1$$
$$j = 0 \ldots J-1$$
$$I \quad \textit{Zeilen}$$
$$J \quad \textit{Spalten}$$

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

Definition: 71 (***Pixel:***)
Die Elemente der Bildmatrix heißen Pixel: (picture elements).

Hexagonal-Rasterung



Hexagonal-Rasterung:

Die Pixel sind (gleichseitige) Sechsecke.

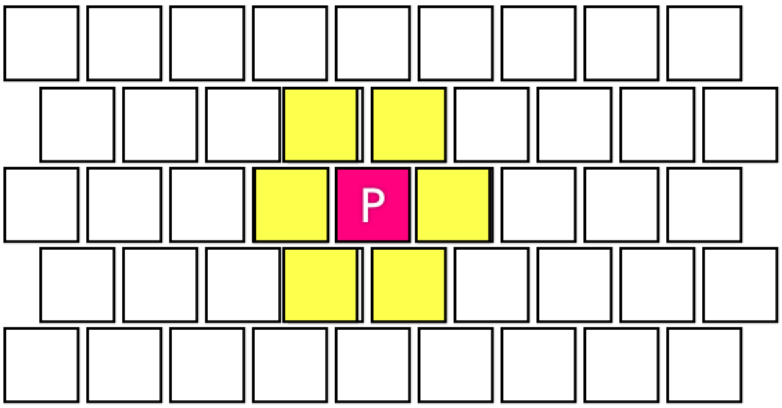
Vorteile:

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

Ein Hexagonalraster läßt sich aus einem Rechteckraster erzeugen, indem jede zweite Zeile der Bildmatrix um einen halben Pixeldurchmesser verschoben wird: So hat jedes Pixel 6 Nachbarn, die ungefähr gleich weit entfernt sind (siehe (Rosenfeld and Kak, 1982)).

Bisher haben wir noch nicht darüber gesprochen, welche Metrik zum Messen der Pixelabstände verwendet wird, und sind stillschweigend vom Euklidschen Abstand ausgegangen. Die Frage der Metrik ist in gerasterten Bildern ein Kapitel für sich .

Erzeugung des Hexagonalrasters aus dem Matrixraster

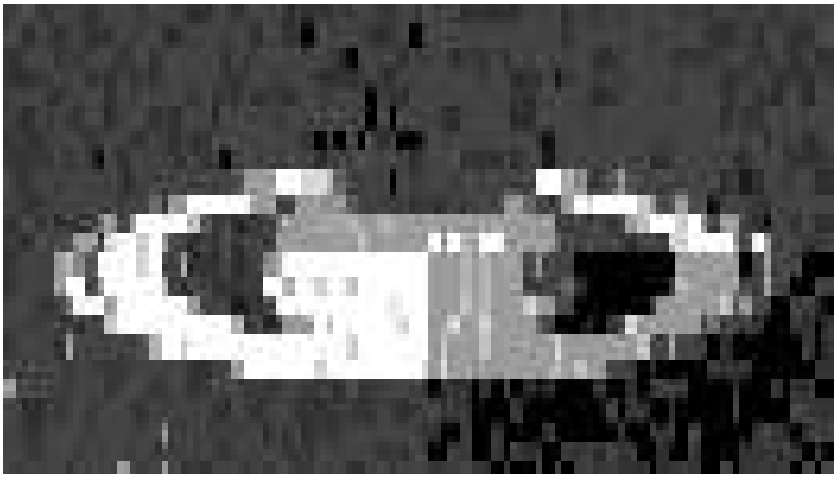


Die Pixel eines Grauwertbildes in Matrix-Rasterung

Jedes Druckzeichen steht für einen bestimmten Grauwert.

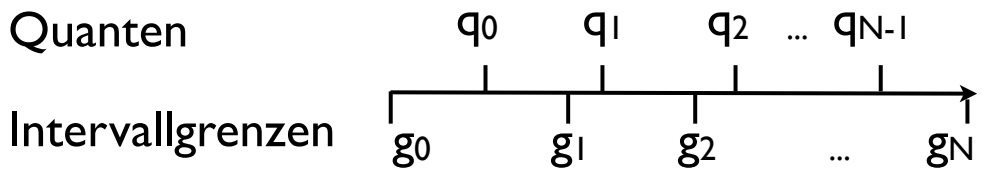
msvjirsndsssssksqsjffssesssoihghpmosrihspssssgmfdlfsisssgsjksisssrmsdoorjsljsnqps
sscsjhnsejkslssddsmssnssfssem . sssssseegs . sssssssssssssshsssssssisssssdfdsessjghsu
jdsssislsssscssisssssssgsdsssssssf . ssssfsdssemkjscsssssssddsssqskgijisssjcsscsfg
gcgkrsrshsesdlgsssssjgsssscsssgdgdfschsssssssskscfds#lsslsgssssssssekjssghhesj
nsssdjosssgsnsssshsemhsjhscssshsss . sffsssssssssshdsssscdsssssdsssihsfkcsscss
olcskshjssdsssdscsssessssisssssss . s . ssssfcsssscdssssgssssssssssfsjhsssss
iselsssessogjrihsgisssssssc . s . s . scs . . sssspsssshbsssssssssosssssssdsgscgs
melhkscsf . shsssskgses . sssssssesrsxssscsssvhgfhrgssassssssdessssdfsssdfdn
sclssossssshessssssswHHHVVVTUNAsoss . slsssnsssssVVVKFqlyHvsssssdsaesshssde
ssgkssssnssCcHHVVVVVHHssHHHEHyspsesiikdsnsssHHAsssHVSVVVVRHsxHwsasssigsssss
gssksenjsDHsHVVVVVPxjscgksEHHHHNHLHHLMHHHIHHHHHJHHHHHlqsssjsHQVVVSVHsHHsc#ssisfe
kffuousHHnTVVSVHtsfsfuisspHHHHHHHHRHHHHHHVHVMHVVHHHHHzsssssscssnRSVVVVHsVHssshgs
gssssHPExVVVVUVssHehcsedsGKVVVVVVVVVVVVVVHHHMHLHJHsHsss . . . . . HHHHHH. sHssr . . .
hssspHVVsHVVVVVOnisskfislVVLVVRVVQVVVVVVVHHHLHOHHHsssszd . . . . . sMHHHHHssHss . . sp
sssmgqCVVHVVVVVVHsssafhsVVVVVVVVVVSVVVHHHQHOHHHSHHst . . . . . gHHHHHHQsHss . . pog .
rsqospsksSRVVVVVVVVVVLfschHHHVQVVVVVUVVHHTHHSHHHHBs . . . c . hsUHHHHHHsCssli . . . je . g
jssssgHsssspvNVVSVVVVVVVVVVVVVVSVVVVVVHQHHHPHHHHHHTHHHHHHHPssHssh . c . . . . . i . c
sqsspsgsnmjhsssrrshessHVVVVVVSVVVVVVVVUHTHHHHHHHHHHQHPHHsro . f . . . . . idid . . . . . gg
Hosssqjsssisusflssgnsssesgisskokguhjnrssh . lih . h . q . d . c . . . . j . . c . lnlm . . . . j . . . h . ck . .
sqedskssmssslgslsjssgfsscssscciemkmjgsmelssikj . . . . j . . . . . s . gd . . e . . . . . n . . . . . smgh
ssdmsstqpssrnssHospsdsnssdslscofmsqsssssdd . . m . . i . . . . i . . . . . e . . hdk . s . . . . . jsie . kns
ssllsssnsossnsprpsqrnhsspsshsjssqsssiqfsq . . nl . . k . ned . . kilcsgm . . f . . f . . . . ggngmi . n .
sjsssshssssHspsHsjlgssscsjtsilsosfdrskrssjioslf . f . . s . p . nsipief . edmh . . s . immkr . s . s

Das Bild in Grauwertdarstellung



28.3 Quantelung

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.

Seien g_0 und g_{max} die Extremwerte der Bildfunktion, zwischen denen die Funktionswerte auf eine endliche Zahl von Repräsentanten q_n abgebildet werden sollen.

Hierzu wird das Intervall $[g_0, g_{max}]$ in eine endliche Zahl N von Teilintervallen zerlegt, die durch die Entscheidungsgrenzen g_n beschrieben werden. Die g_n seien aufsteigend geordnet, d.h. $g_n < g_{n+1}$. Ein Grauwert g, der im Intervall $[g_n, g_{n+1}]$ liegt, wird beim Quantisieren auf den diskreten Quantenwert q_n abgebildet.

Ein *Quantisierer Q* ist damit durch die beiden Mengen $\{g_n\}$ und $\{q_n\}$ vollständig beschrieben. Wie wählt man diese beiden Mengen optimal, wenn die Zahl der zur Verfügung stehenden Quantisierungsstufen N vorgegeben ist? (siehe: Rosenfeld und Kak 76, pp. 100–103)

Zur Notation: Rosenfeld und Kak notieren den Quantisierer durch ein kleines „q“, aber das kann leicht mit den q_i verwechselt werden. Deshalb wird der Quantisierer mit einem großen Buchstaben Q angegeben.

Wir nehmen an, die Wahrscheinlichkeitsdichte $p(g)$ und Zahl der Intervalle N seien bekannt. Weiterhin sei $p(g) = 0$ für alle Grauwerte außerhalb des Intervalls$[g_0, g_N]$.

Dann ist es ein Optimierungsproblem, denjenigen Quantisierer Q zu finden, für den der Quantisierungsfehler, das heißt, ein Maß für die Abweichungen der gewählten Repräsentanten q_n von den Grauwerten, am kleinsten ist.

Als Maß für die Abweichungen wird hier die Summe der quadrierten Differenzen gewählt.

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}.

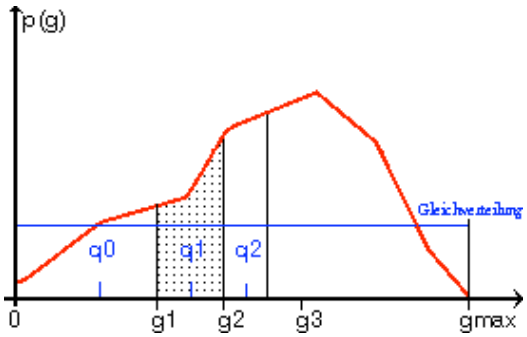
$$\delta_Q^2 = \sum_{n=0}^{N-1} \int_{g_n}^{g_{n+1}} (g - q_n)^2 p(g) dg$$

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

Partielle Ableitungen nach g_n und q_n, Nullstellen:

$$\begin{aligned}
\frac{\partial}{\partial g_n}\delta_Q^2 &= (g_n - q_{n-1})^2 p(g_n) - (g_n - q_n)^2 p(g_n) = 0 \\
& \qquad n = 1 \ldots N-1 \\
\frac{\partial}{\partial q_n}\delta_Q^2 &= -2\int_{g_n}^{g_{n+1}} (g - q_n) p(g) dg = 0 \qquad n = 0 \ldots N-1
\end{aligned}$$

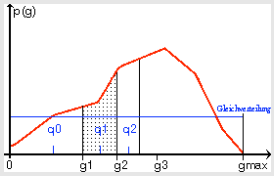
Man beachte, daß nicht nach g_0 und g_N differenziert wird, da die Grenzen des Wertebereichs von g als Randwerte festliegen. q_0 und q_{N-1} dagegen sind nicht a priori festgelegt und müssen aus den angrenzenden Intervallgrenzen ermittelt werden.



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

$$\Rightarrow q_n = \frac{\int_{g_n}^{g_{n+1}} g\, p(g) dg}{\int_{g_n}^{g_{n+1}} p(g) dg} \qquad n = 0 \ldots N-1 \tag{2}$$

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 $p(g) = const$: Gleichmäßige Quantelung.

✎ **Anmerkung:** Die Zeichnung ist nur eine Prizipskizze, die die Bedingungen nicht genau erfüllt!

Für kompliziertere Dichten gibt es meist keine analytischen Lösungen; nur numerische Verfahren.

Numerische Lösungsprozedur

1. Wähle 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.

✎ **Anmerkung:** Numerische Lösungsprozedur:

1. Wähle q_0, $n = 0$.
   In den Lehrbüchern wird keine Heuristik vorgeschlagen, aber es bietet sich an, q_n in Schritten von 1 zu erhöhen. Eine andere Möglichkeit wäre, aus dem Fehler von q_{N-1} abzuleiten, in welche Richtung der Startwert verschoben werden müßte und eine Intervallschachtelung zu machen.

Damit sind g_0 und q_0 festgelegt. Jetzt läßt sich auch g_1 bestimmen (durch numerische integration), denn die zweite Intervallgrenze muß ja so liegen, daß q_0 Gleichung 2 erfüllt.

2. Berechne g_{n+1} mit (2) und setze $n := n + 1$.
   Jetzt sind q_{n-1} und g_n gegeben, und wir können jetzt Gleichung 1 benutzen, um q_n zu finden:
3. Berechne q_n mit (1): $q_n = 2g_n - q_{n-1}$.
4. Falls $q_n > g_N$ weiter bei 7, denn bei diesem Startwert ist keine Lösung mehr möglich.
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 $\in$ 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

Für den Weißabgleich werden die Farben in den einzelnen Kanälen so korrigiert, daß Bereiche, die Weiß erscheinen sollen, keine Farbsättigung mehr haben.

29 Repräsentation von Bildern

29.1 Speicherbedarf der Bildmatrix

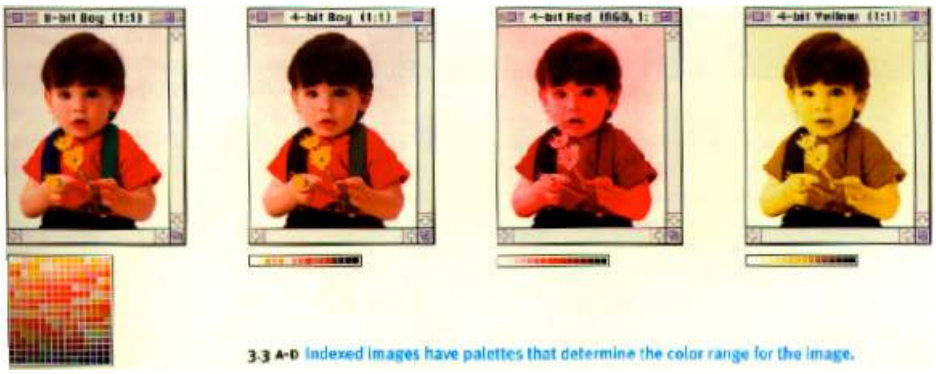
Speicherbedarf der Bildmatrix

- 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 proporional 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 256 Graust.	4 096	262 144	1 048 576	67 108 864
RGB-Farbbild 256^3 Farben Farben	12 288	786 432	3 145 728	201 326 592
Multispektral, 10 Kanäle je 256 Graust.	40 960	2 621 440	10 485 760	671 088 640

Quantelung von Farben: Farbpaletten



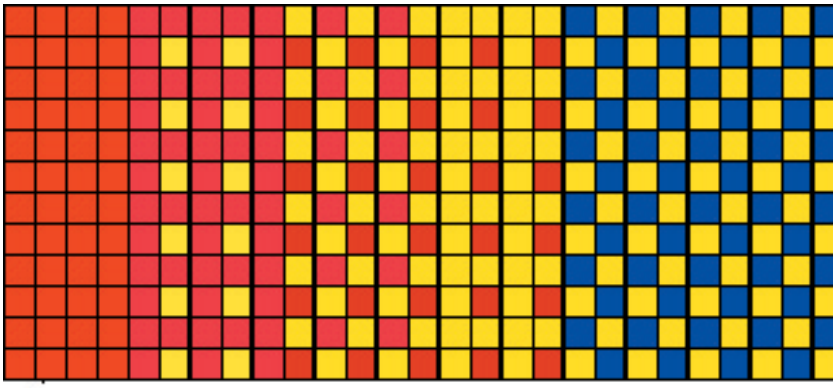
3-3 A-D Indexed images have palettes that determine the color range for the image.

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.



29.1.1 Externe Bildformate

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.

✎ **Anmerkung:** Das XPM-Format für icons

Ein Beispiel:

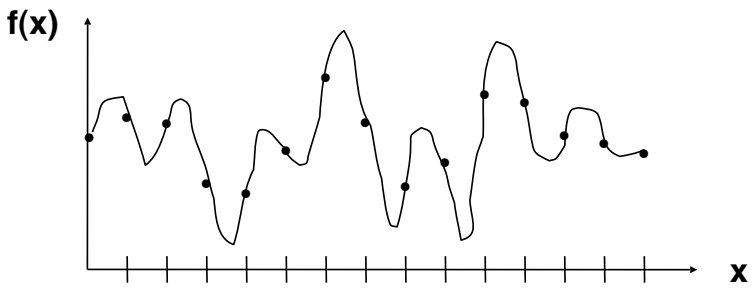
```
/* XPM */
static char * icl8_11432_xpm[] = {
/* width height ncolors chars_per_pixel */
"24 31 9 1",
/* colors */
"       s None  c None",
"o      c #999999",
"O      c black",
"+      c #333333",
"@      c #111111",
"#      c #FF6666",
"$      c #FEFE00",
"%      c #DDDDDD",
"&      c #BBBBBB",
/* pixels */
"           oOoOo                ",
"          O     Oo              ",
"         o        O             ",
"         O        o             ",
"       +++++       O            ",
"       +@@@+       o            ",
"      +@@O@@+       O           ",
"    ++@OOO@O@++     o  #       $",
"   +@@OOOOOOO@@+    O      $  # ",
"  +@OOOOOOOOOOO@+    o  $       ",
"  +@OOOOOO%&oOO@+    O   #  $   ",
" +@OOOOOOOO%&oOO@+    o         ",
" +@OOOOOOOOOOOOO@+     OoO # $ #",
" +@OOOOOOOOO%&oO@+              ",
" +@OOOOOOOOO%&oO@+      $   $   ",
" +@OOOOOOOOO%&oO@+        #  #  ",
" +@OOOOOOOO%&oOO@+     #        ",
" +@OOOOOOOO%&oOO@+    $   $    $",
"  +@OOOOOO%&oOO@+               ",
"  +@OOOOOOOOOOO@+         #     ",
"   +@@OOOOOOOO@@+               ",
"    ++@@@@@@@++                 ",
"      +++++++                   "};
```

Teil XI

Repräsentation im Frequenzraum

30 Auflösung eines Bildes

Auflösung eines Bildes Die Feinheit des Rasters bestimmt, wie gut Details aufgelöst werden. Die folgenden Folien zeigen ein Bild in unterschiedlichen Auflösungen: zuerst sehr grob (1 Pixel entspricht 128×128 Pixeln des Originalbildes), und dann immer feiner aufgelöst (64×64, 16×16, 8×8). Das Testbild ist ein Titelbild der Zeitschrift IEEE-Computer aus dem Jahr 1987, in der ein sehr lesenswerter Rückblick von Brooks auf 20 Jahre Softwarentwicklung erschienen war.

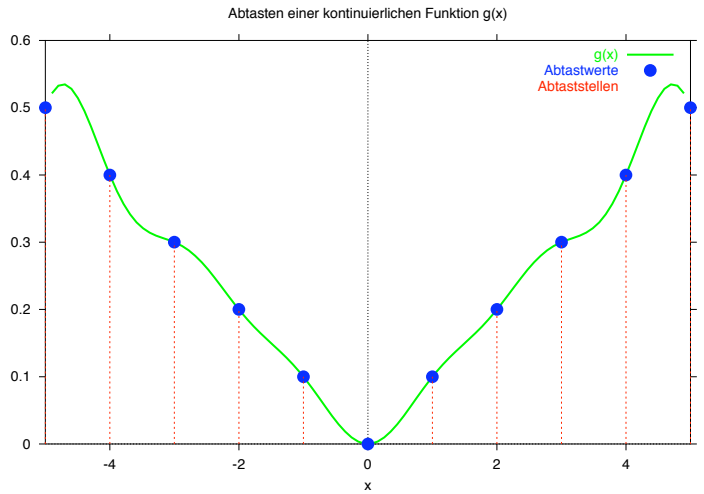


Gegeben sei eine Bildfunktion $g(x, y)$; wieviele Abtaststellen sind für eine optimale Abtastung nötig?

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 *Abtastheorem* (sampling theorem).
- Es basiert auf einer Zerlegung von $g(x, y)$ in Sinus- und Cosinus-Bestandteile.

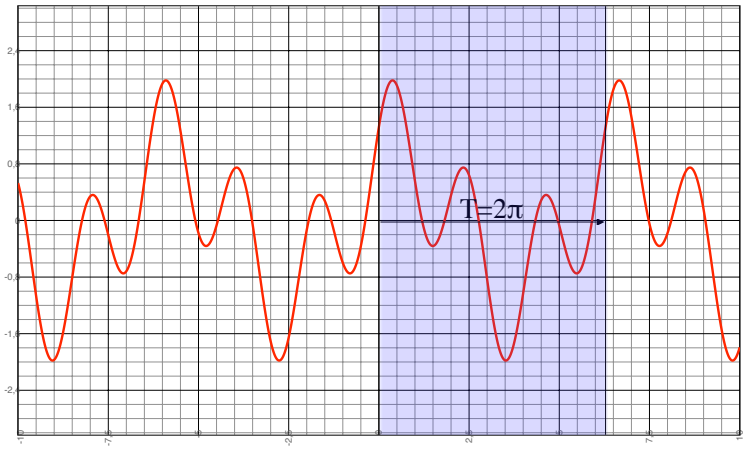
Herleitung: Zunächst *eindimensional*.

30.1 Fourier-Reihenentwicklung

Fourier-Reihenentwicklung

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

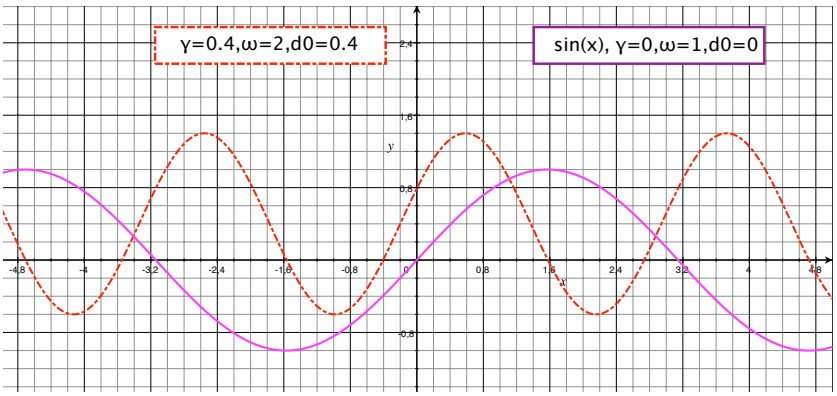
☞ d.h. $g(t) = g(t + T)$.



✎ **Anmerkung:** Die dargestellte Funktion hat die Gleichung

$$y = \sin(1 + \frac{2\pi}{T}x) + \sin(0.5 + \frac{2\pi 3}{T}x))$$

Die Parameter der Sinusfunktion



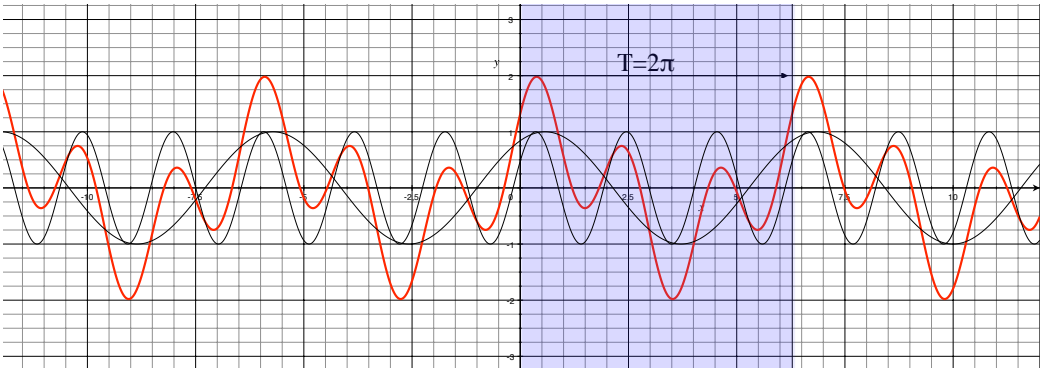
$$f(t) = d\sin(\gamma + \omega t) + d_0$$

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: $\omega_k = \frac{2\pi k}{T}$.

$$\begin{aligned} g(t) &= \sum_{k=1}^{\infty} d_k \sin\left(\gamma_k + \frac{2\pi k}{T} t\right) + d_0 \\ \text{mit} & \quad \text{Winkelsummensatz} \\ & \quad \sin(\alpha+\beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta \\ g(t) &= \frac{a_0}{2} + \sum_{k=1}^{\infty}\left[a_k \cos\left(\frac{2\pi k}{T} t\right) + b_k \sin\left(\frac{2\pi k}{T} t\right)\right] \end{aligned}$$

✎ **Anmerkung:** Zur Notation: In einer Reihe von Bildverarbeitungsbüchern werden Frequenzen immer implizit als Kreisfrequenzen $\frac{2\pi}{T}$ angegeben und dann, wie in der Physik üblich, mit ω bezeichnet, siehe beispielsweise Rosenfeld und Kak. Diese Notation ist sehr praktisch, denn sie erspart einem das Schreiben von vielen, vielen πs. Wir werden uns hier an die Notation von Duda und Hart halten und Frequenzen $\frac{1}{T}$ angeben, wenn nicht explizit anders gesagt. Meist ist es für die Beweise unerheblich, ob Kreisfrequenzen oder Frequenzen gemeint sind, solange man in der Verwendung konsistent bleibt.

Fourierreihe einer kontinuierlichen Funktion

$$
\begin{aligned}
g(t) &= \frac{a_0}{2} + \sum_{k=1}^{\infty}\left[a_k \cos\left(\frac{2\pi k}{T}t\right) + b_k \sin\left(\frac{2\pi k}{T}t\right)\right] \\
e^{ix} &= \cos x + i \sin x, \;\; i = \sqrt{-1} \quad Euler - Formel \\
\cos x &= \frac{1}{2}(e^{ix} + e^{-ix}) \\
\sin x &= \frac{1}{2i}(e^{ix} - e^{-ix}) \Longrightarrow \\
g(t) &= \frac{a_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}\left[(a_k - ib_k)e^{(i2\pi kt/T)}\right. \\
&\qquad \left. + (a_k + ib_k)e^{(-i2\pi kt/T)}\right]
\end{aligned}
$$

$$
\begin{aligned}
g(t) &= \frac{a_0}{2} + \frac{1}{2}\sum_{k=1}^{\infty}\left[(a_k - ib_k)e^{(i2\pi kt/T)} + (a_k + ib_k)e^{(-i2\pi kt/T)}\right] \\
c_k &= \frac{1}{2}(a_k - ib_k) \qquad \textit{komplexer Koeffizient}, \; k > 0 \\
c_{-k} &= c_k^* = \frac{1}{2}(a_k + ib_k) \quad \textit{konjugiert kompl. K.}, \; k > 0 \\
c_0 &= \frac{a_0}{2} \\
g(t) &= \sum_{k=-\infty}^{\infty} c_k e^{i2\pi kt/T}
\end{aligned}
$$

Die Fourier-Reihe in komplexer Darstellung

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

Satz: 73
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.

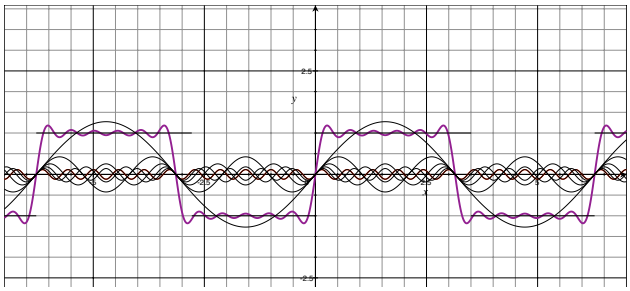
Die Koeffizienten c_k der Fourier-Reihe

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

Die Koeffizienten sind

$$\begin{aligned}
c_k &= \frac{1}{T}\int_0^T g(t)\cdot e^{-\frac{i2\pi kt}{T}}dt, \quad -\infty \le k \le \infty \\
a_k &= \frac{2}{T}\int_0^T g(t)\cdot \cos(\frac{2\pi kt}{T})dt, \quad 0 \le k \le \infty \\
b_k &= \frac{2}{T}\int_0^T g(t)\cdot \sin(\frac{2\pi kt}{T})dt, \quad 0 \le k \le \infty
\end{aligned}$$

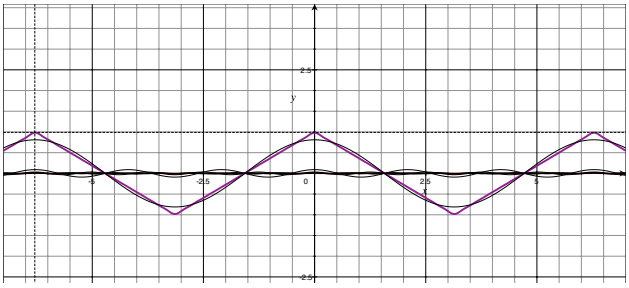
Beispiel: Rechteckfunktion



Rechteckfunktion: Amplitude A, Periode 2π,

$$\begin{aligned}
a_k &= 0 \\
b_k &= \frac{1}{1+2k} \\
g(x) &= \frac{4A}{\pi}\sum_{k=0}^{\infty}\frac{1}{1+2k}\sin((2k+1)x)
\end{aligned}$$

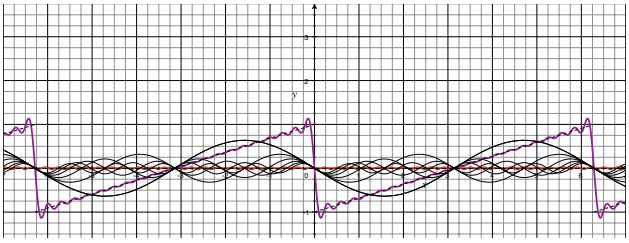
Beispiel: Dreieck-Funktion



Dreieck-Funktion: Amplitude A, Periode 2π,

$$
\begin{aligned}
a_k &= \frac{1}{(1+2k)^2} \\
b_k &= 0 \\
g(x) &= \frac{8A}{\pi\pi}\sum_{k=0}^{\infty}\frac{1}{(1+2k)^2}\cos((2k+1)x)
\end{aligned}
$$

Beispiel: Sägezahn-Funktion



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

$$
\begin{aligned}
a_k &= 0 \\
b_k &= \frac{1}{1+k} \\
g(x) &= \frac{-2A}{\pi}\sum_{k=0}^{\infty}\frac{1}{1+k}\sin((k+1)x)
\end{aligned}
$$

30.2 Fourier-Transformation (1-dimensional)

Eindimensionale Fourier-Transformation

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

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

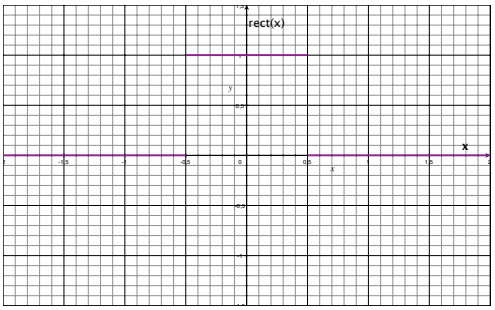
$G(f)$ entspricht den c_k und gibt an, mit welchem Gewicht sinusförmige Bestandteile der Frequenz f in $g(t)$ enthalten sind.

- $G(f)$ 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 folgenden Folien zeigen für ausgewählte Funktionen die Fouriertransformierten. Beachten Sie, daß die Fouriertransformierte der Sinusfunktion nur im Imaginärteil verschieden von Null ist, die der Kosinusfunktion nur im Realteil. Bei (Ballard and Brown, 1982, S. 27) finden Sie weitere Beispiele.

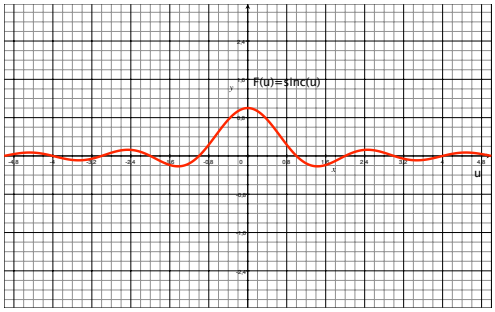
Die Fourier-Transformierte sinc(u) der Rechteck-Funktion rect(x)

$f(x)$
Rechteck-Funktion



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

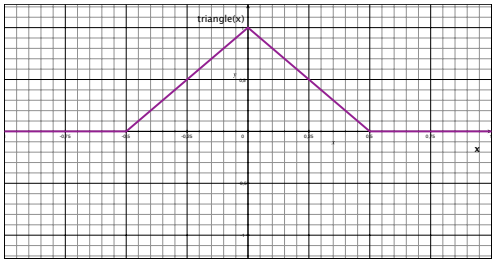
$F(u)$
sinc-Funktion



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

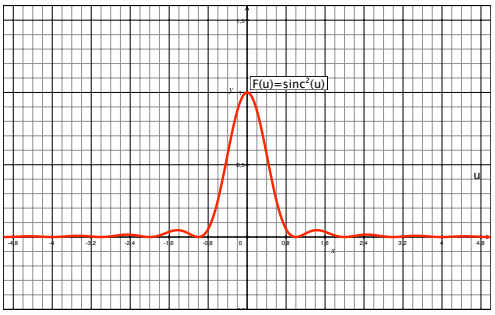
Die Fourier-Transformierte $T(u)$ der Dreieck-Funktion triangle(x)

triangle(x)
Dreieck-Funktion



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

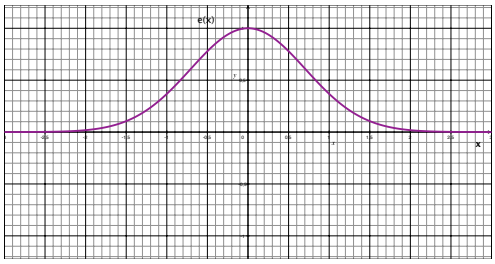
$T(u)$
sinc^2-Funktion



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

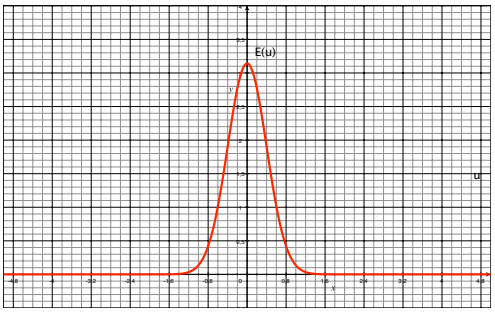
Die Fourier-Transformierte $E(u)$ der Gauß-Funktion e(x)

e(x)
Gauß-Funktion



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

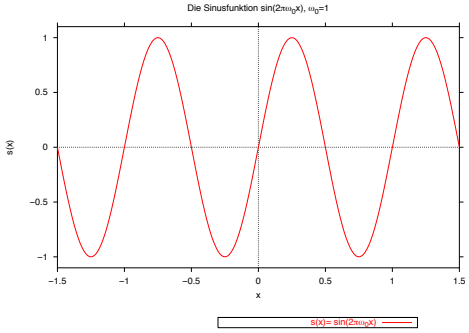
$E(u)$



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

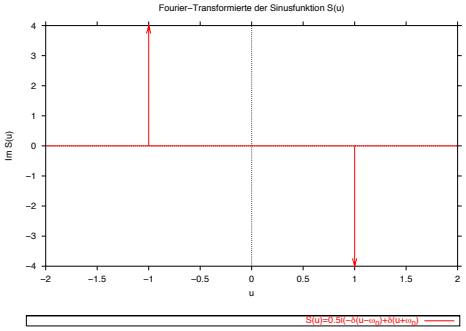
Die Fourier-Transformierte $F(u)$ der Sinus-Funktion $\sin(2\pi\omega_0 x)$

$f(x)$
Sinusfunktion



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

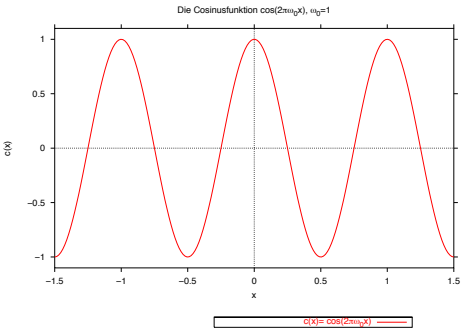
$F(u)$
Delta-Funktion



$\frac{1}{2}i[-\delta(u-\omega_0)+\delta(u+\omega_0)]$

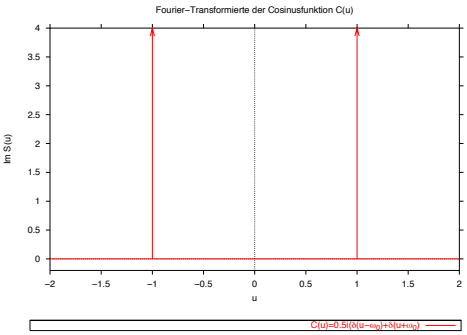
Die Fourier-Transformierte $F(u)$ der Kosinus-Funktion $\cos(2\pi\omega_0 x)$

$f(x)$
Kosinusfunktion



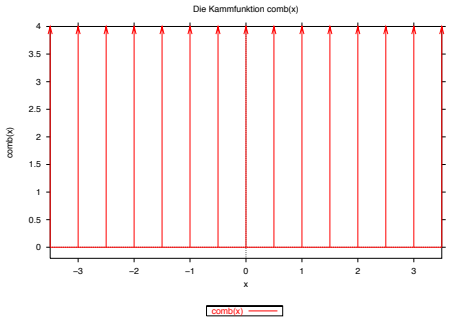
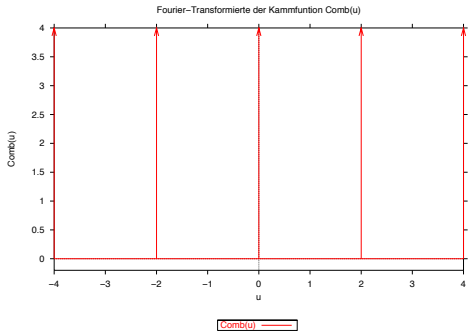
$\cos(2\pi\omega_0 x)$

$F(u)$



$\frac{1}{2}[\delta(u-\omega_0)+\delta(u+\omega_0)]$

Die Fourier-Transformierte $Comb(u)$ der Kamm-Funktion $comb(x)$

$f(x)$ Kammfunktion	$F(u)$ Kammfunktion
	
$\sum_{n=-\infty}^{\infty} \delta(x - x_0)$	$\sum_{n=-\infty}^{\infty} \delta(u - \frac{n}{x_0})$

Faltung und Faltungstheorem

Definition: 74 (*Faltung*)
*Die Faltung zweier Funktionen f und g ist eine Funktion h einer Verschiebung y, mit dem Faltungsoperator *:*

$$h(y) = f * g = \int_{-\infty}^{\infty} f(x)g(y - x)dx$$

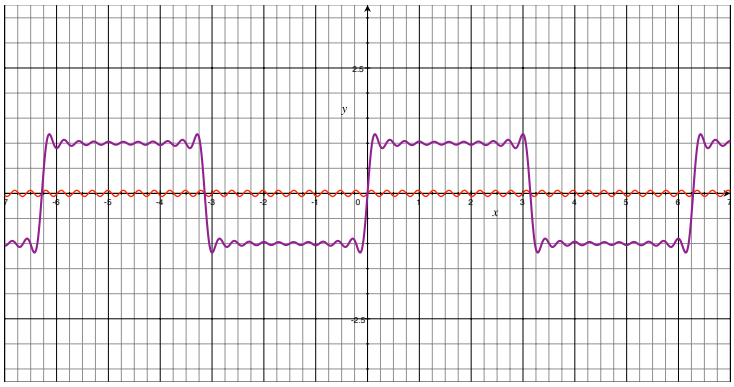
Satz: 75 (Faltungstheorem)
Die Fourier-Transformierte der Faltung zweier Funktionen f und g ist gleich dem Produkt $\mathcal{F}(f) \cdot \mathcal{F}(g)$ der Fourier-Transformierten der beiden Funktionen.

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g) = F \cdot G$$

31 Abtasttheorem (sampling theorem)

Bandbreite und Grenzfrequenz

Definition: 76 (*Grenzfrequenz*)
$g(t)$ ist bandbeschränkt mit Grenzfrequenz W, wenn $G(f) = 0$ für $|f| > W$.



Eine Funktion und ihre Grenzfrequenz

✎ **Anmerkung:** Man sagt auch kurz: „Eine Funktion mit der Bandbreite W“ oder „$g(t)$ ist bandbegrenzt auf W.“

Abtasttheorem (sampling theorem)

Satz: 77 (**Abtastheorem**)
☞ *Eine Funktion mit Bandbreite W kann exakt aus Abtastwerten zurückgewonnen werden, wenn diese höchstens $\frac{1}{2W}$ auseinanderliegen. (Nyquist?Shannon sampling theorem)*

31.1 Ein Beweis

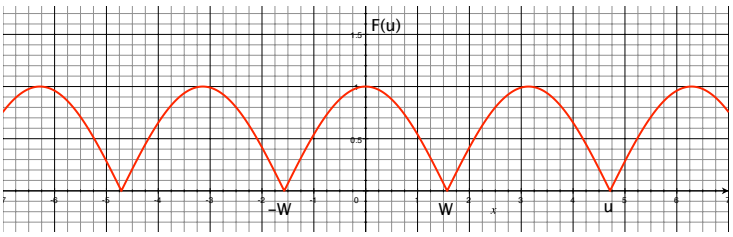
Der Beweis

Beweis: 78

☞ *$g(t)$ sei bandbeschränkt mit Grenzfrequenz W.*

$\Longrightarrow G(f) = 0$ *für* $|f| \geq W$

Setzt man $G(f)$ periodisch fort, kann für die Fourier-Transformierte eine Fourier-Reihenentwicklung angewendet werden.



$$
\begin{aligned}
G(f) &= \sum_{k=-\infty}^{\infty} c_k \cdot e^{-i2\pi k f/(2W)} \qquad G(f) \text{ hat die Periode } 2W \\
\textit{mit } c_k &= \frac{1}{2W}\int_{-W}^{W} G(f) e^{i2\pi k f/(2W)} df \\
&\quad \text{wegen } G(|f| > W) = 0 \quad \Rightarrow \\
&= \frac{1}{2W}\int_{-\infty}^{+\infty} G(f) \cdot e^{i2\pi f k/(2W)} df \\
&\quad \text{wegen Definition von } \mathcal{F}^{-1} \Rightarrow \\
&= \frac{1}{2W} \cdot g\left(\frac{k}{2W}\right) \\
g\left(\frac{k}{2W}\right) &= \text{diskrete Abtastwerte im Abstand } \frac{1}{2W}
\end{aligned}
$$

q.e.d.

Rekonstruktion aus Abtastwerten

$$
\begin{aligned}
g(t) &= \mathcal{F}^{-1}\{G(f)\} = \mathcal{F}^{-1}\left\{\frac{1}{2W}\sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right)\cdot e^{-i2\pi f k/(2W)}\right\} \\
&= \int_{-W}^{+W}\frac{1}{2W}\sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right) e^{-i2\pi f k(2W\ )}\cdot e^{i2\pi f t}df \\
&= \sum_{k=-\infty}^{\infty}\frac{1}{2W}\cdot g\left(\frac{k}{2W}\right)\underbrace{\int_{-W}^{+W} e^{i2\pi f(t-\frac{k}{2W})}df}_{S^*} \\
S^* &= \frac{1}{i2\pi(t-\frac{k}{2W})}\cdot e^{i2\pi f(t-\frac{k}{2W})}\Bigg|_{-W}^{+W} \quad \textit{mit der Euler-Formel:}
\end{aligned}
$$

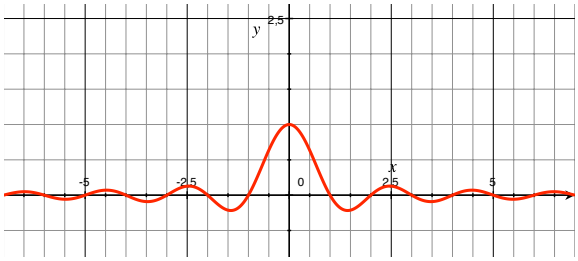
Rekonstruktion als Fourier-Reihe

$$
\begin{aligned}
Sei \qquad & \alpha = 2\pi\left(t-\frac{k}{2W}\right) \\
S^* &= \frac{1}{\alpha}\{-i\cos(f\alpha)+\sin(f\alpha)\}\Bigg|_{-W}^{+W} \\
& \cos \textit{ -Term verschwindet wegen } \cos(W) = \cos(-W) \\
g(t) &= \sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right)\frac{\sin[2\pi W(t-\frac{k}{2W})]}{2\pi W(t-\frac{k}{2W})}, \ \operatorname{sinc}(u) = \frac{\sin(\pi u)}{\pi u}
\end{aligned}
$$

☞

$$
\boxed{g(t) = \sum_{k=-\infty}^{\infty} g\left(\frac{k}{2W}\right)\operatorname{sinc}\left[2W\left(t-\frac{k}{2W}\right)\right]}
$$

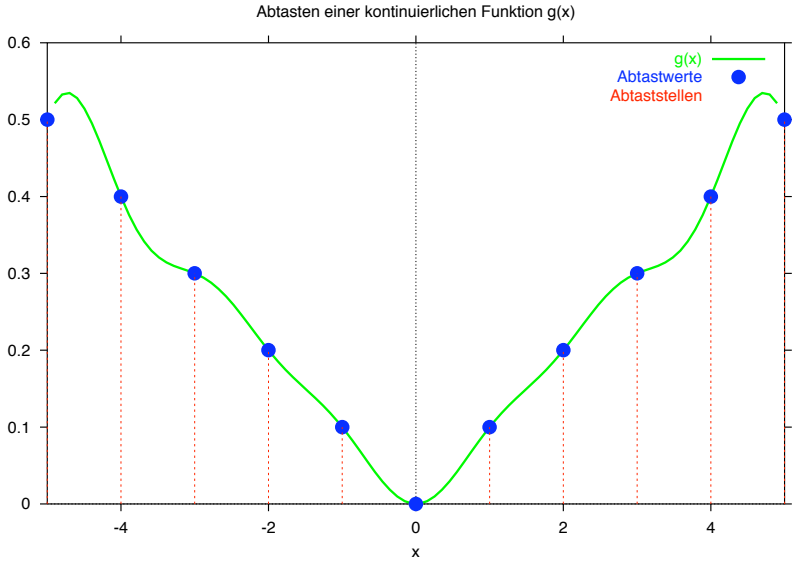
Rekonstruktion des Bildsignals



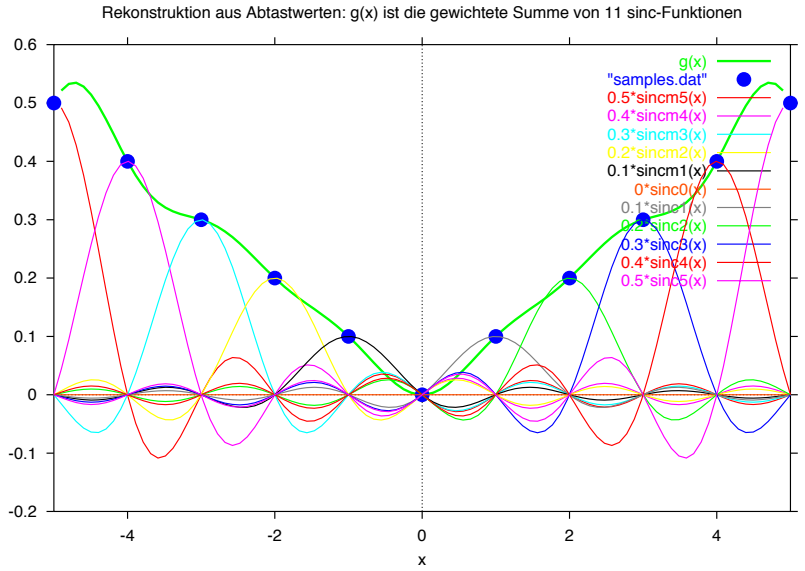
☞ Zur Rekonstruktion von $g(t)$ aus Abtastwerten

- 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



31.2 Scheinfrequenzen: Aliasing

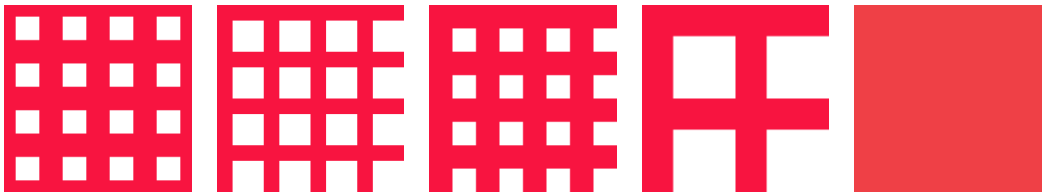
Scheinfrequenzen: Aliasing

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).

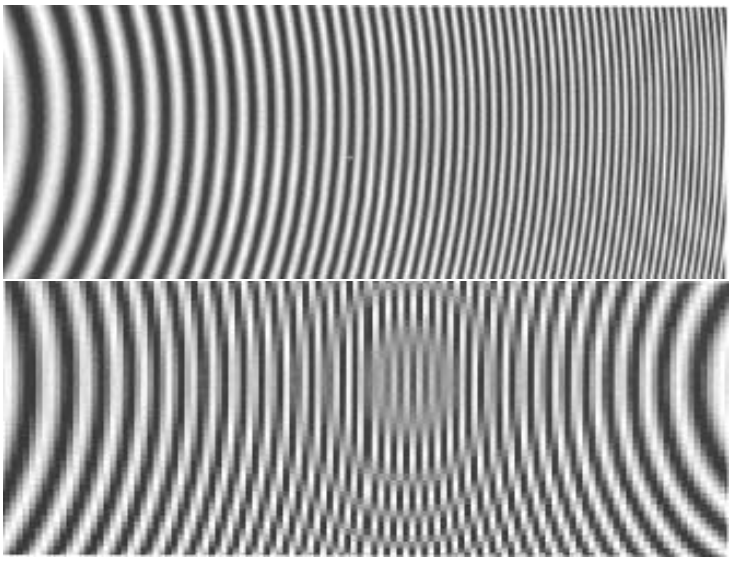
- Dieser Effekt heißt *aliasing* und tritt nicht nur beim Abtasten echter Signale auf sondern auch bei virtuellen (synthetischen) Signalen, z.B. beim *ray tracing*.
- Aliasing muß unbedingt vermieden werden, denn solche Bilddaten sind wegen der zufälligen Scheinfrequenzen *physikalisch sinnlos*, wenn sie auch graphisch gelegentlich sehr ansprechend sind.

Aliasing: Ein Beispiel

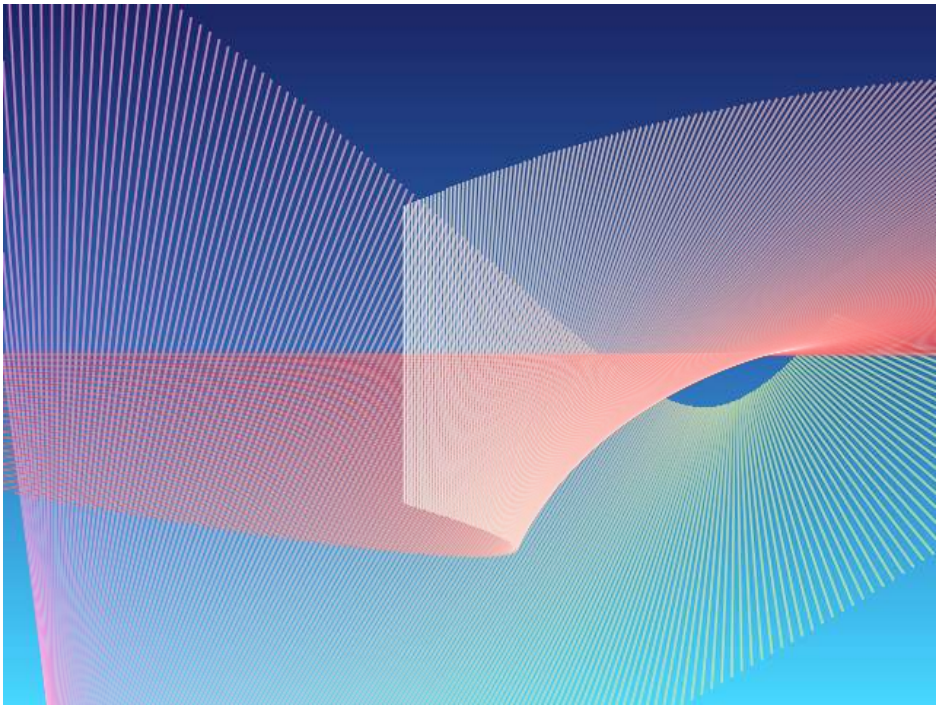
Ein 16×16-Bild und vier unterabgetastete Bilder in den Größen $12 \times 12, 8 \times 8, 5 \times 5$ und 4×4



Scheinfrequenzen



Beispiel: Aliasing



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movqtzAliasing1.html

Zur Entstehung von Scheinfrequenzen:

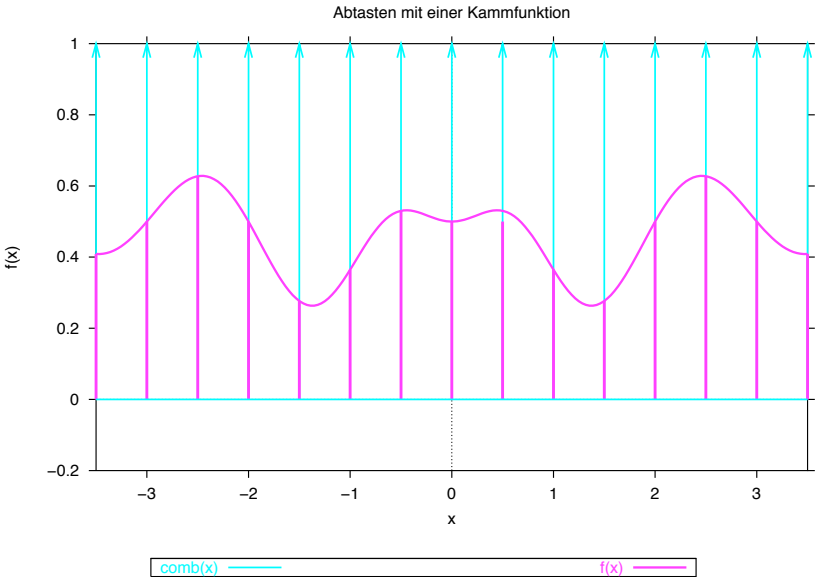
Wir können das Abtasten als ein *Produkt* der Bildfunktion f mit einer *Kammfunktion* modellieren:

- 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

$$
\begin{aligned}
f_s &= f(x)\sum_n \delta(x - nx_0) \\
f_s &= \sum_n f(nx_0)\delta(x - nx_0) \\
F_s(u) &= F(u) * \frac{1}{x_0}\sum_n \delta(u - \frac{n}{x_0}) \\
\text{wegen} \quad & \int_{-\infty}^{\infty} f(x)\delta(x - a)dx = f(a) \\
F(u) * \delta(u - \frac{n}{x_0}) &= F(u - \frac{n}{x_0}) \\
F_s(u) &= \frac{1}{x_0}\sum_n F(u - \frac{n}{x_0})
\end{aligned}
$$

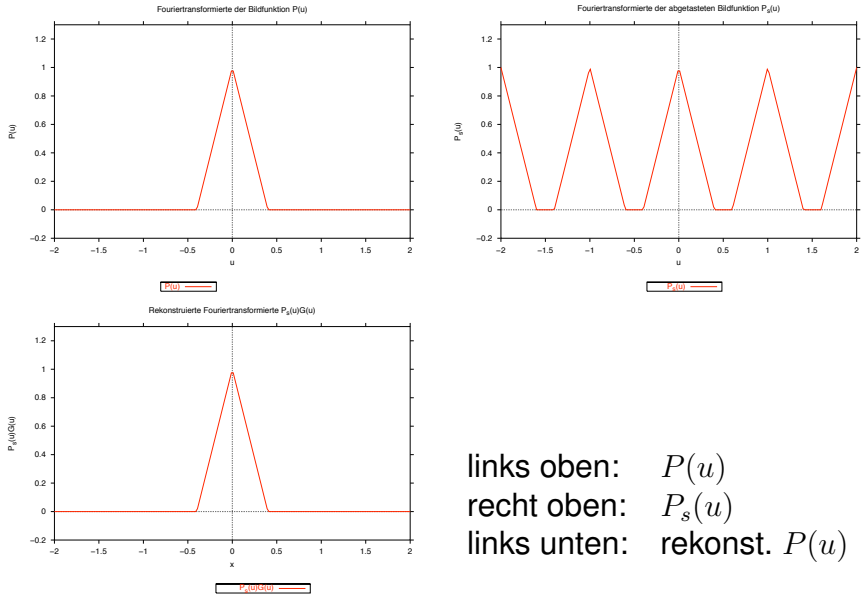
Fensterung der Fourier-Transformierten

- ☞ 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)

$$G(u) = \begin{cases} 1 & |u| < \frac{1}{2x_0} \\ 0 & \text{sonst} \end{cases}$$

$$f(x) = \mathcal{F}^{-1}[F_s(u)G(u)]$$

Abtastung mit geeigneter Schrittweite

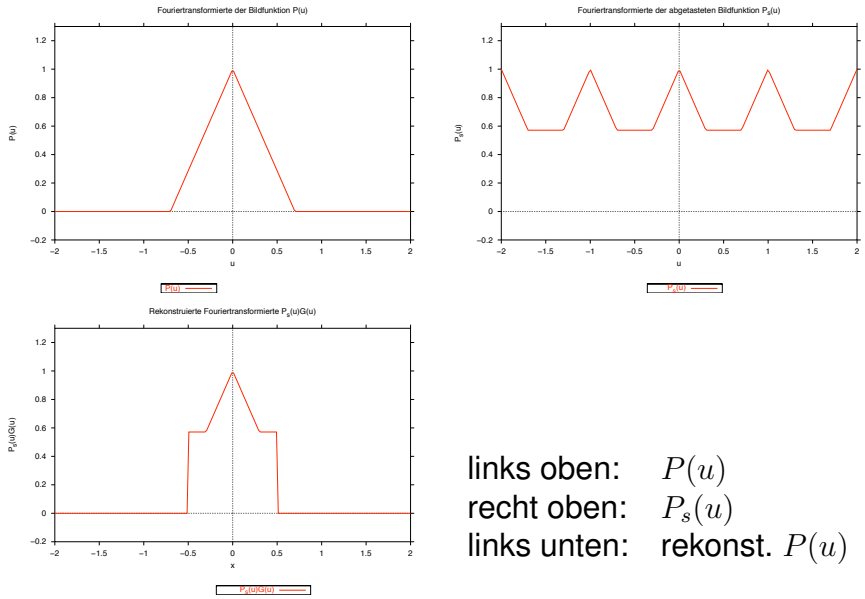


links oben: $P(u)$
recht oben: $P_s(u)$
links unten: rekonst. $P(u)$

Grenzfrequenz richtig

Auf der folgenden Folie sehen Sie, wie bei einer zu hohen Grenzfrequenz sich die Beiträge von $F(u)$ in $F_s(u)$ überlappen. Im Teil des Spektrums, der mit der Fensterfunktion $G(u)$ ausgeschnitten wird, sind noch Beiträge der weiteren Summanden in der Reihe vorhanden. Diese Beiträge sehen wir als Scheinfrequenzen, eben das Aliasing.

Abtastung mit zu großer Schrittweite



links oben: $P(u)$
recht oben: $P_s(u)$
links unten: rekonst. $P(u)$

Grenzfrequenz falsch

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:

$$\Longrightarrow |g\prime(t)| \leq 2\pi W M$$

d.h. die erste Ableitung des Signals ist ebenfalls beschränkt .

Variation eines TV-Signals

Beispiel: 79
Typisches TV-Signal 5 MHz

Bandbreite W: $5\ MHz$

Abstand zweier Abtastwerte: $\frac{1}{2W} = \frac{1}{2 \cdot 5MHz} = 100\ ns$

Helligkeitsbereich:

$$\begin{aligned} g(t) &= -128 \cdots 128 \\ |g\prime(t)| &\leq 2\pi \cdot 5 \cdot 5 \cdot MHz \cdot 128 = 4021.24MHz \end{aligned}$$

Differenz zwischen Abtastwerten Δg:

$$|\Delta g| \leq |g\prime(t)| \cdot 100ns \approx 400$$

Zu Bernsteins Ungleichung siehe (Duda and Hart, 1973, S. 304). Aus Bernsteins Ungleichung folgt, daß ein Digitalbild keine beliebig steilen Kanten enthalten kann.

Antialiasing bei synthetischen Bildern

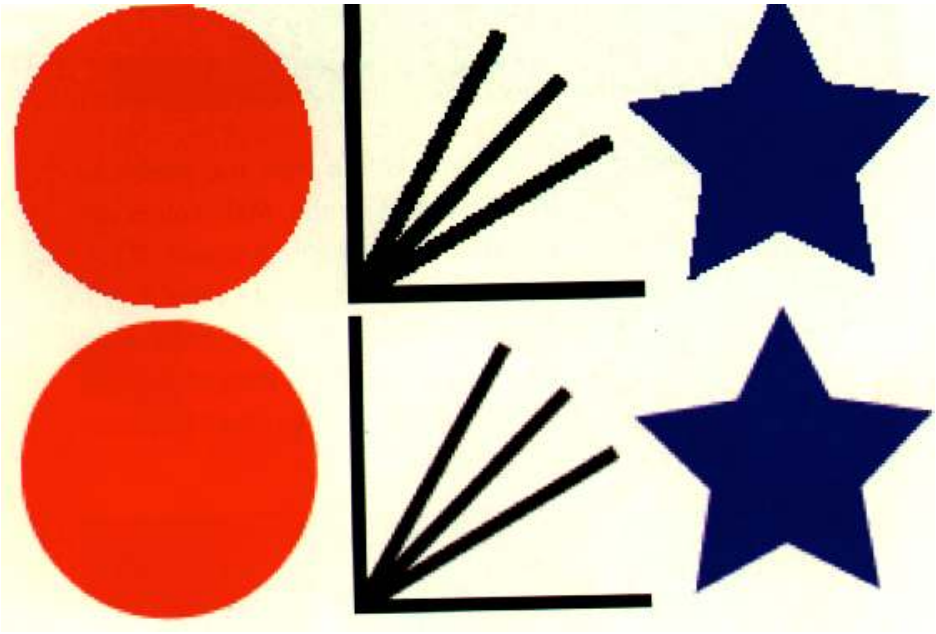
Synthetische Bilder verletzen oft das Abtasttheorem, indem sie steile Helligkeitssprünge enthalten, die bei natürlichen, bandbegrenzten Signalen nicht möglich wären.

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

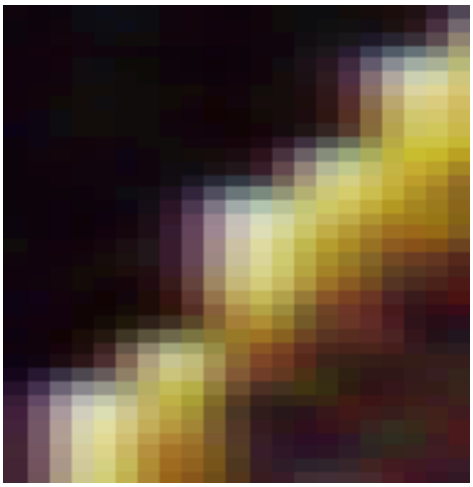


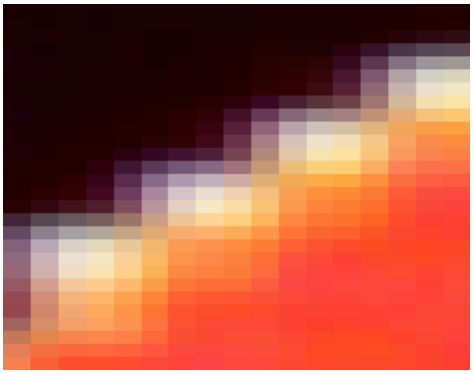
Diese und die folgenden Beispiele stammen aus (Siegel, 1996).

Halos: Falsches Antialiasing

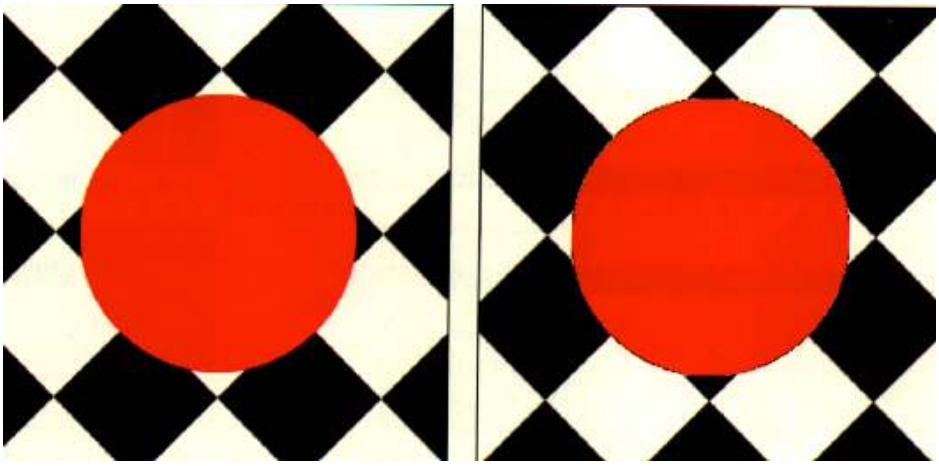
Wenn nach dem Antialiasing der Hintergrund gewechselt wird, entstehen häßliche Halos:

Ausschnitte aus den Halos





Antialiasing bei strukturiertem Hintergrund



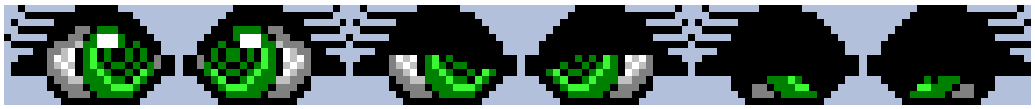
Korrektes Antialisasing falsches Antialiasing

Aliasing bei Blue Screen Matting

Blue Screen Matting mit Anti-Aliasing

31.3 Abtasten in der Zeit

Abtasten in der Zeit: Animationen, Filme



Bildfolgen

Definition: 80 (*Bildfolgen*)
sind Folgen von Bildern (Filme), die durch eine zeitliche Abtastung einer Szene oder Synthetisieren einer Szene (Animationen) zu mehreren Zeitpunkten entstehen.

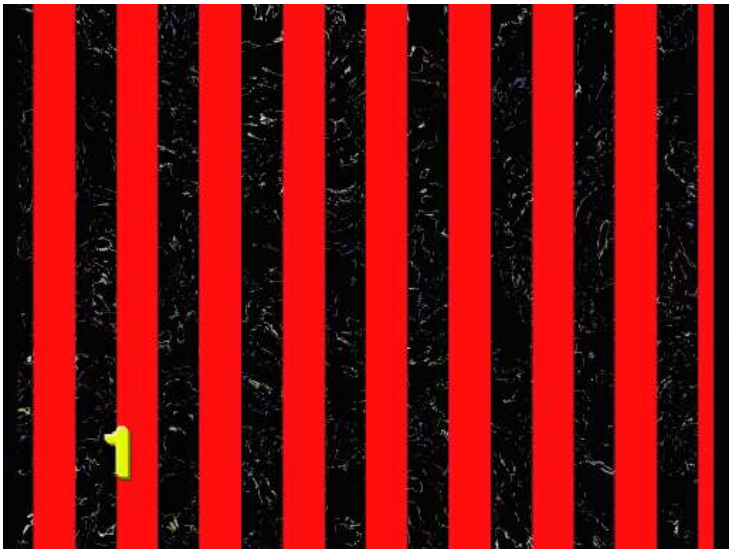
- ☞ Auch beim Abtasten in der Zeit ist das Abtasttheorem unbedingt zu beachten.

  Sonst kann aliasing zu *Scheinbewegungen* führen.

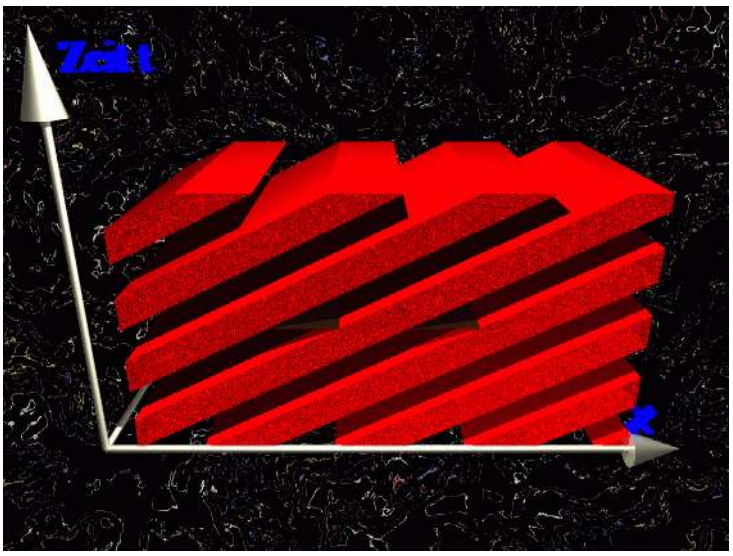
- ☞ Bei zu geringer Abtastfrequenz können sich Bewegungen umkehren oder Objekte scheinbar stillstehen.

✎ **Anmerkung:** Animationen: Das Abtasttheorem gilt natürlich nicht nur für das Abtasten von Bildsignalen, sondern ist auch wichtig für die Erzeugung von Animationen. Da hier hier für jeden Zeitpunkt ein Bild synthetisiert werden muß und die Erzeugung der Einzelbilder meistens sehr rechenaufwendig ist, sollte man die Grenzfrequenzen sehr genau kennen, damit nicht unnötig viele Bilder gerechnet werden.

Verschiebung eines periodischen Musters



Ein zeitveränderliches analoges Bildsignal

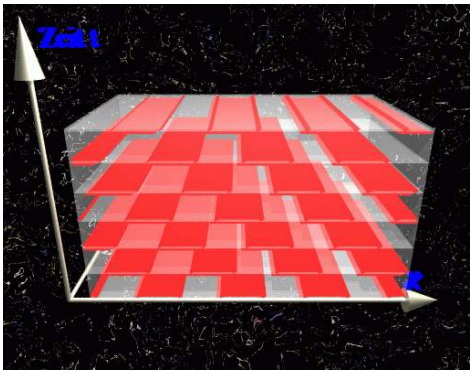


Bildfolge als 3D-Bild

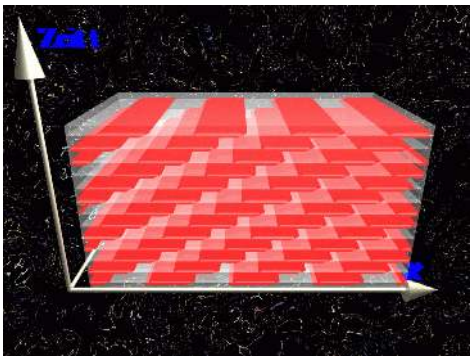
- Eine Bildfolge kann als dreidimensionales Bild betrachtet werden, das neben den Zeilen- und Spaltenkoordinaten noch eine zusätzliche Zeitkoordinate hat.
- Ein bewegtes Streifenmuster bildet im dreidimensionalen Orts-Zeitraum eine periodische Folge von gescherten Quadern.

  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

Auf der folgenden Folie sehen Sie ein Rad mit zerbrochenen Speichen. Wird dieses Rad in Rotation versetzt und die Bewegung abgetastet, können Sie bei einer zu kleinen Abtastfrequenz den äußeren Bereich der Speichen schneller rotieren sehen als den inneren. Das Rad bewegt sich dann scheinbar nicht mehr als ein starrer Körper, sondern zerfällt in mehrere unabhängig voneinander rotierende Teile.

Die „broken wheel illusion“

Quicktime movie:http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/wheelQTSmall.html

Quartz Composer Animation: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/RotorscheibeBrokenWheel.html

Teil XII

Verarbeitung im Frequenzraum

32 2D-Fourier-Transformation

32.1 2D-Fourier-Transformation: Kontinuierliche Signale

2D-Fourier-Transformation

$$
\begin{aligned}
x, y &= \text{Orts } \textit{koordinaten} \\
f_x, f_y &= \text{Orts } \textit{frequenzen} \\
\lambda = \frac{c}{f} &= \text{Orts } \textit{wellenlängen} \\
G(f_x, f_y) &= \mathcal{F}\{g(x,y)\} \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x,y) e^{-2\pi i (f_x x + f_y y)} dx dy \\
g(x,y) &= \mathcal{F}^{-1}\{G(f_x, f_y)\} \\
&= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} G(f_x, f_y) e^{2\pi i (f_x x + f_y y)} df_x df_y
\end{aligned}
$$

32.1.1 Interpretation des Spektrums

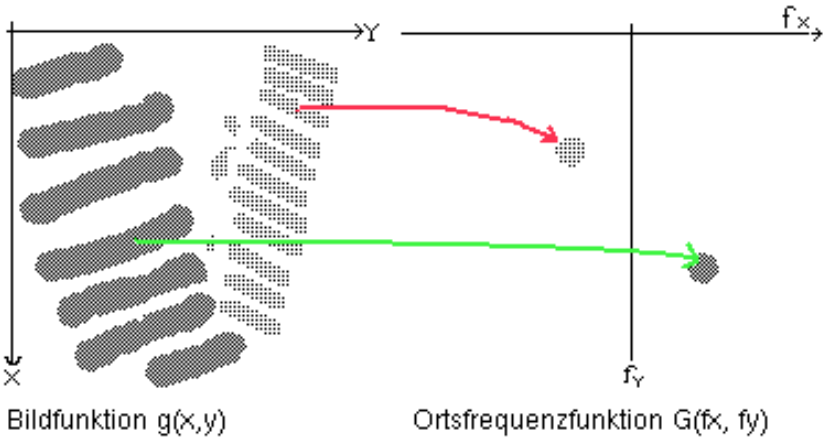
Interpretation des Spektrums

Bildfunktion: $g(x, y)$

Ortsfrequenzfunktion: $G(f_x, f_y)$

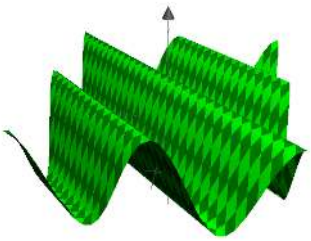
Interpretation: Der Funktionswert an der Stelle (f_x, f_y) gibt *Phase* und *Betrag* der sinusförmigen Grauwertkomponente in $g(x, y)$ mit Frequenz in x-Richtung f_x und Frequenz in y-Richtung f_y an.

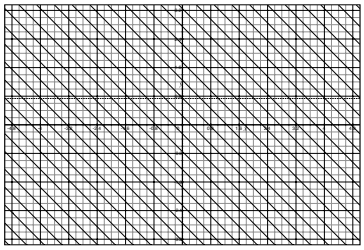
Ein schematisches Beispiel



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.

Nullstellen: Mit $u = 0.5$ und $v = 0.25$ ergeben sich die Perioden in X und Y von $T_x = 2$ und $T_y = 4$. Der Abstand der Geraden ist 1.7888544.

Bei dieser reinen cos-Funktion ist die $G(f_x, f_y)$ nur an zwei Stellen verschieden von Null: bei den Ortsfrequenzen (u,v) und (-u,-v).

32.1.2 Eigenschaften

Eigenschaften der Fourier-Transformation

Linearität:

$$\mathcal{F}\{a \cdot g_1(x,y) + b \cdot g_2(x,y)\}$$
$$= a \cdot \mathcal{F}\{g_1(x,y)\} + b \cdot \mathcal{F}\{g_2(x,y)\}$$

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: 81 (Faltungstheorem)

$$\mathcal{F}\left\{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g_1(\xi,\eta) \cdot g_2(x-\xi, y-\eta) d\xi d\eta\right\} =$$
$$\mathcal{F}\{g_1(x,y)\} \cdot \mathcal{F}\{g_2(x,y)\} = G_1(f_x, f_y) \cdot G_2(f_x, f_y)$$

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 $g(x,y)$.

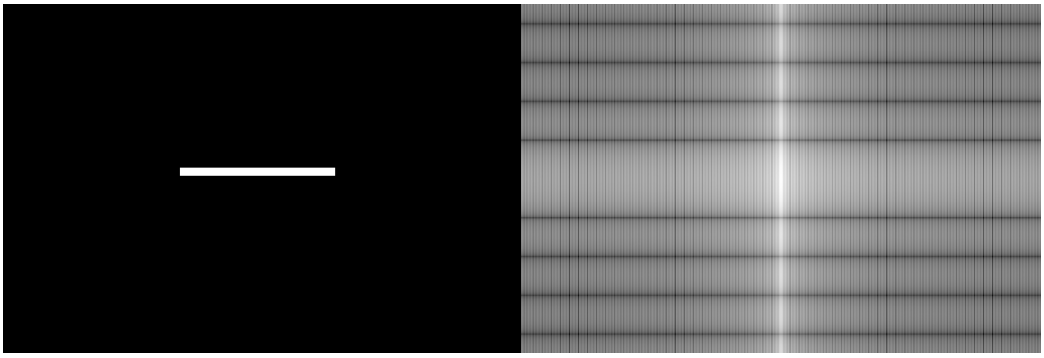
Betragsspektrum bzw. Amplitudenspektrum) von $g(x,y)$:

$$|G(f_x, f_y)| = \sqrt{\text{Re}\,[G(f_x, f_y)]^2 + \text{Im}[G(f_x, f_y)]^2}$$

Energiespektrum bzw. Powerspektrum von $g(x,y)$:

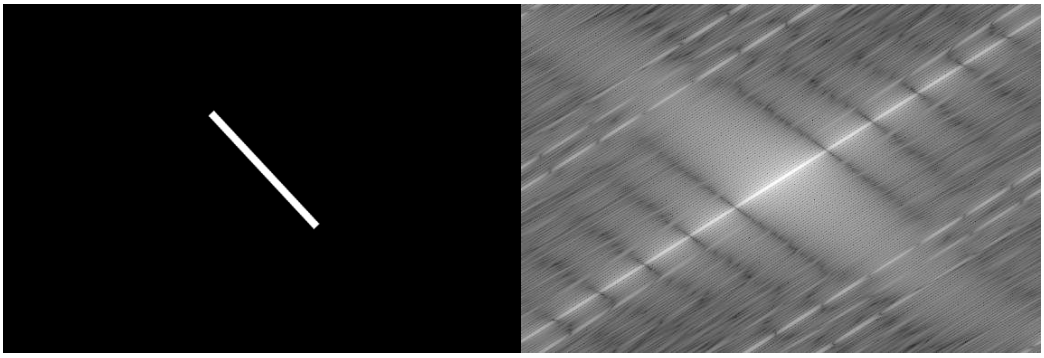
$$|G(f_x, f_y)|^2 = G(f_x, f_y) \cdot G^*(f_x, f_y)$$

Rotationseigenschaften der Fouriertransformation I



Ein einfaches Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

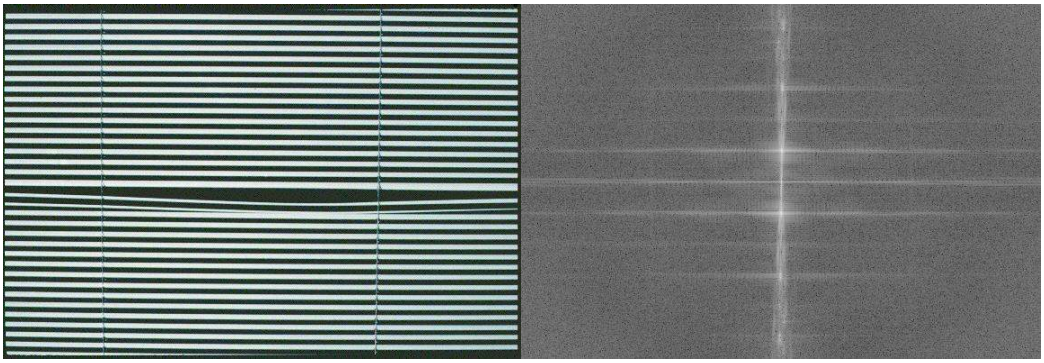
Rotationseigenschaften der Fouriertransformation II



Das rotierte Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

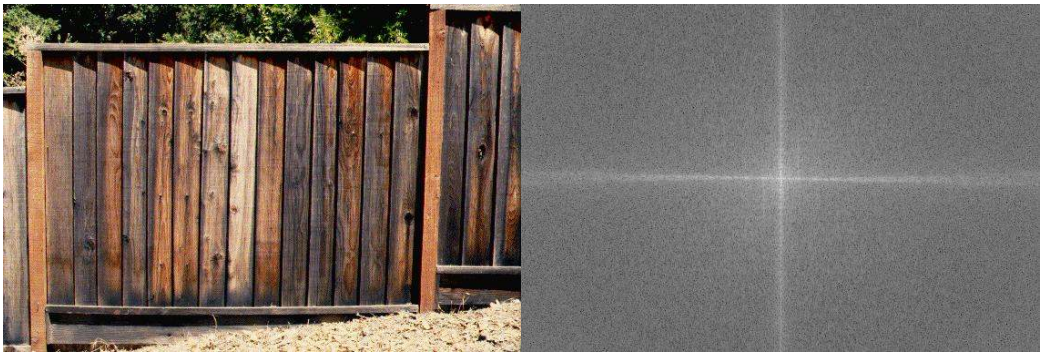
Die folgenden weiteren Beispielbilder wurden von Benjamin Seppke zur Verfügung gestellt.

Weitere Beispiele: Jalousie



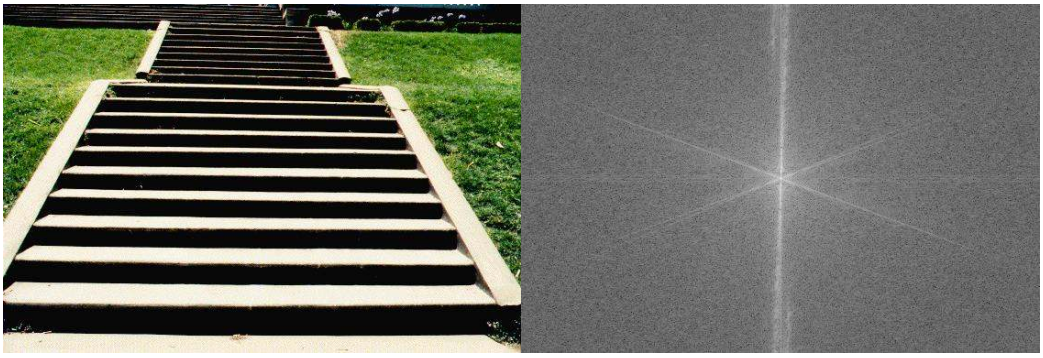
Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Weitere Beispiele: Zaun



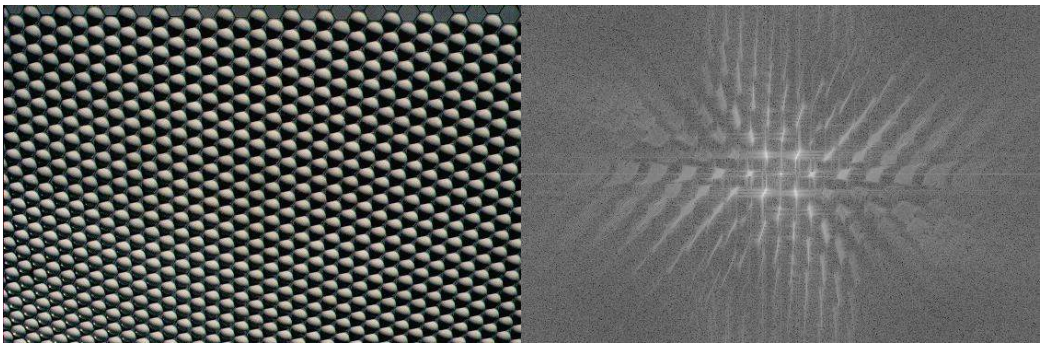
Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Weitere Beispiele: Treppe



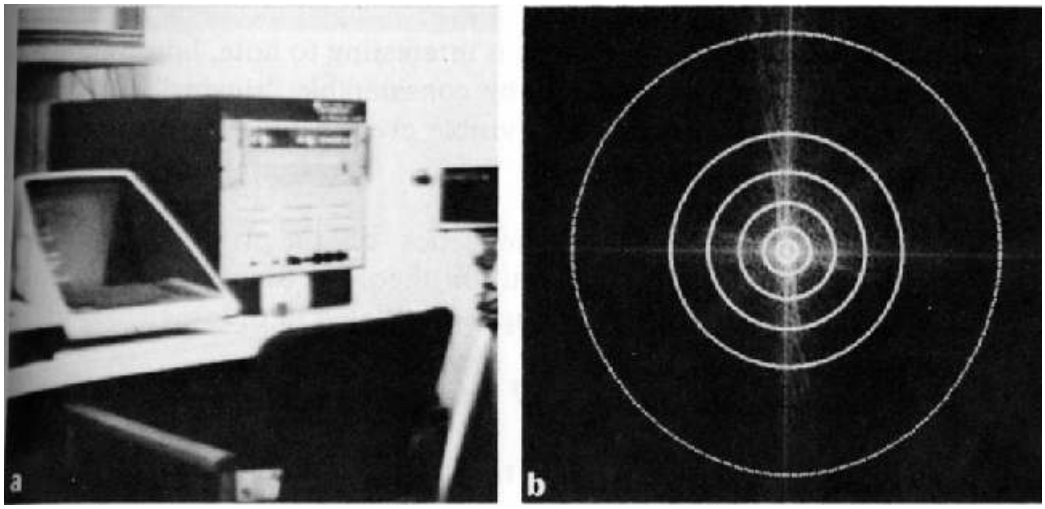
Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

Weitere Beispiele: Gitter



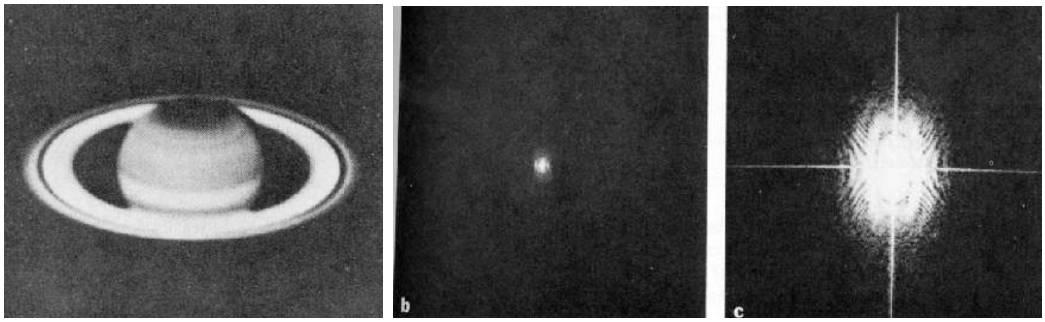
Grauwertbild und das dazugehörige Powerspektrum (logarithmiert).

256 $\times$ 256 Grauwertbild und Fourierspektrum



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: $\log(1+|F(u,v)|)$

33 Filter

33.1 Linearität von Filtern

Lineares Übertragungsverhalten von Filtern

Aus der *Nachrichtentechnik:*
Viele Bausteine zur Signalübertragung zeigen (in Grenzen) *lineares* Übertragungsverhalten:

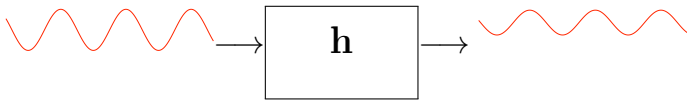
- Sei h ein Filter und seien $s_1(t), s_2(t)$ Signale am Eingang des Filters zum Zeitpunkt t,
- $s_{1h}(t), s_{2h}(t)$: Ausgang des Filters zum Zeitpunkt t:

$$
\begin{array}{ccccc}
s_1(t) & \longrightarrow & \boxed{\mathbf{h}} & \longrightarrow & s_{1h}(t) \\
s_2(t) & \longrightarrow & \boxed{\mathbf{h}} & \longrightarrow & s_{2h}(t) \\
\alpha s_1(t) + \beta s_2(t) & \longrightarrow & \boxed{\mathbf{h}} & \longrightarrow & \alpha s_{1h}(t) + \beta s_{2h}(t)
\end{array}
$$

Übertragungsfunktion H(F)

Definition: 82 (*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:

$$s_h(t) = \underbrace{\int_{-\infty}^{\infty} h(t-\tau) \cdot s(\tau) d\tau}_{\textit{Faltungsintegral, “convolution”}}$$

$$= h(t) * s(t)$$

Filter werden charakterisiert durch:

☞ Übertragungsfunktion $H(f)$, „Filterfunktion“

☞ Impulsantwort $h(t)$

$$\int_{-\infty}^{\infty} h(t-\tau) \cdot \delta(\tau) d\tau = h(t)$$

☞ $\delta =$ Impuls = Delta-Funktion = Dirac-Stoß

$$\delta(t) = \begin{cases} \infty & \text{wenn } t = 0 \\ 0 & \text{andernfalls} \end{cases}$$

33.2 Filtern von Bildern

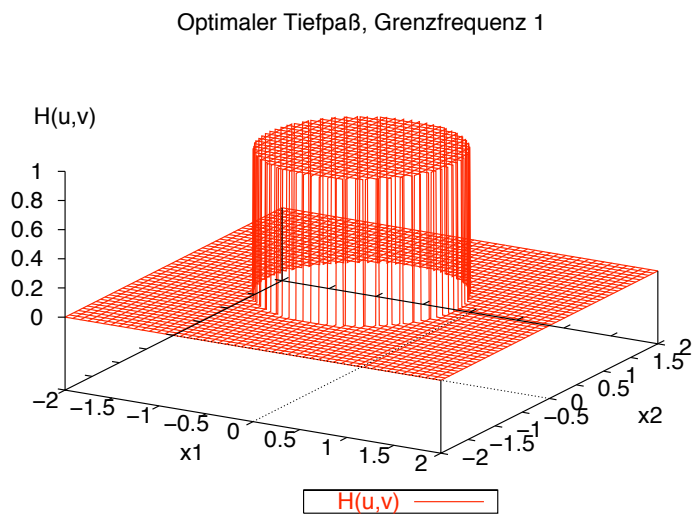
33.2.1 Tiefpaßfilter

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

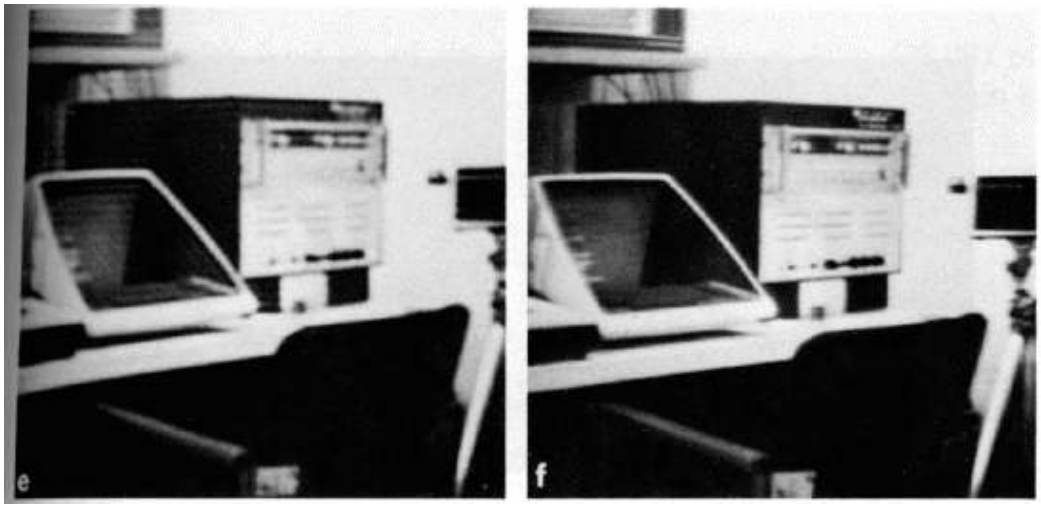


Tiefpaßfilter

Tiefpaßfilter:
sperren oder dämpfen hohe Frequenzen.

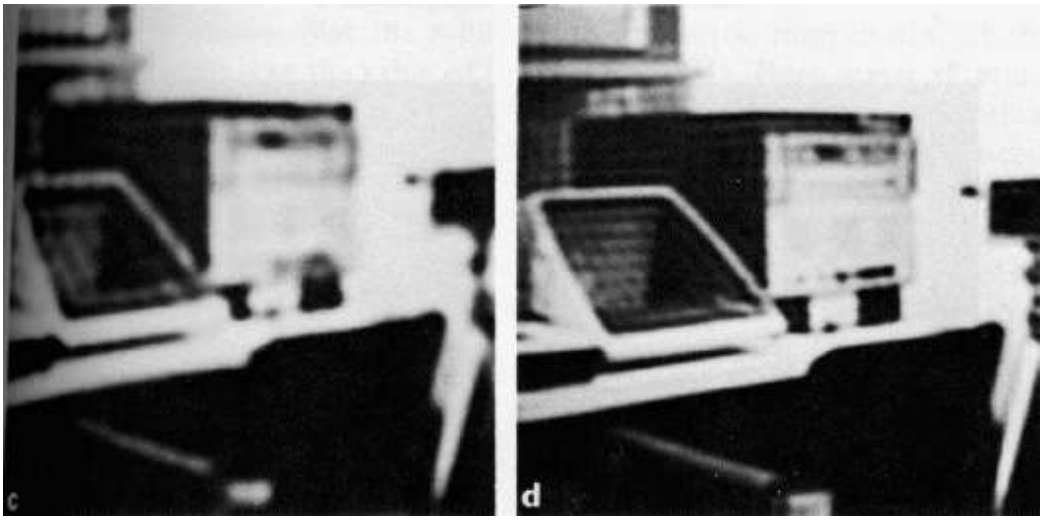
- Sie werden verwendet, um Bilder zu glätten (Rauschunterdrückung, Weichzeichnung)
- oder um eine bestimmte Grenzfrequenz sicherzustellen, z.B. vor einer Unterabtastung.
- Die vorherige Folie zeigt ein *optimales Tiefpaßfilter:* Alle Frequenzen jenseits der Grenzfrequenz werden vollständig gesperrt, alle anderen werden vollständig durchgelassen.

Beispiel: Optimales Tiefpaßfilter I



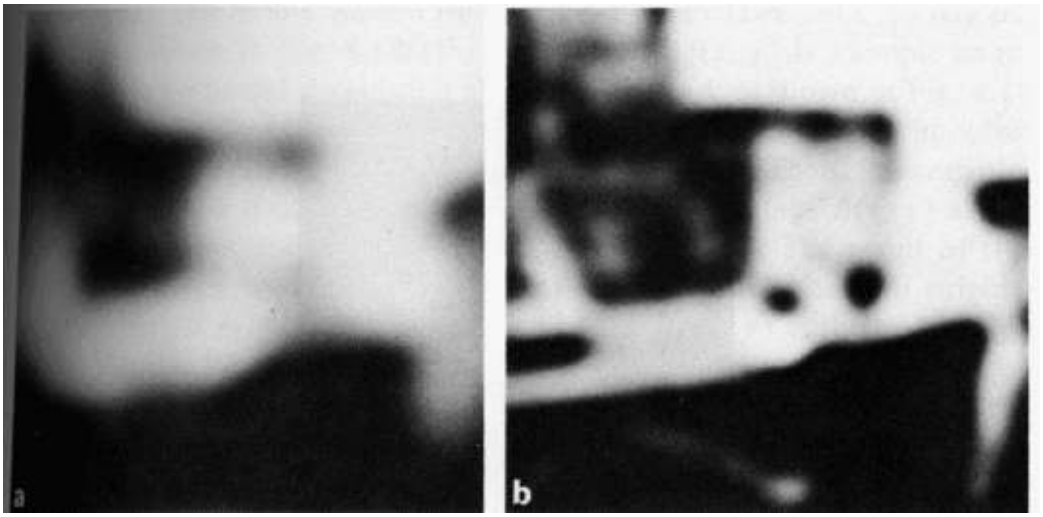
links) Hohe Grenzfrequenz, leichte Weichzeichnung
rechts) Originalbild

Beispiel: Optimales Tiefpaßfilter II



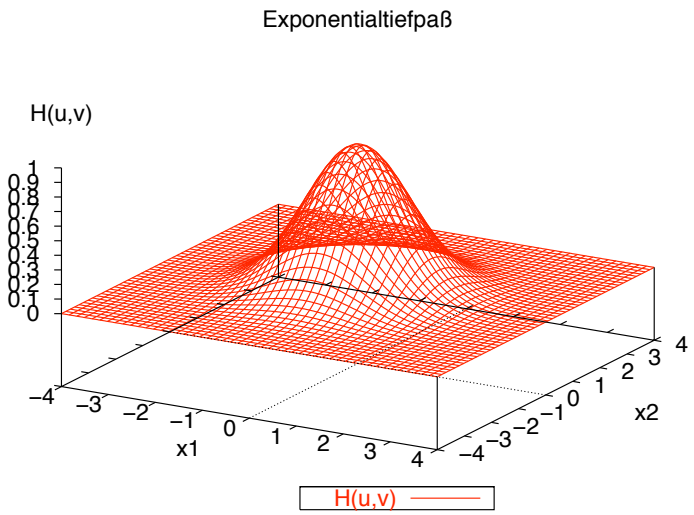
Filterung bei mittleren Grenzfrequenzen, merklicher Weichzeichnereffekt, deutliches *ringing*.

Beispiel: Optimales Tiefpaßfilter III



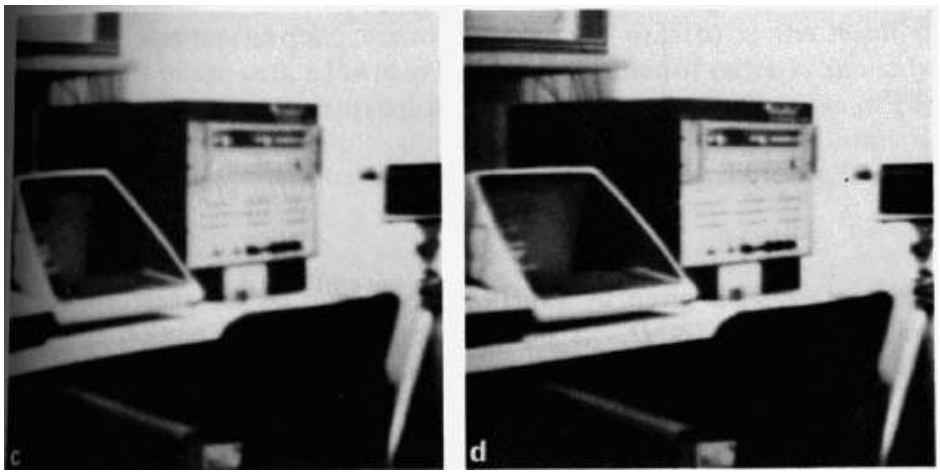
Sehr niedrige Grenzfrequenz, übertriebene Unschärfe

Exponential-Tiefpaß



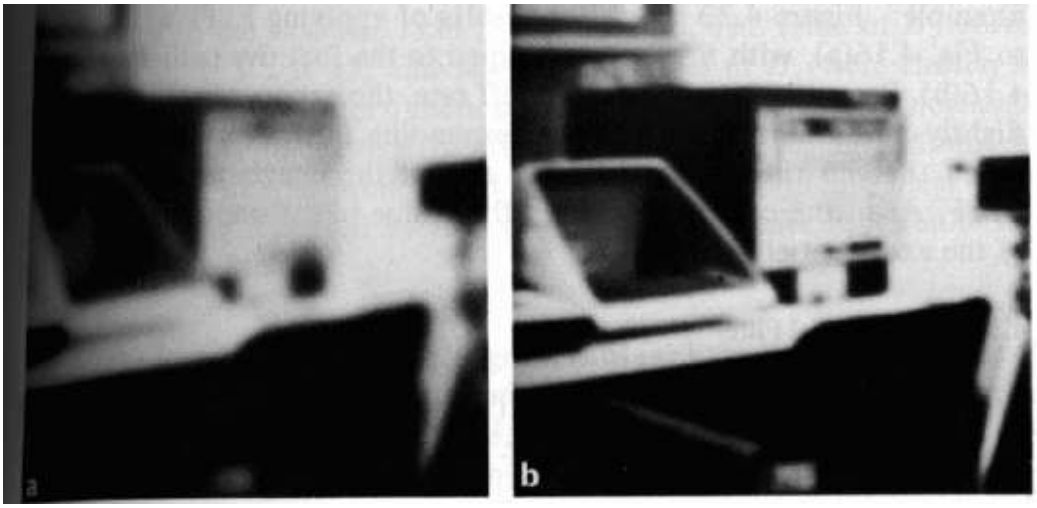
Filter mit glattem Verlauf mildern das *ringing*, z.B. Exponentialfilter, Butterworth-Filter.

Beispiel: Exponential-Tiefpaßfilter



Filterung bei mittlerer Grenzfrequenz, das *ringing* wird vermieden.

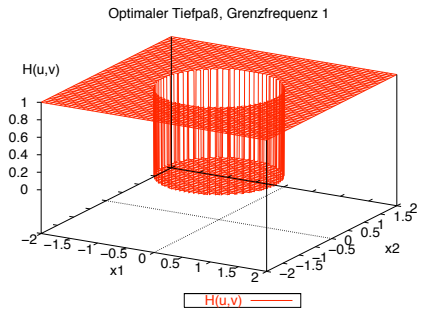
Beispiel: Exponential-Tiefpaßfilter II

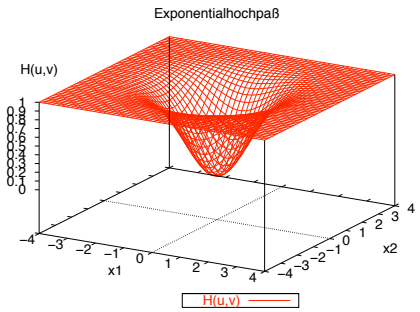


Filterung bei sehr niedriger Grenzfrequenz, deutliche Unschärfe.

33.2.2 Hochpaßfilter

Hochpaßfilter

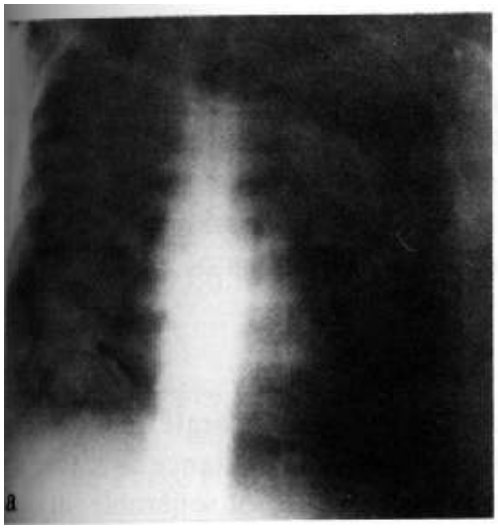




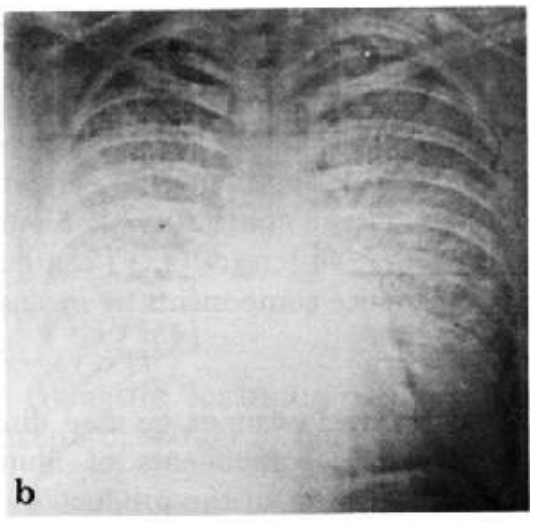
Hochpaßfilter

- unterdrücken oder dämpfen niedrige Frequenzen.
- Sie lassen Bilder schärfer erscheinen.
- Grautöne in glatten Bildbereichen erscheinen ausgewaschen.
- Über Hochpaßfilter können Konturen gefunden werden-

Beispiel: Hochpaßfilterung I

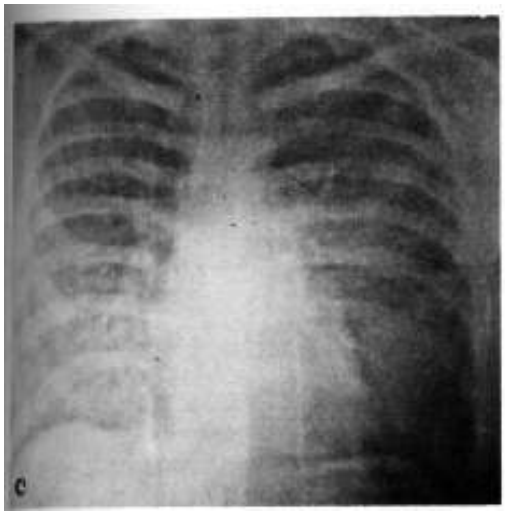


a) Originalbild

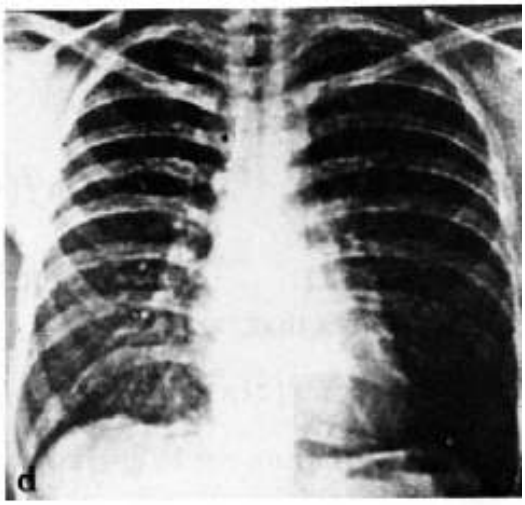


b) Hochpaß-Exponentialfilter

Hochpaßfilterung II



c) High frequency emphasis filter



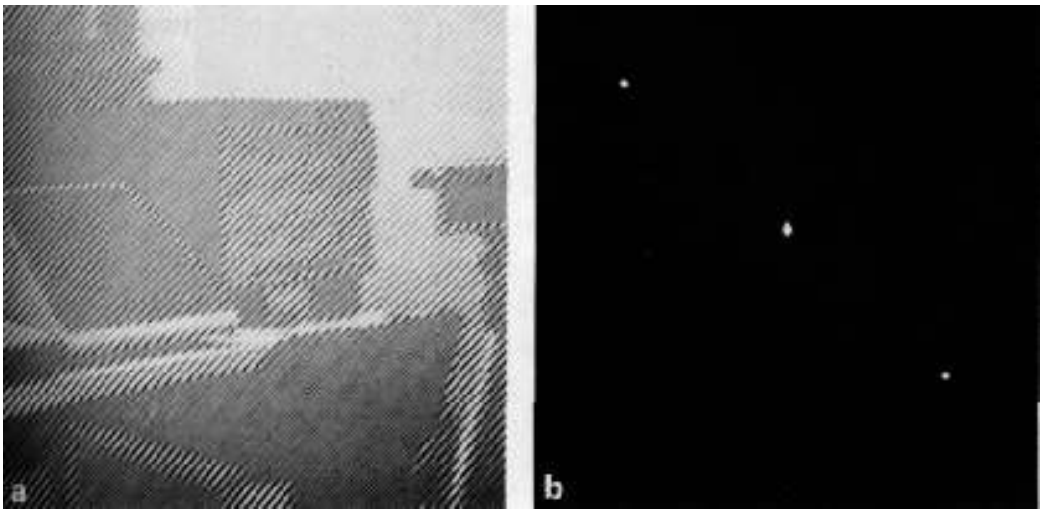
d) wie c), aber kontrastentzerrt

33.2.3 Bandsperre

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



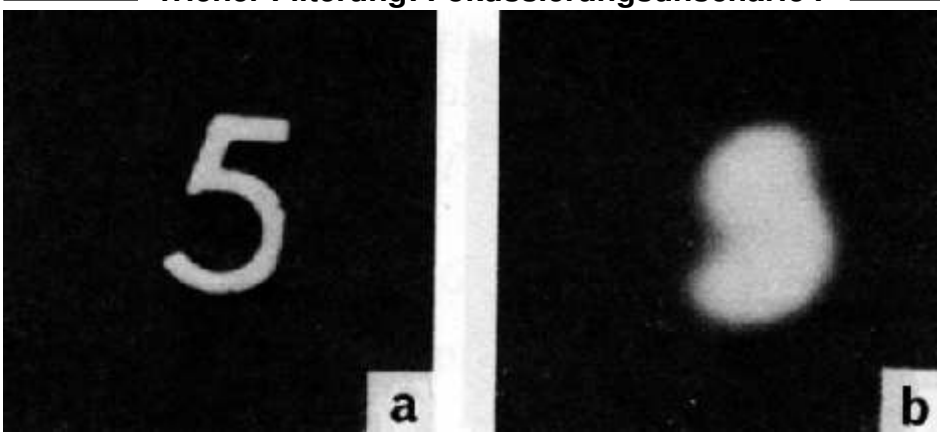
Bildrestauration durch Bandsperre

Eine Bandsperre unterdrückt die Störfrequenz.

Wiener Filterung: Bewegungsunschärfe

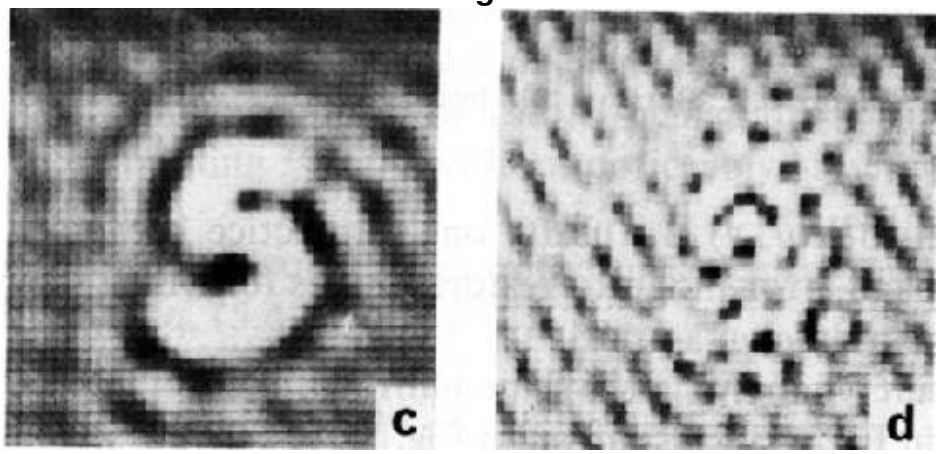
Originalbild und mit inverser Wiener Filterung restaurierte Version Version

Wiener Filterung: Fokussierungsunschärfe I



Originalbild und unscharfe Version

Fokussierungsunschärfe II



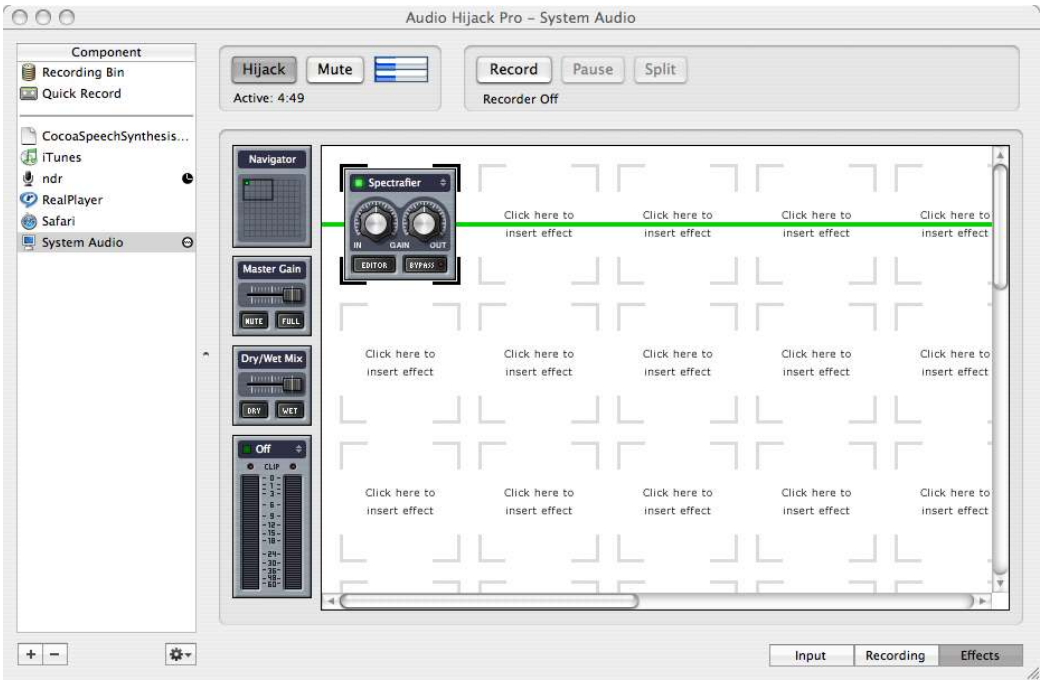
Restaurierte Bilder mit unterschiedlichen Modellen der Unschärfe

33.3 Audio-Filter

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

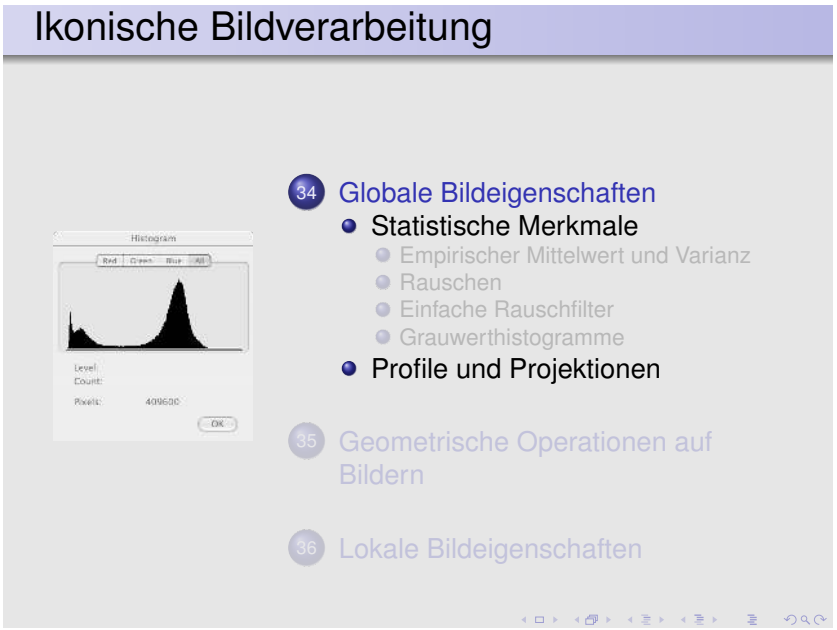


Teil XIII

Ikonische Algorithmen

34 Globale Bildeigenschaften

34.1 Statistische Merkmale



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

34.1.1 Empirischer Mittelwert und Varianz

Empirischer Mittelwert

Definition: 83 (***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: 84 (*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 $\bar{g}$

$$\begin{aligned} \bar{g}_0 &= 0 \\ \bar{g}_k &= \frac{\bar{g}_{k-1}(k-1) + g_k}{k}, k = 2 \cdots K \end{aligned}$$

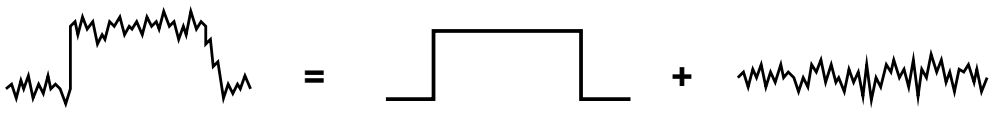
Empirische Varianz σ^2

$$\begin{aligned} \sigma_0^2 &= 0 \\ \sigma_k^2 &= \frac{(\sigma_{k-1}^2 + \bar{g}_{k-1}^2)(k-1) + g_k^2}{k} \\ &\quad -(\frac{\bar{g}_{k-1}(k-1) + g_k}{k})^2, k = 2 \cdots K \end{aligned}$$

34.1.2 Rauschen

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:

$$E[r_{ij,t_1} r_{ij,t_2}] = 0, \text{ für } t_1 \neq t_2$$

✎ **Anmerkung:** Neben dem weißen Rauschen gibt es viele weitere Modelle für das Rauschen - welches Rauschmodell das richtige ist, hängt von den Aufnahmebedingungen ab. Insbesondere kann das Rauschen multiplikativ von der Signalstärke abhängen oder ortsabhängig sein.

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.

Nachtaufnahme mit schlechtem Signal/Rausch-Verhältnis

Stark vergrößerter Ausschnitt

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.

$$\hat{r_k} = \frac{1}{k}\sum_{i=1}^{k} r_i \to 0$$

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 $t = 1 \cdots T$.

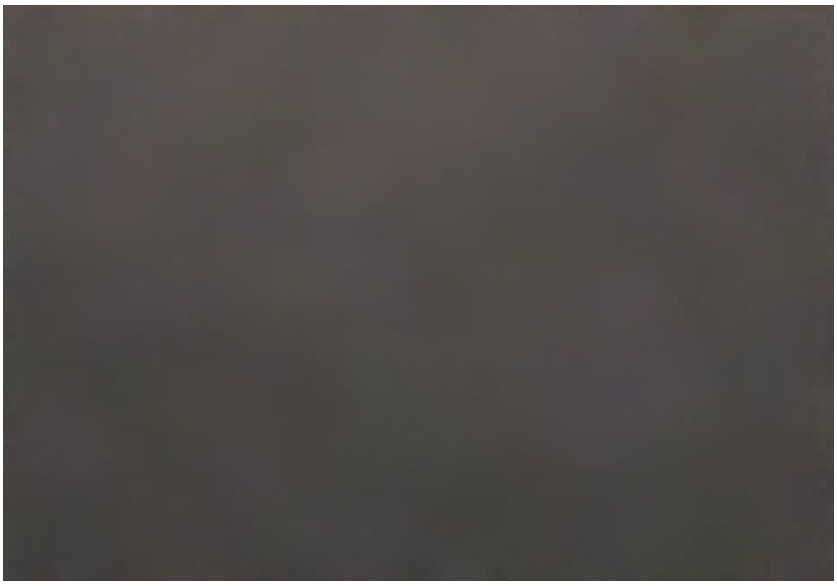
Unschärfe durch Mitteln

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 85 stark gemittelt (10×10).



Wie stark muß gemittelt werden?

Der geschätzte Rauschwert $\hat{r_k}$ nach k Mittelungschritten ist eine Zufallsvariable, deren Mittelwert und Varianz von k abhängt.

$$
\begin{aligned}
\hat{r_k} &= \frac{1}{k}\sum_{i=1}^{k} r_i \\
E[\hat{r_k}] &= \frac{1}{k}\sum_{i=1}^{k} E[r_i] = 0 \quad \text{Mittelwert} \\
E[(\hat{r_k} - E[\hat{r_k}])^2] &= E[\hat{r_k}^2] = E[\frac{1}{k^2}(\sum_{i=1}^{k} r_i)^2] \quad \text{Varianz} \\
&= \frac{1}{k^2}\sum_{i=1}^{k} E[r_k^2] = \frac{\sigma^2}{k}
\end{aligned}
$$

Beispiel

Um die Standardabweichung des Rauschens durch Mitteln zu halbieren, müssen vier Grauwerte gemittelt werden.

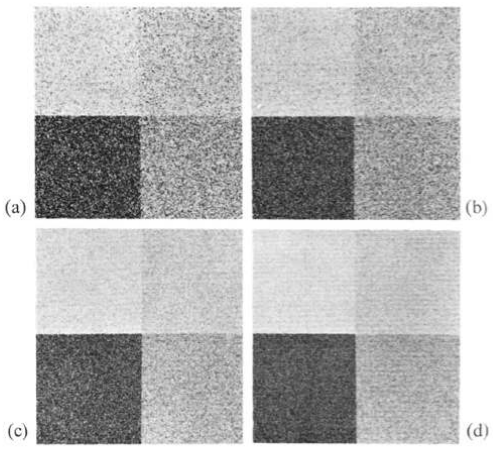


Figure 6.9. Smoothing by superposition of multiple copies. The numbers of copies in parts (a-d) are 2, 4, 8, and 16, respectively. [The pictures that were superimposed are versions of Figure 6.7(a).]

Die folgenden Folien in englischer Sprache sind aus dem Skript zur „Bildverarbeitung 1“ von Bernd Neumann entnommen.

Beispiel für Mittelung

Example of Averaging

intensity averaging with
5 x 5 mask

$$\frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

34.1.3 Einfache Rauschfilter

Glätten eines Bildes: Einfache Rauschfilter

Simple Smoothing Operations

1. Averaging

$$\hat{g}_{ij} = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$$

D is region around g_{ij}

Example of
3-by-3 region D

	ij	

2. Removal of outliers

$$\hat{g}_{ij} = \begin{cases} \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} & \text{if } \left| g_{ij} - \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} \right| \geq S \\ g_{ij} & \end{cases}$$

S is threshold

3. Weighted average

$$\hat{g}_{ij} = \frac{1}{\sum w_k} \sum_{g_k \in D} w_k g_k$$

w_k = weights in D

Example of weights
in 3-by-3 region

1	2	1
2	3	2
1	2	1

Note that these operations are heuristics and not well founded!

Bimodales Mitteln

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. $A = \{g_k \text{ with } g_k \geq \bar{g}_D\}$ $\quad B = \{g_k \text{ with } g_k < \bar{g}_D\}$

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
15	19	26

A

$\bar{g}_D = 16{,}7$ ➡ A, B ➡ $g_D' = 13$

Glätten mit rotierender Maske

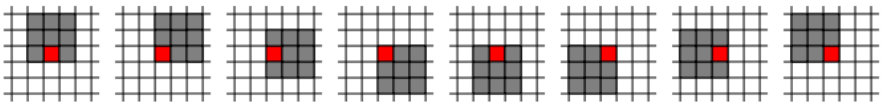
Averaging with Rotating Mask

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:

$$\bar{g}_{ij} = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn} \qquad \sigma_{ij}^2 = \frac{1}{|D|} \sum_{g_{mn} \in D} (g_{mn} - \bar{g}_{ij})^2$$

Possible rotated masks in 5 x 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}

Medianfilter

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
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.

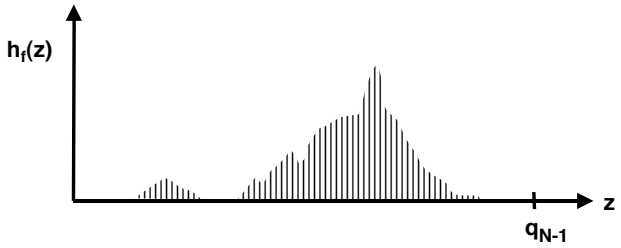
Expansion der hellen Bereiche: Wähle das Pixel mit dem höchsten Rang.

34.1.4 Grauwerthistogramme

Definition: 86 (*Grauwerthistogram*)
Ein Grauwerthistogram $h_f(z)$ eines Bildes f beschreibt die Häufigkeit der Grauwerte z im Bild.

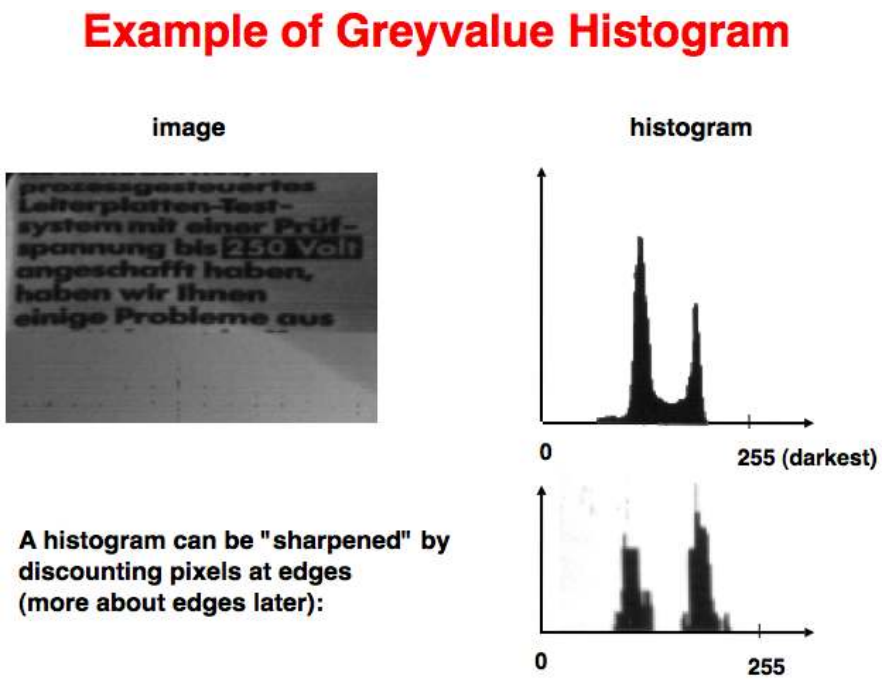
Sei f ein Bild mit N Quantisierungsstufen:

- ☞ *das Histogram ist dann ein Vektor (array) mit N Elementen.*



- ☞ Ein *Histogram* beschreibt *diskrete* Werte,
- ☞ eine *Verteilung kontinuierliche*.

Beispiel



Maskierte Histogramme

Definition: 87 (*Maskierte Histogramme*)
Maskierte Histogramme sind Histogramme über Bildpunkte, die ein Prädikat Q befriedigen:

$$h_a = |g_{ij} = a \wedge Q(g_{ij})|$$

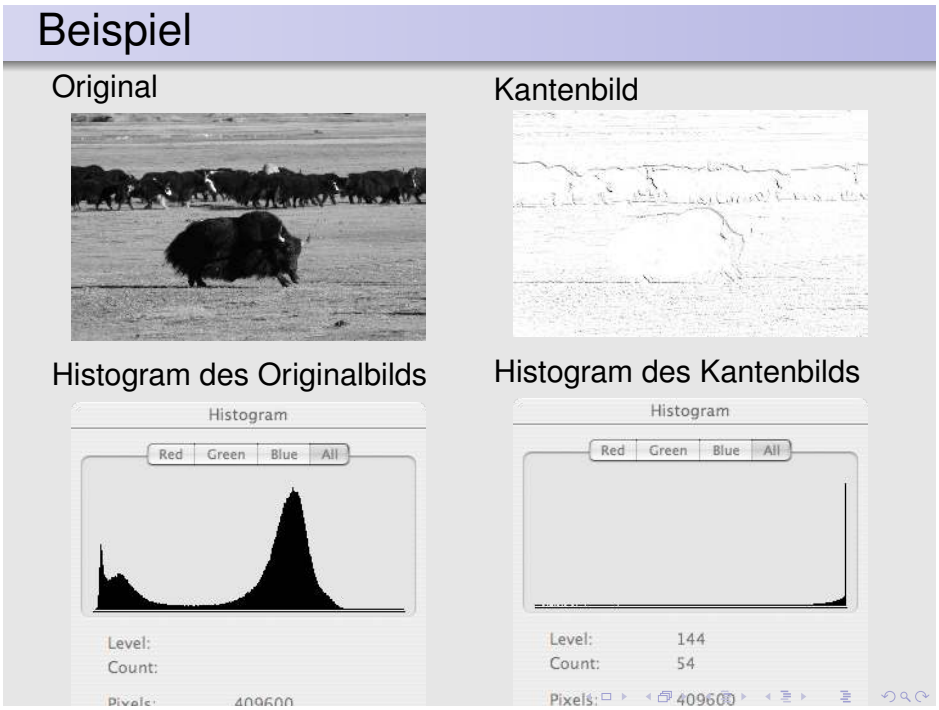
Beispiel: 88 („Histogram schärfen“:)
Es werden nur Pixel berücksichtigt, an denen der Grauwert wenig variiert.

$$Q(g_{ij}) = |g_{ij} - g_{i-1,j}| + |g_{ij} - g_{i,j-1}| < \delta$$

Histogramme abgeleiteter Größen:

Beispiel: 89
Gradientenhistogram: Zähle die Häufigkeit der Gradientenwerte:

$$h(|g_{ij} - g_{i-1,j}| + |g_{ij} - g_{i,j-1}|)$$



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 = 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

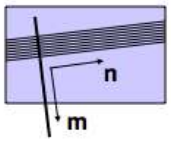
- Schwellenwertbestimmung
- Bereichszerlegung
- Quantisierung
- Bildwiedergabe

34.2 Profile und Projektionen

Projektionen

Projections

A projection of greyvalues in an image is the sum of all greyvalues orthogonal to a base line:

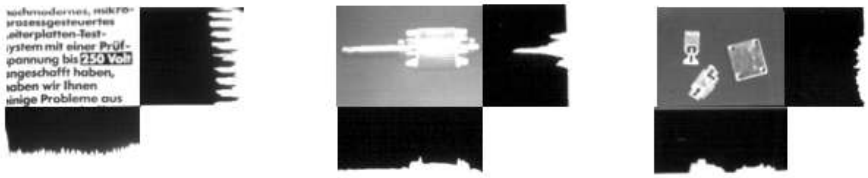


$$p_m = \sum_n g_{mn}$$

Often used:

"row profile" = row vector of all (normalized) column sums

"column profile" = column vector of all (normalized) row sums

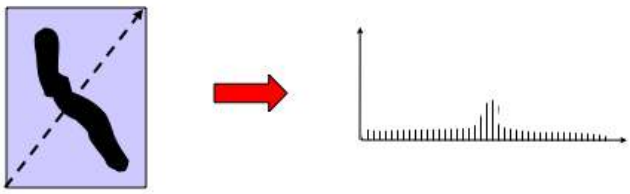


Querschnitt

Cross-sections

A cross-section of a greyvalue 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



35 Geometrische Operationen auf Bildern

35.1 Abstandsmaße für Bildpunkte

35.1.1 Diskrete Metriken

Diskrete Abstandsmaße

Ein Abstandsmaß $d(x, y)$ für zwei Punkte x und y sollte eine *Metrik* sein:

Positiv:

$$\begin{aligned} d(x,y) &\geqslant 0 \\ d(x,y) = 0 &\rightarrow x = y \end{aligned}$$

Symmetrisch:

$$d(x,y) = d(y,x)$$

Dreiecksungleichung:

$$d(x,y) + d(y,z) \geqslant d(x,z)$$

Abstandsmaße in Digitalbildern

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

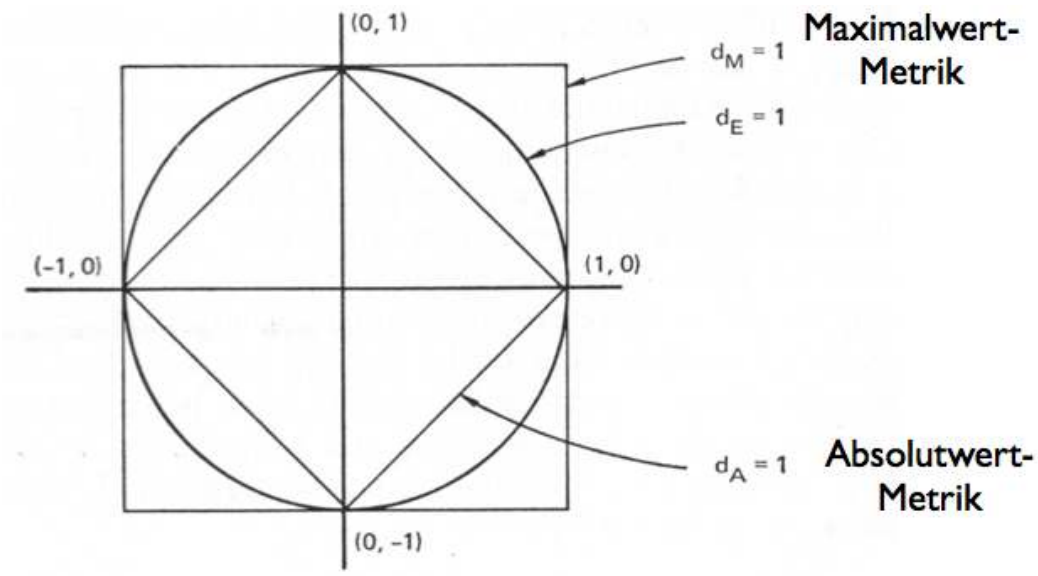
City block distance

$D_4((i,j)(h,k)) = |i-h| + |j-k|$ — 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.

35.1.2 Keine diskrete Metriken

Keine Metriken

Satz: 90 (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: 91 (Gerundeter Euklidischer Abstand:)
Der gerundete Euklidische Abstand $round(d_e(x,y))$ ist keine Metrik.

Beweis: 92
Gegenbeispiel:

$$\begin{aligned}
\textit{Sei} \quad & x=(0,0),\ y=(1,1),\ z=(2,2) \\
d(x,y) &= round(\sqrt{2}) = round(1.41421\dots) = 1 \\
d(y,z) &= round(\sqrt{2}) = round(1.41421\dots) = 1 \\
d(x,z) &= round(2\cdot\sqrt{2}) = round(2.82\dots) = 3 \\
d(x,y) &+ d(y,z) < d(x,z)
\end{aligned}$$

Die Dreiecksungleichung ist nicht erfüllt.

Satz: 93 (Quadrierter Euklidischer Abstand:)
Der quadrierte Euklidische Abstand $d_e(x,y)^2$ ist keine Metrik.

Beweis: 94
Gegenbeispiel:

$$\begin{aligned}
\textit{Sei} \quad & x=(1,0),\ y=(2,0),\ z=(3,0) \\
d(x,y) &= 1 \\
d(y,z) &= = 1 \\
d(x,z) &= 2\cdot 2 = 4 \\
d(x,y)+d(y,z) = 2 &< d(x,z)
\end{aligned}$$

Die Dreiecksungleichung ist nicht erfüllt.

Außerdem: Die Abstände sind nicht notwendigerweise ganze Zahlen.

Satz: 95 (Abgeschnittener Euklidischer Abstand:)
Der abgeschnittene Euklidische Abstand $\lfloor d_e(x,y) \rfloor$ ist keine Metrik.

Beweis: 96
Gegenbeispiel:

$$\begin{aligned} \textit{Sei} \quad & x = (0,0),\ y = (1,1),\ z = (3,4) \\ d(x,y) &= \lfloor \sqrt{2} \rfloor = \lfloor 1.41\ldots \rfloor = 1 \\ d(y,z) &= \lfloor \sqrt{13} \rfloor = \lfloor 3.6\ldots \rfloor = 3 \\ d(x,z) &= \lfloor \sqrt{25} \rfloor = 5 \\ d(x,y) + d(y,z) = 4 &< d(x,z) \end{aligned}$$

Die Dreiecksungleichung ist nicht erfüllt.

35.2 Geometrische Transformationen

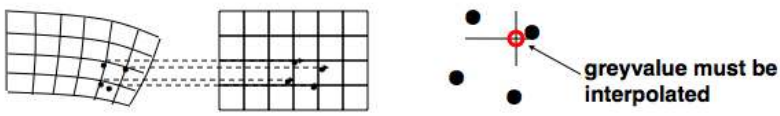
35.2.1 Polynomiale Koordinatentransformation

Geometrische Transformationen

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 ... 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:

$$x' = a_{00} + a_{10}x + a_{01}y + a_{11}xy + a_{20}x^2 + a_{02}y^2$$
$$y' = b_{00} + b_{10}x + b_{01}y + b_{11}xy + b_{20}x^2 + b_{02}y^2$$

at least 6 corresponding pairs needed

Approximation by affine transformation:

$$x' = a_{00} + a_{10}x + a_{01}y$$
$$y' = b_{00} + b_{10}x + b_{01}y$$

at least 3 corresponding pairs needed

Zur Erinnerung: Affine 2D-Transformationen

Translation, Rotation, Scaling, Skewing

Translation by vector $\underline{t}$:

$\underline{v}' = \underline{v} + \underline{t}$ with $\underline{v}' = \begin{bmatrix} x' \\ y' \end{bmatrix}$ $\underline{v} = \begin{bmatrix} x \\ y \end{bmatrix}$ $\underline{t} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$

Rotation of image coordinates by angle α:

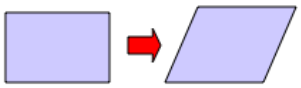
$\underline{v}' = R\,\underline{v}$ with $R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$

Scaling by factor a in x-direction and factor b in y-direction:

$\underline{v}' = S\,\underline{v}$ with $S = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$

Skewing by angle β:

$\underline{v}' = W\,\underline{v}$ with $W = \begin{bmatrix} 1 & \tan\beta \\ 0 & 1 \end{bmatrix}$



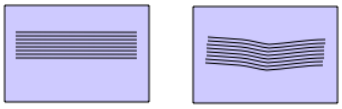
siehe hierzu auch das Kapitel über 2D-Transformationen, Kap. 7, Seite 97 und folgende.

Geomerische Korrektur von Verzeichnungen

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:

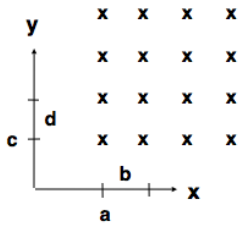
$x_{mn} = a + mb, \; y_{mn} = c + nd$

$m = 0 \ldots M-1$

$n = 0 \ldots N-1$

Actual mark positions:

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

Minimize $$E = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x_{mn} - x'_{mn})^2 + (y_{mn} - y'_{mn})^2$$

$$= \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (a + mb - x'_{mn})^2 + (c + nd - y'_{mn})^2$$

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:

$a = 1/2\,(x'_{00} + x'_{01})$

$b = 1/2\,(x'_{10} - x'_{00} + x'_{11} - x'_{01})$

$c = 1/2\,(y'_{00} + y'_{01})$

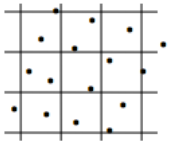
$d = 1/2\,(y'_{01} - y'_{00} + y'_{11} - y'_{10})$

35.2.2 Grauwertinterpolation

Grauwertinterpolation

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

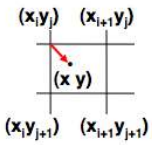
We will take view B:

Compute greyvalues between grid from greyvalues at grid locations.

Grauwertinterpolation: Nächster Nachbar

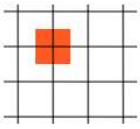
Nearest Neighbour Greyvalue Interpolation

Assign to (x y) greyvalue of nearest grid location

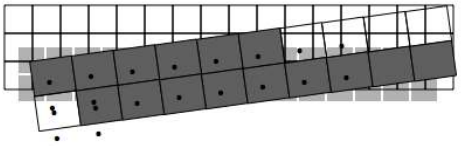


$(x_i y_j)$ $(x_{i+1} y_j)$ $(x_i y_{j+1})$ $(x_{i+1} y_{j+1})$	grid locations
(x y)	location between grid with $x_i \le x \le x_{i+1}$, $y_j \le y \le y_{j+1}$

Each grid location represents the greyvalues in a rectangle centered around this location:



Straight lines or edges may appear step-like after this transformation:



29

Bilineare Grauwertinterpolation

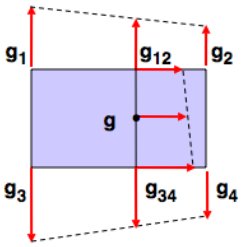
Bilinear Greyvalue Interpolation

The greyvalue at location (x y) between 4 grid points $(x_i y_j)$ $(x_{i+1} y_j)$ $(x_i y_{j+1})$ $(x_{i+1} y_{j+1})$ is computed by linear interpolation in both directions:

$$g(x,y) = \frac{1}{(x_{i+1} - x_i)(y_{j+1} - y_i)} \Big\{ (x_{i+1} - x)(y_{j+1} - y) g(x_i y_j) + (x - x_i)(y_{j+1} - y) g(x_{i+1} y_j) + (x_{i+1} - x)(y - y_j) g(x_i y_{j+1}) + (x - x_i)(y - y_j) g(x_{i+1} y_{j+1}) \Big\}$$

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}



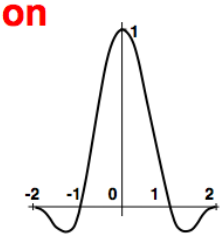
The step-like boundary effect is reduced.
But bilear interpolation may blur sharp edges.

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.

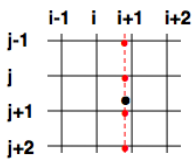
$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{for } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{for } 1 < |x| < 2 \\ 0 & \text{otherwise} \end{cases}$$



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] ... [u, j+2] (red dots), followed by
- 1 vertical interpolation (between red dots) to obtain greyvalue at [u, v].

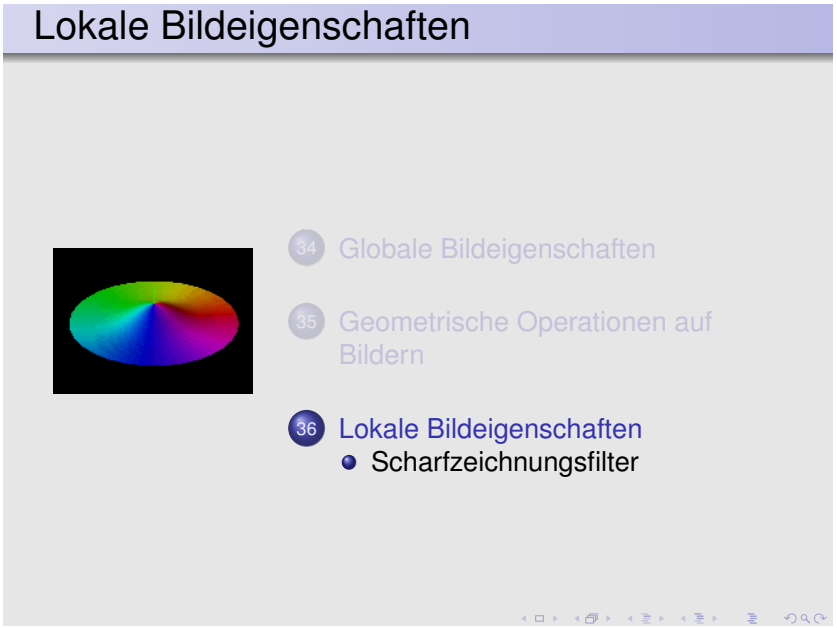


Note:
For an image with constant geyvalues g_0 the interpolated greyvalues at all points between the grid lines are also g_0.

31

36 Lokale Bildeigenschaften

36.1 Scharfzeichnungsfilter



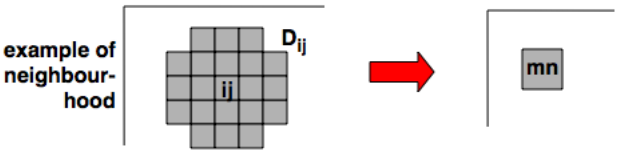
Lokale Bildoperatoren

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

$$\hat{g}_{mn} = f(g_1, g_2, \ldots, g_K) \qquad g_1, g_2, \ldots, g_K \in D_{ij}$$



Pixel indices i, j may be incremented by steps larger than 1 to obtain reduced new image.

Scharfzeichnungsfilter: Unscharfes Maskieren

intensity sharpening with 3 x 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

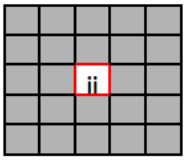
Heuristischer Zugang:
Effekt des Mittelns rückgängig machen

$$\text{Mitteln} \;: g'_{ij} = 1/M \sum_{g_k \in D_{ij}} g_k$$

$$\text{Schärfen} \;: g_{ij} = M \cdot g'_{ij} - \sum_{g_k \in \hat{D}_{ij}} g_k, \quad \hat{D}_{ij} = D_{ij} - \{g_{ij}\}$$

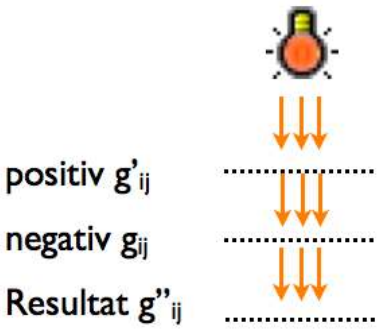
$$\text{geschätzt} : \approx M \cdot g'_{ij} - \sum_{g'_k \in \hat{D}_{ij}} g'_k$$

$\hat{D}_{ij}$, M=25



Fotografische Lösung

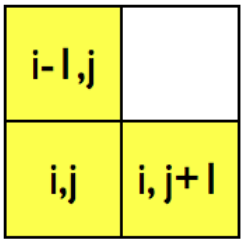
„Unscharfes Maskieren“:
Beim Herstellen von Positiv-Abzügen ein unscharfes Negativbild in den Lichtstrahl einbringen und damit das scharfe Negativ belichten:

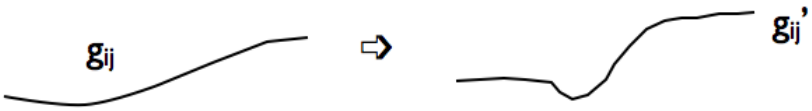


$$g''_{ij} \approx g_{ij} - g'_{ij}$$
$$= g_{ij} - 1/M \sum_{g_k \in \hat{D}_{ij}} g_k$$

Scharfzeichnen: Kanten überhöhen

- **Kanten überhöhen:**
  - einfacher Kantenoperator mißt Kantenhöhe:
    $dg = g_{ij} - g_{i-1j} + g_{ij} - g_{ij+1}$
    $g_{ij}' = g_{ij} + \alpha dg$

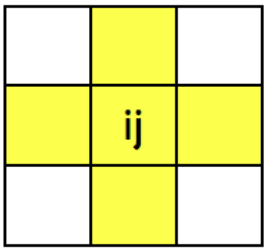


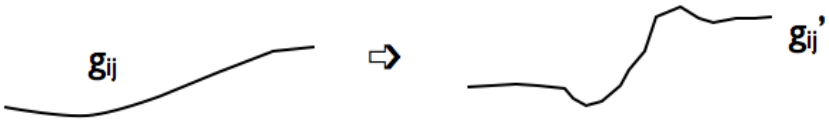


Symmetrische Lösung: Laplace-Operator

- Symmetrische Lösung:

  $dg = 4g_{ij} - g_{i-1j} - g_{ij-1} - g_{ij+1}$
  $g_{ij}' = g_{ij} + \alpha dg$





IVC Interactive Visual Computing: Computer Vision

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 6:

Binarisieren

Universität Hamburg
Fachbereich Informatik

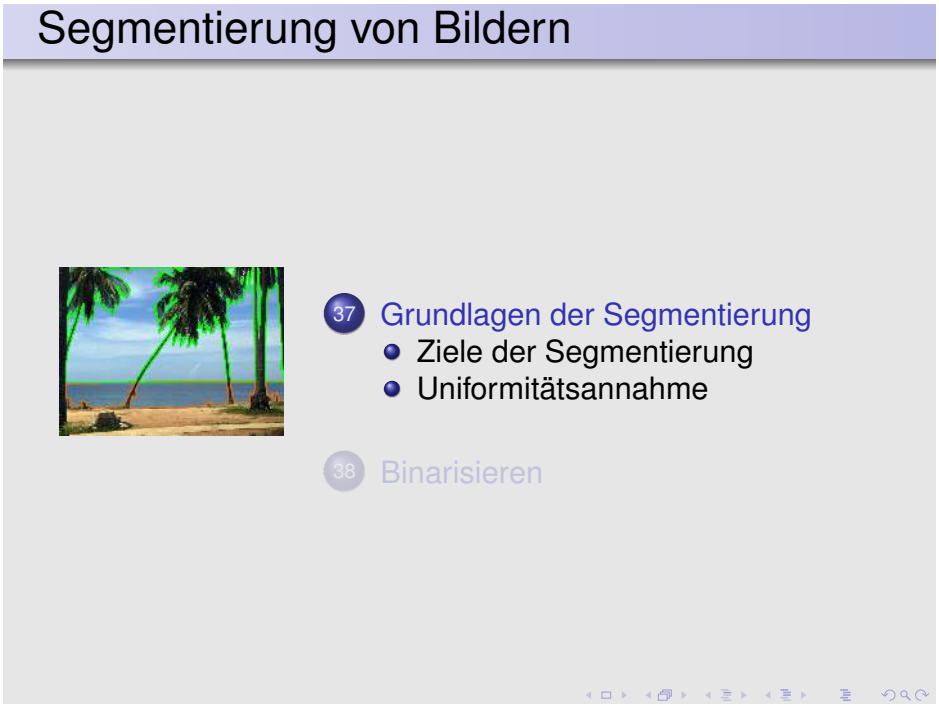
WS 2014/2015

Teil XIV

Binarisieren

37 Grundlagen der Segmentierung

37.1 Ziele der Segmentierung



Segmentierung

Definition: 97 (*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

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

Weiterverabeitende 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:

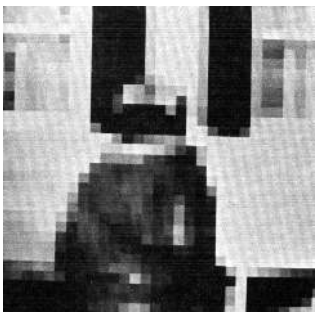
- Detailbeschreibung,
- Stereoskopie,
- Bewegungsanalyse

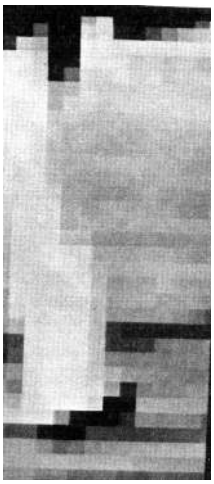
Datenreduktion

- 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





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 Kontexwissen erforderlich.

37.2 Uniformitätsannahme

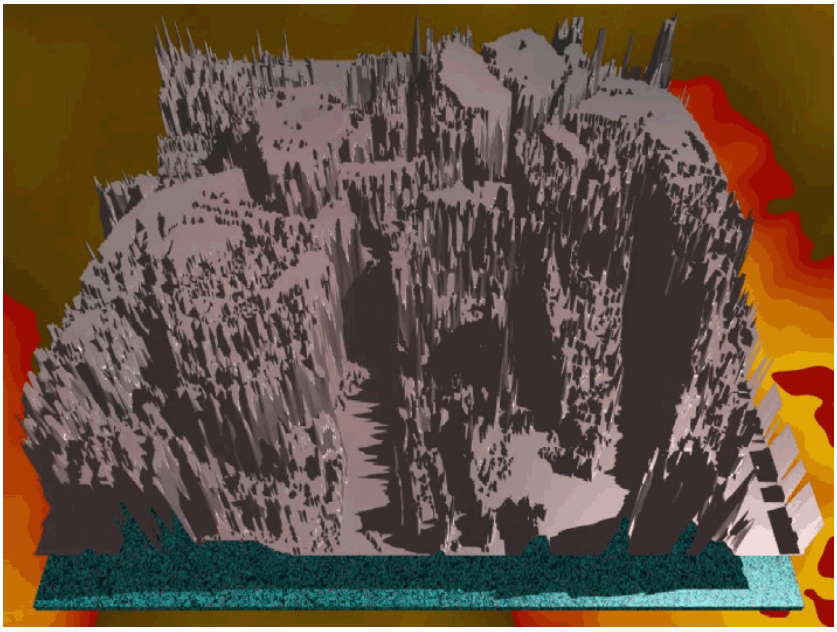
Uniformitätsannahme

Uniformitätsannahme

Den meisten Segmentierungsverfahren liegt eine Uniformitätsannahme für die Bildfunktion zugrunde:

- Objekten entsprechen Bildbereiche, für die ein Uniformitätsprädikat gilt → *Bereichszerlegung*
- Duale Aussage: Objektgrenzen ensprechen 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

38 Binarisieren

Binarisieren

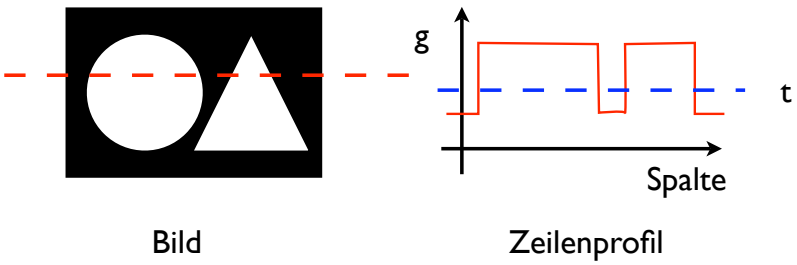
Definition: 98 (*Binarisieren*)
Binarisierung ist die Transformation einer Bildfunktion in ein Binärbild.

Für viele Anwendungen ist es ausreichend, nur zwischen *zwei* Grauwerten zu unterscheiden:

- „SCHWARZ“ und „WEISS“
- oder „1 “ und „0“.

38.1 Voraussetzungen für Schwellenwertverfahren

Vorgehen: Schwellenwertverfahren



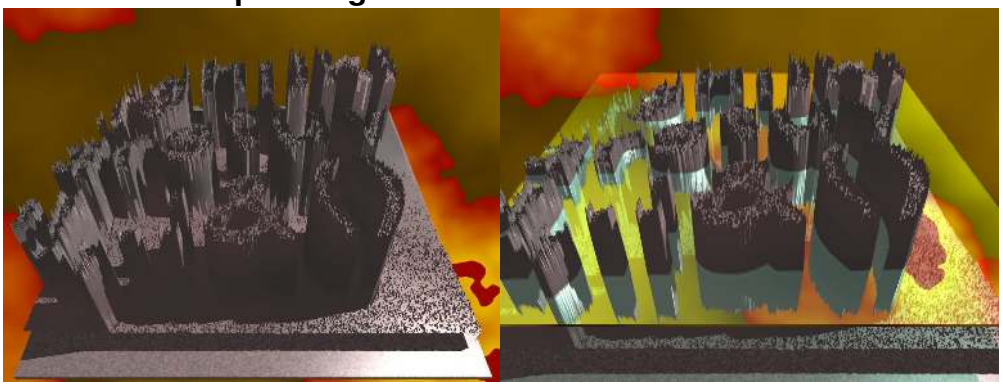
Schwellenwertverfahren

Der Grauwert $g(x, y)$ wird auf „0“ abgebildet, wenn er kleiner als eine Schwellenwert t ist, sonst auf „1“.

$$g^t(x, y) = \begin{cases} 0, & g(x, y) < t \\ 1, & g(x, y) \geqslant t \end{cases}$$

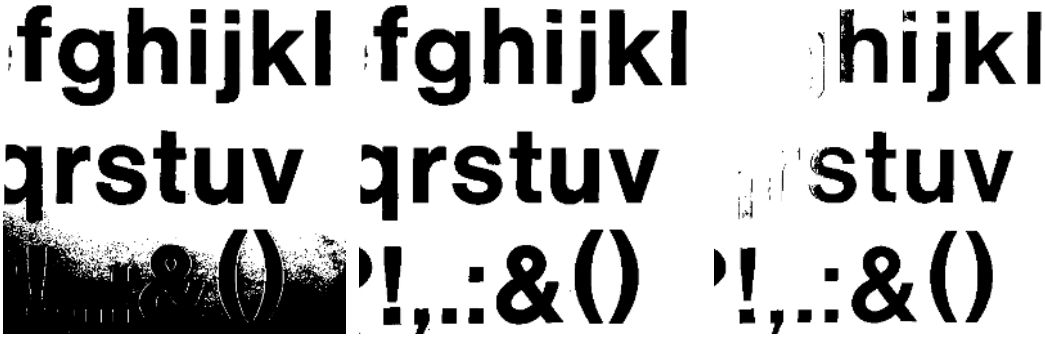
Beispiel: Segmentieren von Schriftzeichen

Beispiel: Segmentieren von Schriftzeichen



Grauwertgebirge und Schwellenwert

Beispiel: Segmentieren von Schriftzeichen



t zu niedrig t optimal t zu hoch

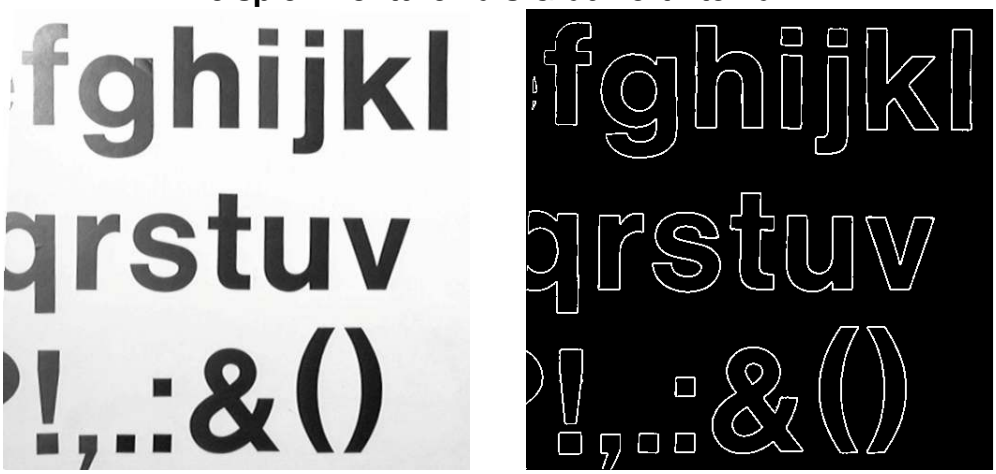
(Negativ vom vorherigen Bild für bessere Druckbarkeit)

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“.

$$g^{u,v}(x, y) = \begin{cases} 1, & u \geqslant g(x, y) \geqslant v \\ 0, & sonst \end{cases}$$

Beispiel: Konturen als Grauwertintervall



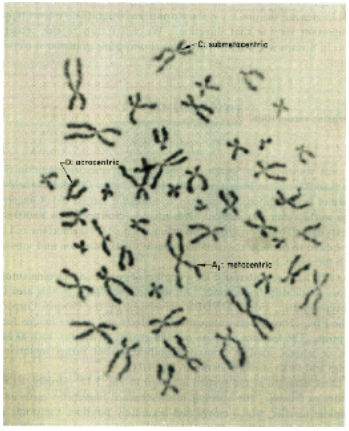
Originalbild Konturen

Allgemein

☞ Schwellenwertverfahren sind anwendbar, um Objekt und Hintergrund zu trennen,

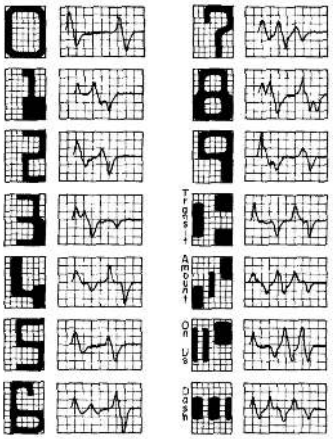
- wenn die Grauwerte im Objekt einen charakteristischen Wert haben,
- oder wenn eine Funktion der Grauwerte im Objekt einen charakteristischen Wert hat.

Beispiele für geeignete Anwendungsszenarien



Chromosomen

Fingerabdrücke



OCR-Schrift

Vorgehen: Charakeristische Funktion der Grauwerte

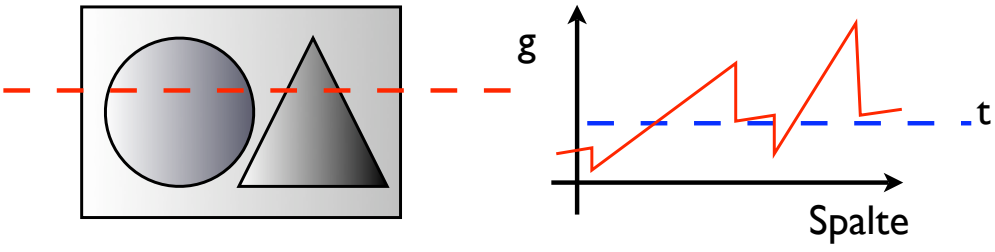
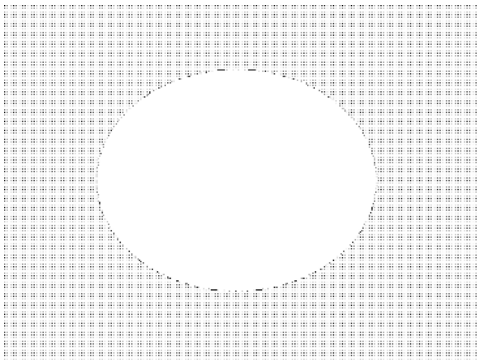
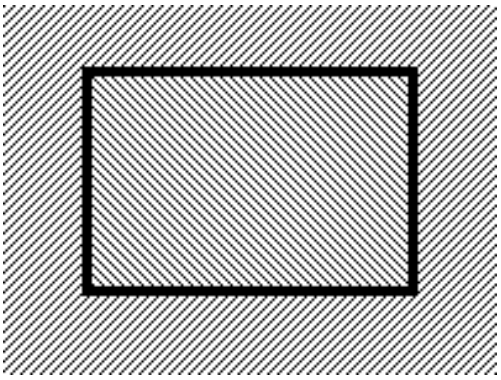


Bild Zeilenprofil

In diesem Beispiel ist der Gradient der Grauwerte charakteristisch für Objekt und Hintergrund.

Segmentieren von texturierten Bildern





Mögliche Homogenitätskriterien:

- Mittlerer Grauwert
- Richtung der Linien

38.2 Schwellenwertwahl

Verfahren zur Schwellenwertwahl

Manuell: systematisches Probieren, eventuell sehr zeitaufwendig

2-Klassen-Klassifikationsproblem: Klassifiziere 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.

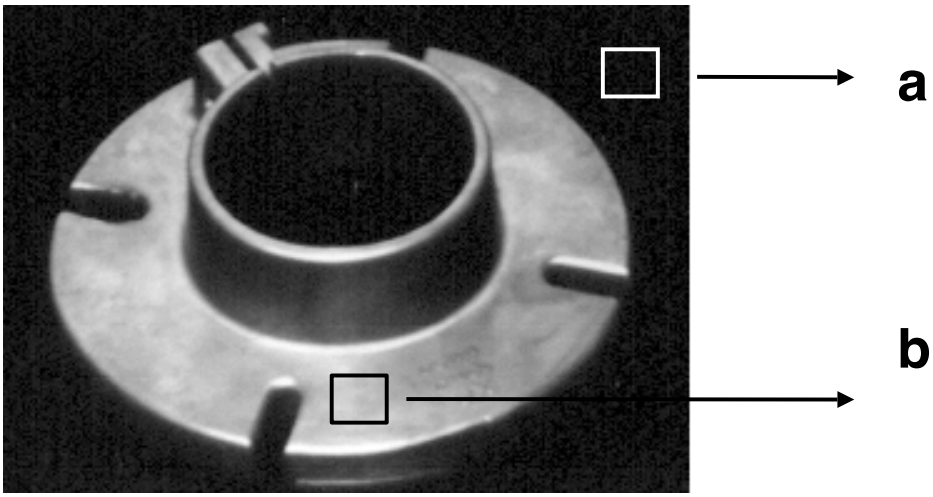
Vorgehen: Variiere systematisch den Schwellenwert (beispielsweise mit binärer Suche), bis das resulierende Binärbild die Kriteriumsfunktion erfüllt.

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:

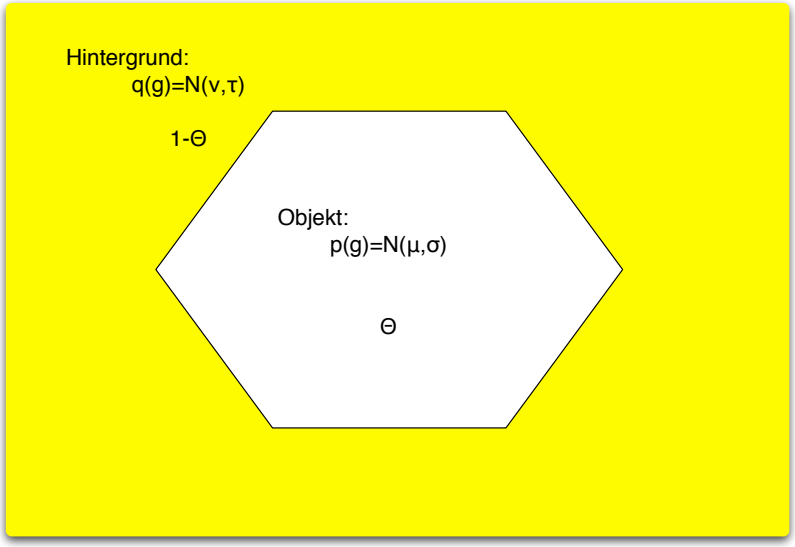
$$t = \frac{a+b}{2}$$

Beispiel: Teile auf dem Fließband in bekannter Position



Verteilungsbasierte Schwellenwertwahl

Annahme: die Dichtefunktionen der Grauwerte im Objektund 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.

$$p(g) = N(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(g-\mu)^2}{2\sigma^2}}$$

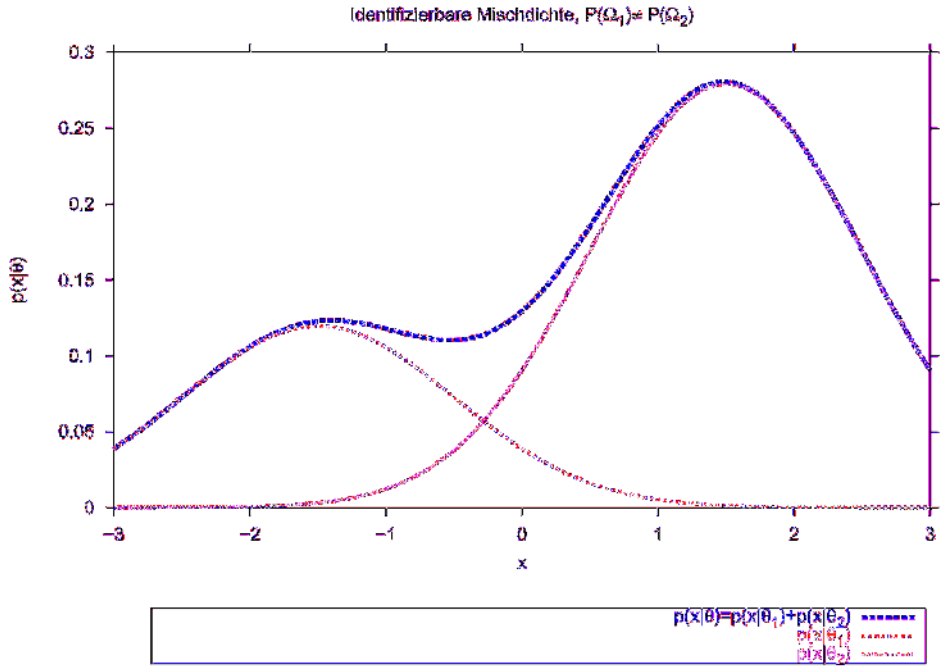
Wahrscheinlichkeitsdichte der Hintergrundgrauwerte: Normalverteilt mit Mittelwert ν und Varianz τ. $\mu < \nu$

$$q(g) = N(\nu, \tau) = \frac{1}{\sqrt{2\pi\tau}} e^{\frac{-(g-\nu)^2}{2\tau^2}}$$

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

$$(1 - \theta) \cdot q(t) = \theta \cdot p(t)$$

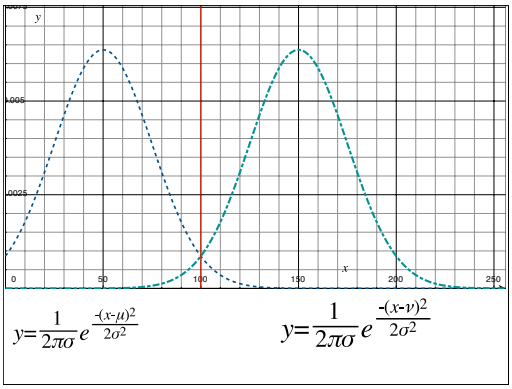
$$
\begin{aligned}
(1-\theta)\cdot q(t) &= \theta \cdot p(t) \\
p(t) &= \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \quad \text{Objekt} \\
q(t) &= \frac{1}{\sqrt{2\pi\tau}} e^{\frac{-(t-\nu)^2}{2\tau^2}} \quad \text{Hintergrund}
\end{aligned}
$$

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: $\theta = \frac{1}{2}, \tau = \sigma$

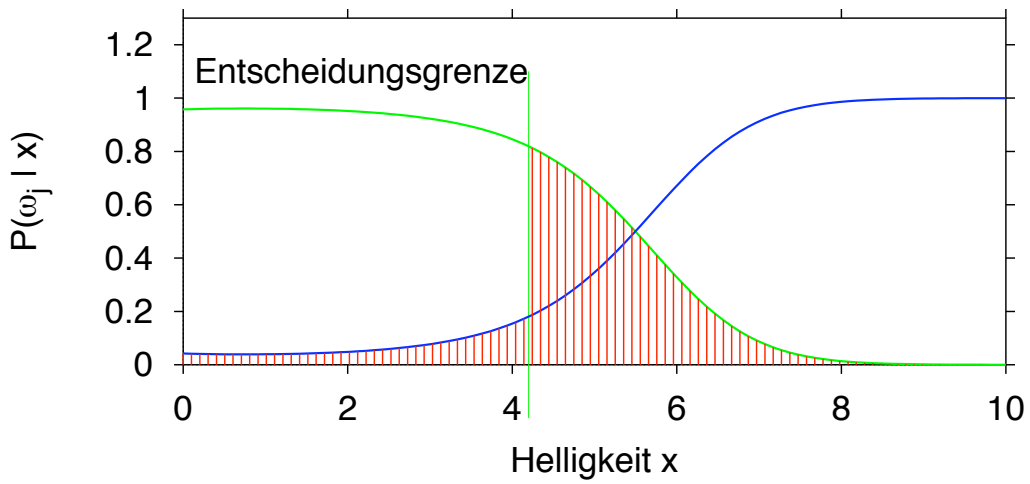
$$
\begin{aligned}
(t-\mu)^2 &= (t-\nu)^2 \\
t &= \frac{\mu+\nu}{2}
\end{aligned}
$$



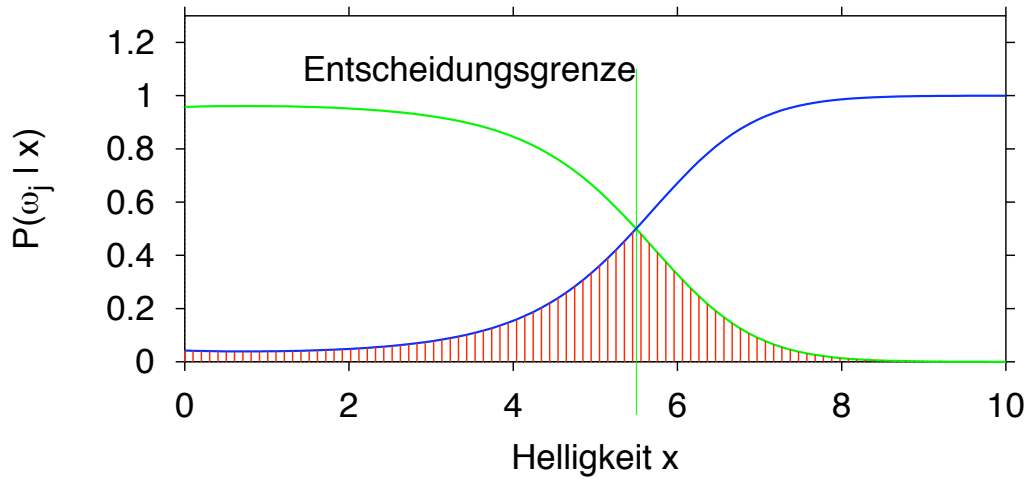
Beispiel: $\mu = 50, \nu = 150, \sigma = \tau = 20, t = 100$

✎ **Anmerkung:** Der Spezialfall nimmt an, daß Objekt und Hintergrund eine gleich große Fläche im Bild einnehmen und daß die Grauwerte im Objekt und Hintergrund in gleicher Weise um den Mittelwert variieren. Das ist eine sinnvolle Annahme, wenn das Rauschen im Bild nicht signalabhängig ist.

Komponenten der Fehlerwahrscheinlichkeit



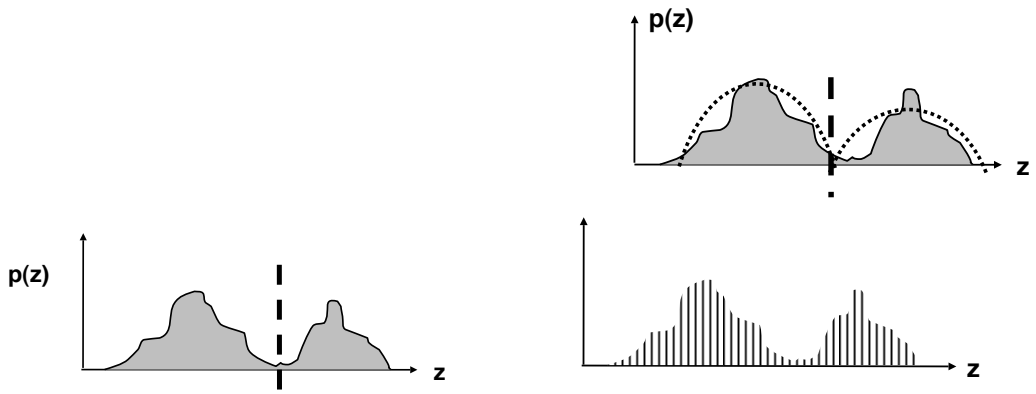
Optimale Schwelle im Schnittpunkt der Dichten



✎ **Anmerkung:** Komponenten der Fehlerwahrscheinlichkeit: Die Folie zeigt die Komponenten, die zum Klassifikationsfehler Objekt-Hintergrund bei falscher Schwellenwertwahl beitragen: Die beiden Flächen unter den Kurven der a posteriori Wahrscheinlichkeiten der beiden Klassen „Objekt“ und „Hintergrund“von $-\infty$ bis zum Schwellenwert für die eine Klasse und von $+\infty$ bis zum Schwellenwert für die zweite Klasse. Dieses ist gerade diejenige Fläche, die von der X-Achse und den Kurven der beiden a posterio Wahrscheinlichkeiten begrenzt wird plus der Fläche, die von den Kurven der a posterio Wahrscheinlichkeiten und dem Schwellenwert t begrenzt wird. *Nur dieser zweite Anteil hängt davon ab, wo die Schwelle liegt.* Dieser zweite Anteil ist gleich 0, wenn die Schwelle entsprechend der Bayesschen Entscheidungsregel genau im Schnittpunkt der beiden Kurven liegt. Kleiner kann die Fehlerwahrscheinlichkeit nicht werden.

Schätzung der Dichten über Grauwerthistogramme

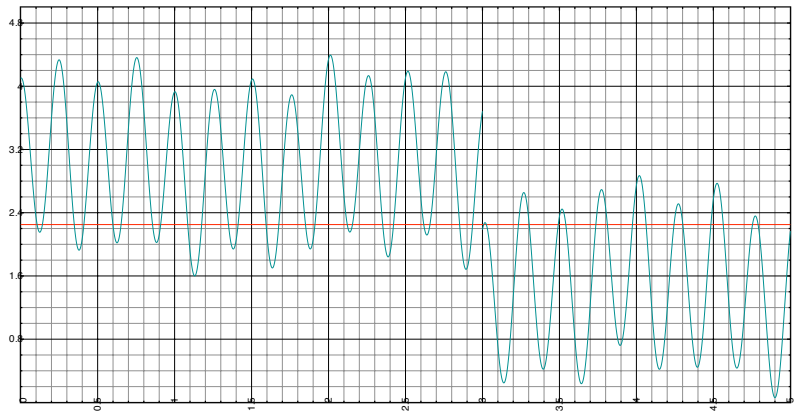
Heuristik: Suche das Tal zwischen den beiden Maxima der Grauwertdichte



Histogramme geben eine diskrete Approximation der Grauwertdichte.

38.3 Bei stark verrauschten Bildern

Zeilenprofil eines stark verrauschten Bildes



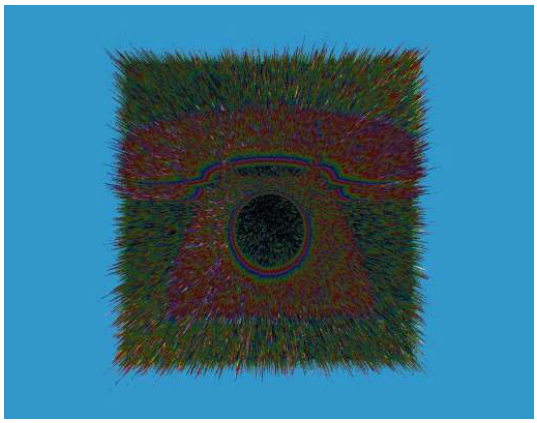
Bei stark verrauschten Bildern

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

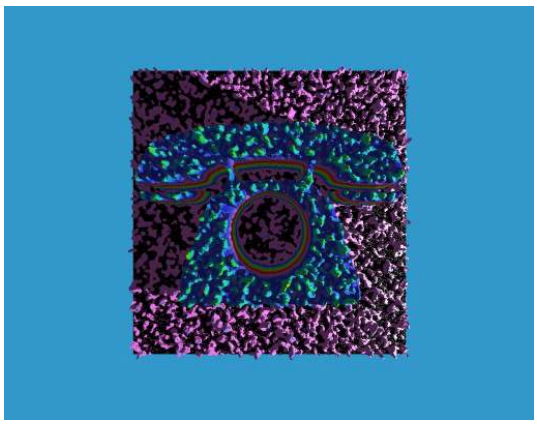
Verrauschtes Bild



Grauwertgebirge

Beispiel: Das Bild mit Medianfilter, Rang 5

Geglättetes Bild



Grauwertgebirge

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

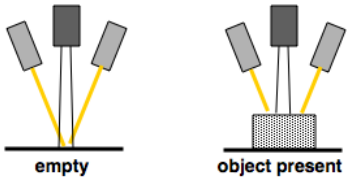
Image Capturing for Thresholding

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.

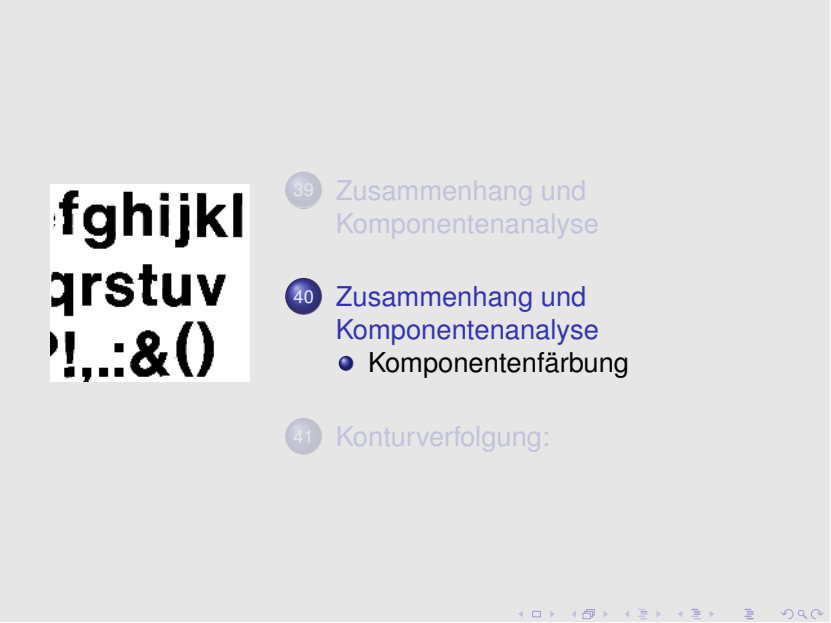


Teil XV

Komponentenanalyse

39 Zusammenhang und Komponentenanalyse

39.1 Zusammenhangsrelationen

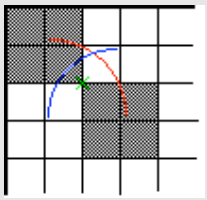


Zusammenhang

Definition: 99 (*Zusammenhang*)
Der Zusammenhang ist eine Äquivalenzrelation Z zwischen Pixeln P mit den folgenden Eigenschaften:

$$
\begin{aligned}
& Z \subset P \times P \\
\textit{Reflexiv} \quad & \forall_{x \in P} Z(x,x) \\
\textit{Symmetrisch} \quad & \forall_{x,y \in P} Z(x,y) \rightarrow Z(y,x) \\
\textit{Transitiv} \quad & \forall_{x,y,z \in P} Z(x,y) \wedge Z(y,z) \rightarrow Z(x,z)
\end{aligned}
$$

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.

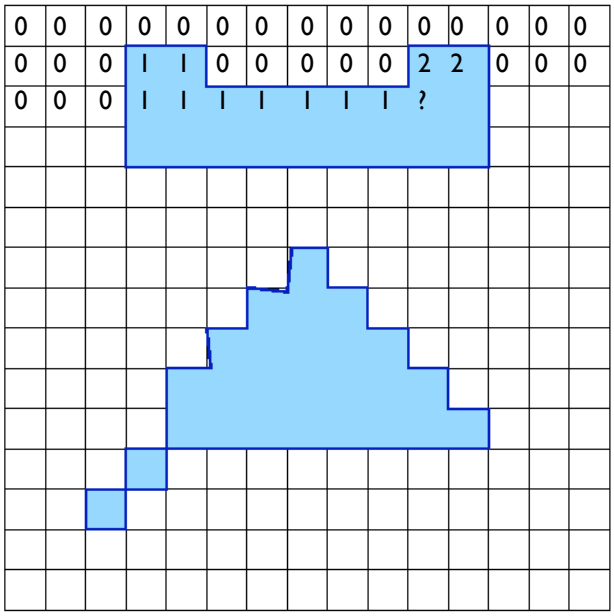
✎ **Anmerkung:** Die Frage des Zusammenhang ist wichtig für Algorithmen, die Objekte in Bildern isolieren und erkennen sollen. Objekte werden oft daran erkannt, daß sie im Bild als zusammenhängende Bildstrukturen abgebildet werden. Widersprüchliche Zusammenhangsrelationen können dann zu widersprüchlichen Aussagen über im Bild enthaltene Objekte führen.

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

39.2 Komponentenfärbung

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.

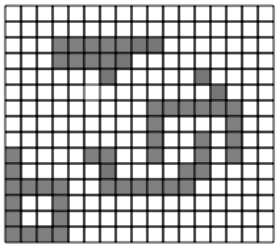
Die Trägheitsmomente: Summiere die Quadrate der Zeilen- und Spaltenkoordinaten der Komponenten.

Rauschunterdrückung

- Komponenten mit sehr kleiner Fläche gehen vermutlich auf Rauschen zurück.
- Diese können dem Hintergrund zugeordnet werden.

Component Labelling

Determining connected regions in B/W images (line fromColumn toColumn)



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)

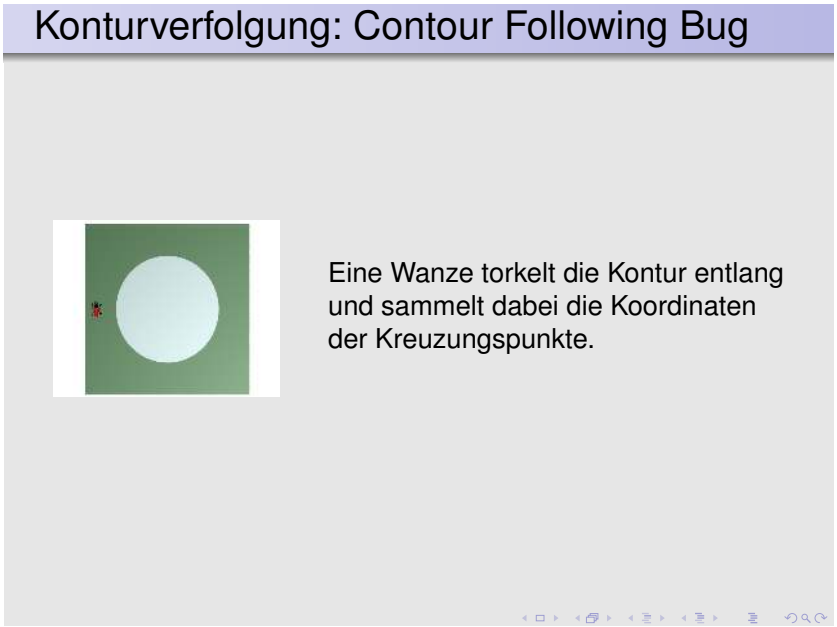
In this example: component descriptions using run-length coding

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

9

40 Konturverfolgung:



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movcontour-bug.html

Konturverfolgung

Verfahren bei kontinuierlichen Bildfunktionen:

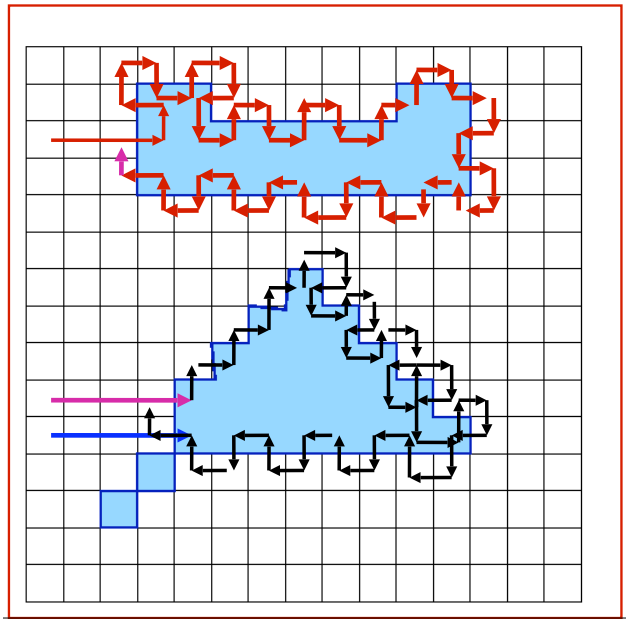
- Gehe nach rechts, bis die Kontur überschritten wird.
- In der Figur drehe entgegen dem Uhrzeigersinn.
- Außerhalb der Figur drehe im Uhrzeigersinn.
- Beim Überqueren der Kontur notiere den Kreuzungspunkt.

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 gekrümmt ist als der Wendekreis der Wanze, wird sie falsch abgetastet.

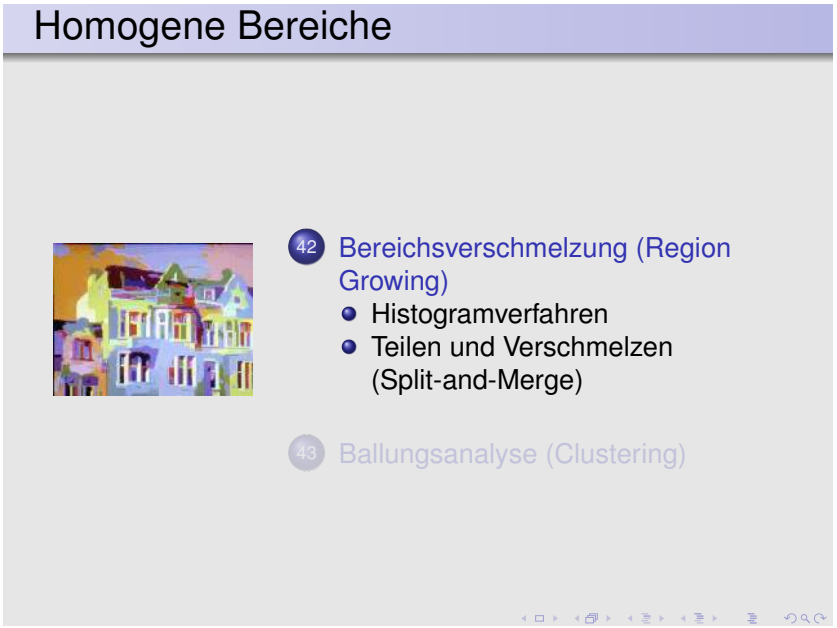
Löcher im Objekt müssen separat untersucht werden.

Mehrdeutig bzgl. 4er- und 8er-Zusammenhang.

Teil XVI

Bereichsfinder

41 Bereichsverschmelzung (Region Growing)



Bereichsverschmelzer (Region Growing)

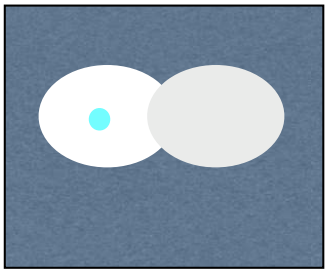
Kleine Saatbereichen können zu größeren Bereichen verschmolzen werden, wenn diese ein *Homogenitätskriterium* erfüllen.

Homogenitätskriterien

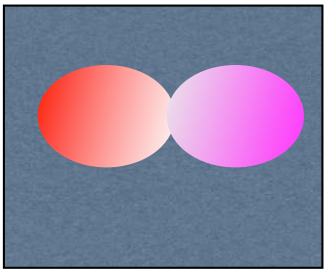
① Verschmelze zwei benachbarte Bereiche, wenn der *verschmolzene Bereich* ein Homogenitätskriterium erfüllt, beispielsweise die Varianz der Grauwerte klein genug ist.

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



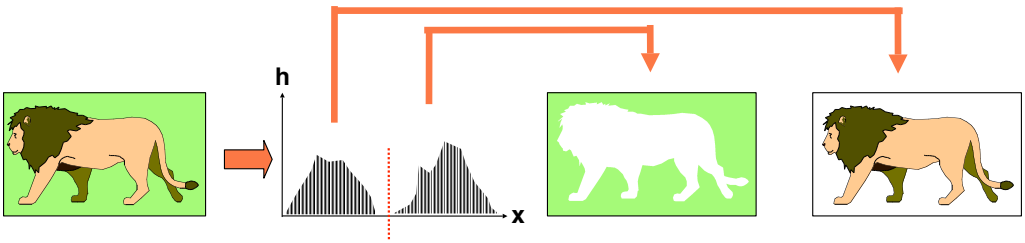
Schwacher Kontrast

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.

41.1 Histogramverfahren

Rekursive Zerlegung mit Histogrammen



- Berechne Histogramme für verschiedenen Pixelmerkmale (z. B: R, G, B, Intensitäts-Histogramme).
- Wähle das klarste Histogram für eine Binarisierung.
- Wende das Verfahren rekursiv auf die Teilbereiche an.

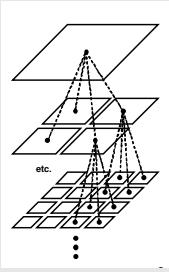
Schwierigkeiten der Histogramzerlegung

- 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.

41.2 Teilen und Verschmelzen (Split-and-Merge)

Split-and-Merge

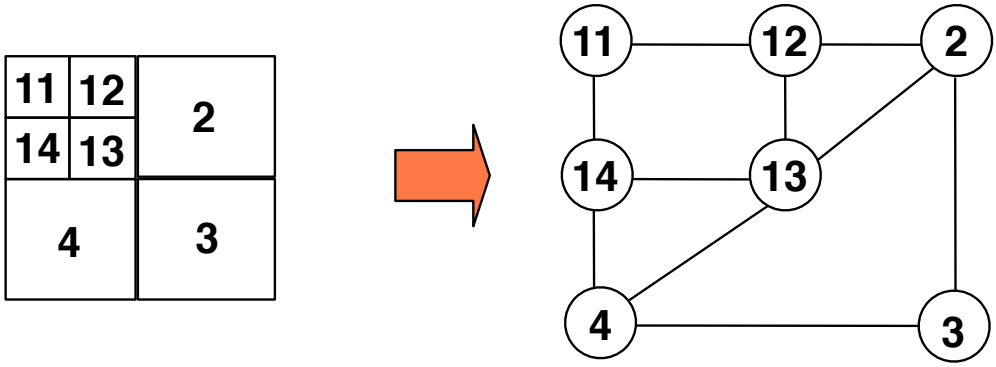
Bereichsgrenzen werden entlang von quadtree-Grenzen festgelegt.



- 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.



42 Ballungsanalyse (Clustering)

42.1 Segmentierung als Klassifikationsproblem

42.1.1 Der Merkmalsraum

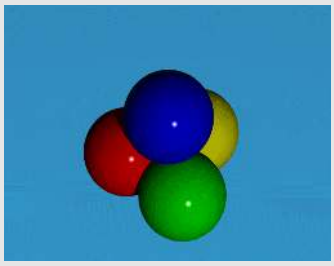
Ballungsanalyse (Clustering)

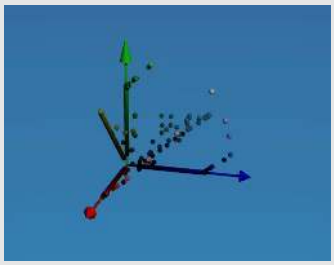
42 Bereichsverschmelzung (Region Growing)

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.

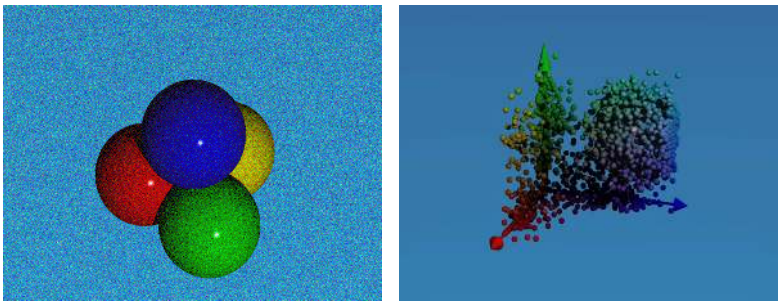




Ein Farbbild und die Pixel im RGB-Merkmalsraum

✎ **Anmerkung:** Ansatz: Die Proben, die zu einer Klasse gehören, bilden Ballungen im Merkmalsraum. Clustering-Verfahren zerlegen den Merkmalsraum in Bereiche, die solchen Ballungen entsprechen. Einige Ballungsanalysealgorithmen sind auch dann anwendbar, wenn die Form der Ballungen nicht bekannt ist oder wenn die Zahl der gesuchten Klassen unbekannt ist.

Rauschen vergrößert die Ballungen



Ein verauschtes Bild und die Pixel im RGB-Merkmalsraum

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movporscheNoisePatSpaceQT.html

Teil XVII

Kantenfinder

43 Objektgrenzen und Kanten

43.1 Entstehung von Kanten

Kantenfinder

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.

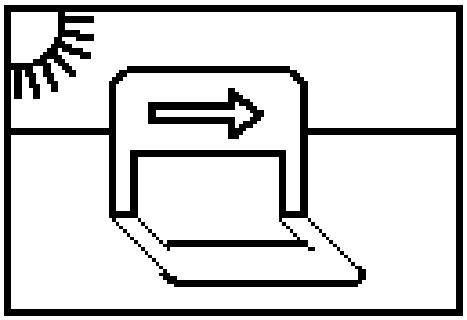
Vorsicht!

Nicht alle Diskontinuitäten im Bild gehen auf Materialgrenzen zurück!

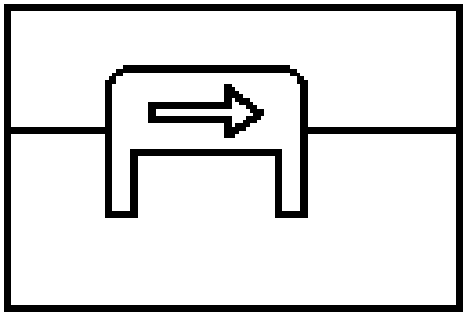
Weitere Ursachen für Grauwertdiskontinuitäten:

- Änderung der Oberflächeneigenschaften (Farbe, Material, Glattheit)
- Änderung der Oberflächenorientierung zum Betrachter
- Änderung der Beleuchtung (Schatten, Sekundärbeleuchtung)
- Rauschen, Quantisierung

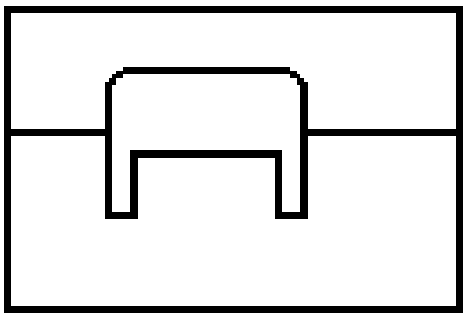
Entstehung von Kanten



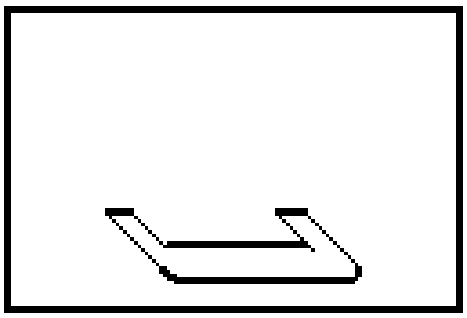
Ein Bild



Reflektanzkanten



Orientierungskanten



Beleuchtungskanten

Kontrastumkehr
Kanten in Realweltbildern

©Wolfgang Förstner

Der Kontrast entlang einer Kante kann abhängig von der Beleuchtung stark variieren und sich gelegentlich auch umkehren: siehe die senkrechte Kante in Bildmitte.

Ansätze zur kantenbasierten Segmentierung

Definition: 100 (***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)

44 Gradientenbasierte Verfahren

44.1 Stufenkanten

Vorgehensweise: Ableitungsorientiert

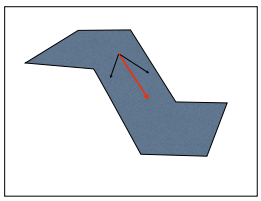
Pixel mit starker Variation der Bildfunktion erkennen wir an hohen Werten der *ersten Ableitung* der Bildfunktion.

Vorgehensweise zur Kantendetektion:

- Berechne den Gradienten der Bildfunktion.
- Suche lokale Maxima des Gradientenbetrages.
- Die so gefundenen Bildpunkte sind Kandidaten für Kantenelemente (*edgel*).

Zwischen den Objektkonturen im Bild und dem Hintergrund variiert die Bildhelligkeit stark:

Wir finden solche Helligkeitssprünge (Kanten), indem wir die Bildfunktion differenzieren und Maxima der ersten Ableitung suchen. Da das Bild eine Funktion in zwei Variablen ist, berechnen wir den Gradienten der Bildfunktion, indem wir den Nabla-Operator darauf anwenden. Wir erhalten als Resultat einen Vektor, der in Richtung des stärksten Grauwertgefälles zeigt und dessen Betrag die Stärke des Grauwertgefälles angibt. Dort, wo der Gradientenbetrag am größten ist, finden wir die steilsten Kanten im Bild.



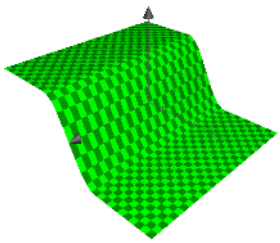
Gegeben: Eine Bildfunktion $g(x, y)$.

$$\text{Sei } \vec{\nabla}_{(x,y)} = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \quad \text{der Gradientenoperator.}$$

$$||\vec{\nabla}_{(x,y)}\ g(x, y)|| = \text{ein Maß für die Variation von } g(x, y).$$

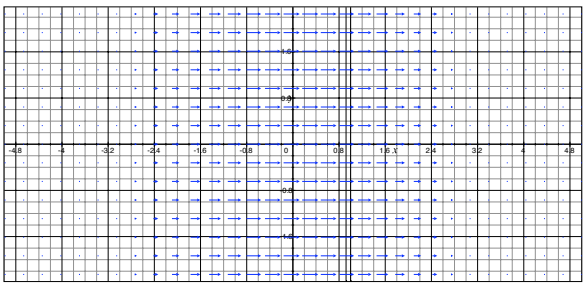
$\vec{\nabla}_{(x,y)} g(x, y)$ zeigt in die Richtung des stärksten Grauwertgefälles.

Beispiel: Stufenkante

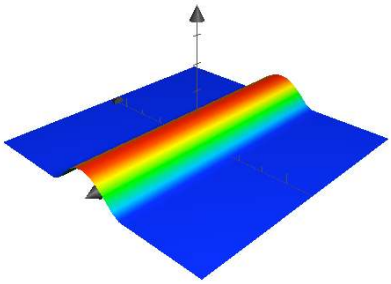


✎ **Anmerkung:** In realen Bildern können wegen der Bandbegrenztheit keine perfekten Stufenkanten vorkommen. Daher sind die Übergänge an Objektgrenzen immer stetig und differenzierbar.

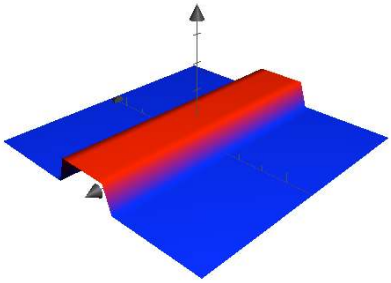
Gradientenfeld der Stufenkante



Gradientenbetrag der Stufenkante



Gradientenbetrag mit Schwellenwert



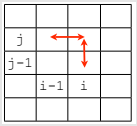
Gradienten im Digitalbild Der Gradient ist nur für kontinuierliche, *differenzierbare* Funktionen definiert. Um in einem Digitalbild Gradienten berechnen zu können, müssen wir das Bild

- entweder durch eine differenzierbare Funktion approximieren
- oder eine diskrete Schätzung für den Gradienten berechnen. Dazu approximieren wir die Differentiale dx, dy durch Differenzen $\Delta x, \Delta y$, die in zwei *orthogonalen Richtungen* gemessen werden. Eine solche Approximation an den Gradienten berechnet der *Roberts cross operator:* $\mathcal{R}(x, y)$. Dieser sehr simple Operator ist anwendbar, wenn die Bilder wenig verrauscht und die Kontraste sehr gut sind.

44.2 Digitale Approximationen des Gradienten

Gradienten im Digitalbild

Diskrete Ableitungen in x, y und in Richtung θ:
Differenzen statt Differentiale



Bildausschnitt

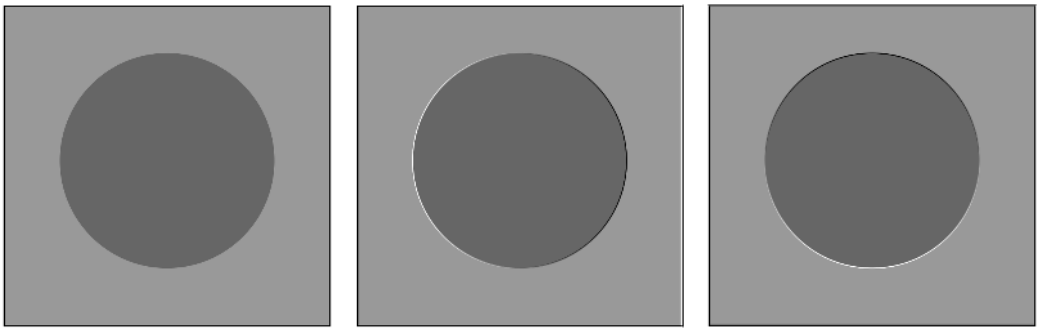
$$\begin{aligned}
\Delta_x g(i,j) &= g(i,j) - g(i-1,j) \\
\Delta_y g(i,j) &= g(i,j) - g(i,j-1) \\
\Delta_\theta g(i,j) &= \Delta_x g(i,j) \cdot \cos\theta + \Delta_y g(i,j) \cdot \sin\theta
\end{aligned}$$

Digitaler Gradient: (Betrag)

$$\begin{aligned}
||\vec{\nabla}_(x,y) g(i,j)|| &= \sqrt{(\Delta_x g(i,j))^2 + (\Delta_y g(i,j))^2} \\
\text{oft vereinfacht zu} &\approx |\Delta_x g(i,j)| + |\Delta_y g(i,j)| \\
\text{oder} &\approx \max(|\Delta_x g(i,j)|, |\Delta_y g(i,j)|)
\end{aligned}$$

Genauigkeit 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.

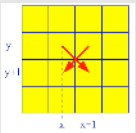


Original $\Delta_x g(i,j)$ $\Delta_y g(i,j)$
dem Original überlagert.

44.2.1 Roberts-Cross-Operator

Roberts-Cross-Operator

Ein symmetrischer digitaler Gradient:
Roberts cross operator $\mathcal{R}(x, y)$:

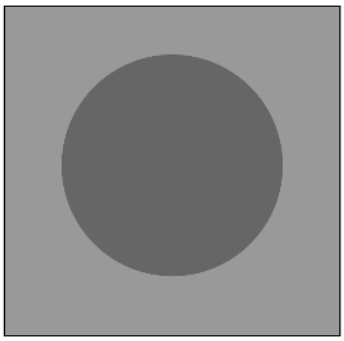


$$\begin{aligned} \Delta_1 &= g(x,y) - g(x+1,y+1) \\ \Delta_2 &= g(x,y+1) - g(x+1,y) \\ ||\vec{\nabla}_(x,y)g(x,y)|| &\approx \mathcal{R}(x,y) \end{aligned}$$

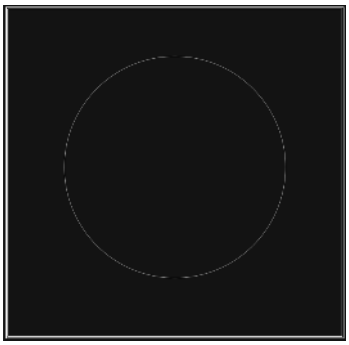
Näherungen

$$\begin{aligned} \mathcal{R}(x,y) &= \sqrt{\Delta_1^2 + \Delta_2^2} \\ \mathcal{R}(x,y) &\approx \mathcal{F}(x,y) = |\Delta_1| + |\Delta_2| \end{aligned}$$

Beispiel: Roberts-Cross-Operator



Original



Roberts-Cross-Operator-Maxima

✎ **Anmerkung:** Roberts-Cross-Operator

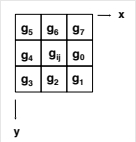
Da wir nur die Maxima des Gradienten finden wollen, sind wir nur an den relativen Gradientenbeträgen interessiert, nicht an den absoluten Werten. In der Praxis wird daher meist mit der Approximation $\mathcal{F}(x, y)$ gearbeitet, da zur Berechnung nur die Addition und Subtraktion notwendig sind. (Duda and Hart, 1973, Seite 268)

$$\mathcal{R}(x, y) \leq \mathcal{F}(x, y) \leq \sqrt{2}\mathcal{R}(x, y)$$

Der Roberts-Cross-Operator ist extrem rauschempfindlich und daher nur bei sehr gutem Signal-Rausch-Verhältnis anwendbar.

44.2.2 Sobel-Operator

Sobel-Operator



- Berechnet die Gradientenkomponenten Δ_x und Δ_y anhand einer 3×3-Umgebung.
- Die Bildfunktion wird gleichzeitig durch mittenbetontes Mitteln geglättet.

$$\begin{aligned}
\Delta g_x &= (g_1 + 2g_0 + g_7) - (g_3 + 2g_4 + g_5) \\
\Delta g_y &= (g_1 + 2g_2 + g_3) - (g_7 + 2g_6 + g_5) \\
||\vec{\nabla}_(x, y) g_{ij}|| &= \sqrt{\Delta g_x^2 + \Delta g_y^2} \\
\tan\gamma &= \frac{\Delta g_y}{\Delta g_x}
\end{aligned}$$

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.

Die Maskendarstellung ist typisch für linare Faltungsoperatoren. die Koeffizienten der Maske geben an, mit welchem Wert jedes Pixel multipliziert werden muß, um den Operatorwert zu berechnen.

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

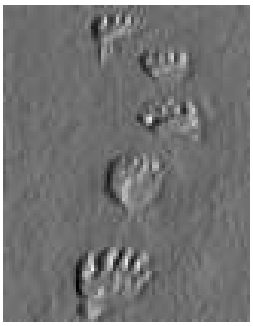
Pixel des Bildes

$$
\begin{aligned}
\Delta g_x &= -0-0-0+2+4+2=8 \\
\Delta g_y &= -0-2-2+0+2+2=0 \\
\tan\gamma &= \frac{\Delta g_y}{\Delta g_x}=0 \\
\gamma &= 0^\circ
\end{aligned}
$$

Beispiel: Sobel-Operator 3

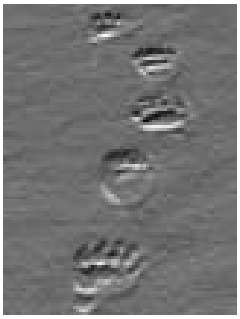
g(x, y)
greyvalue image

0 = black
255 = white



Δg_x
x-component of greyvalue gradient

0 = greyvalue 128



Δg_y
y-component of greyvalue gradient

0 = greyvalue 128

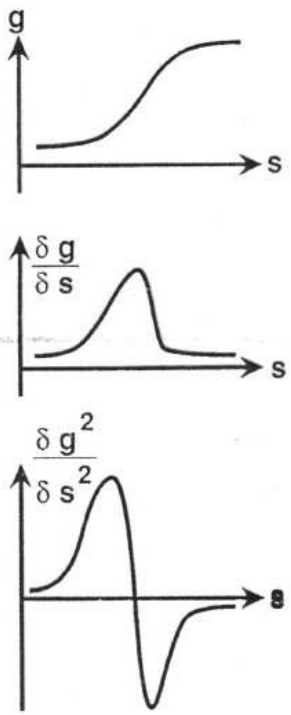
✎ **Anmerkung:** Das obige Beispiel wurde von den Folien der Bildverarbeitung 1 Vorlesung von Bernd Neumann entnommen.

45 Laplace-basierte Operatoren

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")

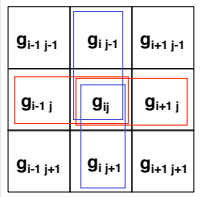
Verlauf des Gradienten-Betrages

Verlauf der 2. Ableitung

45.1 Der Laplace-Operator

Diskrete Approximation des Laplace-Operators

Zur diskreten Approximation der partiellen 2. Ableitungen werden Differenzen von Differenzen berechnet.



$$\begin{aligned}\nabla^2 g_{ij} &= (g_{i+1j} - g_{ij}) - (g_{ij} - g_{i-1j}) \\ &\quad +(g_{ij+1} - g_{ij}) - (g_{ij} - g_{ij-1}) \\ &= g_{i+1j} + g_{i-1j} + g_{ij+1} + g_{ij-1} - 4g_{ij}\end{aligned}$$

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.

$$\begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \qquad \nabla^2 f = 4 - 4 = 0$$

$$\begin{matrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{matrix} \qquad \nabla^2 f = 2 + 2 + 1 + 3 - 8 = 0$$

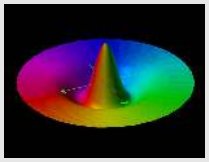
$$\begin{matrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 1 & 2 & 1 \end{matrix} \qquad \nabla^2 f = 6 - 8 = -2$$

$$\begin{matrix} 1 & 1 & 3 \\ 1 & 1 & 3 \\ 1 & 1 & 3 \end{matrix} \qquad \nabla^2 f = 6 - 4 = 2$$

45.2 Der Mexican-Hat Operator

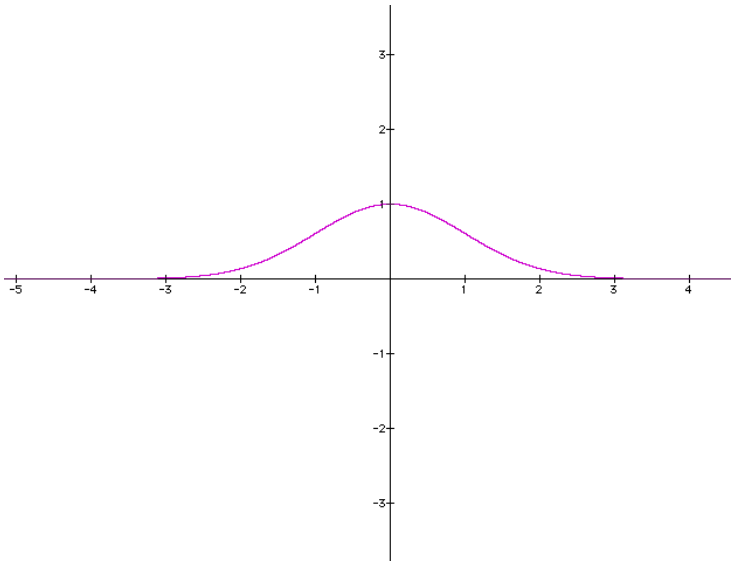
Der Mexican-Hat Operator

Marr und Hildreth 1980



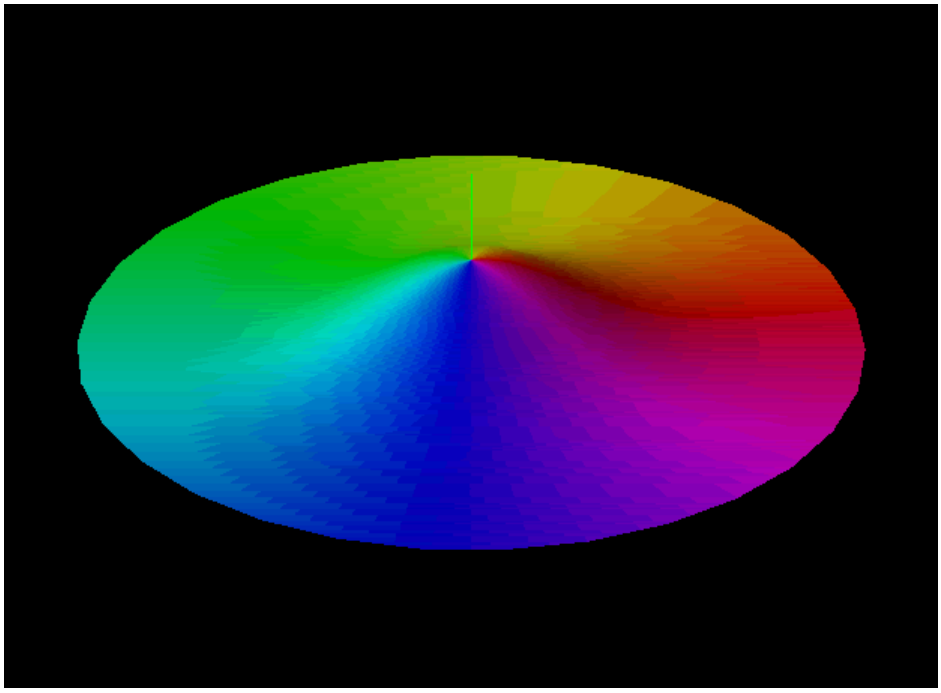
- Glätten des Bildes mit einem Gauß-Filter
- Lokalisation der Extremwerte des Gradienten durch die Nulldurchgänge der 2. Ableitung

Querschnitt eines Gauß-Filters, $\sigma = 1$

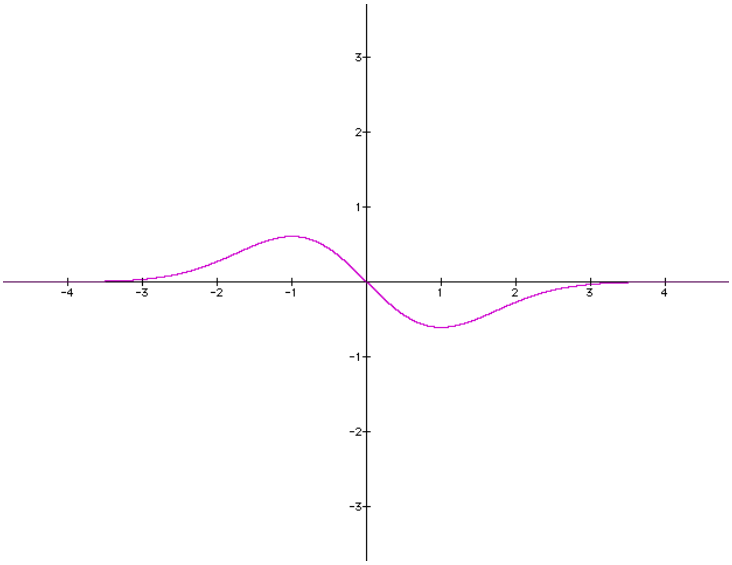


$$z = e^{-\frac{r^2}{2}}$$

2D-Gauß-Filter, $\sigma = 1$

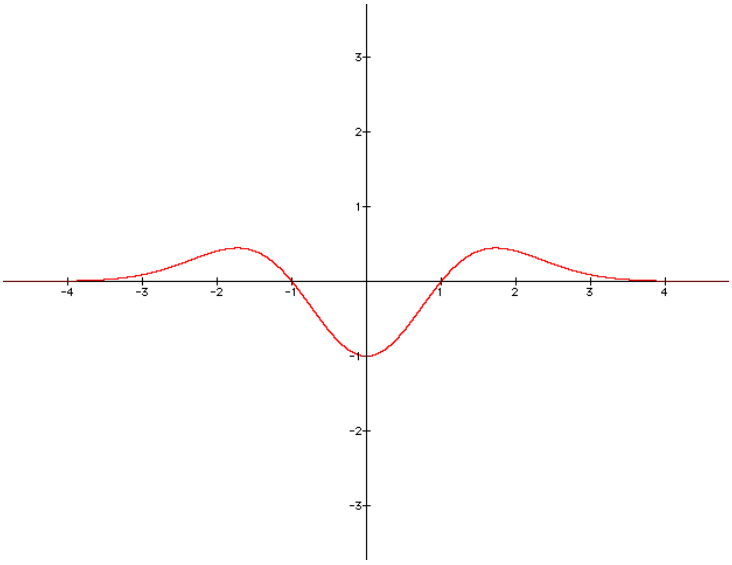


Erste Ableitung der Gauß-Funktion, $\sigma = 1$



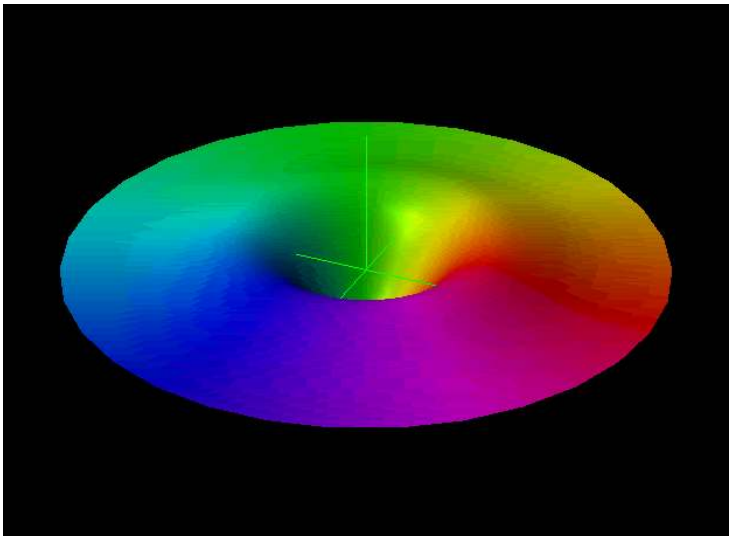
$$z = -r \cdot e^{-\frac{r^2}{2}}$$

Zweite Ableitung der Gauß-Funktion, $\sigma = 1$

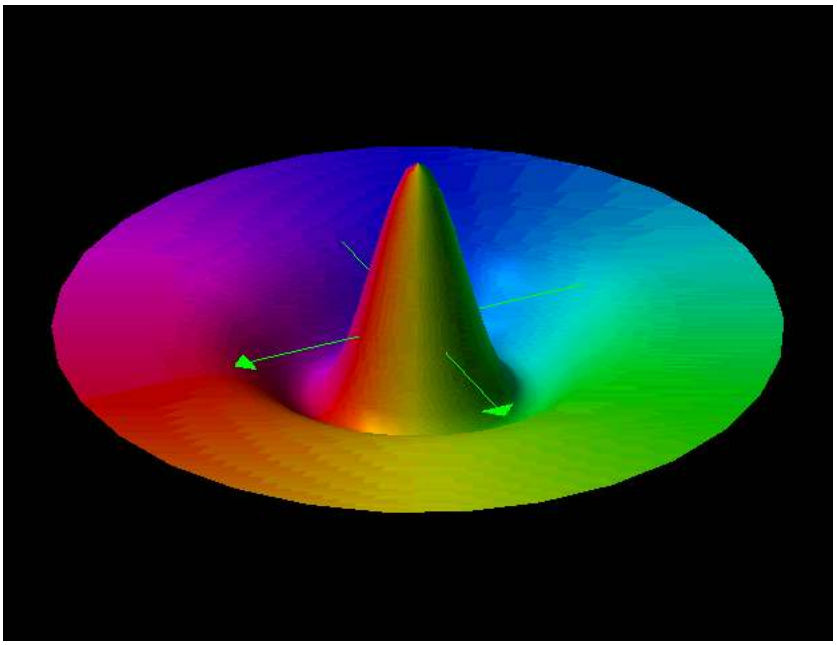


$$z = (r^2 - 1) \cdot e^{-\frac{r^2}{2}}$$

Zweite Ableitung der Gauß-Funktion, $\sigma = 1$

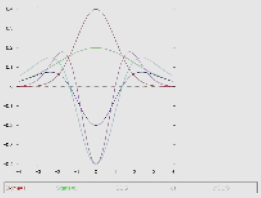


Der Mexican-Hat Operator: Ein umgekehrter Sombrero



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movmexican-hat.html

Eine andere Interpretation: DoG-Operator

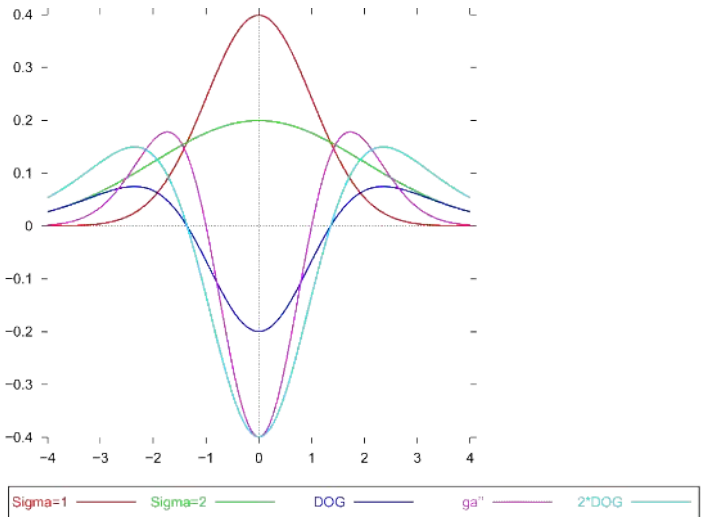


Difference of Gaussians (DOG):

- 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]

(Marr, 1976)

Differenz zweier Gauß-Funktionen mit unterschiedlicher Varianz



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movdogQ.html

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:

$$\nabla^2[f(x,y,\sigma)]\cdot g(x,y) = h(x,y)\cdot g(x,y)$$

$$h(x,y) = c(\frac{x^2+y^2-\sigma^2}{\sigma^4})e^{\frac{x^2+y^2}{2\sigma^2}}$$

c normalisiert die Summe der Gewichte zu Null.

Diskrete Approximation von $h(x,y)$

$$\begin{vmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{vmatrix} \approx h(x,y) = c(\frac{x^2+y^2-\sigma^2}{\sigma^4})e^{\frac{x^2+y^2}{2\sigma^2}}$$

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

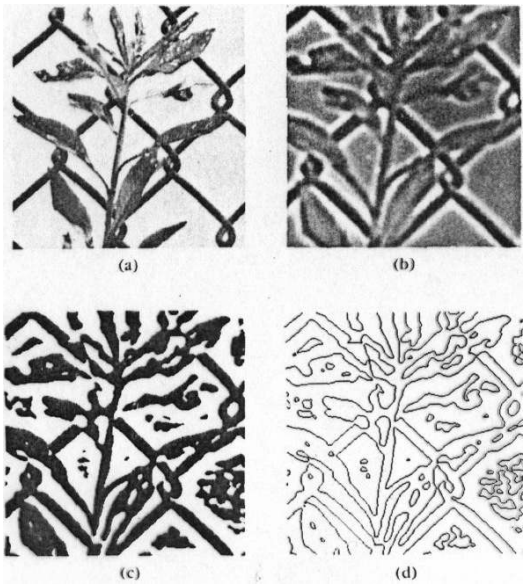
Originalbild

DoG-Filterung mit $\sigma_1 = 1, \sigma_2 = 1.6$

Kantensuche mit dem Mexican-Hat-Operator

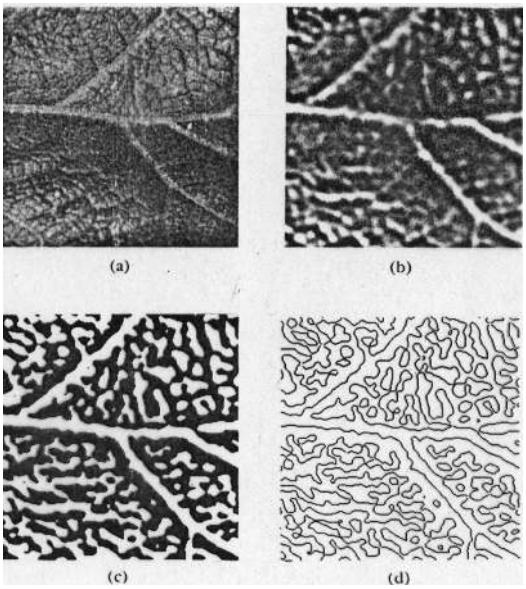
- Die Nulldurchgänge des mit dem Mexican-Hat-Operator gefilterten Bildes weisen auf Extremwerte des Gradienten hin.
- Kantenzüge werden gefunden, indem Nulldurchgangslinien verfolgt werden.
- Das Verfahren liefert also nicht isolierte edgel, sondern geschlossene Kantenzüge.

Beispiel: Maschendrahtzaun



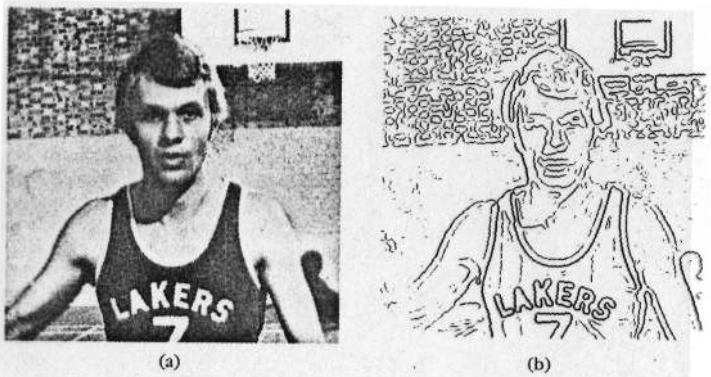
Original, Mexican-Hat-Resultat, positive und negative Mexican-Hat-Resultate, Nulldurchgänge

Beispiel: Blattstrukturen



Original, Mexican-Hat-Resultat, positive und negative Mexican-Hat-Resultate, Nulldurchgänge

Beispiel

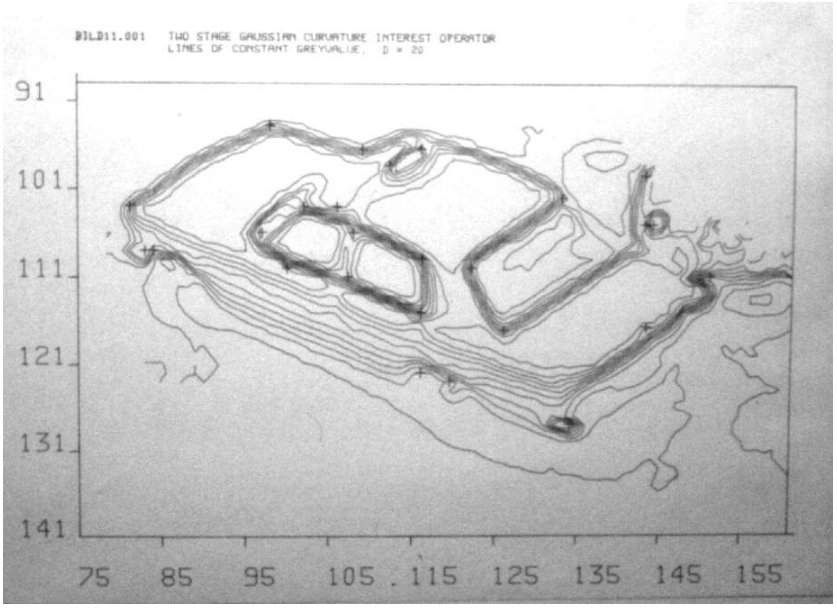


(a) (b)

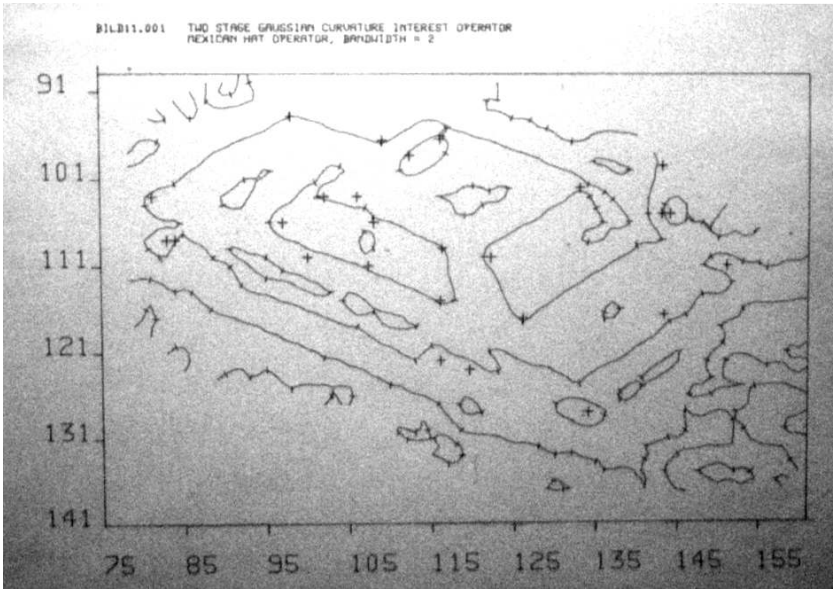
Figure 2–15. Another example of zero-crossings; here, the intensity of the lines has been made to vary with the slope of the zero-crossing, so that it is easier to see which lines correspond to the greater contrast. (Courtesy BBC Horizon.)

Beispiel: Taxi-Szene, Originalbild

Beispiel: Taxi-Szene: Höhenlinien des Grauwertgebirges



Beispiel: Taxi-Szene



Nulldurchgangslinien des Mexican-Hat-Operators, Durchmesser 5

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.

45.3 Der Canny-Operator

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, gGrauwertbild:

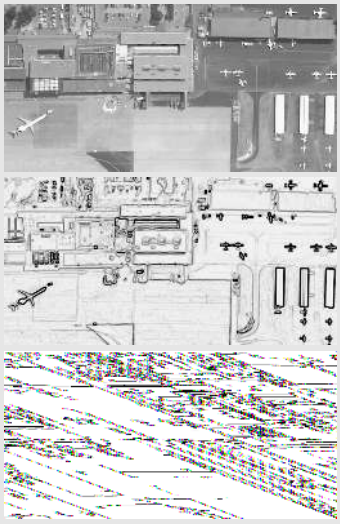
$$\vec{n} = \frac{\nabla(f \cdot g)}{|\nabla(f \cdot g)|}$$

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



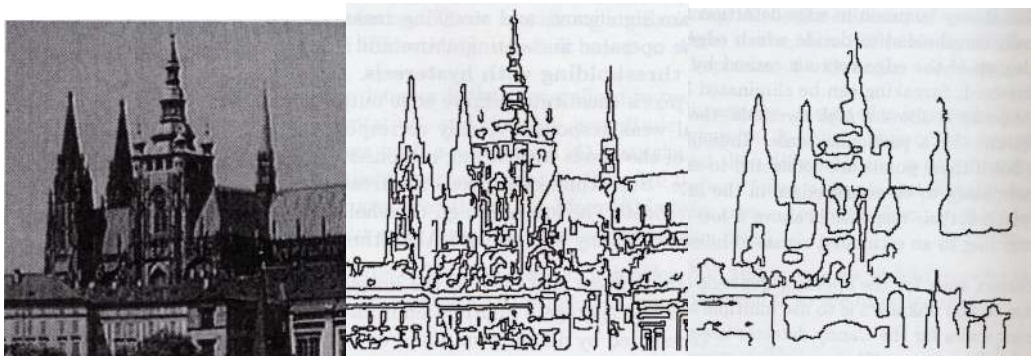
Beispiel (Flughafen Nürnberg)

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 agregiert die gefundenen Kanten über einen speziellen *feature synthesis*-Algorithmus.

Skalenvergleich



Original | Canny Operator $\sigma = 1$ | Canny Operator $\sigma = 2.8$

Hysteresis-Schwellenwert

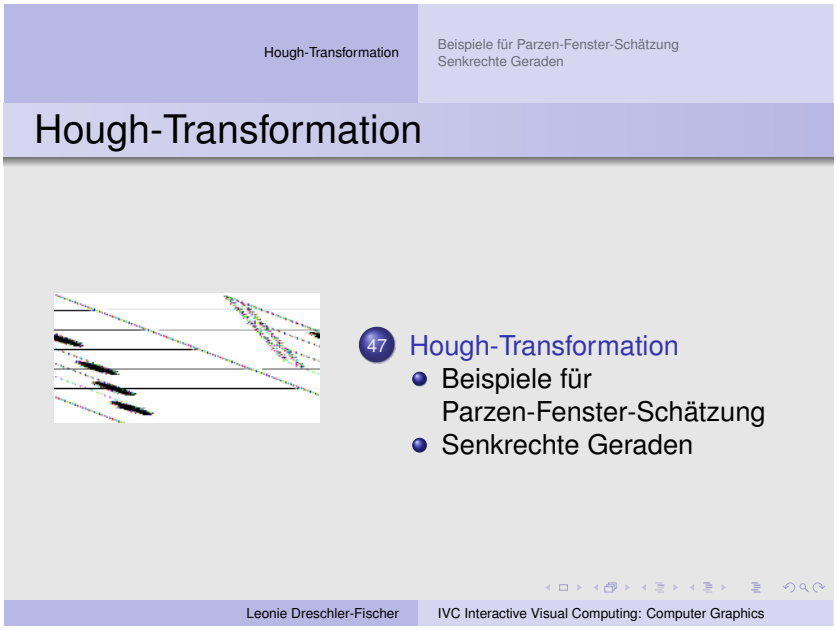
Alle edgel

gefiltert

Teil XVIII

Gruppierung von Kanten

46 Hough-Transformation



Geradensuche

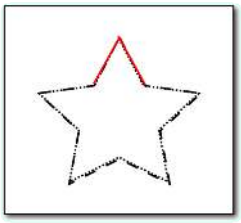
Für viele Bildverarbeitungsaufgaben ist es nötig, Geraden oder andere regelmäßige Linien erkennen zu können, beispielsweise zur

Geraden sind wichtig zur

- Erkennung von Gebäuden und Straßen,
- Analyse von Dokumenten,
- Auswertung von Fluchtpunkten und Perspektive,
- Erkennung natürliche Objekte, wie Baumstämme.

Edgel ⇒ Geraden: Hough-Transformation

Bild



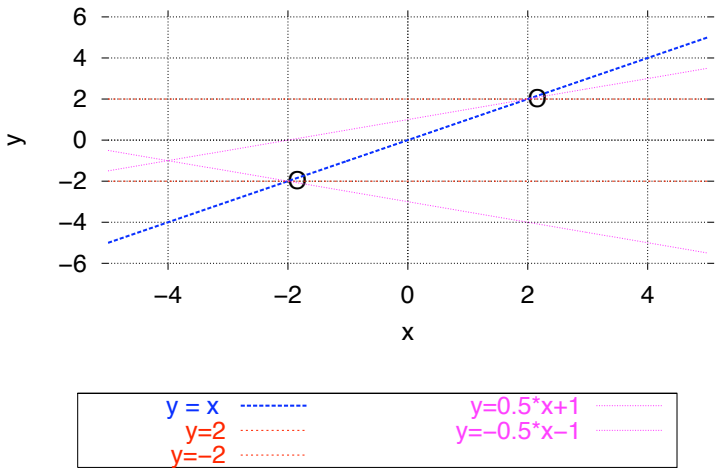
edgel und einige Geraden

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 Geradenbüschel

Gesucht:
Die Gerade, die im Durchschnitt aller Geradenbüschel der edgel liegt.

✎ **Anmerkung:** Wenn wir ein edgel gefunden haben, dann kennen wir einen Punkt im Bild, durch den vermutlich ein Kantenzug geht, aber wir kennen nicht die Richtung der Kante und wissen nicht, mit welchen anderen Segmenten dieser Punkt zu verbinden ist. Hier nehmen wir an, daß wir das Vorwissen haben, daß die Kanten von Geraden stammen.

Durch jeden Bildpunkt können beliebig viele Geraden gehen. Ein edgel definiert ein *Geradenbüschel* von möglichen Kandidaten. Wenn wir mehrere edgel haben, dann liegt die gesuchte Gerade im Durchschnitt der Geradenbüschel, die von diesen edgeln definiert werden.

Gruppieren von edgeln: Hough-Transformation

Wir überlagern dem Bildraster ein $x - y$-System. Dann können wir Geraden im Bild durch Gleichungen der Form

$$y = \underbrace{m}_{Steigung} x + \underbrace{c}_{Achsenabschnitt} \quad \text{beschreiben.}$$

Sei (x_0, y_0) ein Bildpunkt, an dem eine Kante gefunden wurde — ein *edgel.* Alle Geraden durch diesen Punkt erfüllen die Gleichung:

$$\begin{aligned} y_0 &= m x_0 + c \\ c &= \underbrace{-x_0}_{Steigung} \cdot m + \underbrace{y_0}_{Achsenabschnitt} \end{aligned}$$

(Hough, 1962)

✎ **Anmerkung:** Die Hough-Transformation (Hough, 1962) wurde ursprünglich zur Detektion von Geraden entwickelt und wird jetzt aber generell zur Detektion parametrischer Kurven und Flächen (Kreise, Ellipsen, ElLipsoide) eingesetzt.

Parameterraum

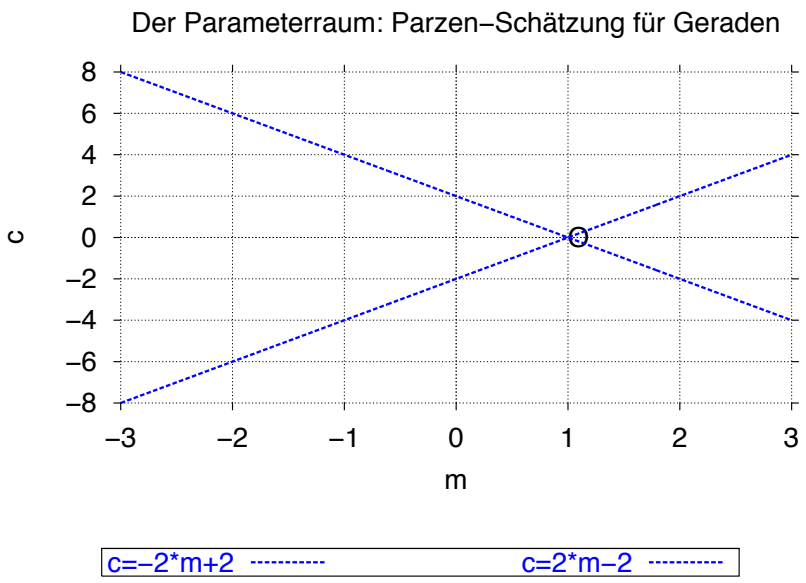
$$c = \underbrace{-x_0}_{Steigung} \cdot m + \underbrace{y_0}_{Achsenabschnitt}$$

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 $M \times C$



Parameterraum

Wir wollen herausfinden, welche Geraden im Bild am wahrscheinlichsten sind. Das sind offensichtlich solche Paare von (m, c) im Parameterraum, in deren Nähe die Geraden von vielen Geradenbüscheln liegen. Wir suchen also eine Schätzung für $p(m, c)$.

Wir gehen wie folgt vor: Jedes edgel x_i, y_i definiert ein Geradenbüschel im Bildraum und damit eine Gerade g_i durch den MC-Parameterraum. Die Menge dieser Geraden ist unsere Stichprobe.

Jetzt benötigen wir noch ein Abstandsmaß, mit dem wir gewichten können, wie gut eine solche Gerade die Wahrscheinlichkeit für eine bestimmte Kombination (m, c) stützt. Jede Probe gibt uns gleich eine ganze Menge von Punkten (die Gerade im MC-Raum). Wir könnten jeden Punkt der Geraden einzeln berücksichtigen, aber ökonomischer ist es, einfach vom Punkt (m, c) aus das Lot auf die Gerade zu fällen und den kürzesten Abstand des Punkts (m, c) von der Geraden als Gewicht zu verwenden. Der Punkt (x_i, y_i) aus dem Bildraum definiert im MC-Parameterraum die Gerade $g_i : c_i = -x_i m_i + y_i$. Ein Punkt (m, c) hat von dieser Geraden den Abstand:

$$\Delta((m, c), g_i) = \left| \frac{-x_i \cdot m - c + y_i}{\sqrt{x_i^2 + 1}} \right|$$

Diesen Abstand verwenden wir als Teil der Evidenz für eine Gerade mit den Parametern (m, c).

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.

In der Bildverarbeitung wird die Hough-Transformation meist als Histogram-Verfahren angewendet, siehe (Ballard and Brown, 1982). Das Histogram-Verfahren hat den Nachteil, daß jede Probe nur innerhalb eines festen Fensters zur Schätzung beiträgt und die Dichte nur in diskreten Schritten geschätzt werden kann. Die hier vorgeschlagene Variante mit dem Parzenfenster interpoliert über die Fenstergrenzen hinweg und gibt uns eine kontinuierliche Dichtefunktion, allerdings ist der Rechenaufwand vergleichsweise hoch.

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.

$$p(m, c) = \frac{1}{n} \sum_{i=1}^{n} \delta_n(\Delta((m, c), g_i))$$

✎ **Anmerkung:** δ_n ist eine Fensterfunktion.

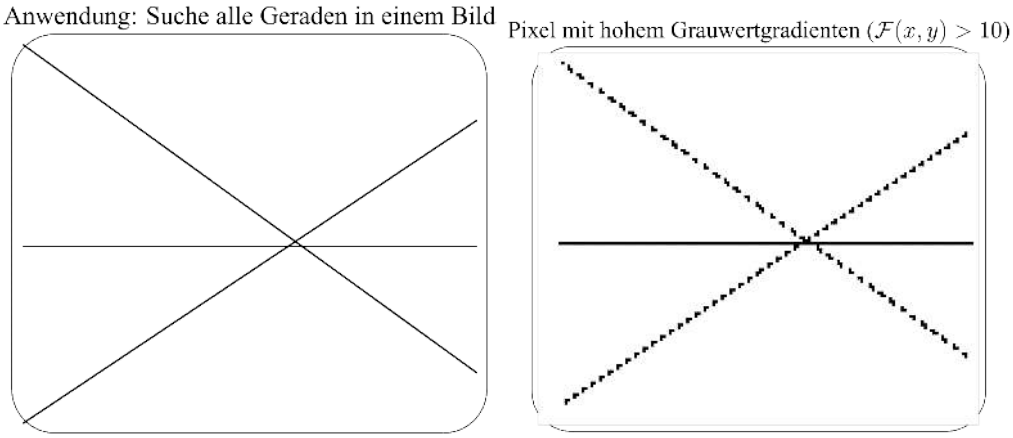
46.1 Beispiele für Parzen-Fenster-Schätzung

Auf den folgenden Folien sehen Sie die Anwendung dieses Verfahrens auf zwei Bilder, die Geraden enthalten. Damit die beiden Koordinaten m und c im Parameterraum in vergleichbaren Größenordnungen bleiben, wurden die Bildkordinaten bei der Abbildung von Zeilen und Spalten auf x und y so skaliert, daß die x-Koordinaten im Intervall $[0, 1]$ liegen. Da die Steigung m für sehr steile Geraden gegen ∞ geht, wurden nur Steigungen zwischen -2 und 2 berücksichtigt. Senkrechte Geraden können so also nicht gefunden werden.

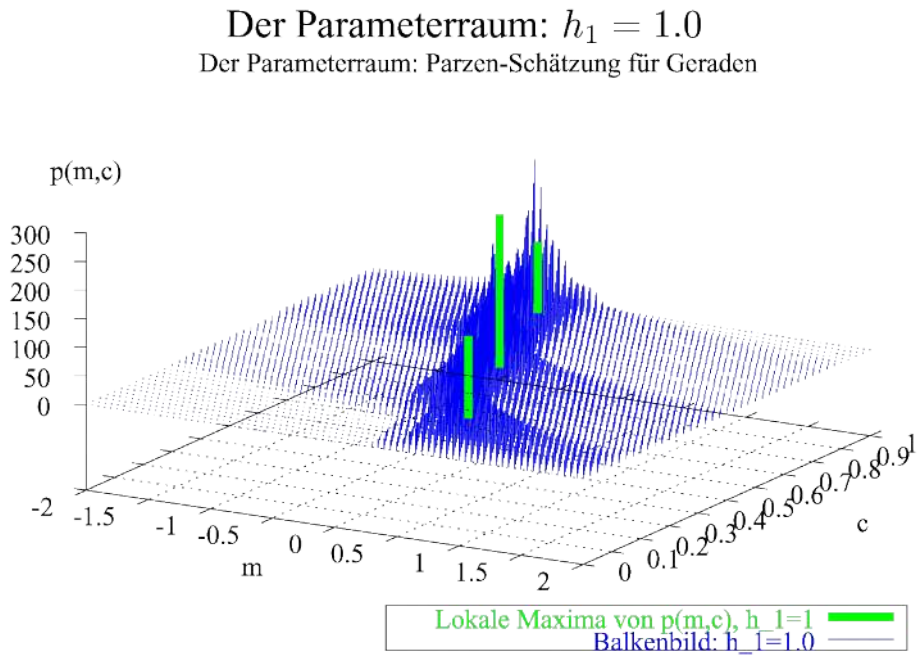
Für jedes Bild wurden zwei Versuche mit unterschiedlichen Werten von h_1 (Breite der Fensterfunktion) durchgeführt. Beim einfachen Testbild mit nur drei deutlich getrennten Geraden zeichnen sich alle drei Geraden als deutliche Maxima von $p(m, c)$ im Parameterraum ab, unabhängig von h. Beim etwas schwierigeren Notenbild können die Notenlinien bei $h_1 = 5$ nicht mehr aufgelöst werden. Bei einer so großen Fensterbreite ist die Mittelung zu stark.

Ein einfaches Beispiel

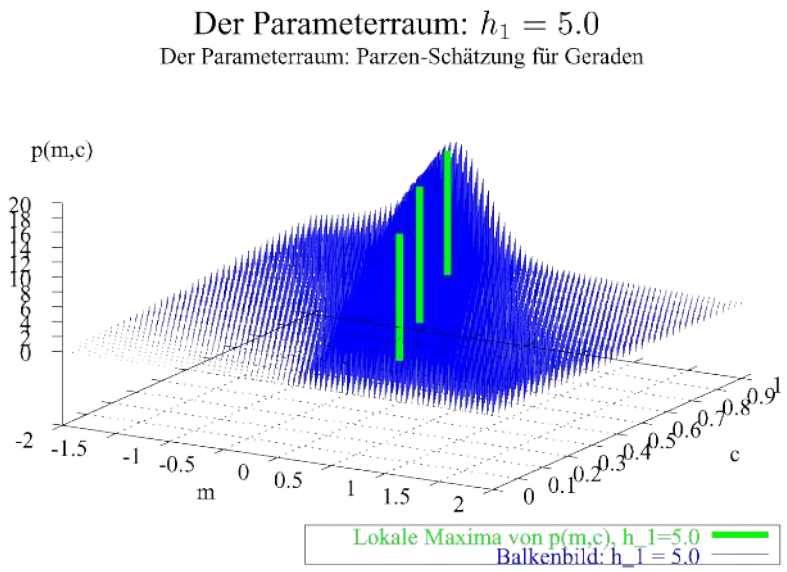
Beispiel: 101
Ein einfaches Bild und die berechneten edgel



Dichtefunktion, Fensterbreite $h = 1$

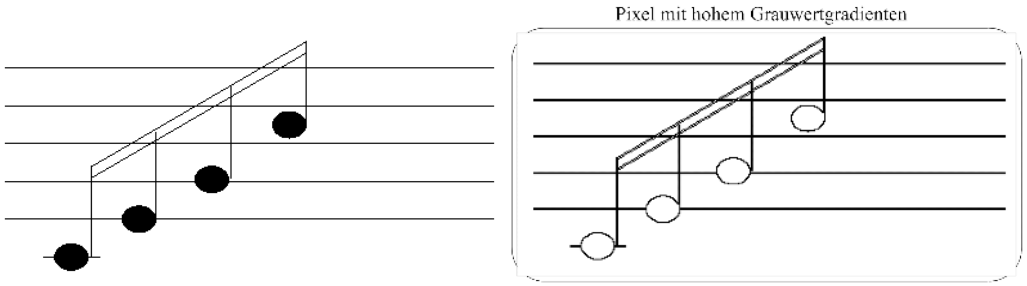


Dichtefunktion, Fensterbreite $h = 5$

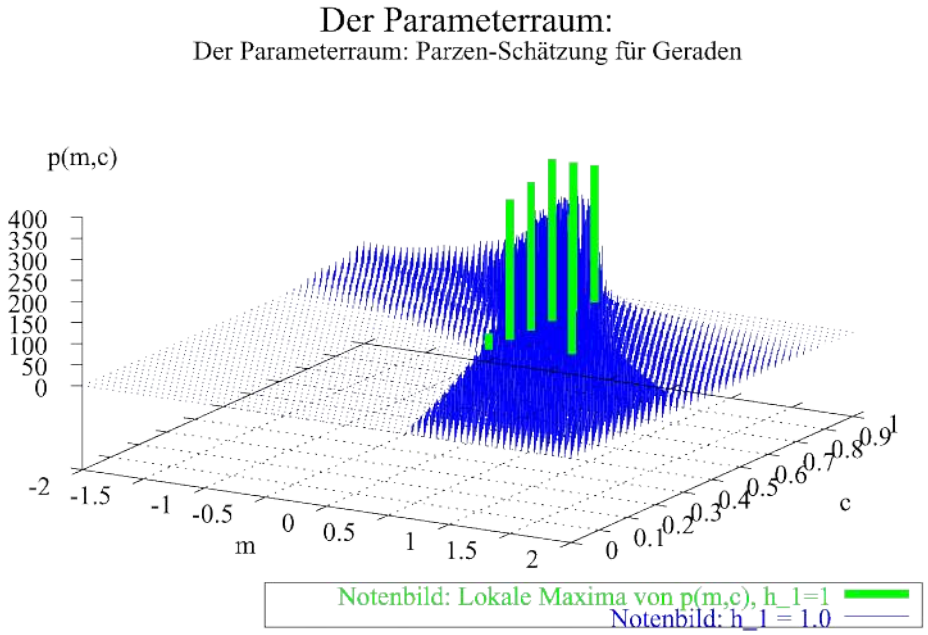


Ein weiteres Beispiel

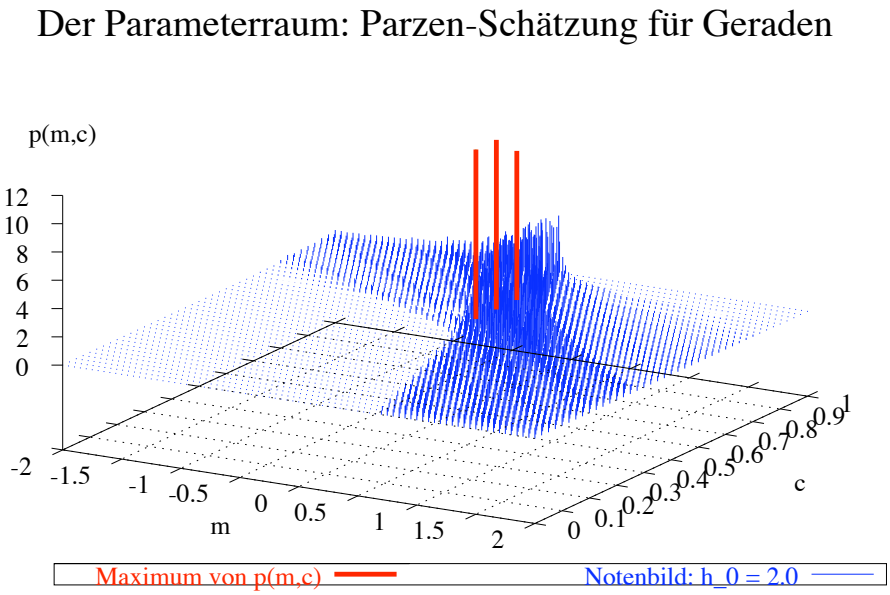
Beispiel: 102
Ein Notenbild und die berechneten edgel



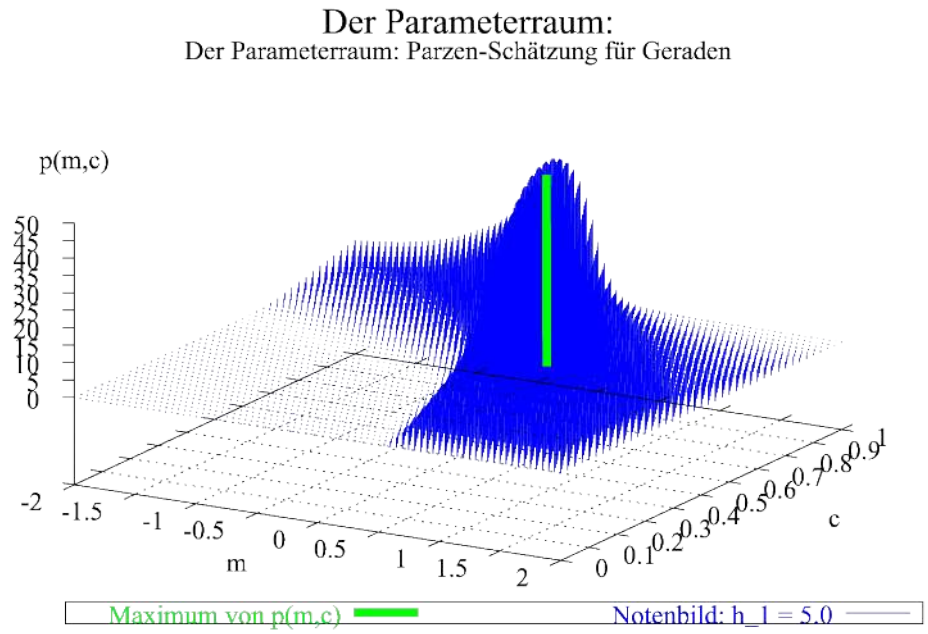
Dichtefunktion, Fensterbreite $h = 1$



Dichtefunktion, Fensterbreite $h = 2$



Dichtefunktion, Fensterbreite $h = 5$



✎ **Anmerkung:** Der Einfluß der Fensterbreite h

Die Geraden des Notensystems bilden deutlich getrennte Maxima, die beiden schrägen Balken bilden ein gemeinsames Maximum, da ihr Abstand bei dieser Wahl von h_1 zu klein ist.

Es fällt auf, daß mit $h_1 = 1$ sogar das winzige Geradenstück durch die Note „c“ gefunden wurde, aber nicht die vergleichsweise deutlich längeren Notenhälse. Das liegt daran, daß die Notenhälse senkrechte Geraden sind mit $m = \infty$. Diese können wir in einem endlichen Parameterraum nicht berücksichtigen.

46.2 Senkrechte Geraden

Senkrechte Geraden

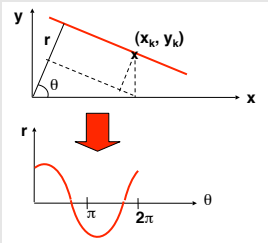
Darstellung der Geraden

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



Geradendarstellungen

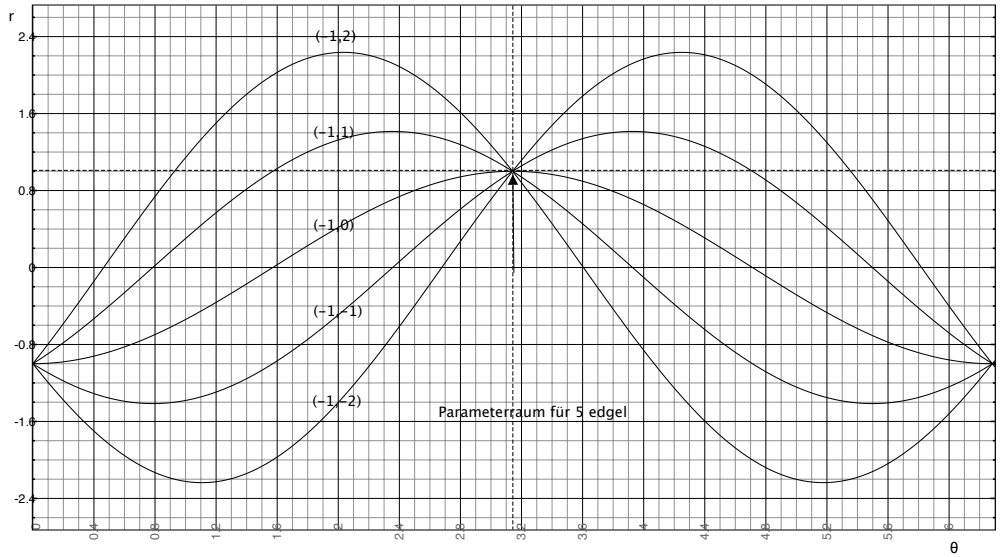
r: Abstand vom Ursprung

θ: Richtung der Normalen

Im Parameterraum ergibt sich eine sinusähnliche Kurve.

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.

$$\begin{aligned}(x_k, y_k, \phi) &= \text{edgel mit Gradientenrichtung } \phi \\ x_k \cos\theta + y_k \sin\theta &= r, \phi - \delta \leqslant \theta \leqslant \phi + \delta \text{ im Parameterraum}\end{aligned}$$

Hough-Transformation für Kreise

Der Parameterraum:

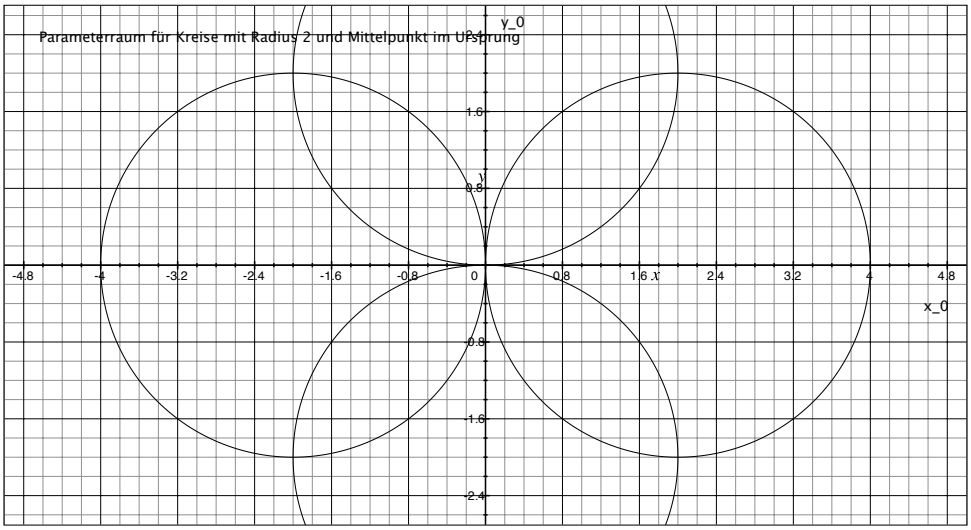
r: Radius

x_0, y_0: Koordinaten des Mittelpunkts

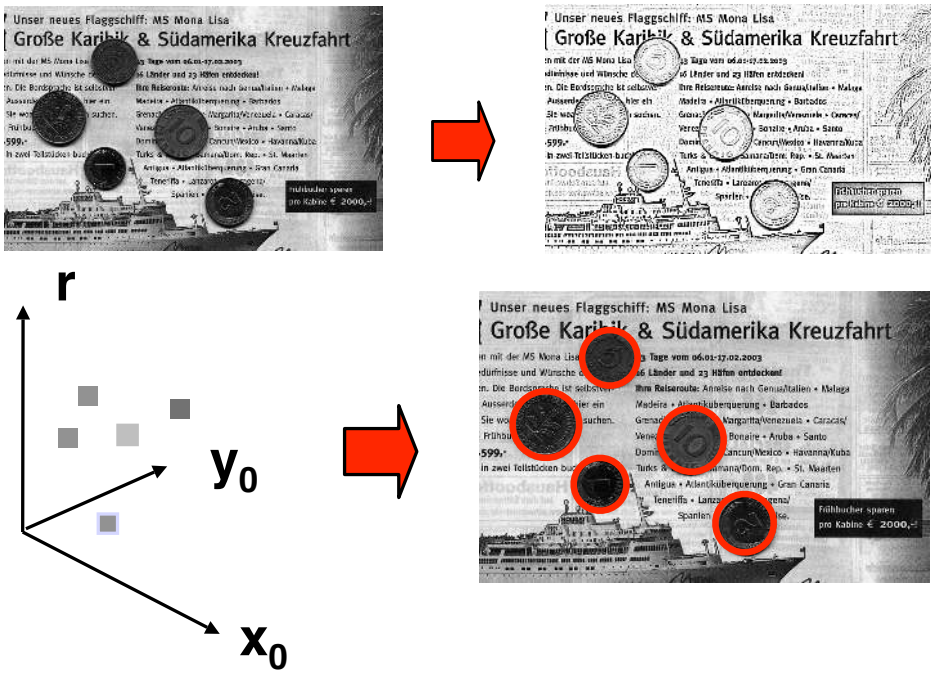
x_k, y_k: Koordinaten des edgels

$$(x_k - x_0)^2 + (y_k - y_0)^2 = r^2$$

Beispiel: Parameterraum für Kreis mit Radius 2 und Mittelpunkt im Ursprung



Anwendung: Münzen auf strukturiertem Untergrund



aus dem Skript „Bildverarbeitung 1“ von Bernd Neumann

IVC Interactive Visual Computing: Computer Vision

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 7:

Geometrische Szenenrekonstruktion

Universität Hamburg
Fachbereich Informatik

WS 2014/2015

Teil XIX

Geometrische Szenenrekonstruktion

47 Geometrische Szenenrekonstruktion

Während des letzten Jahrzehnts wurden beim Verständnis der Probleme der geometrischen Szenenrekonstruktion beachtliche Fortschritte erreicht. Die entwickelten Verfahren lassen sich unter dem Namen „shape-from. . . “ zusammenfassen – „shape-from-shading“, „shape-from-texture“, „shape-from-contour“, „shape-from-stereo“ usw. Die erste umfassende Diskussion der grundlegenden Probleme der Szenenrekonstruktion und der shape-from-verfahren erschien 1981 in einem Sammelband der Zeitschrift „Artificial Intelligenz“, Band 17.

„shape-from-shading“ ermöglicht es, die Orientierung von Oberflächenelementen zu ermitteln, wenn deren *bidirektionale Reflektanzfunktion* und die Beleuchtung bekannt sind (Horn, 1986); ebenso läßt sich die Orientierung von Oberflächen relativ zum Betrachter durch „shape-from-texture“ ermitteln, wenn die Texturgradienten auf perspektivische Effekte zurückzuführen sind. Einige der Verfahren sind auf Einzelbilder anwendbar (monokulare Verfahren), während andere mehrfache Objektansichten erfordern. Wir unterscheiden außerdem zwischen trigonometrischen und photometrischen Rekonstruktionsverfahren. Trigonometrische Verfahren basieren auf einer Auswertung der Ortsinformation, während photometrische Verfahren die spektralen Eigenschaften auswerten. Wir konzentrieren uns in diesem Kapitel auf Einzelbildverfahren und im folgenden auf Mehrbildverfahren und zugleich auf die trigonometrischen Ansätze, wie Stereoskopie und Bewegungsstereo.

In diesem Kapitel (ein Auszug aus (Görz, 2005, Kapitel 6)) wird vertiefend auf Verfahren eingegangen, die zur geometrischen und photometrischen Rekonstruktion natürlicher Szenen eingesetzt werden können — also die Algorithmen und Repräsentationen der zweiten Schicht eines allgemeinen bildverstehenden Systems. Die hier vorgestellten Modelle und Algorithmen formalisieren den Prozeß der Bildentstehung und sind damit nicht nur für die Bildverabeitung relevant, sondern auch für die Computergrafik. Als Nachschlagewerk für die hier angesprochenen Themen empfehle ich das sehr ausführliche zweibändige Lehrbuch von Glassner, 1995.

Wir werden hier vorrangig eine ingenieursmäßige Sicht auf das Problem vertreten, weniger eine kognitive Sicht. Eine umfassende Darstellung der kognitiven Aspekte finden Sie im Kapitel 6.2 der zweiten Auflage dieses Buches. Das menschliche Sehsystem dient uns hier als Maßstab für die Leistungsanforderungen, nicht jedoch als operationales Vorbild, wie es von einer Reihe von Forschern, vor allem am MIT, angestrebt wird. Die Entschlüsselung des menschlichen Sehsystems ist sicherlich ein wichtiges und faszinierendes Projekt der KI, durch welches für das Problem des Maschinensehens viele Anregungen gewonnen werden können – aber gerade aus der Sicht der Robotik erscheint es als vordringliche Aufgabe, zunächst die *dem räumlichen Sehen zugrundeliegenden Probleme* zu verstehen. Dies wäre eine der Vorausetzungen dafür, je nach Aufgabe und Anforderungen, für jedes Problem einen maßgeschneiderten Sehapparat zu entwerfen. Auch Marr and Nishihara, 1978, zwei Hauptverteter des kognitiven Paradigmas der Bilddeutung, haben festgestellt:

> *For human vision to be explained by a computational theory, the first question is plain: What are the problems the brain solves when we see?*

Grundlegender Schritt im Prozeß der Deutung eines Bildes ist – wie von Neumann in (Görz, 2005, Kapitel 6.1) dargestellt – die Generierung einer *geometrisch-temporalen Szenenbeschreibung*, die die relative räumliche Anordnung der Objekte einer Szene, den räumlichen Verlauf ihrer Oberflächen und deren photometrische Eigenschaften, die Bewegungsabläufe und die zeitlichen Veränderungen der Objekte beschreibt. Der Vorgang der Generierung einer solchen Beschreibung wird als *geometrische Szenenrekonstruktion* bezeichnet.

Im folgenden soll zunächst das grundlegende Problem der Szenenrekonstruktion beschrieben werden.

Als eine Möglichkeit zur Lösung dieses Problems wird dann auf trigonometrische Ansätze näher eingegangen, die aus der Ortsinformation mehrerer Ansichten einer Szene deren dreidimensionale Struktur zu rekonstruieren versuchen. Nach einer Einführung des dabei zu lösenden Korrespondenzproblems werden verschieden Eigenschaften sowohl der Objekte unserer natürlichen Umwelt als auch der für die Aufnahme von Stereobildern verwendeten Kamerasysteme untersucht, die eine Reduzierung der bei der Korrespondenzanalyse auftretenden Mehrdeutigkeiten gestatten.

Es sei darauf hingewiesen, daß in unserer Definition der „geometrischen Szenenrekonstruktion“ (im Gegensatz zur Definition von Neumann) der temporale Aspekt betont wird, die Erkennung von Objekten als Instanzen von Objektmodellen aber nicht mit einbezogen wird, da hierfür szenenspezifisches Wissen erforderlich wäre.

Inversion des Bildentstehungsprozesses: Die grundlegende Schwierigkeit bei der Szenenrekonstruktion kann als *Inversionsproblem* betrachtet werden. Die mit Hilfe eines bildgebenden Prozesses gewonnenen Bilddaten sind durch die Eigenschaften der abgebildeten Szene, durch die Charakteristik des verwendeten Sensors, durch die Beleuchtungsbedingungen und die Aufnahmegeometrie vollständig bestimmt. Eine korrekte Deutung des Bildes verlangt nun die Inversion des Bildenstehungsprozesses. Durch die Projektion der beobachteten Szene auf eine Ebene geht jedoch die Tiefeninformation verloren, der Projektionsvorgang ist somit nicht eindeutig umkehrbar: eine unendliche Anzahl von Kombinationen der für die Bildentstehung relevanten Faktoren kann sich bei der Abbildung in übereinstimmenden Bildstrukturen ausprägen. Zur Rekonstruktion der Szene, d.h. zur Auflösung dieser Mehrdeutigkeiten, müssen deshalb einschränkende Randbedingungen herangezogen werden; das sind typischerweise die Gesetze der projektiven Geometrie, der Photometrie und die physikalischen Eigenschaften von Objekten und ihren Oberflächen.

47.1 Tiefen- und Formhinweise

Szenenrekonstruktion

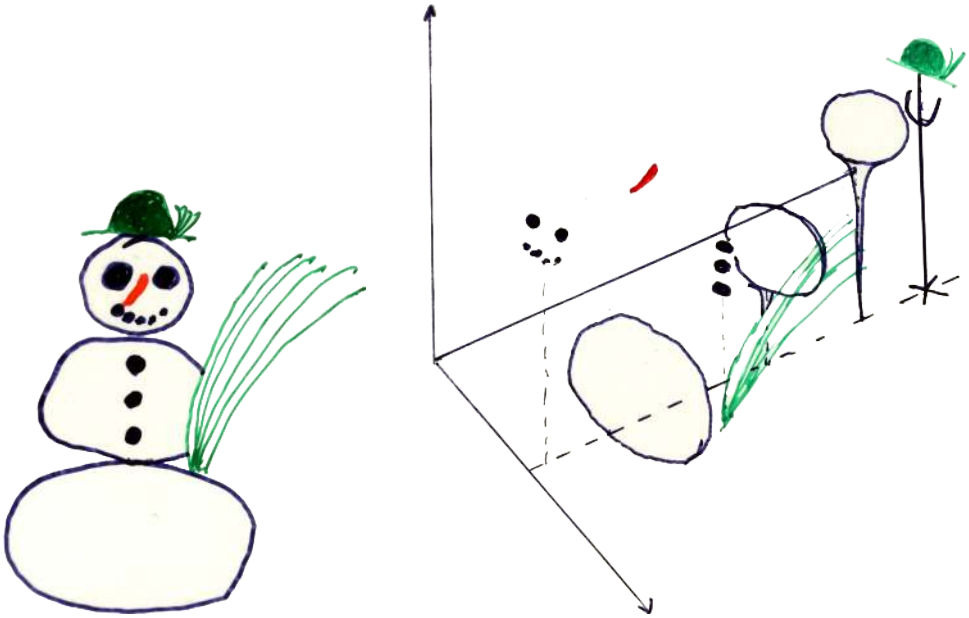
Alte und junge Frau

48 Geometrische Szenenrekonstruktion
- Tiefen- und Formhinweise
  - Motorische Hinweise
  - Scheinbare Größe
  - Texturgradienten
- Shape from Contour

49 Photometrische Szenenrekonstruktion

50 Das Generalitätsprinzip

Mehrdeutigkeit bzgl. der Tiefe



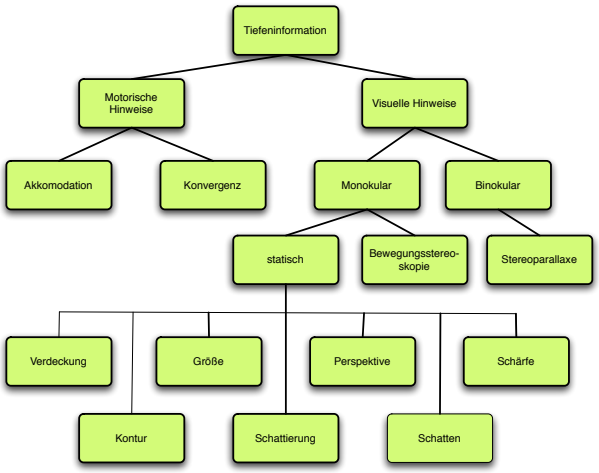
Ansicht 1 Ansicht 2

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 solche Tiefenhinweisen können wir entweder absolute Entfernungen, relative Entfernungen oder Orientierungen von Oberflächen ableiten.
- ☞ Wenn wir bestimmte Annahmen machen müssen, sprechen wir von *indirekten* Tiefenhinweisen, sonst von *direkten* Tiefenhinweisen.

Tiefenhinweise



47.1.1 Motorische Hinweise

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.

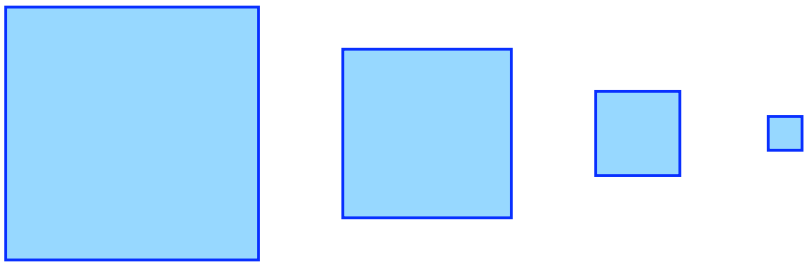
Beides sind *direkte* Tiefenhinweise, da sie unabhängig von der abgebildeten Szene anwendbar sind.

Akkomodation: Auf welche Entfernung fokussiert die Linse?

Vergenz: Unter welchem Winkel wird ein Objekt fixiert?

47.1.2 Scheinbare Größe

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:

$$\frac{\text{scheinbare Geschwindigkeit}}{\text{absolute Geschwindigkeit}} \quad \text{oder} \quad \frac{\text{scheinbare Helligkeit}}{\text{absolute Helligkeit}}.$$

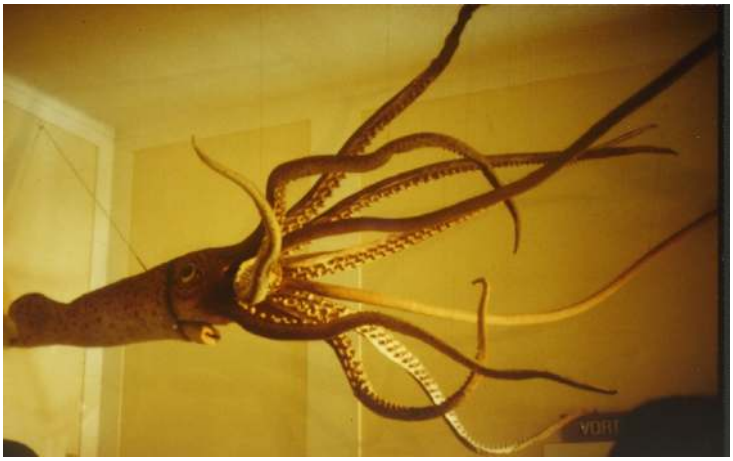
✎ **Anmerkung:** Auf Folie 532 sehen sie den Gasthof Hilligenley auf der Hallig Langeneß abgebildet. Wenn Sie die typische Größe eines solchen Gasthofs kennen (vertraute Größe), dann können Sie schätzen, aus welcher Entfernung das Bild aufgenommen wurde.

Tiefenhinweis: Vertraute Größe

Die folgende Folie zeigt das Bild eines Baumstumpfes. Auch hier können Sie die Entfernung schätzen, indem Sie das Verhältnis von bekannter absoluter zu scheinbarer Größe auswerten. Aber Vorsicht! Verwenden Sie nicht das Modell der typischen Baumstümpfe in deutschen Wäldern — dieser spezielle Baumstumpf steht im Sequoia-Nationalpark und ist knapp 10 m hoch. Wenn Sie genau hinsehen, können Sie mich am Fuß des Baumstumpfs erkennen.

Tiefenhinweis: Vertraute Größe 2

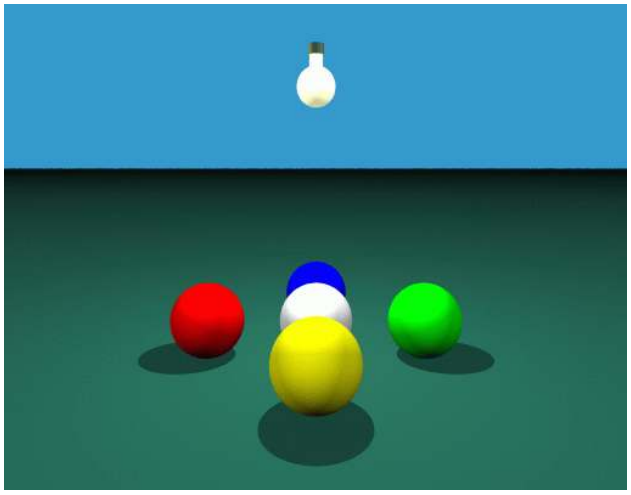
Tiefenhinweis: Verdeckung



✎ **Anmerkung:** Ein Objekt, das ein anderes verdeckt, muß sich dichter am Betrachter befinden als das verdeckte Objekt.

Weitere Tiefenhinweise geben in diesem Bild vor allem die sichtbaren Konturen. Dieses Bild ließe sich auch gut räumlich interpretieren, wenn nur die Konturen zu sehen wären (siehe Kapitel 47.2.

Tiefenhinweis: Schattierung und Schattenwurf



Wenn wir wissen, daß die Variation in der Schattierung einer Oberfläche allein auf die Beleuchtung zurückgeht, und wenn wir die Eigenschaften der Lichtquelle kennen, dann läßt sich der Verlauf der Oberfläche rekonstruieren. Auf die photometrischen Hinweise werden wir jetzt nicht näher eingehen, sonderm beim Thema „photometrische Modellierung“ darauf zurückkommen.

Tiefenhinweis: Schattierung und Schattenwurf 2

- 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

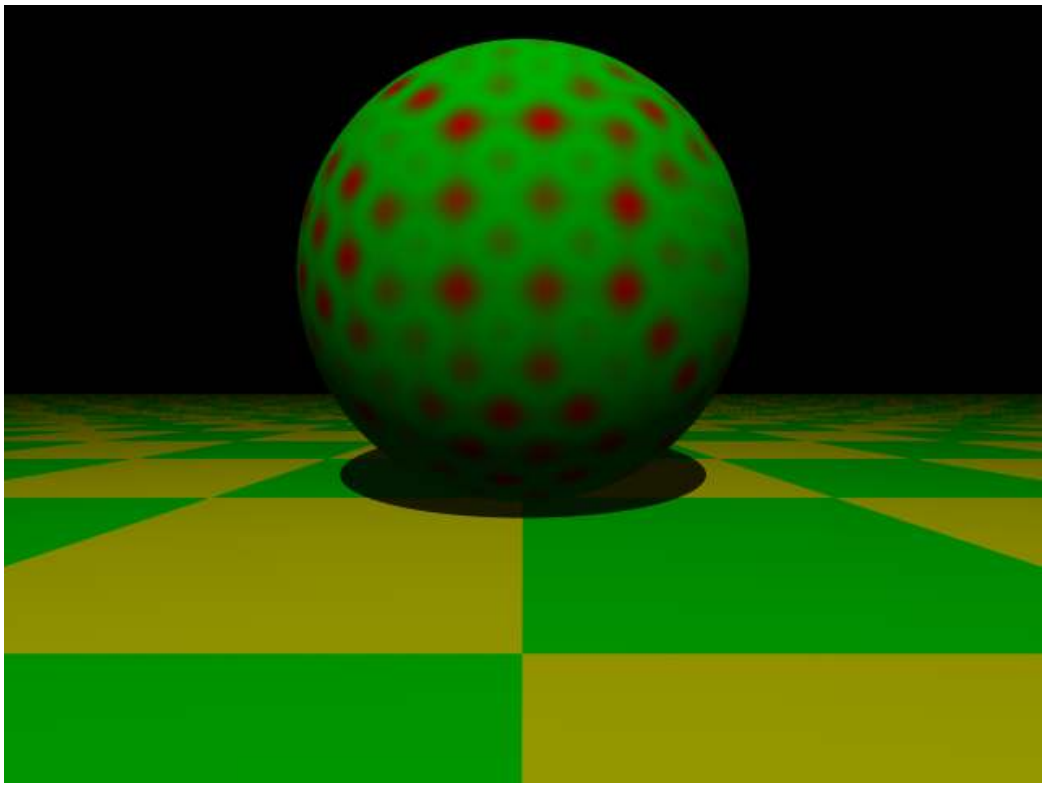
Atmosphärische Trübung

- Das von der Atmosphäre gestreute Licht (Luftlicht) bildet einen Schleier, der das Objekt umso weißer erscheinen läßt, je weiter es entfernt ist.

Atmosphärische Trübung

47.1.3 Texturgradienten

Tiefenhinweis: Texturgradient



Texturgradienten

Die *perspektivische Verkürzung* beeinflußt die Abbildung von texturierten Flächen.

Größe der Texel: Die Texel werden mit zunehmender Entfernung *kleiner*.

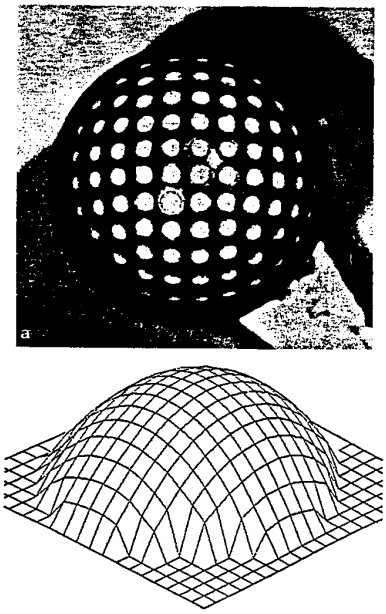
Form der Texel: Die Form der Texel hängt von der Blickrichtung ab — Kreise werden als Ellipsen abgebildet.

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 nach Barrow und Tenenbaum 1981

Texturgradient

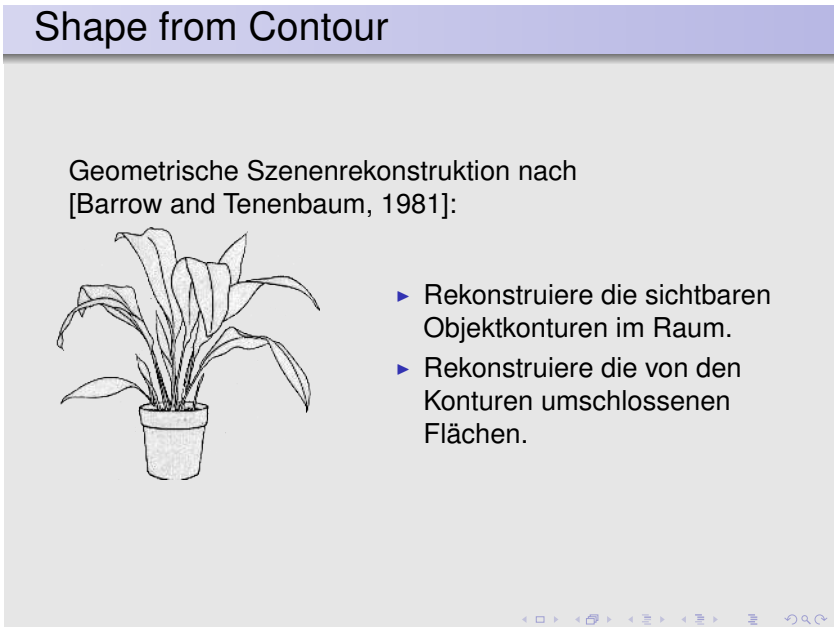
Perspektivische Verkürzung der Texel

Texturgradient:

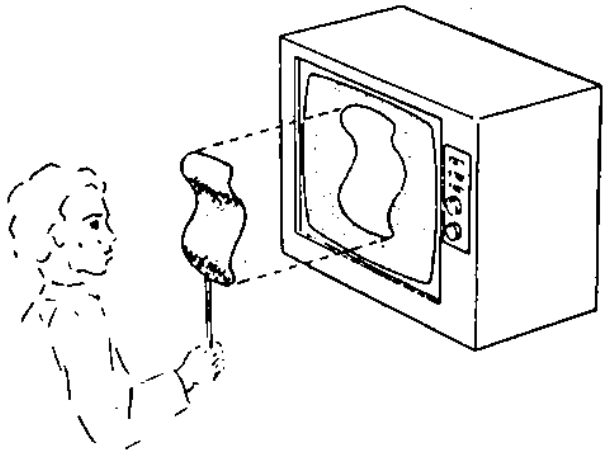
Perspektivische Verkürzung der Blüten und der Abstände

47.2 Shape from Contour

47.2.1 Rekonstruktion von Objektkonturen



Konturen und Flächen



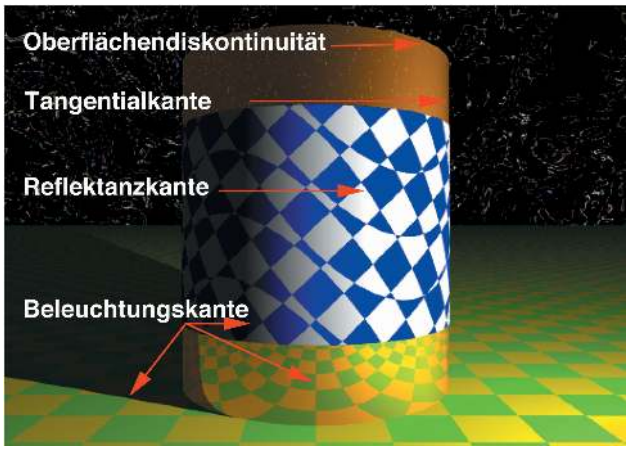
Unser Sehsystem füllt Linienzüge automatisch mit glatten Flächen.

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 Diskontinuiäten
  - der Objektoberfläche (Orientierung, Pigmentierung)
  - oder der Beleuchtung.

Shape from Contour



Arten von Bildstrukturen

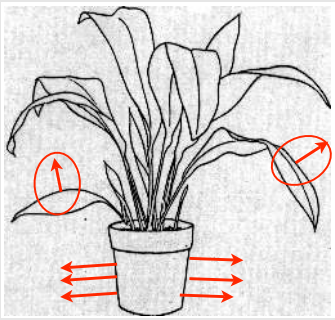
Oberflächendiskontinuitäten: (discontinuity boundaries) Die Oberflächennormale einer Fläche ändert sich sprunghaft.

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.

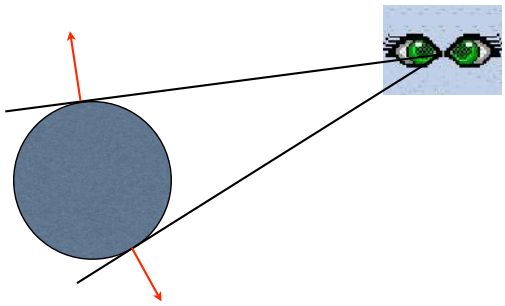
Oberflächendiskontinuitäten und Tangentialkanten



boundaries

- Tangentialkanten (extremal boundaries)
- Oberflächendiskontiniutäten (discontinuity boundaries)

Oberflächendiskontinuitäten und Tangentialkanten



Tangentialkanten

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.

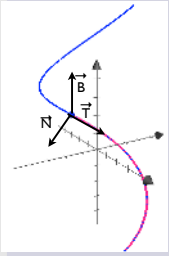
Ob eine Linie eine Tangentialkante ist, muß aus dem Kontext der anderen Bildstrukturen erschlossen werden: Symmetrien, Schattierung der Fläche usw, siehe (Barrow and Tenenbaum, 1981).

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: 103 (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* $\chi(s)$.

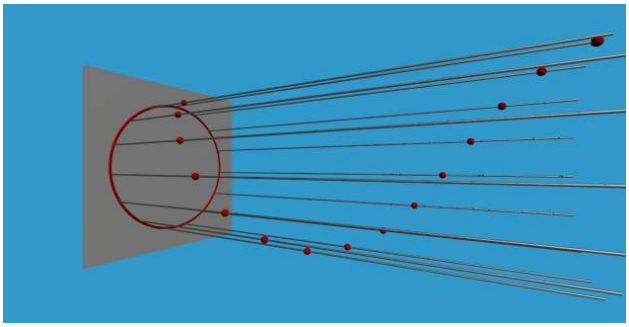
Die Krümmung beschreibt die Richtungsänderung des Tangentenvektors in der Schmiegeebene der Kurve.

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:

$$\int |\chi'(s)| ds = Minimum$$

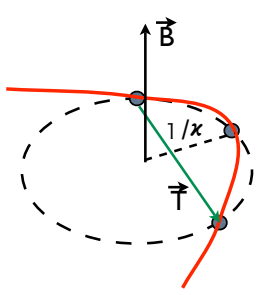
Forderung 2:

$$\int \tau(s) ds = Minimum$$

$$\int (\frac{d(\chi(s) \cdot \vec{B})}{ds})^2 ds = \int (\chi'(s)^2 + \chi(s)^2 \tau^2) ds$$

Minimisierung dieses Ausdrucks bedeutet, diejenige Raumkurve zu finden, die einem ebenen Kreisbogen am ähnlichsten ist.

Lokale Optimierung

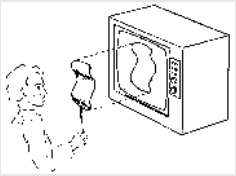


- Aus drei Punkten entlang der Kontur lassen sich lokal $\vec{T}, \vec{B}$ und χ schätzen.
- Verschiebe lokal die Punkte entlang der Projektionsstrahlen, so daß das Integral lokal minimiert wird.

Dieses Verfahren rekonstruiert Ellipsen als gekippte Kreise.

47.2.2 Rekonstruktion von Oberflächen

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

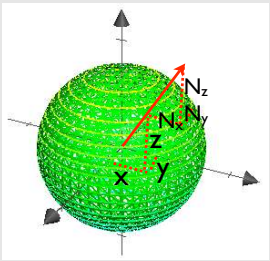
Elastische Membranen, Seifenfilme: $\chi_1 + \chi_2 = 0$

Gebogene Papierbögen: $\chi_1 \cdot \chi_2 = 0$

Ebenen, Zylinder, Kugeln: $\chi_1 =$ const., $\chi_2 =$ 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

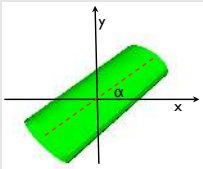


Kugeloberfläche

$$\begin{aligned} N_x &= x/R \\ N_y &= y/R \\ N_z &= z/R \end{aligned}$$

Normalenvektor $\vec{N}$:
Die Komponenten des Normalenvektors sind linear von den Bildkoordinaten x und y abhängig.

Normalen eines Zylinders mit Radius R



Zylinderoberfläche

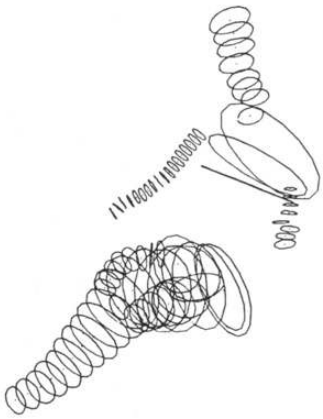
$$
\begin{aligned}
N_x &= ((y\cos\alpha - x\sin\alpha)\cdot\sin\alpha)/R \\
N_y &= ((y\cos\alpha - x\sin\alpha)\cdot\cos\alpha)/R \\
N_z &= 1 - {N_x}^2 - {N_y}^2
\end{aligned}
$$

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

(a)



(b)

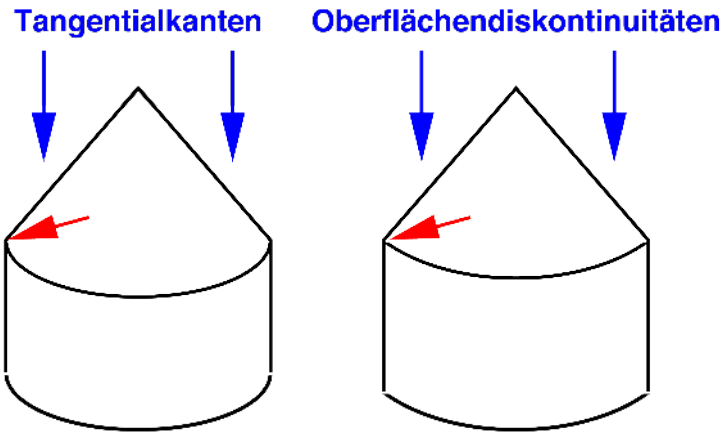
aus (Ballard and Brown, 1982) nach Agin-1972

47.2.3 Klassifikation von Konturen

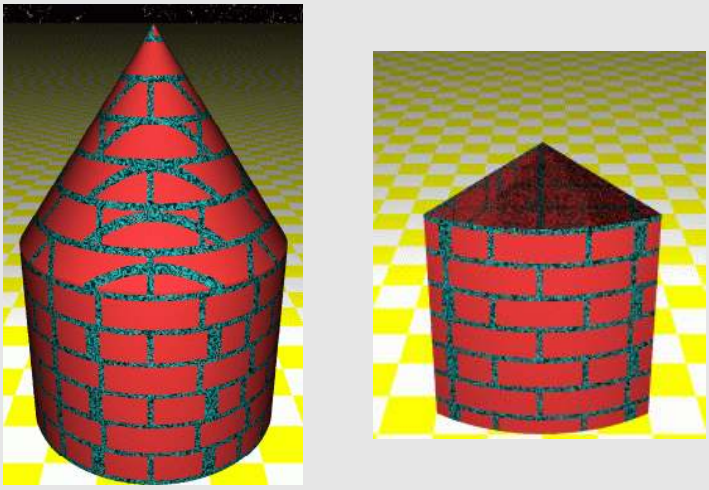
Klassifikation der Konturen

- Damit das Flächeninterpolationsverfahren angewendet werden kann, müssen Tangentialkanten erkannt werden.
- Indizien für Tangentialkanten sind beispielsweise
  - Symmetrie
  - Schattierung
  - Textur

Türmchen oder Torte?



Die von den Konturen begrenzten Flächen



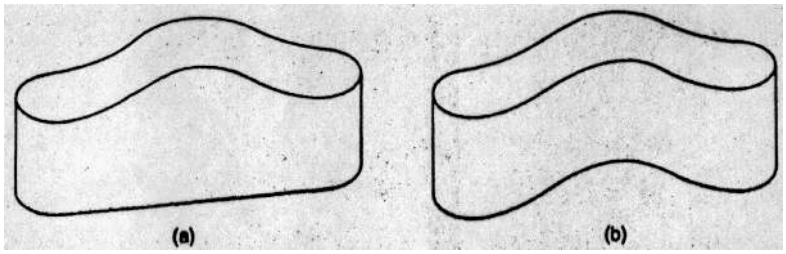
✎ **Anmerkung:** Türmchen oder Torte?

Die Konturen allein reichen nicht immer zur Klassifikation von Kanten aus, da die Details bei schlechter Auflösung schwer zu erkennen sein können. Die auf der vorherigen Folie dargestellten Konturen kann man wahlweise als das Bild eines Türmchens oder als das Bild einer Tortenscheibe interpretieren. Im ersten Fall ist das Objekt rotationssymmetrisch und die Außenkonturen sind Tangentialkanten, im zweiten Fall handelt es sich dagegen um Oberflächendiskontinuitäten. Die linke Figur stellt in der Tat ein Türmchen dar, was man daran erkennen kann, daß die Außenkonturen des Zylinders Tangenten an die Ellipsen in der Grundfläche des Zylinders darstellen, während sie diese im Fall der Tortenscheibe schneiden. Bei schlechter Bildauflösung sind solche Hinweise nicht zu erkennen.

Der Kontext einer Kontur Die folgenden Bilder zeigen, daß für die Interpretation einer Kontur der Kontext wichtig ist. Benachbarte Linien entscheiden, ob eine Linie als eben oder gekrümmt gesehen wird.

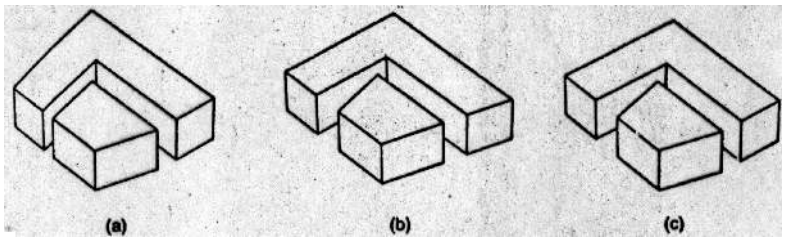
Bei der Fläche, die auf einer Seite von einer gekrümmten Linie und auf der gegenüberliegenden Seite von einer Geraden begrenzt wird, sehen wir die gekrümmte Linie als koplanar mit der Geraden, im zweiten Beispiel, wo beide Seiten von parallelen gekrümmten Linien begrenzt werden, scheinen die Linien auf einer gekrümmten Fläche zu liegen.

Der Kontext einer Kontur



planare Seitenfläche gekrümmte Seitenfläche

Im Kontext eines benachbarten Objektes



Die drei Linienzeichnungen der drei Trapeze a-c sind identisch. Der Kontext des L-Blockes entscheidet, ob wir den Winkel zwischen den bei im Trapez rechte Winkel sehen oder nicht.

Beim Vexierbild der „alten und der jungen Frau“ ändern sich die sichtbaren Orientierungen abhängig von der Interpretation des Bildes. Dieselbe Fläche kann – je nach Interpretation – einmal uns zugewendet und einmal von uns weggewendet erscheinen. Die Nase der alten Frau zeigt auf uns zu, aber wenn wir dieselbe Struktur als Wange der jungen Frau sehen, dann zeigt sie von uns weg, ohne daß sich etwas am Kontext der Linien im Bild geändert hat.

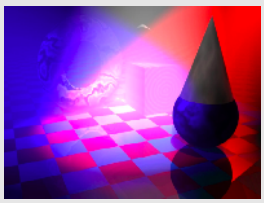
Abhängigkeit von der Interpretation

Die alte und die junge Frau

48 Photometrische Szenenrekonstruktion

48.1 Photometrische Bildentstehung (image formation)

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.

Die hier besprochenen Verfahren zur photometrische Szenenrekonstruktion gehen auf die grundlegenden Arbeiten von Horn und seinen Kollegen am MIT zurück, siehe dazu vor allem das Lehrbuch (Horn, 1986, Kap. 10,11), (Woodham, 1981), (Ikeuchi and Horn, 1981).

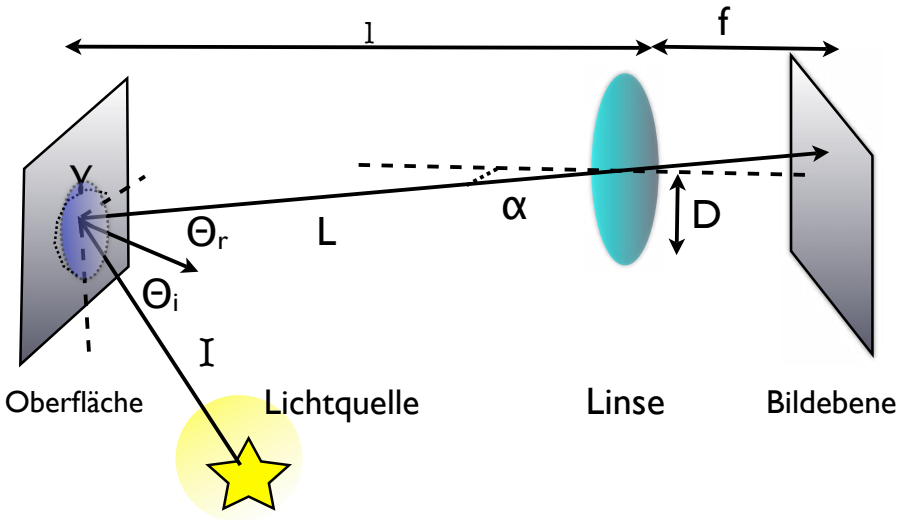
Um aus den Bildhelligkeiten auf die Oberflächenform zurückschließen zu können, müssen wir wissen, wie die Bildhelligkeiten entstehen:

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



48.1.1 Messen von Strahlungswerten

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.

Bei Glassner, 1995 finden Sie eine ausführliche Übersicht zu den drei Systemen. Glassner diskutiert auch den Nachteil, der sich aus der Verwendung von skalaren Größen ergibt: Alle Werte müssen explizit auf Raumwinkel bezogen werden und die flächenbezogenen Größen müssen auf die wirksamen Flächenquerschnitte projiziert werden. All dieses macht den Umgang mit diesen Größen sehr, sehr umständlich.

Sie sollten das photometrische System verwenden, wenn sie es mit von Menschen empfundenen Intensitäten zu tun haben, insbesondere die Helligkeiten von Lampen und die Empfindlichkeit von Kameras wird häufig in photometrischen Größen angegeben. Im Unterschied zu den physikalischen Größen wie Energie und Leistung wird hier nicht die Strahlungsleistung im gesamten Spektrum berücksichtigt sondern nur die als sichtbares Licht abgestrahlte Energie.

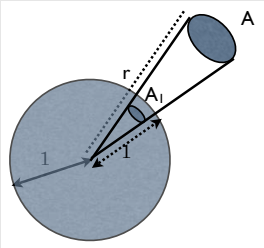
Für die folgenden Betrachtungen werden wir das radiometrisches System verwenden.

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):

$$P = \frac{dQ}{dt}$$

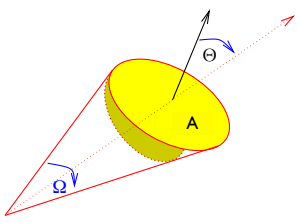
Strahlstärke I: Strahlungsleistung je Raumwinkelelement (radient intensity)

$$\mathrm{I} = \frac{dP}{d\omega}$$

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 Raumwinkelelement ω. und pro Flächenelement A senkrecht zum Strahl an.

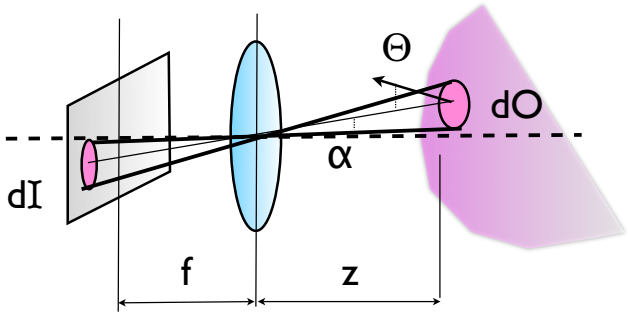
Strahldichte L (radiance)

$$L = \frac{d^2 P}{d\omega dA}$$

$$\omega = \frac{A \cos \vartheta}{R^2}$$

48.1.2 Modellierung des Sensors

Modellierung des Sensors: Bildintensität



- Die Helligkeit (oder Intensität) einer Fläche im Bild ist proportional zu der auf diese Fläche eingestrahlten Leistung (Bestrahlungsstärke E_i).

$$\text{Intensität} \sim \text{Bestrahlungsstärke } E_i$$

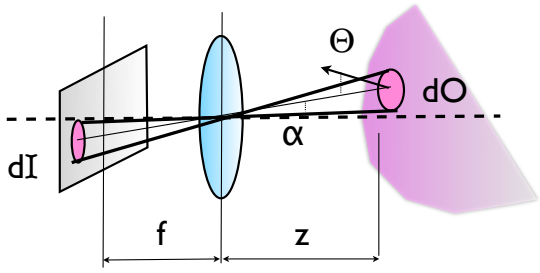
Berechnung der Strahlstärke E_i

Problem: 104 (Gesucht:)
Bestrahlstärke E_i der Bildfläche dI

Gegeben: Sei

- L : Strahldichte L (radiance) 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 $d\omega_i = d\omega_o$



$$
\begin{aligned}
d\omega_i &= d\omega_o \\
d\omega_i &= \frac{dI \cdot \cos\alpha}{(f/\cos\alpha)^2} \\
d\omega_o &= \frac{dO \cdot \cos\vartheta}{(z/\cos\alpha)^2} \\
\frac{dO}{dI} &= \frac{\cos\alpha}{\cos\vartheta}(\frac{z}{f})^2
\end{aligned}
$$

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:

$$
\omega = \frac{\pi}{4} \cdot \frac{d^2 \cos\alpha}{(z/\cos\alpha)^2} = \frac{\pi}{4} \cdot (\frac{d}{z})^2 \cdot \cos^3\alpha
$$

- Die Strahlungsleistung dP, die von der Objektfläche dO aus unter dem Winkel ϑ durch die Linse geht:

$$
\begin{aligned}
dP &= L \cdot \omega \cdot dO \cdot \cos\vartheta \\
&= L \cdot dO \cdot \frac{\pi}{4} \cdot (\frac{d}{z})^2 \cdot \cos^3\alpha \cos\vartheta
\end{aligned}
$$

Die Bestrahlstärke E_i des Bildelements $d\mathrm{I}$

Die Linse richtet die Strahlungsleistung dP auf ein Bildelement mit der Fläche $d\mathrm{I}$:

- Die Bestrahlstärke E_i der Bildfläche $d\mathrm{I}$:

$$\begin{aligned} E_i = \frac{dP}{d\mathrm{I}} &= L \cdot \underbrace{\frac{dO}{d\mathrm{I}}}_{\frac{\cos\alpha}{\cos\vartheta}(\frac{z}{f})^2} \cdot \frac{\pi}{4} \cdot (\frac{d}{z})^2 \cdot \cos^3\alpha \cos\vartheta \\ E_i &= L \cdot \frac{\pi}{4} \cdot (\frac{d}{f})^2 \cdot \cos^4\alpha \end{aligned}$$

Ergebnis

Satz: 105 (Bestrahlstärke E_i eines Bildelements)

$$E_i = L \cdot \frac{\pi}{4} \cdot (\frac{d}{f})^2 \cdot \cos^4\alpha$$

Die Bestrahlstärke E_i eines Bildelements $d\mathrm{I}$ 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: 106 (*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.

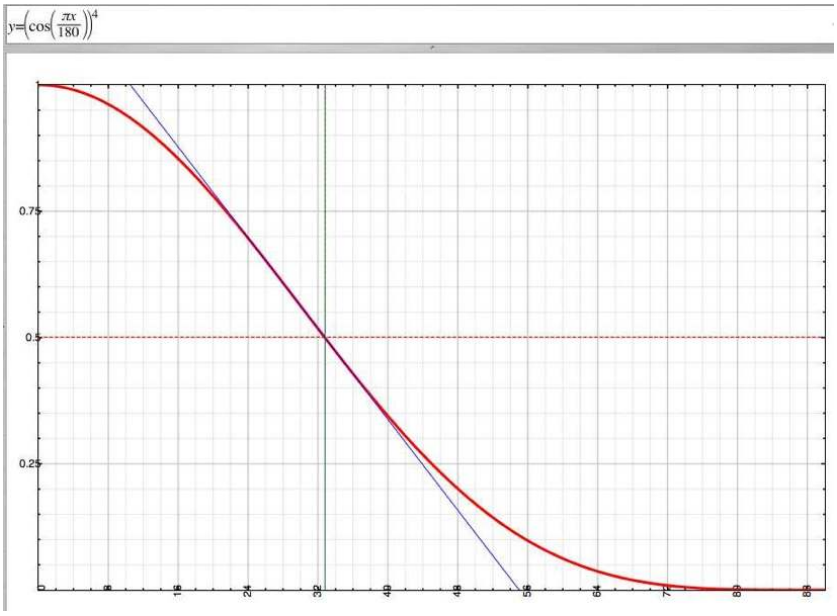
Vignetierung

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: 107 (*Natürliche Vignetierung*)
Die Randverdunklung proportional zu $\cos^4\alpha$ heißt natürliche Vignetierung.

Natürliche Vignetierung: $\cos^4 \alpha$



Wir wissen jetzt, wie ein Objektiv die durch die Öffnung einfallende Strahldichte L in Bildhelligkeiten proportional zur Bestrahlstärke E_i am Bildwandler umsetzt.

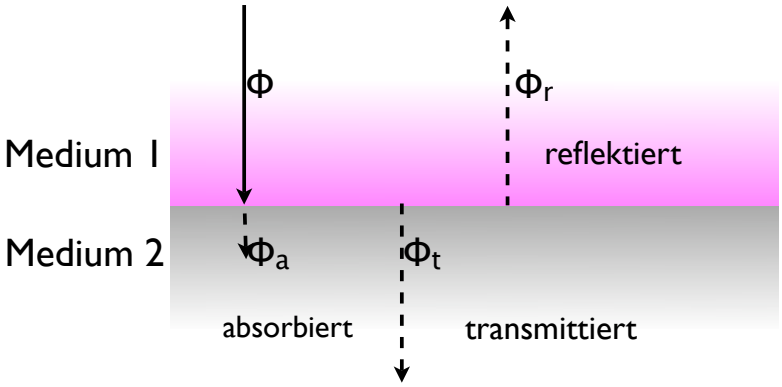
Jetzt soll modelliert werden, wovon die am Objekt abgestrahlte Strahldichte L abhängt. Hierfür müssen wir ein lokales Beleuchtungsmodell definieren, das die Reflektanz des Objektes beschreibt. Für die photometrische Szenenrekonstruktion ist eine sehr genaue Kenntnis des Reflexionsverhaltens der Objektoberflächen nötig.

48.1.3 Modellierung der Oberfläche

Strahlungsfluß an der Grenzfläche zweier Medien

Der einfallende Strahlungsfluß wird entweder

- reflektiert, absorbiert oder transmittiert.



$$\Phi = \Phi_r + \Phi_a + \Phi_t$$

Abhängigkeit von der Wellenlänge, Beispiel Laub

Laub reflektiert stark im Infrarotbereich und absorbiert stark im Rotkanal.

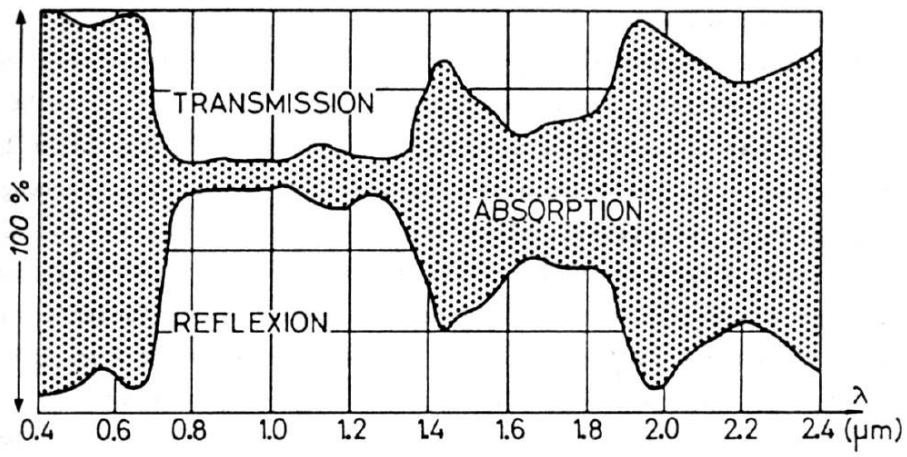
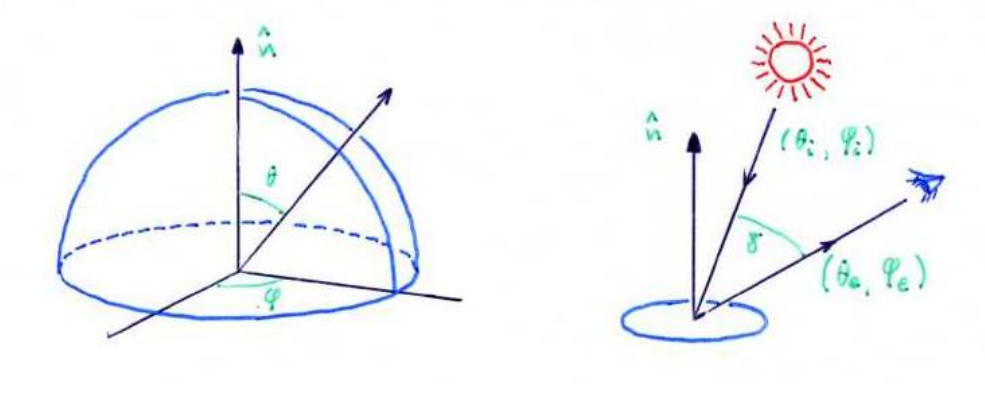


Bild 2.5-10: Reflektierte, absorbierte und transmittierte Strahlung als Anteile der auf ein Laubbaumblatt auftreffenden Strahlung (Manual of Remote Sensing, 1983)

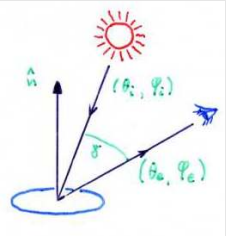
Die am Objekt reflektierte Strahldichte L



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.

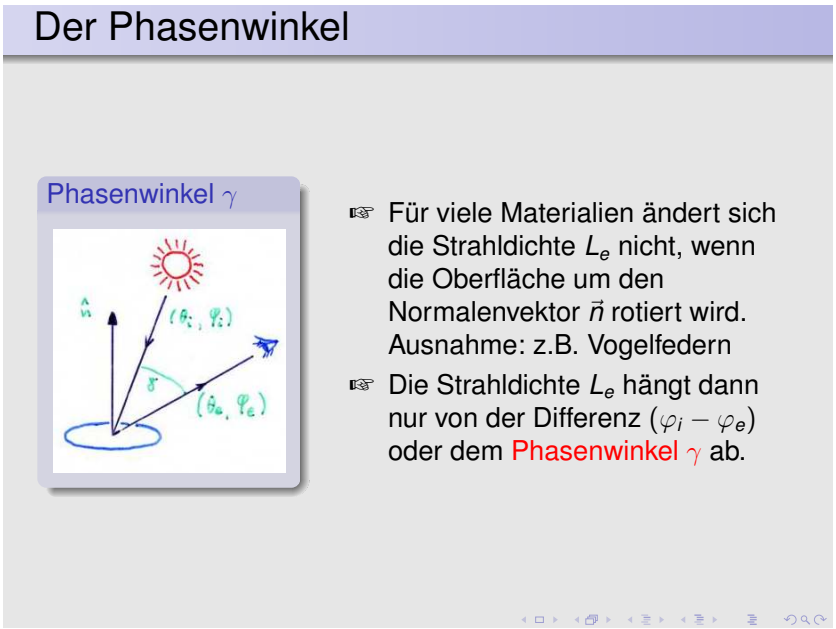
Die Reflexionsfunktion (BDRF)

Definition: 108 (***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 $L_e(\vartheta_e, \varphi_e)$.



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

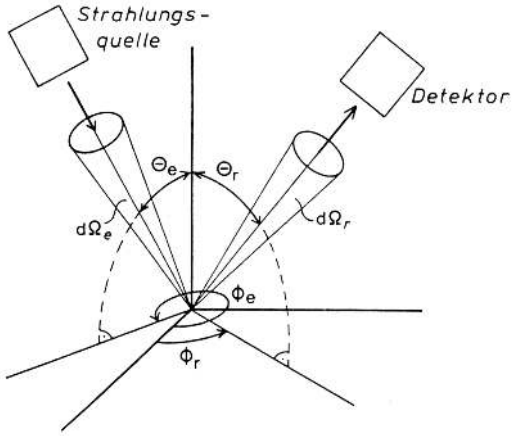


Bild 2.5-5: Messung der Reflexionsfunktion

Die BDRF muß durch Messung der Strahldichte für alle Winkelkombinationen $\vartheta_i, \vartheta_e, \varphi_i, \varphi_e)$ ermittelt werden.

48.1.4 Beispiele für Reflexionsfunktionen

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.

$$\begin{aligned}
\vartheta_i &= \vartheta_e \\
\gamma &= \pi \\
\phi(\vartheta_i, \vartheta_e, \varphi_i - \varphi_e) &= \delta(\vartheta_i - \vartheta_e) \cdot \delta(\pi - (\varphi_i - \varphi_e)) \\
\text{Kronneckersymbol } \delta(x) &= \begin{cases} 0 & x \neq 0 \\ 1 & x = 0 \end{cases}
\end{aligned}$$

Lambert'scher Strahler

Die ideale matte Oberfläche
(Lambert'scher Strahler):

$$\phi(\vartheta_i, \vartheta_e, \gamma) \quad = \quad \cos(\vartheta_i)$$

- 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 Ausfallswinkel 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 Ausfallswinkel 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:

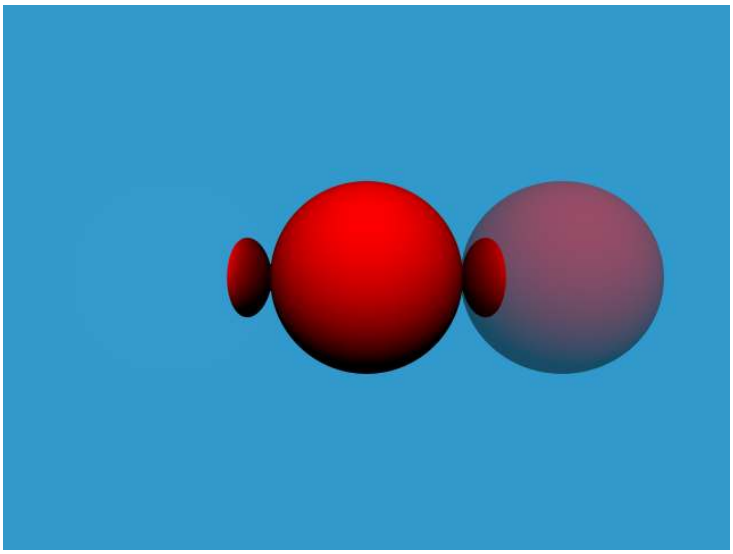
Beispiel: 109

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}
```

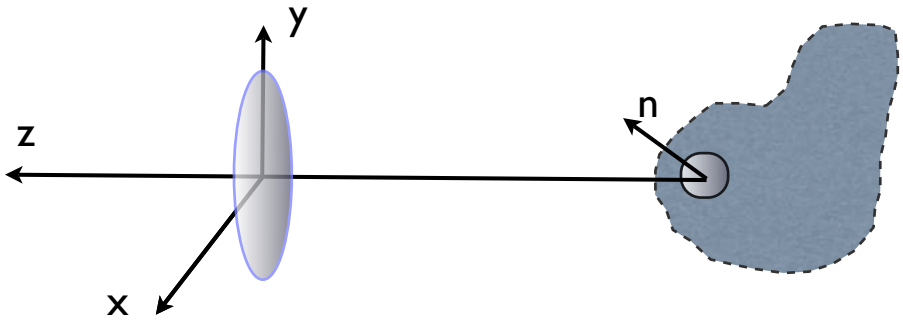
Beispiel Drei Kugeln mit den Oberflächen

```
finish{Mirror}, finish{Lambert}, finish{Mixed}
```



48.1.5 Die Reflektanzkarte

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 $\vec{s} \approx \hat{s} = (-p_s, -q_s, 1)^T$:

$$\begin{aligned}
\cos\angle(\vec{a}, \vec{b}) &= \frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|} \\
\cos\vartheta_e = \cos\angle(\vec{n}, \vec{v}) &= \frac{1}{\sqrt{1 + p^2 + q^2}} \\
\cos\vartheta_i = \cos\angle(\vec{n}, \vec{s}) &= \frac{1}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}}
\end{aligned}$$

Die Reflexionsfunktion ist materialspezifisch, aber unabhängig von einer speziellen Beleuchtungs- und Betrachterorientierung. Die Reflektanzkarte dagegen gibt die Strahldichte für eine bestimmte Orientierung von Betrachter und Lichtquelle an.

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:

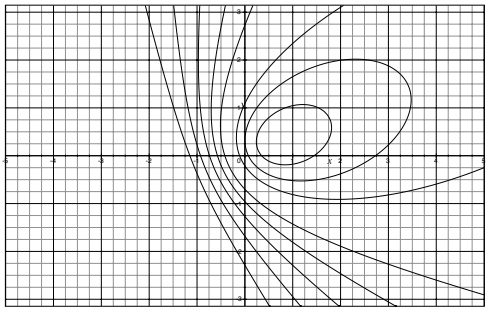
$$R(p,q) = \frac{1 + p_s p + q_s q}{\sqrt{1+p_s^2+q_s^2}}$$

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 $(p,q) = (p_s, q_s)$,

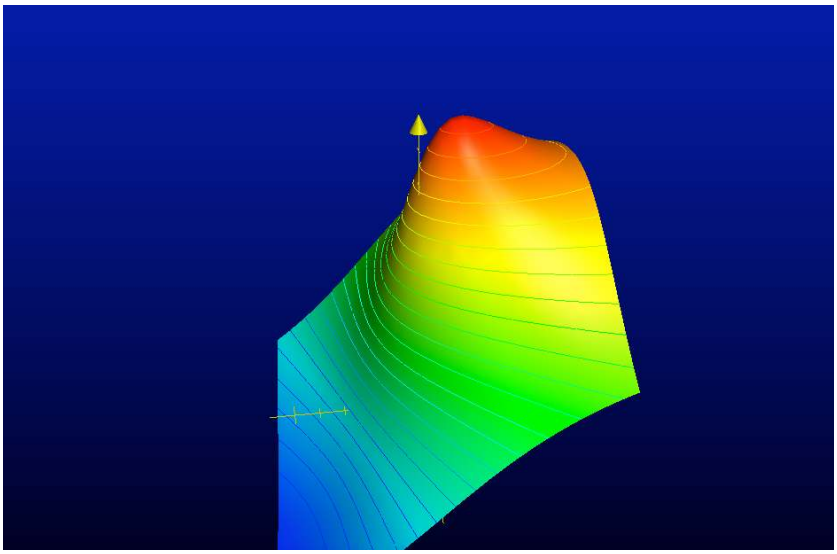
Beispiel: $R_L(p,q)$ Lambert'scher Strahler



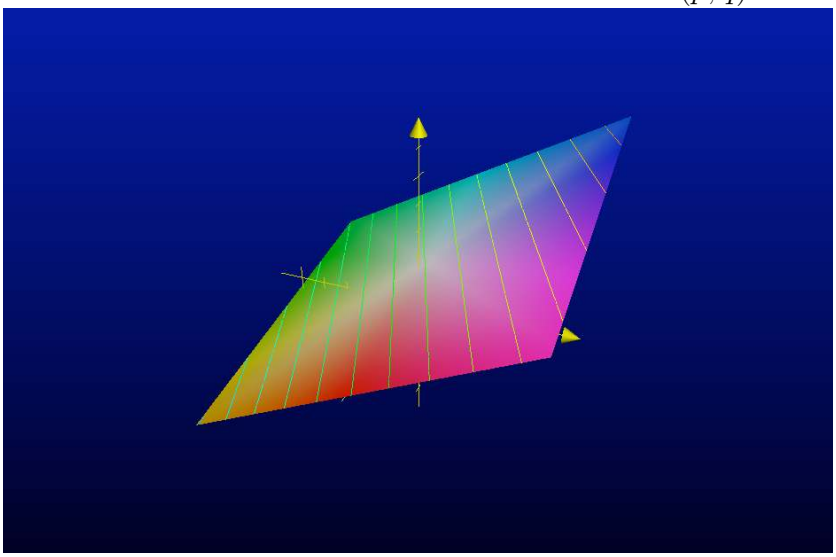
$R_L(p,q)$ mit $\rho = 1.0, p_s = 0.7, q_s = 0.3$

nach (Woodham, 1981)

3D-Plot der Reflektanzkarte $R_L(p, q)$



3D-Plot der Reflektanzkarte $R(p, q)$



48.2 Photometrisches Stereo

Rekonstruktion der Oberflächenorientierung

Satz: 110 (Bildintensität und Reflektanzkarte)
Die Intensität $\mathrm{I}(x, y)$ *im Bild ist proportional zur Reflektanzkarte* $R(p, q)$*:*

$$\mathrm{I}(x, y) = c \cdot R(p, q)$$

☞ 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.

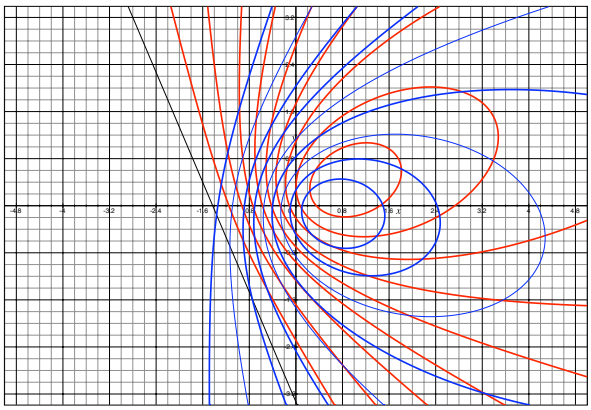
Photometrisches Stereo

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]

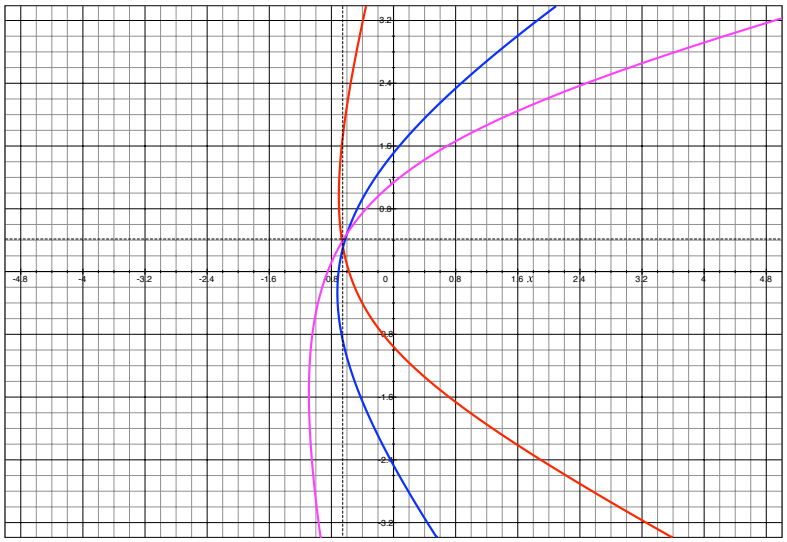
- Wenn die Reflektanzkarte linear ist und die Isolinien sich schneiden, existiert eine eindeutige Lösung.
- Bei nicht-linearen Reflektanzkarten können weitere Aufnahmen erforderlich sein.

Photometrisches Stereo



$R_L(p, q)$ mit $\rho = 1.0, p_{s1} = 0.7, q_{s1} = 0.3, p_{s2} = 0.6, q_{s2} = -0.1$

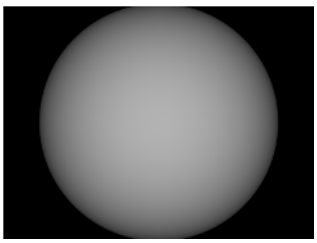
Eindeutige Lösung mit drei Aufnahmen



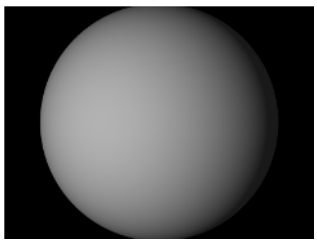
$R_L(p, q)$ mit $\rho = 1.0, p_{s1} = 0.7, q_{s1} = 0.3,$

$p_{s2} = 0.6, q_{s2} = -0.1, p_{s3} = 0.5, q_{s3} = -0.3$

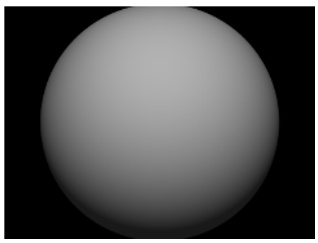
Beispiel: Eine Kugel mit drei Reflektanzkarten



$< 0, 0, -20 >$



$< -10, 0, -20 >$



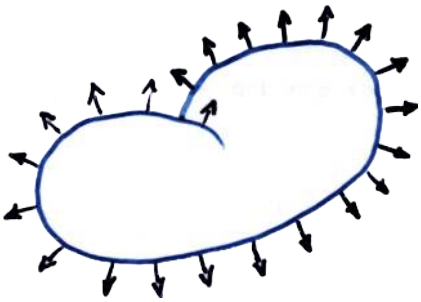
$< 0, 10, -20 >$

Als RGB-Mischung der drei Bilder:

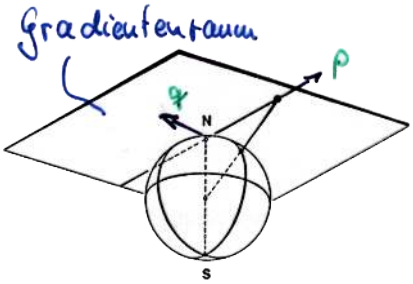
48.3 Shape from Shading

Rekonstruktion mit nur einer Aufnahme

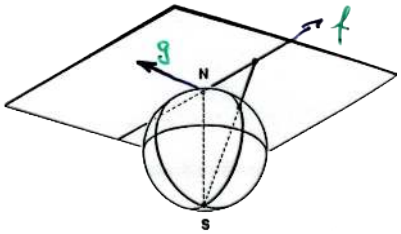
- ☞ Im Allgemeinen gibt es mit nur einem Bild keine eindeutige Rekonstruktion der Oberfläche: *ill posed problem.*
- ☞ Wir nehmen zusätzlich an, daß die Oberfläche glatt ist und die Objektränder Tangentialkanten sind.
- ☞ Dann ist die Oberflächenorientierung an den Rändern bekannt und kann als Startwert zur Rekonstruktion der Fläche genutzt werden.
- ☞ Im Gradientenraum ist allerdings ausgerechnet diese Orientierung an der Kontur nicht darstellbar.



Stereografische Projektion



gnomonisch



stereografisch

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: 111 (Image Irradiance Equation)

$$I(x,y) \quad = \quad R(f,g)$$

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: 112 (*Regularisierung*)

$$\begin{array}{ccc} I(x,y) & = & R(f,g) \\ \textit{ill-posed Problem} & \overbrace{\Longrightarrow}^{\textit{Regularisierung}} & \textit{well- posed Problem} \end{array}$$

Die Regularisierung schränkt den Lösungsraum auf „glatte“ Lösungen ein.

Problem: 113 (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: 114
Eine mögliche Glattheitsforderung ist:

$$e_s = \int\int((f_x^2 + f_y^2) + (g_x^2 + g_y^2))dxdy \rightarrow \textit{Minimum}$$

$$f_x, f_y, g_x, g_y : \textit{Ableitungen nach} x, y.$$

Lösungsverfahren

Sowohl die Bildintensitäten als auch die Reflektanzkarte sind fehlerbehaftet. Daher suchen wir keine exakte Lösung sondern minimieren den quadratischen Fehler:

$$e_i = \int\int (I(x,y) - R(f,g))^2 dxdy \rightarrow \text{Minimum}$$

durch Minimierung des Integrals

$$e_i = \int\int F(f, g, f_x, f_y, g_x, g_y) dxdy$$

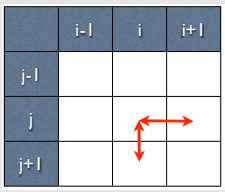
mit

$$F = (f_x^2 + f_y^2) + (g_x^2 + g_y^2) + \lambda(I(x,y) - R(f,g))^2$$

und λ als Wichtungsfaktor.

Der Faktor λ gewichtet, wie stark der relative Einfluß der Glattheitsforderung und der Bildinformation ist.

Der diskrete Fall



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:

$$r_{i,j} = (I_{i,j} - R(fi,j, gi,j))^2$$

Der zu minimierende Gesamtfehler e:

$$e = \sum_i \sum_j (s_{ij} + \lambda r_{i,j}) \rightarrow \text{Minimum}$$

✎ **Anmerkung:** Minimieren des Fehlers: Als nächstes werden diejenigen Werte f, g gesucht, die das Fehlermaß e minimieren. Dazu wird das Fehlermaß nach den Zeilen und Spalten differenziert. Für die Zeilen und Spalten werden die Koordinaten k und l eingeführt, um diese nicht mit den Indizes i, j der Fehlersumme zu verwechseln. Das shape-from-shading Verfahren und die zugrunde liegende Variationsrechnung sind besonders gut im Lehrbuch von Horn, 1986 beschrieben, das einen sehr nützlichen mathematischen Anhang hat.

Minimierung des Fehlers

Die Ableitung von e nach $f_{k,l}$ und $g_{k,l}$:

$\frac{1}{4}\cdot$

	-1	
-1	4	-1
	-1	

$$\begin{aligned}
\frac{\partial e}{\partial f_{k,l}} &= 2(f_{k,l} - \bar{f}_{k,l}) - 2\lambda(I_{k,l} - R(f_{k,l}, g_{k,l}))\frac{\partial R}{\partial f} \\
\frac{\partial e}{\partial g_{k,l}} &= 2(g_{k,l} - \bar{g}_{k,l}) - 2\lambda(I_{k,l} - R(f_{k,l}, g_{k,l}))\frac{\partial R}{\partial g}
\end{aligned}$$

mit

$$\begin{aligned}
\bar{f}_{i,j} &= \frac{1}{4}(f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1}) \\
\bar{g}_{i,j} &= \frac{1}{4}(g_{i+1,j} + g_{i,j+1} + g_{i-1,j} + g_{i,j-1})
\end{aligned}$$

$f_{i,j} - \bar{f}_{i,j}$ ist eine Approximation des Laplaceoperators $\nabla 2 f$.

Iterative Lösung: Relaxation

Die Ableitungen werden gleich Null gesetzt und die Gleichungen umgestellt:

$$\begin{aligned}
f_{k,l}^{n+1} &= \bar{f}_{i,j}^{n} + \lambda(I_{k,l} - R(f_{i,j}^{n}, g_{i,j}^{n}))\frac{\partial R}{\partial f} \\
g_{k,l}^{n+1} &= \bar{g}_{i,j}^{n} + \lambda(I_{k,l} - R(f_{i,j}^{n}, g_{i,j}^{n}))\frac{\partial R}{\partial g}
\end{aligned}$$

☞ In jedem Iterationsschritt werden $f_{i,j}^{n}, g_{i,j}^{n}$ durch $\bar{f}_{i,j}^{n}, \bar{g}_{i,j}^{n}$ an die Umgebung angepaßt.

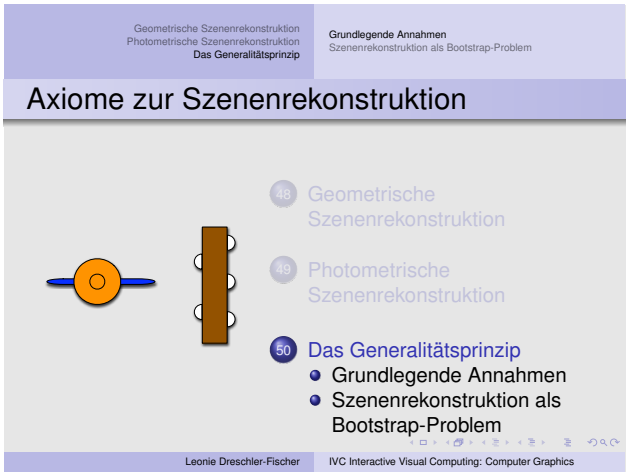
☞ Die Startwerte an den Tangentialkanten werden so in die glatten Gebiete des Bildes propagiert.

✎ **Anmerkung:** Regularisierung und Relaxation

Die Regularisierung über eine Glattheitsforderung ist typisch für die shape-from Verfahren. In ähnlicher Weise können auch shape-from-texture und shape-from-contour regularisiert werden. Horn hat dieses Verfahren auch zur Lösung des Blendenproblems eingesetzt. Wir werden in der im Kapitel über den optischen Fluß (siehe Kapitel Abschnitt 50 darauf zurückkommen.

49 Das Generalitätsprinzip

49.1 Grundlegende Annahmen



Eine wichtige Grundlage für die Formulierung von Regeln zur niederen Bilddeutung, die auch zur Einschränkung von möglichen Zuordnungen von Bildmerkmalen beim dynamischen Stereo verwendet werden können, bildet das von Binford, 1981 begründete *Generalitätsprinzip*.[5] Dieses besagt, daß die in einem Bild detektierten Strukturen jeweils durch das allgemeinste mögliche Modell erklärt werden sollten.

[5] Das zugrundeliegende Prinzip, daß die beste Erklärung eines Phänomens immer die einfachste, auf am wenigsten Annahmen beruhende sei, ist seit dem 14. Jahrhundert nach dem englischen Philosophen William of Ockham bekannt als *Ockham's razor*.

Hieraus lassen sich insbesondere zwei Heuristiken ableiten:

- *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.

Hierauf aufbauend können wir eine weitere Heuristik formulieren, die bei der Auswertung von Bild*sequenzen* hilfreich ist:

- *Das Prinzip der allgemeinen Bewegung*
  Eine spezielle Sicht des Beobachters oder eine spezielle Position der Lichtquelle ist für Bilder einer Sequenz die Ausnahme, nicht die Regel. (Dreschler-Fischer, 1987)

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)

Diese Regeln wurden von Neumann, 1982 für einen Übersichtsvortrag auf der GWAI-1982 zusammengestellt.

Kohärenz

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.

Persistenz

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 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.

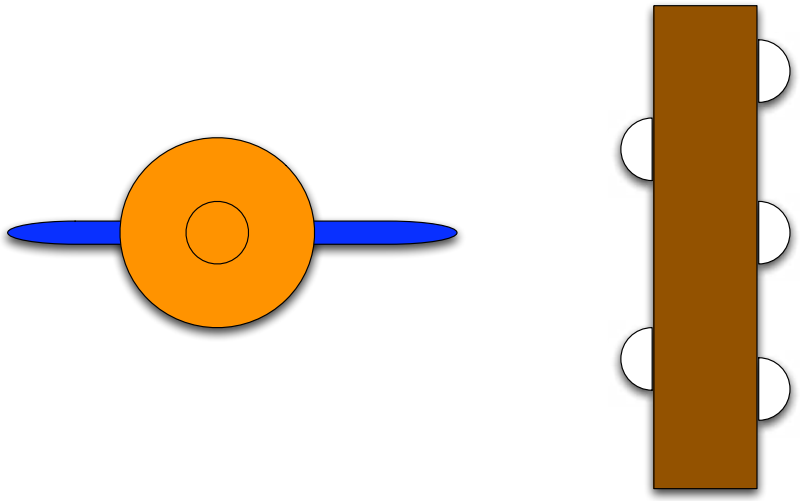
Ockham's razor

William of Ockham:

$\approx$ 1285-1349 Die beste Erklärung eines Phänomens ist immer die einfachste, auf am wenigsten Annahmen beruhende.

Untypische Ansichten

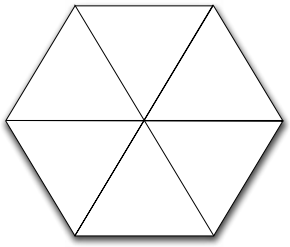
Untypische Ansichten sind auch für Menschen schwer zu deuten:



Das vorherige Bild skizziert beliebte „Drudel“:

- ein Mexikaner auf dem Fahrrad, von oben gesehen, so daß nur der Sombrero zu sehen ist.
- ein Bär, der einen Baum hochklettert, von der abgewandten Seite gesehen — nur ein Ohr und die vier Tatzen sind zu erkennen.
-

Beispiel: eine symmetrische Figur

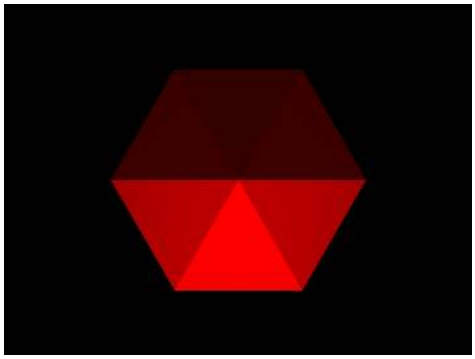


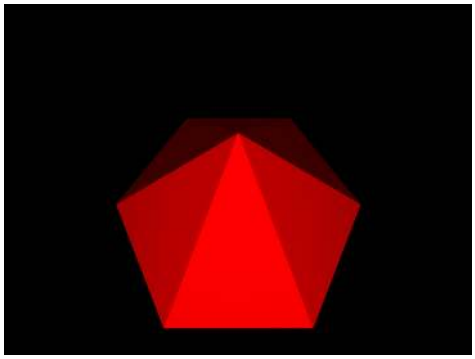
Wenn dieses Bild von einem *allgemeinen Standort* aus aufgenommen wurde, kann es keine Pyramide sein:

- ☞ 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





Allgemeines Modell

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*.

Oberflächen

Annahme 6: Glattheit von Oberflächen
Physikalische Oberflächen sind i.A. glatt.

- Diese Annahme ist die Basis für die Bildsegmentierung über Homogenitätskriterien.

Parallelität

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.

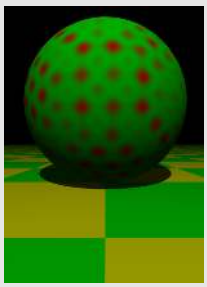
Zur Ames-Raum-Illusion siehe Unterabschnitt 13.2, Seite 138, `http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/amesPerspective.html`.

Annahme 8:
Parallele Kurven im bleiben parallel, wenn sie auf der Oberfläche leicht verschoben werden.

Annahme 9:
Parallele Kurven liegen lokal auf einem Zylinder.

Beiden Annahmen folgen aus der Glattheit von Oberflächen.

Textur



48 Geometrische Szenenrekonstruktion

49 Photometrische Szenenrekonstruktion

50 Das Generalitätsprinzip
- Grundlegende Annahmen
- Szenenrekonstruktion als Bootstrap-Problem

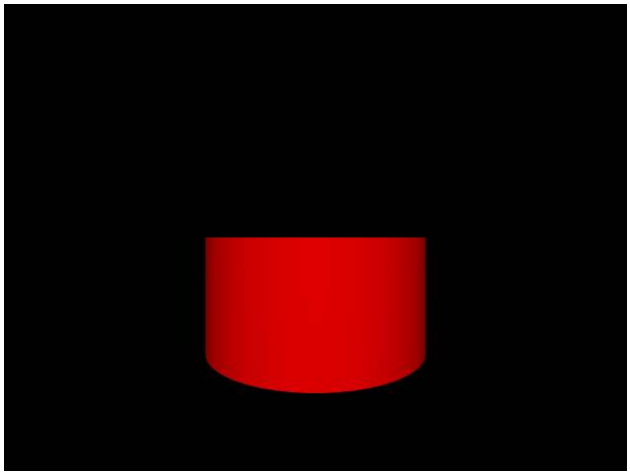
☞ Ein Texturgradient geht in der Regel auf perspektivische Verkürzung zurück, nicht auf spezielle Muster der Oberfläche.

Geraden

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

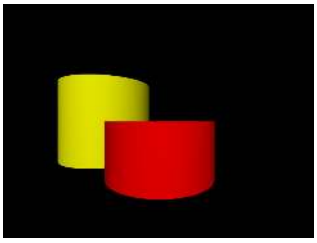


Der obere Deckel des Zylinders und die optische Achse der Kamera liegen in einer Ebene, so daß der Rand des Deckels als Gerade abgebildet wird.

Annahme 12: Parallele Geraden
Parallele Geraden im Bild sind auch im Raum parallel.

Diese Annahme folgt aus Annahme 7 über parallele Kurven.

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.

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.

Zur Illusion des unmöglichen Objektes: siehe 13.3, Seite 141. http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/unmoeglich.html

Starrheit

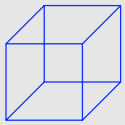
Annahme 15: Starrheit
Physikalische Objekte sind starre Körper, solange nichts auf das Gegenteil hindeutet.

- 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 her wechseln.

- Ein berühmtes Beispiel ist der Necker-Würfel, die Zeichnung eines Drahtrahmenwürfels in Parallelprojektion, benannt nach dem Schweizer Naturforscher L. A. Necker (1832).

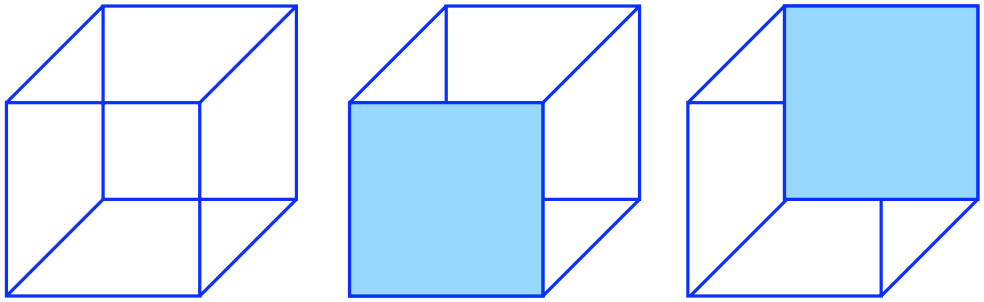


- Dieses Bild hat zwei verschiedenen Interpretationen als Kubus.
- Die Mehrdeutigkeit geht darauf zurück, daß die Flächen fehlen, sie liegt nicht an der Parallelprojektion.

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/necker.html

Weitere Illusionen siehe: http://www.michaelbach.de/ot/sze_Necker/index.html

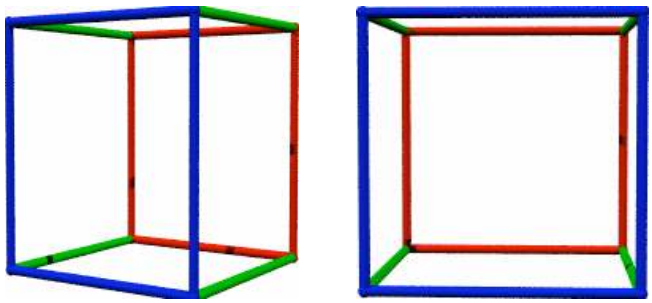
Mögliche Interpretationen



Eine ähnliche Mehrdeutigkeit tritt auf, wenn wir einen echten Drahtrahmenwürfel mit nur einem Auge betrachten. Wir können ihn dann wahlweise als Würfel oder als Pyramidenstumpf sehen.

Ein Drahtrahmenwürfel in Zentralprojektion

Wenn wir die rote Fläche vorne liegen sehen, dann sehen wir anstelle des Würfels einen Pyramidenstumpf.



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.

Bewegung

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.

49.2 Szenenrekonstruktion als 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. Diese Verifikation ist bei szenenspezifischen Verfahren und eingeschränkten Szenen in der Regel unproblematisch, aber bei natürlichen Szenen ein noch ungelöstes Problem. Beispielsweise ist einer Textur mit einer systematischen Variation der Dichte nicht ohne weiteres anzusehen, ob die beobachtete Dichtevariation durch perspektivische Effekte verursacht wird oder ob die Oberfläche von sich aus unterschiedlich stark texturiert ist, wie beispielsweise das Fell eines Leoparden. Um zu prüfen, ob „shape-from-texture“ anwendbar ist, müßte man zuvor die Orientierung der Oberfläche im Raum ermitteln. Dasselbe trifft in ähnlicher Form auch auf die anderen „shape-from…“ Methoden zu. Das heißt:

> Um Objektstrukturen aus Bildstrukturen geometrisch zu rekonstruieren, müssen die erzeugenden Objektstrukturen zumindestens qualitativ schon bekannt sein.

Da natürliche Szenen in der Regel komplex sind und Objektstrukturen der unterschiedlichsten Art vorkommen können, stellt das Problem der Verifikation einen Teufelskreis dar, der nicht einfach zu durchbrechen zu sein scheint. Dieses Problem wollen wir *Bootstrap-Problem* nennen, da Rekonstruktionen nur auf Rekonstruktionen aufbauen können und es darum gehen muß, einen Einstieg in den Kreis der gegenseitigen Abhängigkeiten zu finden.

Hier hilft die folgende Beobachtung weiter: Die einzelnen Bildstrukturen lassen zwar, für sich allein betrachtet, in der Regel mehrere Interpretationen zu, aber durch den räumlichen und zeitlichen Kontext könnten die Mehrdeutigkeiten eventuell reduziert werden, da die geometrische Interpretation einer Bildstruktur nicht der Interpretation von damit verbundenen Strukturen widersprechen darf.

Je nach Art der Bildmerkmale können also unterschiedliche Rekonstruktionsmethoden eingesetzt werden – trigonometrische Verfahren für genau lokalisierbare Strukturen, wie Kanten und Eckpunkte, und photometrische oder statistische Verfahren für flächenhafte Bildstrukturen. Mittels trigonometrischer Verfahren lassen sich absolute Tiefenmesungen für Punkte und linienhafte Objektstrukturen vornehmen, während die photometrischen Verfahren eher relative Tiefenhinweise geben und sich für flächenhafte Bildelemente eignen. Diese flächenhaften Bildelemente stellen Randbedingungen für die Deutung der sie begrenzenden linienhaften Strukturen dar und umgekehrt.

Aus dieser Sicht ist das Ziel der Szenenrekonstruktion nun, aus den vielfältigen Interpretationsmöglichkeiten für die einzelnen Bildelemente diejenigen auszuwählen, die im Kontext der Nachbarstrukturen einen möglichst großen Teil des Bildes widerspruchsfrei erklären. Das Rekonstruktionsproblem stellt sich somit als ein *Suchproblem in einem extrem großen Suchraum* dar, das dadurch erschwert wird, daß die optimale Lösung nur im Vergleich zu allen Alternativen definiert ist. Keine Teilinterpretation kann abgelehnt werden, solange nicht klar ist, ob es nicht eventuell eine weitere Lösung gibt, die einen noch größeren Teil des Bildes konsistent erklärt.[6]

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

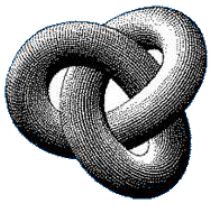
Stereo, Bewegung: Korrespondierende Merkmale

[6]Die Betrachtung des Rekonstruktionsproblems als Suchproblem bedeutet nicht unbedingt, daß dieses auch durch ein Suchverfahren gelöst werden muß. Gerade wegen des extrem großen Suchraums sind suboptimale oder heuristische Ansätze, wie Relaxationsververfahren, denkbar, um zu tragbaren Rechenzeiten zu kommen.

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

IVC Interactive Visual Computing: Computer Vision

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 8:

Mehrfache Objektansichten

Universität Hamburg
Fachbereich Informatik

WS 2014/2015

Teil XX

Mehrfache Objektansichten

50 Bewegungsstereo und optischer Fluß

50.1 Rekonstruktion starrer Körper

Bewegungsstereo und optischer Fluß

Ein Bewegungsfeld

51 Bewegungsstereo und optischer Fluß
- Rekonstruktion starrer Körper
- Bewegungsfeld einer Translation

52 Das Korrespondenzproblem bei Bildfolgen

53 Stereoskopie

Grundlegende Arbeiten zu diesem Themenbereich stammen von Nevatia, 1976, Ullman, 1979 und Clocksin, 1978. Eine gute Zusammenfassung findet sich im Lehrbuch von Horn, 1986.

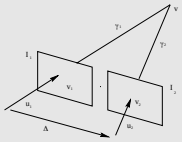
Literatur zum Motion Stereo

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.

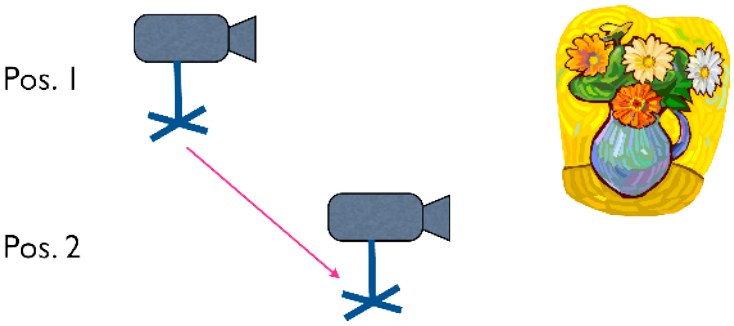
Stereoskopie



Triangulation

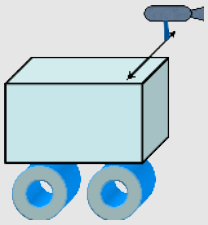
Wenn wir eine stereoskopische Aufnahme haben, können wir bei bekannter Basislinie mittels Triangulation die Tiefe messen.

Stereo mit nur einer Kamera



- Bei stationären Objekten kann die Kamera zur zweiten Position verschoben werden.

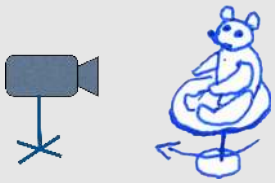
Erweiterung:



- 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

Mehr zum Stanford Cart siehe http://www.stanford.edu/~learnest/cart.htm

Motion Stereo (Nevatia 1976)

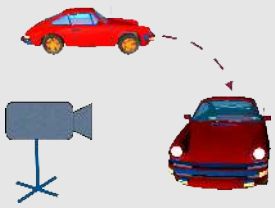


Objekt in kontrollierter Bewegung

Umkehrung der Relativbewegung:

- Das Objekt wird kontrolliert relativ zur Kamera bewegt.
- Mathematisch kein Unterschied zum vorherigen Verfahren.

Freie Objektbewegung



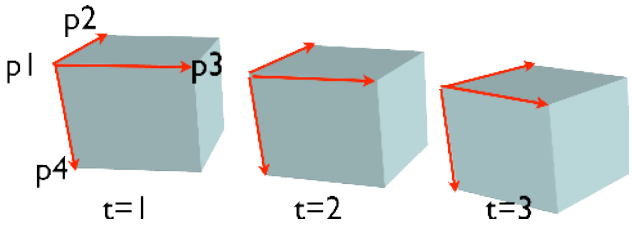
Objekt in freier Bewegung

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: 115

- *starres, unbekanntes Objekt mit den Punkten $p1 \cdots p4$,*
- *bewegungsbedingt unterschiedliche Ansichten,*
- *unbekannte Bewegungsparameter*

Homologe Punkte

Definition: 116 (*Homologe Punkte*)
Homologe Punkte sind Punkte in verschiedenen Bildern, die Abbildungen desselben Objektpunktes sind.

Bewegungsstereo liefert:

Lösung: 117
Structure from Motion

- *Räumliche Struktur (für homologe Bildpunkte)*
- *räumliche Trajektorie (Translation, Rotation)*
- *falls notwendig: Gruppierung in Objekte*

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.

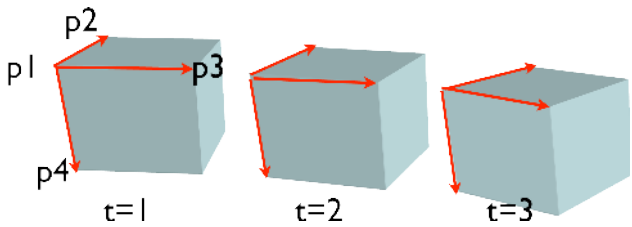
Anwendungen von Bewegungsstereo in der Robotik

Vermessen eines unbekannten Objektes:
- Aufbau eines Objektmodells durch Beobachtung
- Werkstücke, Hindernisse
- Ermitteln von Eigen- und Objektbewegung

Navigatation: Hindernisvermeidung

Ergänzung für andere Shape From- Verfahren

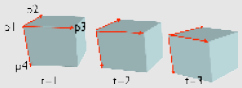
3-D-Analyse von Punktverschiebungen



Problem: 118 (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

Unbekannt:
- $6 \cdot (N-1)$ Bewegungsparameter
- $3 \cdot P$ Punktkoordinaten

Lösung

$$\Longrightarrow P \geq 3 + \frac{2}{2N-3}$$

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

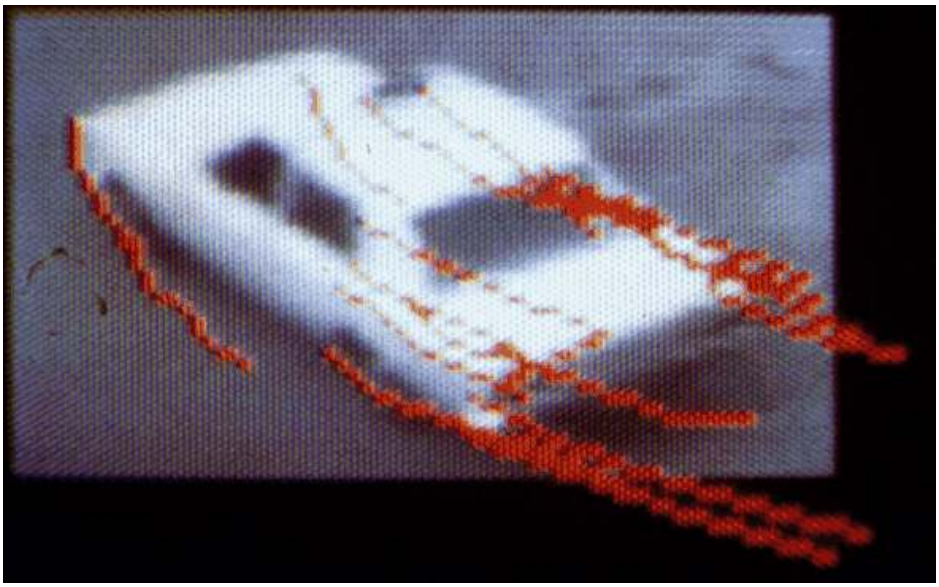
- Keine 3D-Information aus einer reinen Drehung um das Kamerazentrum.
- Keine 3D-Information bei einer reinen Translation entlang des Abbildungsstrahls eines Punktes.

Unser Auge kann anhand der verschiedenen Ansichten eines bewegten Objektes die Form ermitteln. Die folgende Folie zeigt die einen transparenten Zylinder, auf dessen Oberfläche sich zufällig verteilte Punkte befinden. Dieser Zylinder wurde in Paralleprojektion abgebildet. Im statischen Bild ist die Anordnung der Punkte im Raum nicht zu erkennen, aber wenn wir die Drehbewegung zeigen, wird die Zylinderform deutlich.

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/cylinder.html

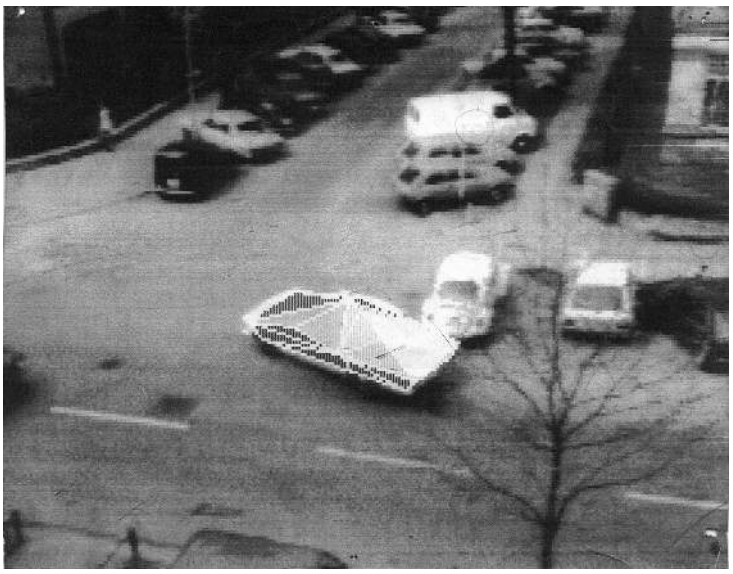
Beispiel: MORIO-System

Bewegungsspuren von markanten Punkten über 21 Bilder einer Folge

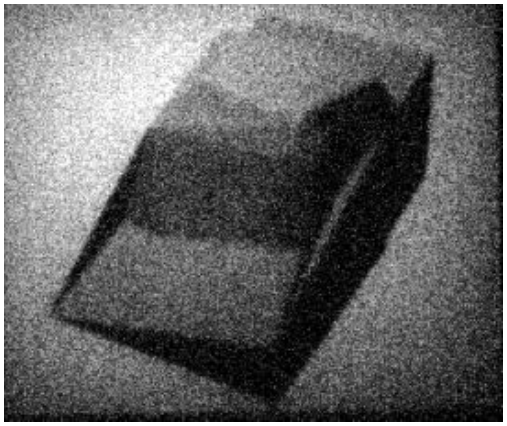


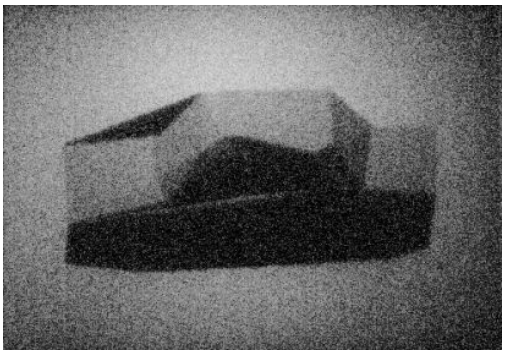
Beispiel: MORIO-System

3D-Rekonstruktion der markanten Punkten als konvexe Hülle, dem ersten Bild der Folge überlagert.

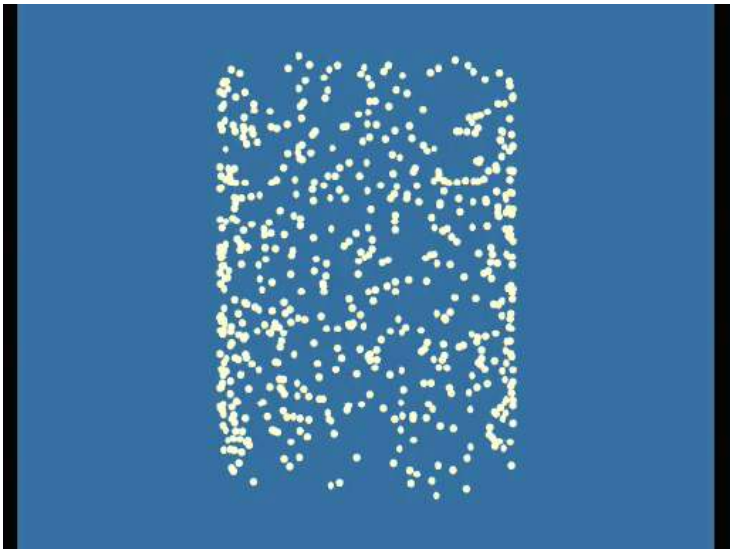


MUSCAT: Modellierung von Konkavitäten





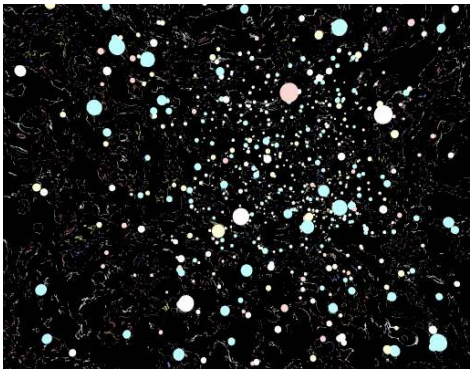
Ein transparenter Zylinder

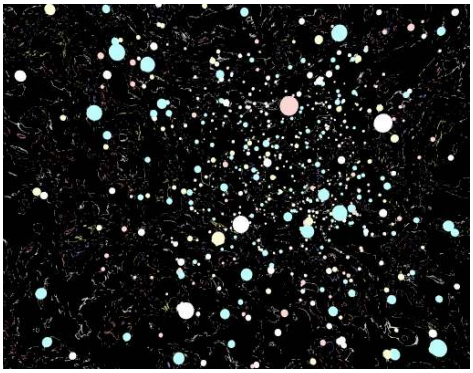


http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/cylinder.html

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.





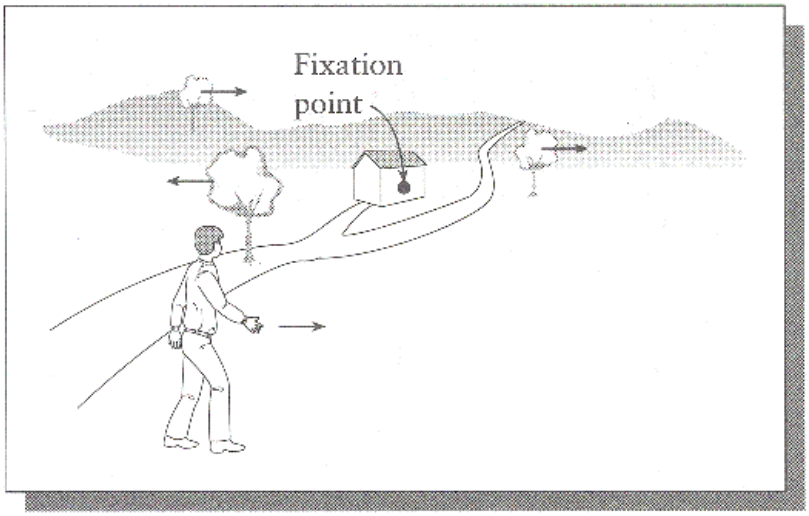
http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movMotion1Komparator.html

50.2 Bewegungsfeld einer Translation

✎ **Anmerkung:** Als nächstes werden wir eine reine Translationsbewegung betrachten und dabei die Bewegungsspuren untersuchen, die die Abbildungen von Objekten in der Bildebene hinterlassen.

Ein bewegter Beobachter

und scheinbare Bewegungen der Umgebung



aus (Ballard and Brown, 1982)

Das Bewegungsfeld

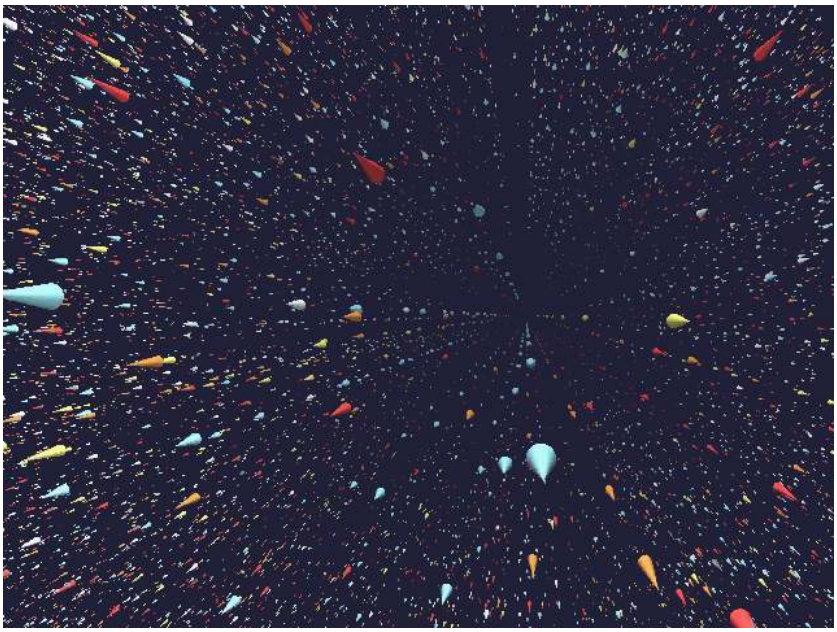
Definition: 119 (***Bewegungsfeld***)
Das Vektorfeld der Verschiebungen vom Zeitpunkt einer Bildaufnahme zum Zeitpunkt der nächsten Aufnahme (Verschiebungsvektoren) heißt „Bewegungsfeld“ (motion field).

☞ Aus dem Bewegungsfeld kann auf die Entfernung und Form eines Objektes geschlossen werden.

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.

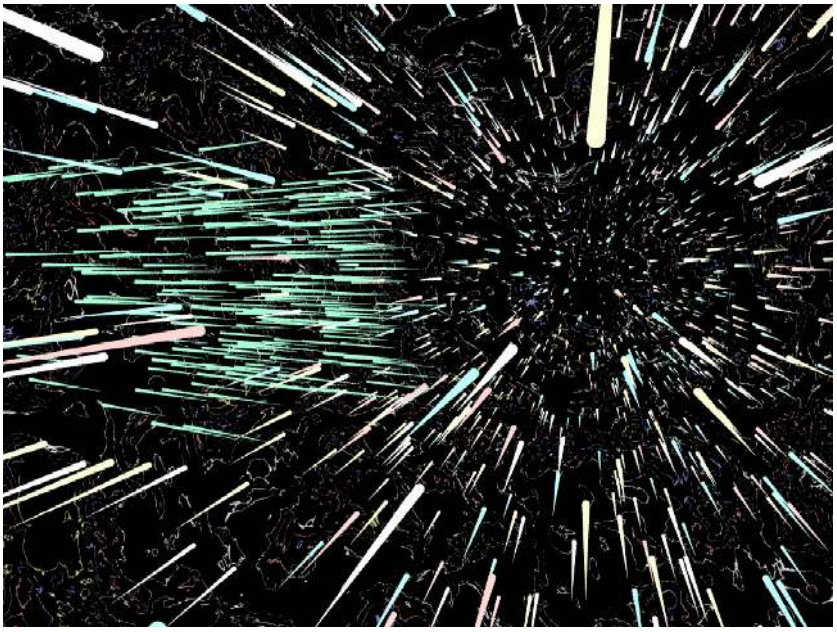
Das Bewegungsfeld einer Translation, Blick seitwärts



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/StarfieldMotionFlowComet.html

✎ **Anmerkung:** Die vorherige Folie zeigt das Vektorfeld des Bewegungsfeldes. Anstelle der bewegten Punkte sind kleine Kegel dargestellt, deren Richtung und Länge der Verschiebung von Bild zu Bild entspricht. Man erkennt, daß der Expansionspunkt (FOE) nichts anderes ist als der Fluchtpunkt der Geraden, auf denen die Translation erfolgt. Wenn die Szenenobjekte eine Eigenbewegung haben, dann kann es mehrere Fluchtpunkte geben.

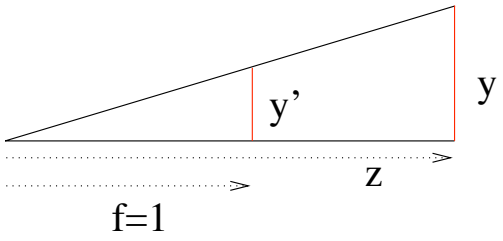
Expansionspunkte für zwei überlagerte Bewegungsfelder



✎ **Anmerkung:** Wo liegt der Expansionspunkt? Um die Lage der Expansionspunkte auszurechnen, werden wir der Einfachheit halber annehmen, daß das Weltkoordinatensystem das Kamerakoordinatensystem ist. Weiterhin nehmen wir an, daß die Brennweite der Kamera f gleich $f = 1$ sei. Die Resultate lassen sich leicht auf andere Brennweiten und ein Weltkoordinatensystem in allgemeiner Lage übertragen. Die Herleitung der folgenden Gleichungen finden Sie bei (Ballard and Brown, 1982).

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:

$$\vec{O} = (-\frac{dx}{dt}, -\frac{dy}{dt}, -\frac{dz}{dt}) = (-u, -v, -w)$$

Der Weltpunkt $P = (x_0, y_0, z_0)$ hat nach der Zeit t den Bildpunkt P'_t mit

$$P'_t = (x', y') = (\frac{x_0 + u \cdot t}{z_0 + w \cdot t}, \frac{y_0 + v \cdot t}{z_0 + w \cdot t})$$

Grenzwertbetrachtung:

Der Ausgangspunkt der Bewegung zum Zeitpunkt $t = -\infty$.

$$P'_{-\infty} = (\frac{u}{w}, \frac{v}{w})$$

Expansionspunkt

Definition: 120 (*Expansionspunkt*)
Der Expansionspunkt (focus of expansion) FOE ist der Bildpunkt des scheinbaren Ausgangspunktes der Bewegung zum Zeitpunkt $t = -\infty$.

$$FOE = (\frac{u}{w}, \frac{v}{w})$$

☞ 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:

$$\underbrace{\frac{D(t)}{|\vec{v}(t)|}}_{Bild,2D} = \underbrace{\frac{z(t)}{w}}_{Welt,3D} \qquad \frac{\text{Distanz}}{\text{Geschwindigkeit}}$$

Satz: 121
Der Abstand im Bild vom Expansionspunkt verhält sich zur Verschiebungsgeschwindigkeit im Bild wie die Tiefe z zur Geschwindigkeitskomponente w in z-Richtung.

Sei für einen Punkt P_1 die Tiefe $z_1(t)$ zum Zeitpunkt t bekannt, dann läßt sie sich für alle anderen Punkte P_i gleicher Geschwindigkeit berechnen:

$$\begin{aligned}
\frac{D_1(t)}{v_1(t)} &= \frac{z_1(t)}{w} \\
\frac{D_i(t)}{v_i(t)} &= \frac{z_i(t)}{w} \;\Big|\quad w \text{ eliminieren} \\
z_i(t) &= z_1(t) \cdot \frac{D_i(t)}{D_1(t)} \cdot \frac{v_1(t)}{v_i(t)}
\end{aligned}$$

Für die x- und y-Koordinaten folgt dann aus den Projektionsgleichungen und der bekannten Tiefe z:

$$\begin{aligned}
z(t) &= w \cdot \frac{D(t)}{v(t)} \\
x(t) &= x'(t) \cdot z(t) = x' \cdot w \cdot \frac{D(t)}{v(t)} \\
y(t) &= y'(t) \cdot z(t) = y' \cdot w \cdot \frac{D(t)}{v(t)}
\end{aligned}$$

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.

Beachte:

- Das Verfahren ist nur anwendbar, wenn eine reine Translation vorliegt.
- Bei sich drehenden Objekten kann das Verfahren nicht angewendet werden.

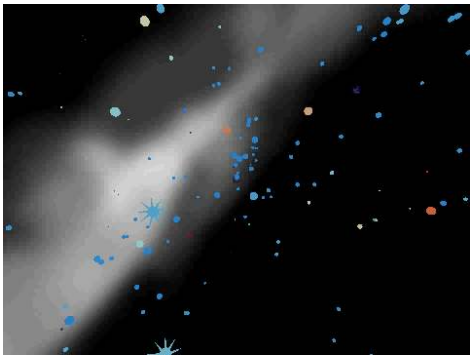
http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movStarfieldWithUFOPan.html

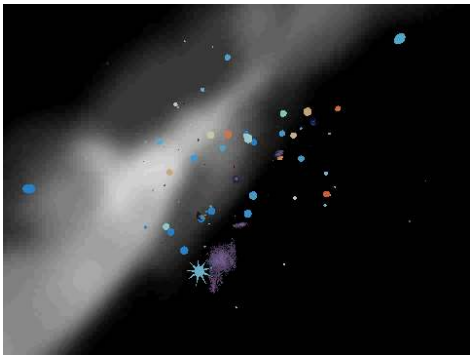
Relativistische Effekte

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





- 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.

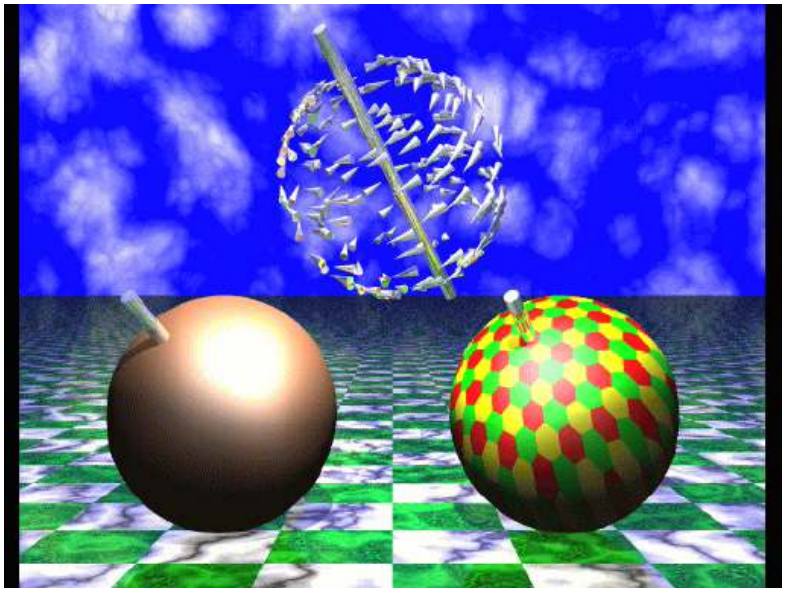
51 Das Korrespondenzproblem bei Bildfolgen

51.1 Bewegungsfeld vs. optischer Fluß

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 Beleuchtungveränderungen das Bild so verändern, daß es wie ein Bewegungsfeld erscheint.

Drei rotierende Kugeln



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movmotionflowbasicQT.html

Das Korrespondenzproblem bei Bildfolgen

Definition: 122 (*Korrospondenzproblem*)
Das Problem in einer Menge von Bildern homologe Punkte zu finden, heißt Korrospondenzproblem.

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

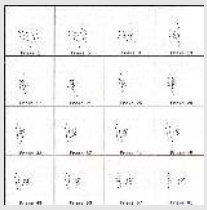
Definition: 123 (*Optischer Fluß*)
Das Vektorfeld der vom visuellen System wahrgenommenen Verschiebungen heißt optischer Fluß (optical flow).

Definition: 124 (*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.

Algorithmen zur Lösung des Korrespondenzproblems können Sie in der Vorlesung "'Bildverarbeitung 1"' kennenlernen.

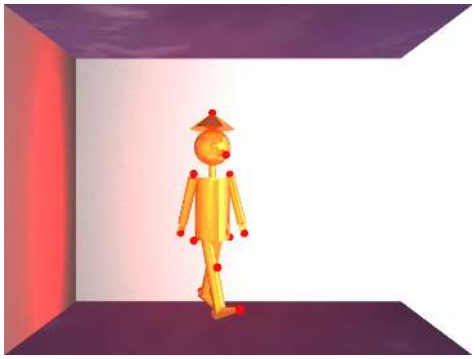
Moving Light Displays MLD

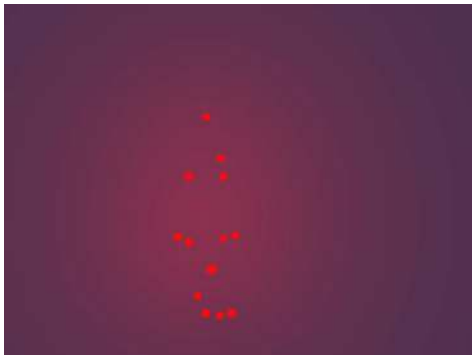


Das menschliche Sehsystem kann sehr gut die Bewegungen einzelner Punkte deuten und die Szene räumlich rekonstruieren.

(Ballard and Brown, 1982)

Ein simuliertes MLD





Am simulierten Roboter wurden an allen strategischen Stellen Lichter angebracht.

Ein simuliertes MLD, Licht aus: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/MLD1.html

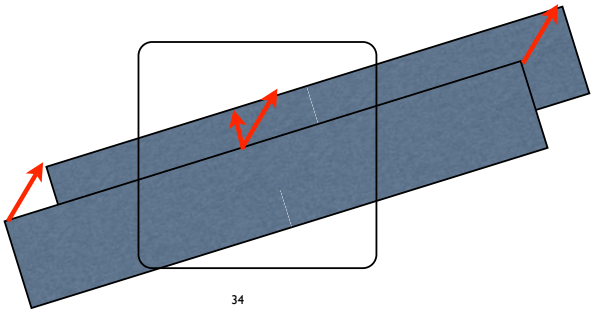
Ein simuliertes MLD, Licht an: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/MLD2.html

✎ **Anmerkung:** Blendeneffekt: Unser Sehsystem kann den optischen Fluß nur dort messen, wo eine Struktur in mindestens 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.

Der Blendeneffekt ist der Grund für die optische Täuschung, die als "barber pole illusion" bekannt ist. Hier sehen wir die Streifen eines rotierenden Zylinders entlang der Achse scheinbar aufsteigen oder absteigen, je nach der Drehgeschwindigkeit des Zylinders. Einähnlicher Effekt tritt auch bei rotierenden Scheiben mit einem Spiralmuster auf.

Solche Illusionen sind wichtige Hinweise darauf, wie wohl unser Sehsystem das Korrespondenzproblem löst.

Das Blendenproblem

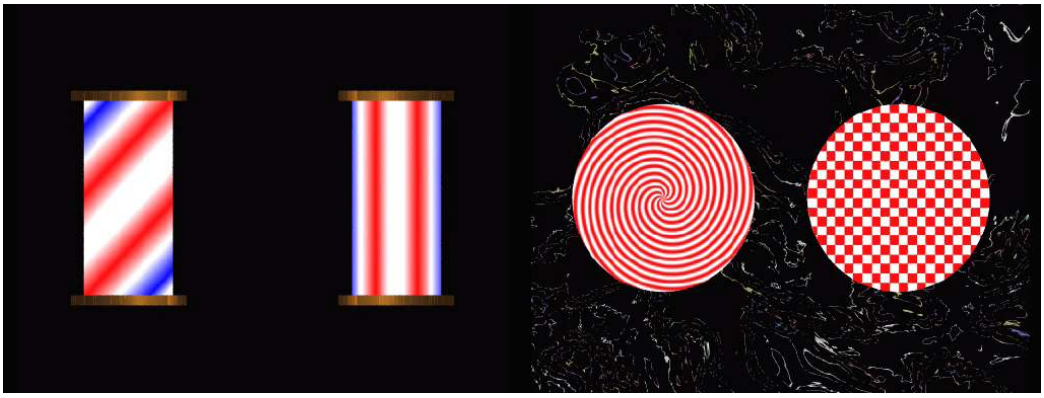


Die Grauwertverschiebungen können generell nur in Richtung des Gradienten bestimmt werden, wenn nur eine lineare, durchgehende Kante im Bild zu sehen ist.

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



http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/barberpole-basicQT.html

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/BV1_Movies/Pages/movRotorscheibeSpirale.html

52 Stereoskopie

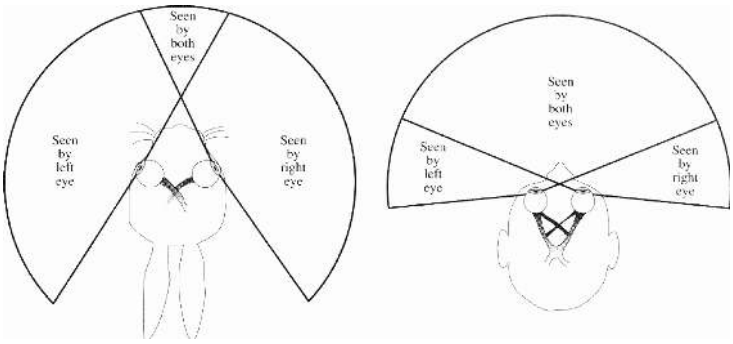
52.1 Binokulares Sehen

Die Bildbeispiele in diesem Kapitel stammen aus dem Bildband (Stereogramm-1994, 1994). Für die Algorithmen lesen Sie bitte in den Lehrbüchern von Sonka et al., 1999 oder Ballard and Brown, 1982 nach.

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.



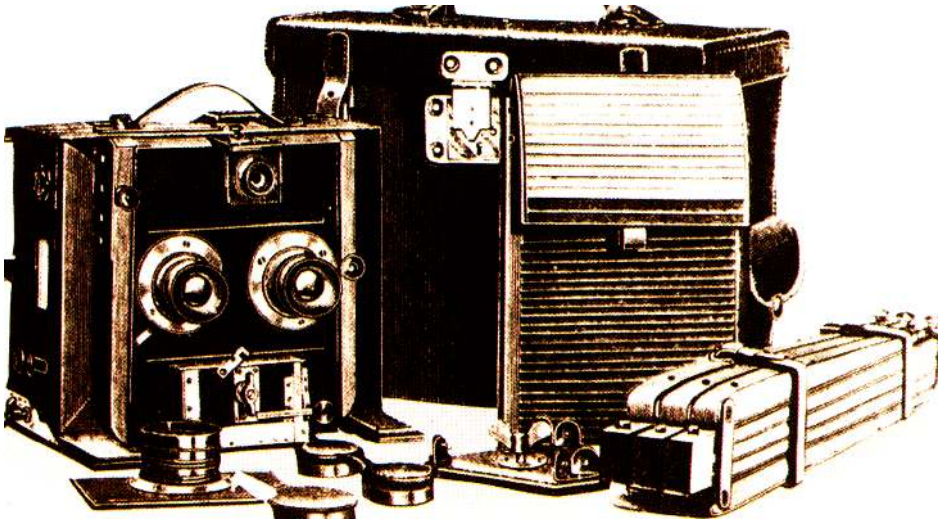
Anordnung der Augen: Kaninchen und Mensch.

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

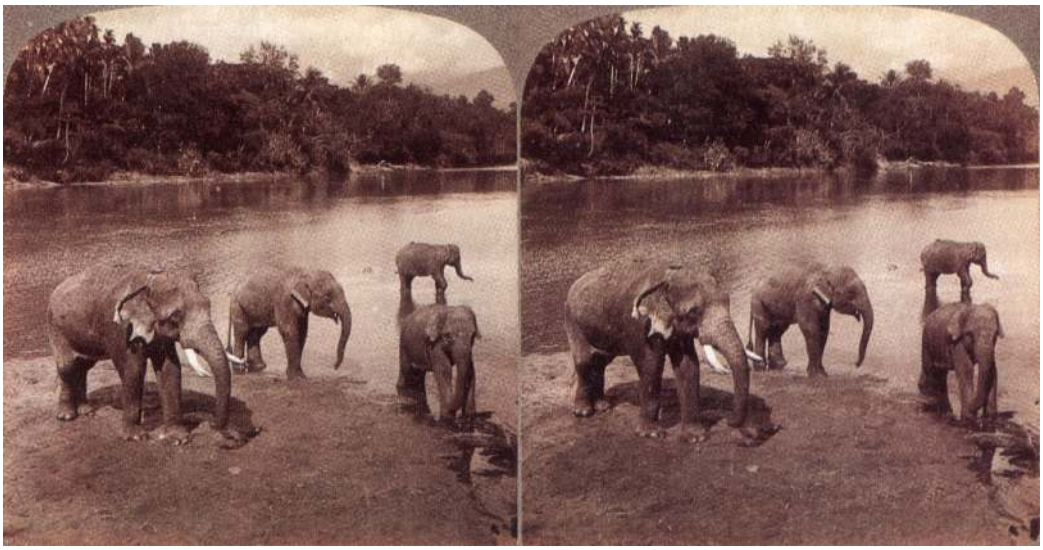
Behälter für Photoplatten (1895)



Historische Stereoaufnahmen

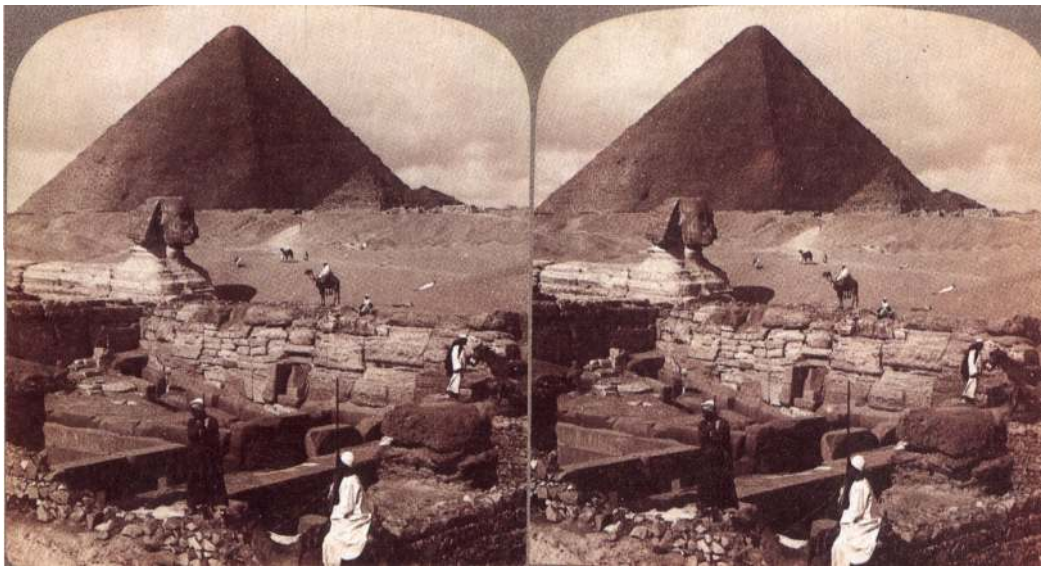
Ein Stereobildpaar aus dem Glaspalast in London (um 1851–1853)

Historische Stereoaufnahmen



Elefanten an einer Wasserstelle auf Ceylon, spätes 19. Jahrhundert

Historische Stereoaufnahmen



Cheops-Pyramide mit Sphinx, spätes 19. Jahrhundert

Alle Stereopaare aus (Stereogramm-1994, 1994)

Stereoskopie-Anwendungen

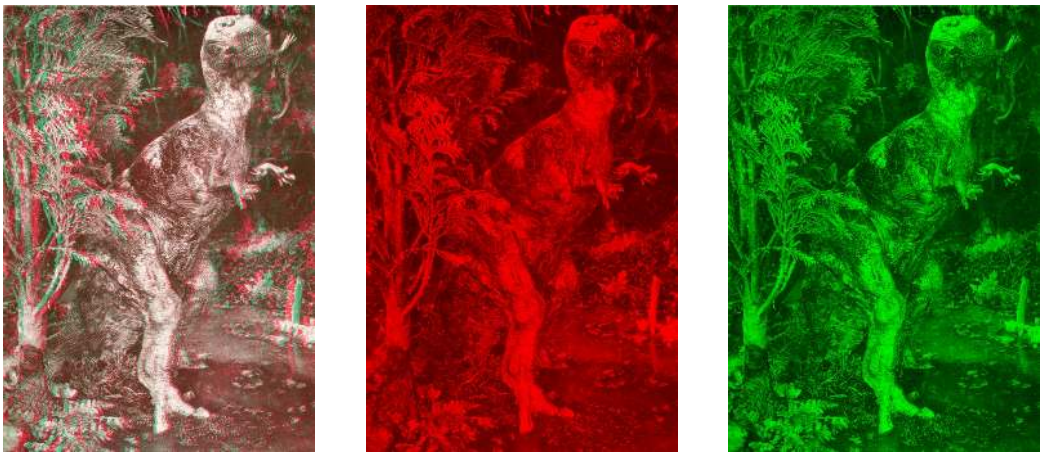
- Kartographie
- Robotik
- Medizin
- Raumfahrt
- Computergrafik
- Kino ...
- ...

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.

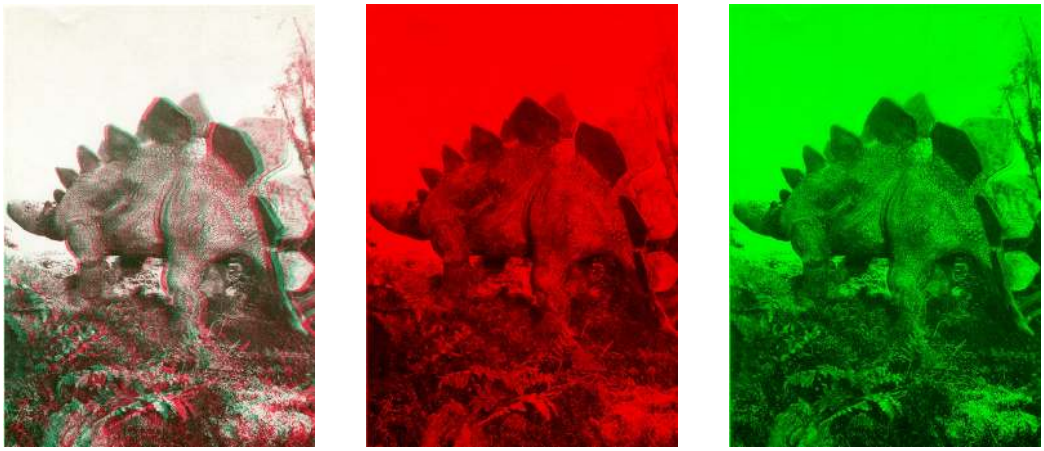
Anaglyphenbrille: Eine Brille mit passend gefärbten Gläsern selektiert die Bilder für das linke und rechte Auge.

Tyrannosaurus Rex in Anaglyphendarstellung



Rhamphorhynchus gemmingi in Anaglyphendarstellung

Stegosaurus stenops



Präsentation von Stereobildpaaren

Stereoskop : Jedes Auge sieht das passende Bild über ein seperates 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

Ohne Zusatzgeräte: **Mit viel Übung und Geduld**

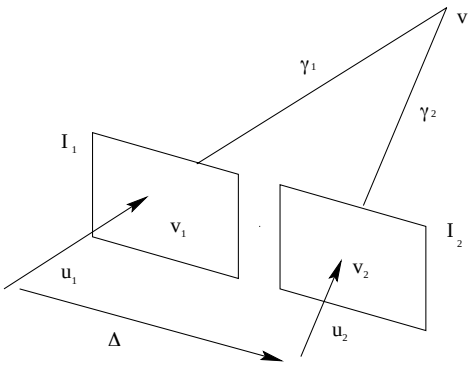
Stereogramme: Wenn die beiden Augen richtig ausgerichtet werden, entweder parallel oder über Kreuz (Schielen), sieht man zwischen den beiden Stereobildern ein räumliches Bild.

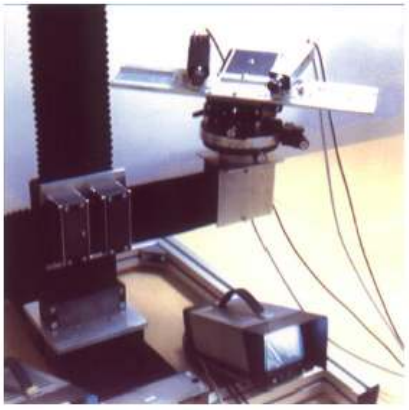
Zufallspunktstereogramme: Testbilder aus der kognitiven Psychologie, die zeigen, daß für das räumliche Sehen keine Bildstrukturen nötig sind.

$\Longrightarrow$ Vorläufer des „Magischen Bildes“.

52.2 Stereoskopische Tiefenrekonstruktion

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 $\gamma_2$sich schneiden, ist $\ell(a_0, b_0) = 0$ sonst:

$$\begin{aligned} a_0 &= \frac{\vec{u}_1 \cdot \Delta - (\vec{u}_1 \cdot \vec{u}_2)(\vec{u}_2 \cdot \Delta)}{1 - (\vec{u}_1 \cdot \vec{u}_2)^2} \\ b_0 &= \frac{(\vec{u}_1 \cdot \vec{u}_2)(\vec{u}_1 \cdot \Delta) - \vec{u}_2 \cdot \Delta}{1 - (\vec{u}_1 \cdot \vec{u}_2)^2} \end{aligned}$$

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

$$\vec{u_1} = \frac{\vec{v_{pi}} - \vec{L_1}}{||\vec{v_{pi}} - \vec{L_1}||}$$

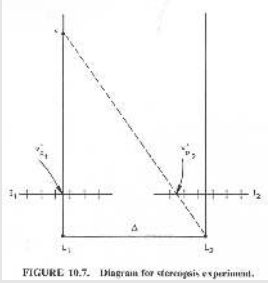
$\vec{L_2}$ und $\vec{v}$ folgen aus der inversen perspektivischen Transformation.
(nach (Duda and Hart, 1973))

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.

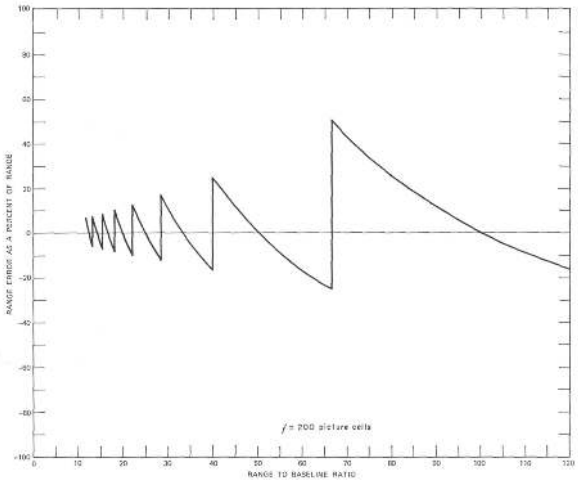
Rekonstruktionsfehler



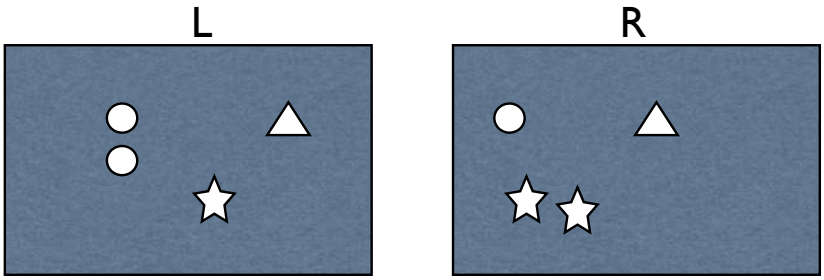
Einfluß der Auflösung

Die Bildunkte v'_{p1} und v'_{p2} werden nur begrenzter Genauigkeit bestimmt.

Tiefenfehler in Prozent der Entfernung

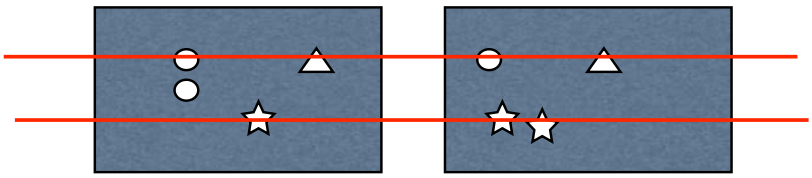


Das Korrespondenzproblem



- 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

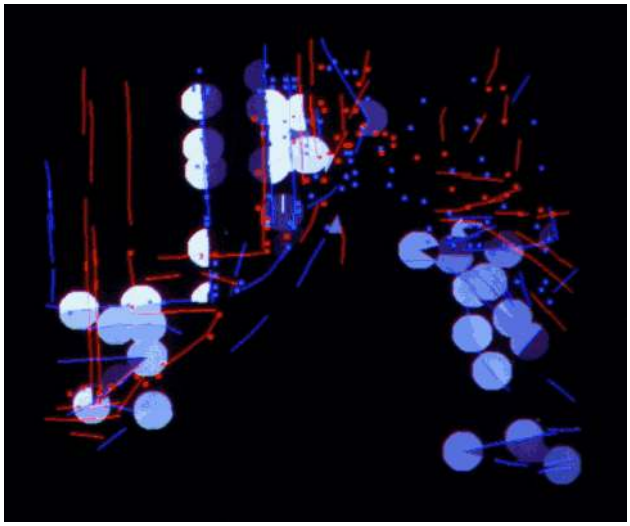
- Die Bildebenen der beiden Kameras sind koplanar und liegen in der xy-Ebene des Kamerakoordinatensystems.
- Beide optischen Zentren liegen auf der x-Achse.
- Dann sind epipolare Linien parallel zur x-Achse und homologe Punkte liegen in derselben Bildzeile.

Stanford Autonomous Land Vehicle

Stereo-Aufnahme einer Straße mit detektierten Kanten (Triendl-Operator)

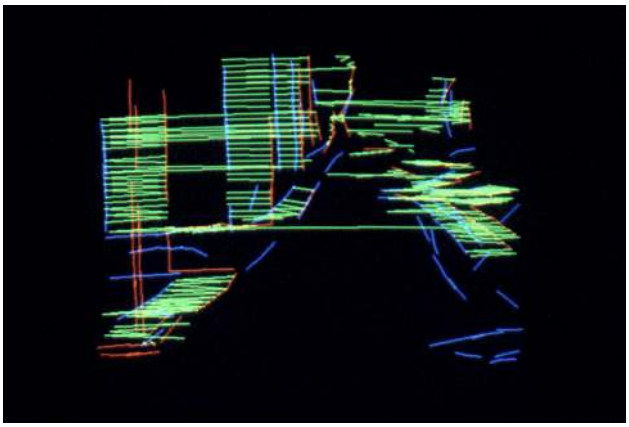
Stanford Autonomous Land Vehicle

Merkmale im linken und rechten Bild



Stanford Autonomous Land Vehicle

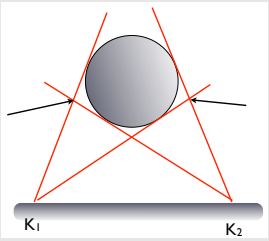
Zuordnung von homologen Punkten



Schwierigkeiten bei der Ähnlichkeitsmessung

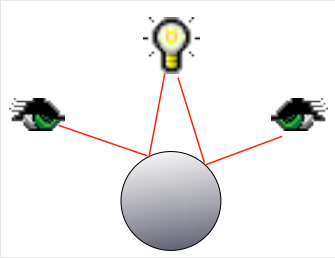
- Unterschiedlicher Blickwinkel: Perspektive, Licht, Winkel,
- Unterschiedlicher Abstand,
- Unterschiedliches Gesichtsfeld,
- Kontrastumkehrung.

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.

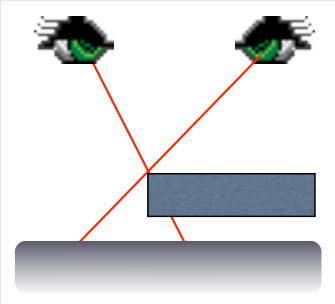
Spiegelungen

- 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

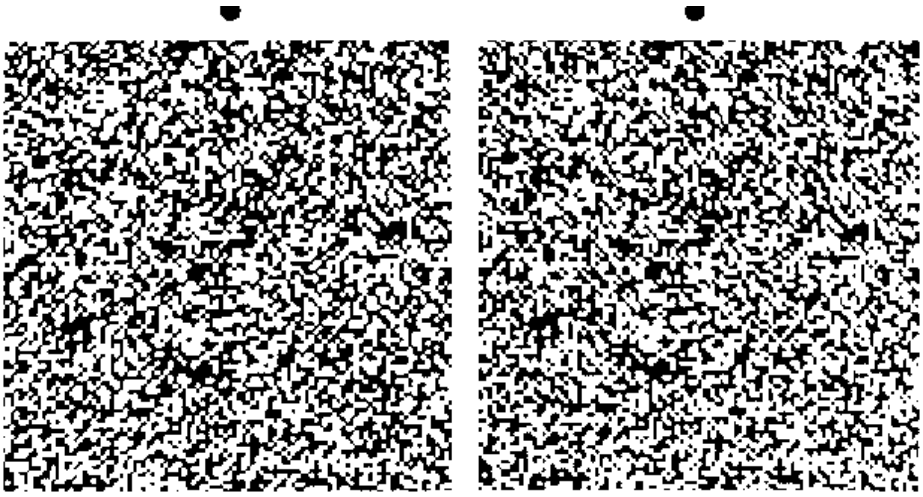
Wenn die Aufnahmebedingungen frei gestaltet werden können, kann das Korrespondenzproblem durch technische Hilfsmittel gelöst werden:

- Markierungspunkte auf der Objektoberfläche
- Markieren mit Lichtpunkten usw.

52.3 Zufallspunktstereogramme

Zufallspunktstereogramme von Bela Julesz (1959)

(Random-dot-stereogram)

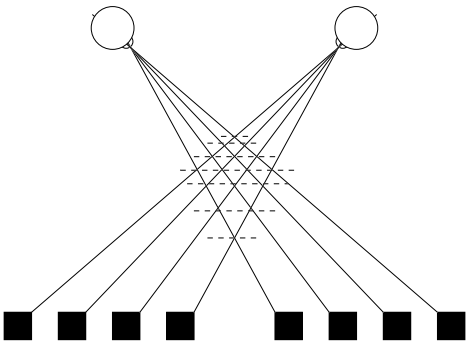


Zur Bedeutung der Zufallspunktstereogramme

- 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)



Mögliche Korrespondenzen:

Bildelemente links Bildelemente rechts

- Das visuelle System des Menschen wählt vermutlich die Lösung, bei der benachbarte Bildelemente eine ähnliche Tiefe haben.

Die obige Grafik zeigt Texel entlang einer epipolaren Linie im linken und rechten Bild. Da die Texel sich nicht unterscheiden, könnte jedes linke mit jedem rechten Texelkorrespondieren. Durch jede Korrespondenz wird die Tiefe festgelegt. Gesucht wird diejenige Teilmenge von korrespondierenden Texeln, bei der eine glatte Oberfläche entsteht und bei der entlang jedem Projektionsstrahl nur höchstens ein Paar korrespondierender Texel liegt.

Zuordnungen zwischen Texeln mit ähnlicher Entfernung unterstützen sich daher, Zuordnungen entlang desselben Projektionsstrahls hemmen sich.

Annahmen

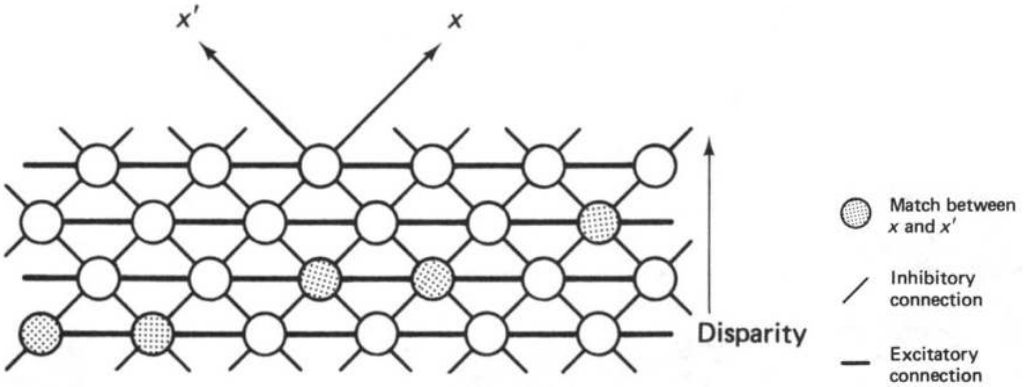
Annahmen:

- 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 Zuodnungen hemmen sich.

Implementation des Netzwerks

Implementation des Netzwerks als binäre Matrix $C(x, y, d)$,

- wobei C(x1,y1,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:

$$
\begin{aligned}
C_{n+1}(x, y, d) &= \Big\{ \sum_{(x',y',d')\in S} C_n(x', y', d') \\
&\quad - \sum_{(x',y',d')\in \Theta} C_n(x', y', d') \\
&\quad + C_0(x', y', d') \Big\} \\
S &= \text{support set: unterstützende Zuordnungen} \\
\Theta &= \text{inhibitory set: konkurrierende Zuordnungen}
\end{aligned}
$$

Der „support set“ enthält benachbarte Zuordnungen gleicher Tiefe d, der „inhibitory set“ benachbarte Zuordnungen unterschiedlicher Tiefe.

Normalisierung

Nach jeder Iteration normalisiere alle Elemente der Matrix $C(x, y, d)$ zu„1“ oder „0“ in Abhängigkeit von einer Schwelle T.

Resultate: Ein Bild mit zwei Tiefenebenen

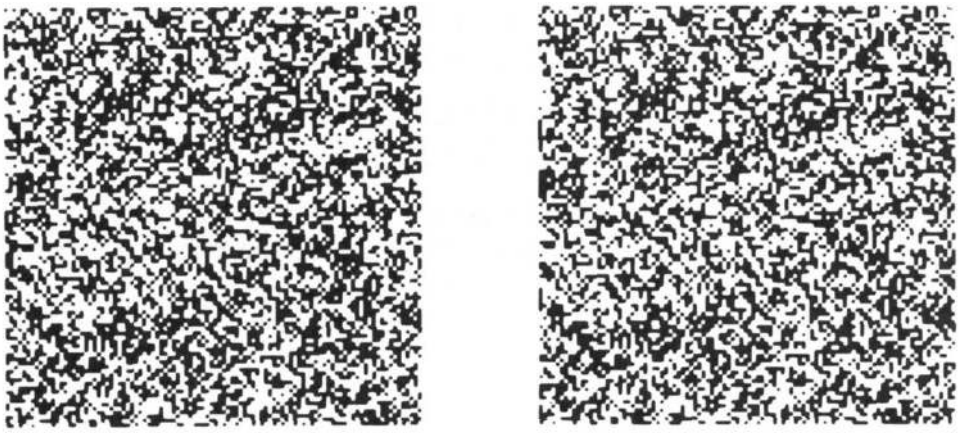
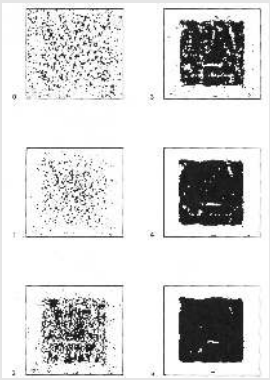


Fig. 3.23 A random-dot stereogram.

Die Testbilder entstanden, indem im linken Bild Pixel zufällig hell oder dunkel gefärbt wurden. für das rechte Bild wurde ein rechteckiger Ausschnitt aus der Bildmitte nach rechts verschoben und der freigelegte Hintergrund mit zufälligen schwarzen Punkten gefüllt.

Resultate



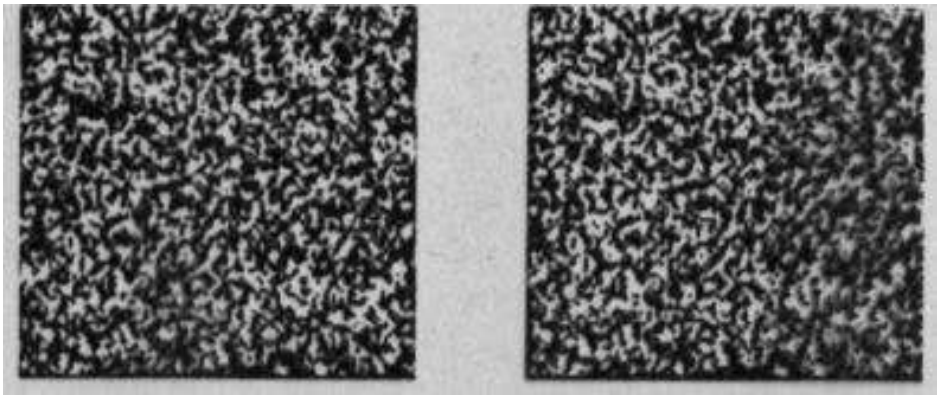
Disparität d

Sechs Iterationen des Stereo-Algorithmus

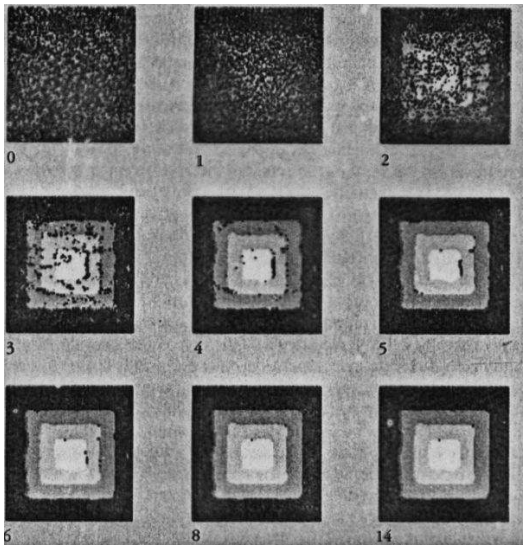
Für das Stereogramm von Folie 1359.

- 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

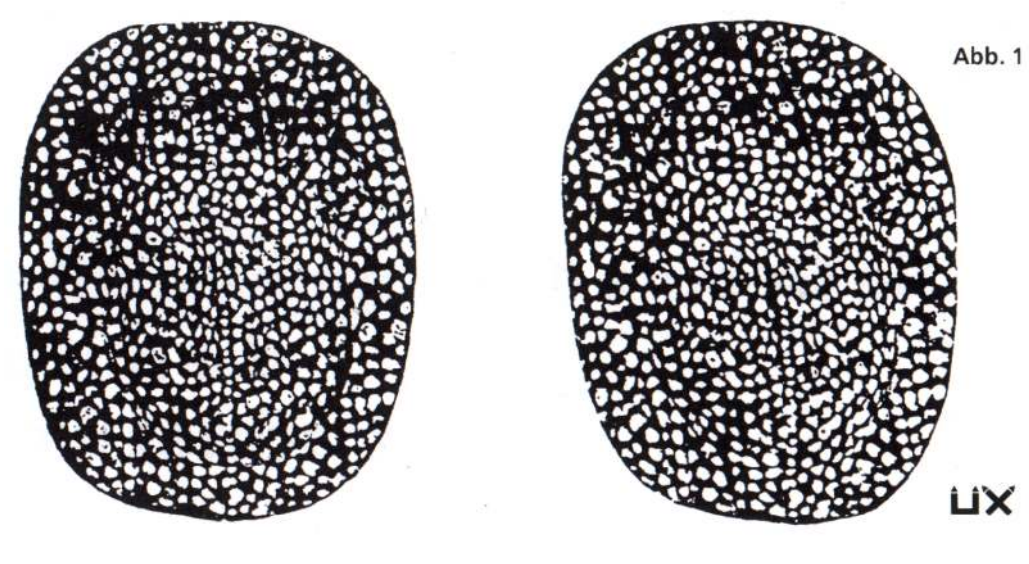


Beispiel 2: Bild mit vier Ebenen



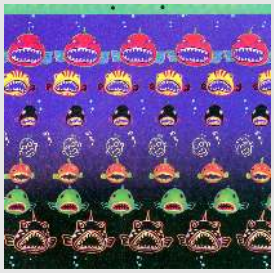
Das erste Stereogramm von Kompaneysky 1939:

Die Venus



52.4 Autostereogramme

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.

Konstruktion eines „Magischen Bildes“

- ☞ Stark vereinfachte Stereogeometrie:

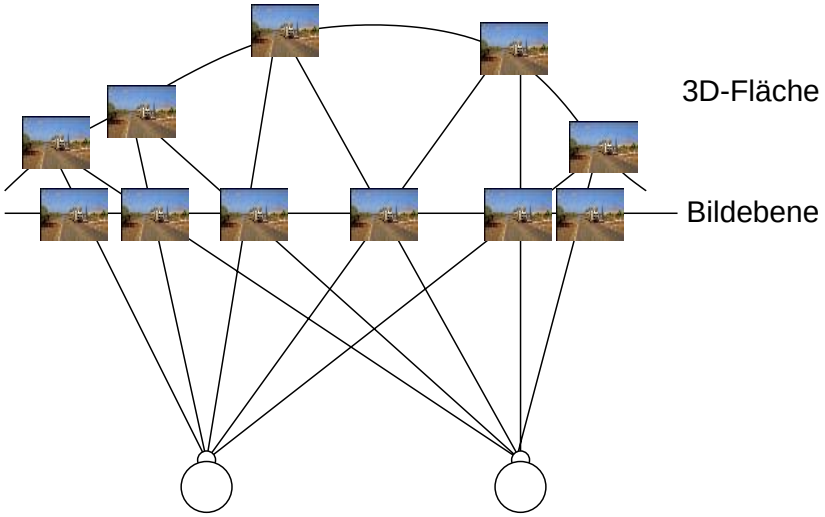
  Die Augen sind parallel gestellt, als solle ein unendlich entferntes Objekt betrachtet werden.

- ☞ 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.

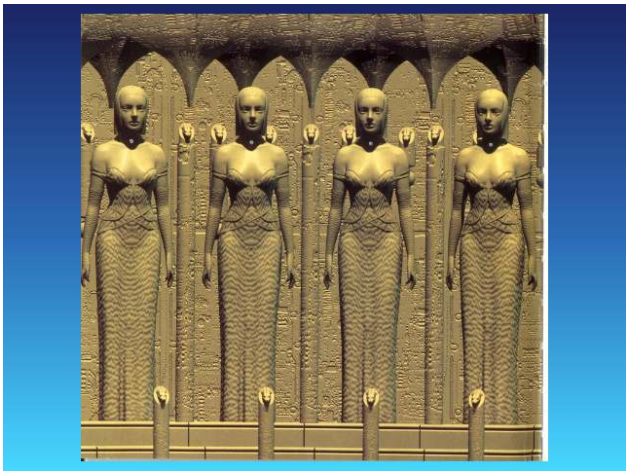
- ☞ Form und Farbe der Texel können für jede Zeile willkürlich gewählt werden.

Die Tapetentechnik



Konstruktion eines Autostereogramms mit der Tapetentechnik

Tapetentechnik: Die drei Grazien



Schweigende Schönheiten, Naoyuki Kato

Farbkorrespondenz

Diese Bild zeigt, daß widersprüchliche Farbe bei der Lösung des Korrespondenzproblems nicht stört. Unser Sehsystem ist in der Lage, die roten und blauen Flächen stereoskopisch zu fusionieren. das Bild hat mehrere Lösungen, je nachdem, ob blaue Streifen mit roten Streifen oder gleichfarbige Streifen korrespondieren.

Künstler-Stereogramme

Seisaku Kano

Künstler-Stereogramme: Salvador Dali

Der Stuhl 1976

Gala von hinten 1972-73, unvollendet

Lichtstiftbilder: McNelly

53 Das Korrespondenzproblem

Im folgenden sollen nun einige Konsequenzen des Generalitätsprinzips betrachtet werden, die hilfreiche Heuristiken zur Bestimmung von Korrespondenzen zwischen Merkmalen verschiedener Bilder darstellen.

Eindeutigkeit Eine der wichtigsten Folgerungen aus dem Prinzip des allgemeinen Standortes ist die Forderung nach der Eindeutigkeit von Zuordnungen der Bildmerkmale in der zeitlichen Folge oder in einem Stereobildpaar, eine Forderung, die bei vielen bekannten Korrespondenzverfahren, sowohl für Stereobilder als auch für Bildsequenzen, erhoben wird Grundlage hierfür ist die Annahme, daß ein Merkmal im Bild durch eine Struktur in der Szene hervorgerufen wird, die eine eindeutige Position in der dreidimensionalen Szene hat. Diese Annahme gilt jedoch nur dann, wenn das Prinzip des allgemeinen Standortes erfüllt ist. Liegt dagegen eine spezielle Sicht des Beobachters vor, bei der die Blickrichtung (ein Projektionsstrahl) z.B. in einer Ebene mit einer Objektoberfläche der Szene liegt, dann liegen auf diesem Projektionsstrahl mehrere Szenenpunkte, die in einem Bild alle auf denselben Bildpunkt abgebildet werden, im anderen Bild jedoch auf mehrere Punkte. Damit ist aber auch die Forderung nach Eindeutigkeit der Zuordnungen von Bildmerkmalen eine Heuristik, bei deren Anwendung vorausgesetzt wird, daß in den Bildern keine spezielle Sicht vorliegt.

Gerade, gekrümmte und kollineare Linien im Bild Eine der interessanten Eigenschaften der perspektivischen Projektion ist es, daß sie Geraden in der dreidimensionalen Szene auf Geraden im Bild abbildet. Der Schluß in die andere Richtung, daß eine Gerade im Bild durch eine Gerade in der Szene zustande kommt, gilt jedoch nicht. Hierbei handelt es sich lediglich um eine Heuristik, die nur dann gültig ist, wenn wir in einem Bild keine spezielle Sicht vorliegen haben. Wir können dies durch folgende Regel formulieren:

$$\text{2D-Kante ist gerade} \wedge \neg(\text{spezielle Sicht auf planare 3D-Kante})$$
$$\Leftrightarrow$$
$$\text{3D-Kante ist gerade}$$

Dies gilt nicht nur für gerade Kanten, sondern auch für Gruppen einzelner Punkte oder Endpunkte von Kanten, die auf einer Geraden liegen, und außerdem für zwei Geraden im Bild, die kollinear sind, also auf einer gemeinsamen Geraden liegen.

Machen wir die Annahme, daß bei einem Bild keine spezielle Ansicht vorliegt, vertrauen wir also auf das Prinzip des allgemeinen Standortes, dann können wir diese Regel als Heuristik verwenden und aus einer detektieren geraden Kante im Bild schließen, daß sie auch im Raume gerade ist. Durch den immer gültigen Schluß in die andere Richtung erhalten wir, daß die zu dieser Kante korrespondierende Kante im anderen Bild ebenfalls eine Gerade sein muß. Hieraus ergibt sich dann die Einschränkung für mögliche Stereozuordnungen:

> Eine Gerade in einem Bild kann in einem anderen Bild nur mit einer Geraden korrespondieren.

Bei dieser Einschränkung handelt es sich um eine Heuristik, die auf das Prinzip des allgemeinen Standortes vertraut. Sie ist für Stereozuordnungen anwendbar, für zeitliche Zuordnungen jedoch nur dann, wenn zusätzlich die Annahme gemacht wird, daß sich eine Gerade in der Szene von einem Zeitpunkt zum nächsten nicht verformt.

Für gekrümmte Linien existiert eine ähnliche Heuristik, die ebenfalls aus den Eigenschaften der perspektivischen Abbildung folgt:

2D-Kante ist gekrümmt
$\Leftrightarrow$
$\neg$(spezielle Sicht auf planare 3D-Kante)
3D-Kante ist gekrümmt

Aus dieser Regel folgt sofort die Einschränkung für mögliche Stereozuordnungen:

> Eine gekrümmte Kante in einem Bild kann in einem anderen Bild nur mit ebenfalls gekrümmten Kante korrespondieren.

Hierbei handelt es sich ebenfalls um eine Heuristik; zwar folgt aus der Gekrümmtheit einer Kante im Bild, daß sie im Dreidimensionalen ebenfalls gekrümmt sein muß, sie kann jedoch planar sein und bei einer speziellen Sicht daher als Gerade auf eine andere Bildebene abgebildet werden.

Parallele Linien im Bild: Häufig wird auch angeführt, daß parallele Kanten im Bild zu parallelen Kanten im Raume gehören,[7] und es wird die Einschränkung formuliert, daß parallele Kanten nur mit parallelen Kanten korrespondieren können. Hierbei handelt es sich jedoch um eine äußerst schwache Heuristik, denn es gilt: Parallele Linien im Raume werden durch die perspektivische Projektion nur dann auf parallele Linien im Bild abgebildet, wenn sie parallel zur Bildebene sind. Dies erfordert also eine spezielle Sicht, die für Bildebenen eines Stereokamerasystemes nur dann erfüllt sein kann, wenn zusätzlich die beiden Bildebenen parallel zueinander sind. Sind die Linien im Raume jedoch nicht parallel, dann können diese nur dann auf parallele Linien im Bild abgebildet werden, wenn eine spezielle Sicht vorliegt. Es ist jedoch äußerst unwahrscheinlich, daß dies für beide Bildebenen eines Stereokamerasystemes der Fall ist. Diese Einschränkung kann daher sinnvoll nicht verwendet werden.

Parallele Linien im Raume: Haben wir aus der 3D-Rekonstruktion von stereoskopischen Kantenzuordnungen Geraden im Raume erhalten, die parallel sind, dann können wir eine weitere Eigenschaft der perspektivischen Projektion ausnutzen: Die Abbilder von parallelen Geraden im Raume konvergieren im Bild gegen einen gemeinsamen *Fluchtpunkt*. Dieser Fluchtpunkt ist dabei nicht abhängig von der Lage einer Geraden im Raume, sondern lediglich von deren Orientierung. Verwenden wir die zusätzliche Annahme, daß Parallelen im Raume über längere Zeit hinweg parallel bleiben und ist bekannt, daß sich ihre Orientierung im Raume mit der Zeit nicht ändert, dann können wir dieses Gesetz der projektiven Geometrie dazu verwenden, für die Bilder darauffolgender Zeitpunkte den Fluchtpunkt zu berechnen und in diesen Bildern dann nach Geraden zu suchen, deren Verlängerungen durch diesen Fluchtpunkt laufen. Für diese Geraden besteht dann die Hypothese, daß sie Bilder derselben Parallelen sind und daher mit diesen korrespondieren.

[7] Mit Parallelen sind hier nicht nur Geraden gemeint, sondern auch gekrümmte Linien, deren Abstand aber konstant ist.

Fluchtpunkte Die soeben beschriebene Eigenschaft der perspektivischen Abbildung, daß in der Szene parallele Geraden in Bildern gegen einen gemeinsamen Fluchtpunkt konvergieren, läßt sich auch noch auf andere Weise ausnutzen: Können in einem Bild Geraden detektiert werden, deren Verlängerungen sich in einem gemeinsamen Punkte schneiden, und nimmt man weiterhin an, daß keine spezielle Sicht auf nicht parallele Geraden im Raume vorliegt, dann kann hieraus die Hypothese gebildet werden, daß es sich um einen Fluchtpunkt einer Schar von im Raume parallelen Geraden handelt.

Aus dem bestimmten Fluchtpunkt kann dann die Orientierung dieser Geradenschar im Raume bestimmt werden und daraus die Position des Fluchtpunktes in einem anderen Bild. Handelt es sich tatsächlich um Parallele, dann müssen in diesem anderen Bild ebenfalls Geraden existieren, deren Verlängerungen sich in dem für dieses Bild berechneten Fluchtpunkt schneiden. Kann man solche Geraden im anderen Bild detektieren, dann ergibt sich hieraus die Hypothese, daß diese Menge von Geraden der beiden Bilder korrespondiert.

Koinzidenz In den beiden vorangegangenen Abschnitten über parallele Geraden im Raum und über Fluchtpunkte haben wir davon gesprochen, daß sich die Verlängerungen von Geraden im Bild in einem gemeinsamen Fluchtpunkt schneiden, denn die Geraden in der Szene werden in der Regel nicht unendlich lang sein und im Bild daher vor Erreichen des Fluchtpunktes enden.

Schneiden sich Geraden im Bild tatsächlich, dann erhalten wir unter Annahme des Prinzips des allgemeinen Standortes die folgende Heuristik: Schneiden sich Geraden im Bild in einem gemeinsamen Punkte, dann schneiden sich die sie erzeugenden Linien in der Szene ebenfalls in einem gemeinsamen Punkt. Eine Ausnahme hiervon existiert lediglich bei Bestehen einer speziellen Sicht, bei der die Endpunkte der Linien in der Szene auf einem gemeinsamen Projektionsstrahl eines Kamerasystems liegen müßten.

Zwei weitere hieraus folgende Heuristiken bilden die Möglichkeit, schon ermittelte Korrespondenzen über Schnittpunkte von Kanten hinweg zu propagieren:

- Hat man zwei Punkte bereits einander zugeordnet, dann korrespondieren auch die von diesen Punkten ausgehenden Kanten.

- Hat man zwei Kanten bereits einander zugeordnet, dann korrespondieren auch die Kanten, mit denen ein gemeinsamer Schnittpunkt besteht.

Vorsicht geboten ist beim Propagieren jedoch an Punkten, an denen zwei Kanten „T-förmig“ aufeinander treffen, da das Auftreten eins solchen Punktes einen starken Verdacht auf eine Verdeckung und damit auf einen Tiefensprung bildet (siehe (Binford, 1981) für diese sogenannte *T-junction rule*):

> Der ‘Stamm’ eines „T“ liegt nicht dichter am Beobachter als dessen ‘Dach’.

Diese Heuristik ist ebenfalls auf das Prinzip des allgemeinen Standorts begründet.

Den hier beschriebenen Heuristiken liegt jeweils das Prinzips des allgemeinen Standortes zugrunde. Es ist offensichtlich, daß aus einem Bild allein nicht ermittelt werden kann, ob in diesem Bild eine spezielle Sicht vorliegt oder nicht. Das Prinzip der allgemeinen Bewegung ergibt jedoch, daß eine spezielle Sicht nicht in mehreren Bildern einer Sequenz existieren wird. Die Auswertung einer Sequenz von Stereobildern ermöglicht es also festzustellen, ob in einem Bild eine spezielle Sicht vorgelegen hat, und falls eine solche vorgelegen haben sollte, die Folgerungen aus dann nicht gültigen Heuristiken eventuell zurückzunehmen.

53.1 Das Korrespondenzproblem

Wie oben bereits angedeutet, ist es Ziel der trigonometrischen Rekonstruktionsverfahren, Tiefeninformation für exakt lokalisierbare Strukturen aus der Kombination mehrfacher Objektansichten zu gewinnen: Aus einem Stereobildpaar kann beispielsweise durch die trigonometrische Parallaxe die Position einzelner Punkte im dreidimensionalen Raume berechnet werden; die Verwendung einer zeitlichen Folge von Bildern ermöglicht dann die Bestimmung der Bewegung von einzelnen Punkten im Raum.

Um eine Rekonstruktion der Tiefe einzelner Szenenpunkte mittels trigonometrischer Parallaxe sowie die Verfolgung von Szenenpunkten in der zeitlichen Folge durchführen zu können, ist es allerdings erforderlich zu ermitteln, welche Bildpunkte sowohl eines Stereobildpaares als auch eines zeitlichen Bildpaares *korrespondieren*, also Abbildungen ein und desselben Szenenpunktes sind. Dieses Problem wurde von Ullman, 1979 unter dem Namen *Korrespondenzproblem* eingeführt.

Das Korrespondenzproblem ist seit seiner Definition durch Ullman in unterschiedlicher Weise angegangen worden: Visuelle Korrespondenz, Grauwertverschiebungen und physikalische Korrespondenz (siehe (Dreschler-Fischer, 1986) für eine detaillierte Diskussion dieser drei Begriffe). *Visuelle Korrespondenz* bezeichnet die Korrespondenz zwischen Merkmalen, wie sie bei der visuellen Wahrnehmung des Menschen beobachtet wird. *Grauwertverschiebungen* geben in Form von Vektorfeldern Auskunft über die zwischen zwei Bildern auftretenden räumlichen Verschiebungen der sichtbaren Grauwertmuster, den sogenannten *optischen Fluß* (gute Übersichten hierzu geben Nagel, 1986 und Sprengel, 1988). Als *physikalische Korrespondenz* werden Korrespondenzen zwischen solchen Bildmerkmalen bezeichnet, die Abbilder desselben physikalischen Objektmerkmals sind.

Merkmalsbasierte Korrespondenzanalyse

Im folgenden soll der Ansatz der sogenannten *merkmalsbasierten Korrespondenzanalyse* als Versuch zur Ermittlung physikalischer Korrespondenzen genauer betrachtet werden. Für eine geometrische Rekonstruktion liegt es nahe, vor allem diejenigen Oberflächenstrukturen zu rekonstruieren, die visuell markant sind. Da der einzige Hinweis auf Strukturen auf den Oberflächen der zu rekonstruierenden Objekte Strukturen in ihren Abbildungen sind, bietet es sich an, das physikalische Korrespondenzproblem merkmalsbasiert zu lösen und nach korrespondierenden Abbildungen von körperfesten Oberflächenstrukturen zu suchen. Das Problem festzustellen, welche Bildpunkte zweier Bilder Abbildungen ein und desselben Szenenpunktes sind, gliedert sich hierbei in zwei Schritte:

Die Merkmalsdetektion: In diesem ersten Schritt wird versucht, in den Bildern Merkmale zu detektieren, die eine eindeutige physikalische Ursache in der beobachteten Szene haben. Bei diesen Merkmalen muß es sich um die Abbilder körperfester Punkte auf den Objekten handeln und sie sollten in allen Bildern, auf die sie abbgebildet werden, auch detektiert werden.

Die Korrespondenzanalyse: Im zweiten Schritt wird dann versucht, diejenigen Merkmale jeweils zweier Bilder einander zuzuordnen, die Abbilder ein und derselben Punkte der Szene sind, welche also korrespondieren.

Die Korrespondenzanalyse stellt nun für sich selbst ein schwieriges Problem dar, da die Zuordnung von Bildmerkmalen – wie das Problem der Szenenrekonstruktion insgesamt – grundsätzlich mehrdeutig ist. Für die Korrespondenzanalyse müssen daher einschränkende Randbedingungen herangezogen werden, die die bestehenden Mehrdeutigkeiten auflösen oder zumindest reduzieren. Hierbei schlägt das Bootstrap-Problem in dreifacher Weise zu:

Merkmalsdetektion: Die Segmentatierung eines Bildes impliziert Merkmalsmodelle, die einer Annahme über das Vorhandensein von Kanten, Regionen oder markanten Punkten zugrunde liegen.

Merkmalsinterpretation: Physikalische Korrespondenz ist nur für körperfeste Oberflächenstrukturen definiert. Nicht alle Bildstrukturen entsprechen solchen körperfesten Objektstrukturen, sondern entstehen beispielsweise durch Beleuchtungseffekte oder Verdeckung. Eine Aussage über die Art der Bildstrukturen impliziert eine zumindest qualitative Vorstellung von den erzeugenden Parametern der Bildentstehung.

Mehrdeutigkeit der Korrespondenz: Wegen der fehlenden Tiefeninformation ist die Korrespondenz für einzelne Bildstrukturen in der Regel mehrdeutig. Hypothesen über korrespondierende Bildelemente implizieren räumliche Interpretationen, während umgekehrt Hypothesen über den geometrischen Aufbau von Objekten die möglichen Korrespondenzen einschränken.

Angesichts dieser Komplexität des physikalischen Korrespondenzproblems liegt es nahe, die Aufgabe der effektiven Reduzierung der Mehrdeutigkeiten durch ein Zusammenspiel aller verfügbaren Rekonstruktionstechniken anzugehen. Hierbei sind die einzelnen Verfahren nicht als voneinander unabhängige Methoden zur Ermittlung von 3-D Information zu betrachten und einzusetzen (wie es beispielsweise Crowley, 1984 vorschlägt). Vielmehr lassen sich die gegenseitigen Wechselwirkungen zwischen den konkurrierenden Methoden zur Überprüfung ihrer Anwendbarkeit nutzen.

Kategorien von Hinweisen zur Reduzierung der Mehrdeutigkeiten

Bei dem Wissen, das zur Reduzierung der bei der Korrespondenzanalyse auftretenden Mehrdeutigkeiten genutzt werden kann, muß zunächst grundsätzlich zwischen physikalischen Gesetzen einerseits und Heuristiken andererseits unterschieden werden. *Heuristiken* dienen zur zielgerichteten Steuerung der Suche, um schnell tragfähige Hypothesen zu finden, während *physikalische Gesetze* zur Verifikation dieser Hypothesen eingesetzt werden. Gemäß dem „Prinzip der kleinsten Verpflichtung"(*principle of least commitment* sollte ein praktisches Verfahren defensiv angelegt sein, d.h. es wird keine Hypothese allein aufgrund von Heuristiken zurückgewiesen, solange keine physikalische Evidenz dagegen vorliegt, sondern allenfalls zugunsten anderer Hypothesen zurückgestellt.

Folgende Kategorien von Hinweisen können zur Reduzierung der auftretenden Mehrdeutigkeiten bei der Korrespondenzanalyse und somit zur Lösung des Korrespondenzproblems beitragen:

Geometrische Eigenschaften: Eine Reihe von geometrischen Beschränkungen für mögliche Zuordnungen von Bildmerkmalen lassen sich aus einer Modellierung des Prozesses der Bildentstehung herleiten: Als erstes ergeben sich aus dem *Generalitätsprinzip* unter der Annahme, daß Objekte nicht aus einer ganz speziellen Blickrichtung betrachtet werden, Einschränkungen der möglichen Form korrespondierender Kanten. Die *epipolare Geometrie* eines Stereokamerasystems ermöglicht weiterhin die Einschränkung des ursprünglich zweidimensionalen Suchbereiches für korrespondierende Merkmale auf ein eindimensionales Intervall.

Objekteigenschaften: Eine Betrachtung der Eigenschaften der Objekte unserer natürlichen Umwelt erlaubt die Formulierung weiterer Einschränkungen für mögliche Merkmalszuordnungen. Aus der Tatsache, daß Objekte massebehaftet sind, folgt beispielsweise, daß sie sich im Raum auf *glatten* Bahnen bewegen. Dies läßt sich zur Ermittlung von zeitlichen Korrespondenzen ausnutzen. Aus der Annahme, daß die Oberflächen der beobachteten Objekte einer Szene in der Regel undurchsichtig und glatt sind, folgen weitere Einschränkungen.

In den folgenden Abschnitten werden wir auf die einzelnen Möglichenkeiten zur Reduzierung der Mehrdeutigkeiten im Detail eingehen. Dabei werden sowohl Heuristiken als auch physikalische Gesetzmäßigkeiten behandelt werden.

53.2 Geometrische Einschränkungen aus der Bildentstehung

Die epipolare Geometrie

[8]

Aus der relativen Orientierung der beiden verwendeten Kameras zum Zeitpunkt der Aufnahme eines Stereobildpaares sowie dem zugrundegelegten Kameramodell folgen einige der wichtigsten Einschränkungen für mögliche Stereozuordnungen, die zusammengenommen unter dem Bergriff *epipolare Geometrie* bekannt sind. Diese Einschränkungen beruhen einzig und allein auf physikalischen Gesetzmäßigkeiten, und es liegen ihnen keinerlei szenenspezifische Annahmen zugrunde. Sie sind damit allgemein einsetzbar und werden auch in allen bekannten Stereosystemen verwendet. Bei der folgenden Einführung der epipolaren Geometrie wollen wir keine Einschränkung der relativen Orientierung der beiden Kameras zueinander vornehmen, sondern zunächst den allgemeinen Fall beschreiben.

Das Kameramodell Das allgemein übliche Modell einer Kamera ist das einer *Lochkamera*: Die Bildebene befindet sich in einem Abstand b hinter einer Lochblende; das Bild eines Punktes der dreidimensionalen Szene ist definiert als der Schnittpunkt der Bildebene mit dem Projektionsstrahl, der durch den Szenenpunkt und die Lochblende geht. Der geometrischen Raumpunkt der Lochblende wird als *optisches Zentrum* bezeichnet, den Abstand b von Bildebene und optischem Zentrum bezeichnen wir, in Anbetracht einer späteren Erweiterung des Kameramodells, jedoch nicht als Brennweite, sondern als *Bildweite*. Derjenige Projektionsstrahl, der senkrecht auf der Bildebene steht, wird als *optische Achse* bezeichnet, sein Durchstoßpunkt durch die Bildebene als *Kamerahauptpunkt*.

Bei der hier beschriebenen Abbildungsgeometrie erhalten wir ein Bild, das auf dem Kopfe steht und seitenverkehrt ist. Wir können die Bildebene jedoch um das optische Zentrum nach vorn spiegeln und erhalten dann ein mathematisch äquivalentes Kameramodell mit seitenrichtigen Bildern. In diesem Kameramodell durchstößt ein von einem Szenenpunkt kommender Projektionsstrahl zuerst die Bildebene, bevor er das optische Zentrum passiert.

[8] Die Abschnitte **??** sowie 53.3 stützen sich auf eine von Carsten Schröder erarbeitete Übersicht.

Jede Kamera hat ihr eigenes *Kamerakoordinatensystem*. Sein Ursprung liegt im optischen Zentrum, die z-Achse fällt mit der optischen Achse zusammen und die x- und y-Achsen liegen in der Bildebene. Der Kamerahauptpunkt hat in diesem System die Koordinate $\mathbf{h} = (0, 0, b)^T$.

Die Stereogeometrie Bei der Aufnahme eines Stereobildpaares werden zwei Kamerasysteme verwendet, die in dem hier zunächst betrachteten allgemeinen Falle beliebig gegeneinander verschoben und rotiert sein können (siehe hierzu Abbildung 1). Im allgemeinen werden sich die beiden Bildebenen dann in einer Geraden schneiden.[9] Die Verbindungsgerade zwischen den beiden optischen Zentren, die sogenannte *Basislinie*, durchstößt im allgemeinen beide Bildebenen. Diese Durchstoßpunkte sind die Bilder des optischen Zentrums der jeweils anderen Kamera und werden als *Epipole* bezeichnet.

Wird nun ein Szenenpunkt auf beide Bildebenen abgebildet, so definieren dieser Szenenpunkt sowie die beiden optischen Zentren und damit auch die beiden Projektionsstrahlen und die Basislinie eine Ebene – die sogenannte *epipolare Ebene*. Eine solche epipolare Ebene schneidet die beiden Bildebenen jeweils in einer Geraden, den sogenannten *epipolaren Linien*. Auf einer epipolaren Linie liegen damit die Bildpunkte all derjenigen Szenenpunkte, die sich irgendwo in der dazugehörigen epipolaren Ebene befinden. Auch der Epipol einer Bildebene liegt als Bild des optischen Zentrums der anderen Kamera auf dieser epipolaren Linie. Da der Epipol einer Bildebene aber ein Punkt auf der Basislinie ist, liegt er in allen möglichen epipolaren Ebenen und damit auch auf allen dazugehörigen epipolaren Linien dieser Bildebene. Anders ausgedrückt bedeutet dies: Alle möglichen epipolaren Linien einer Bildebene bilden eine Geradenbüschel – der gemeinsame Punkt des Geradenbüschels ist Epipol dieser Bildebene.

[9] Mit Bildebenen sind hierbei die Ebenen unendlicher Ausdehnung gemeint, nicht nur deren Ausschnitte, die unsere tatsächlichen Bilder enthalten.

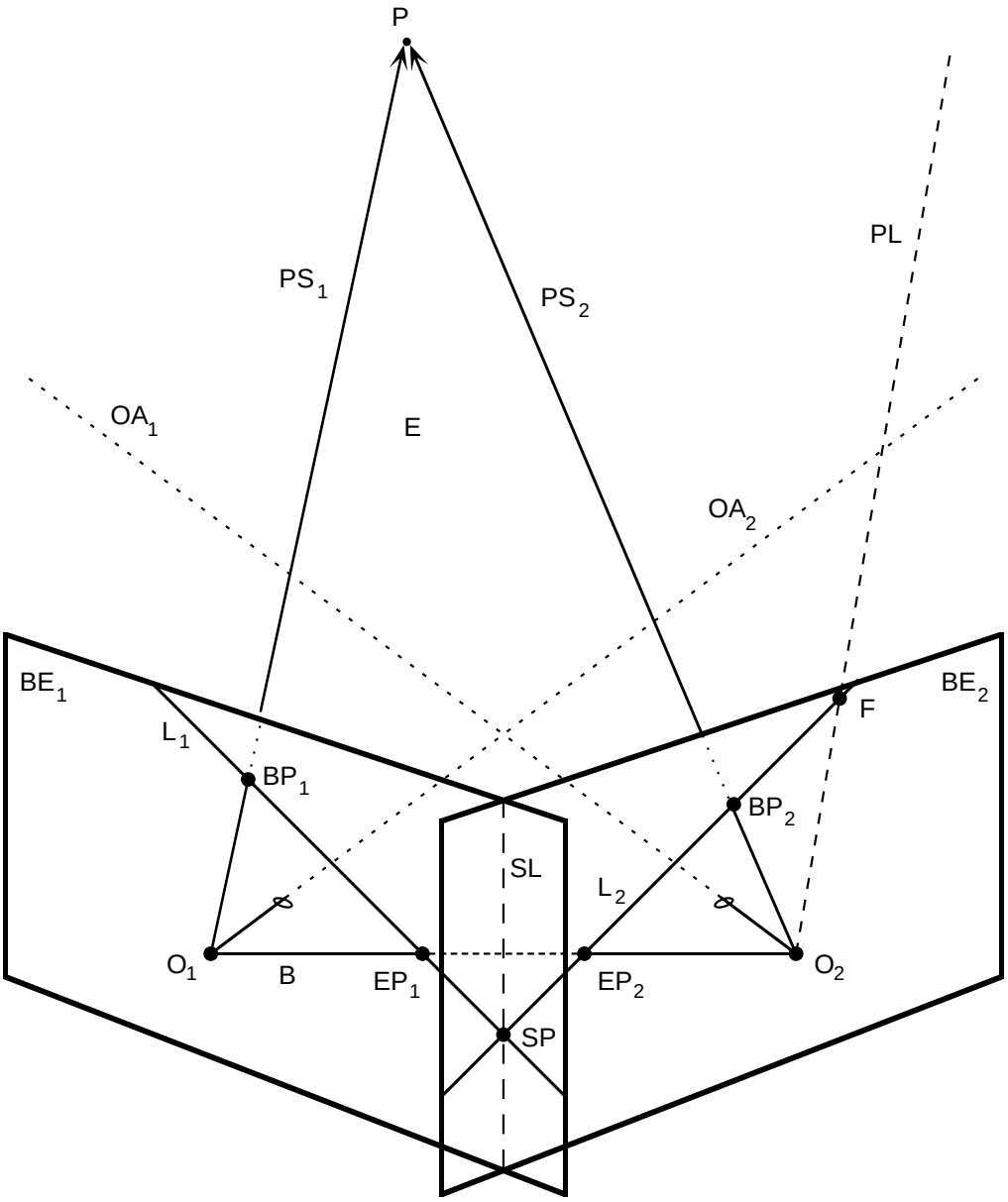


Abbildung 1: Die epipolare Geometrie (*BE* Bildebene, *O* optisches Zentrum, *OA* optische Achse, *B* Basislinie, *SL* Schnittlinie der Bildebenen, *P* Szenenpunkt, *PS* Projektionsstrahl, *BP* Bildpunkt, *E* epipolare Ebene, *L* epipolare Linie, *EP* Epipol, *SP* Schnittpunkt der epipolaren Linien, *PL* Parallele zum Projektionsstrahl, *F* Fluchtpunkt).

Aus der Abbildung 1 können wir auch noch ersehen, daß sich die beiden zu einer epipolaren Ebene gehörenden epipolaren Linien der Bildebenen – nennen wir sie *korrespondierende* epipolare Linien – in einem gemeinsamen Raumpunkt schneiden, wobei dieser Schnittpunkt auf der Schnittgeraden der beiden Bildebenen liegt. Bezeichnen wir die Mengen der Punkte der beiden Bildebenen als BE_1, BE_2, die der epipolaren Ebene als E, die der epipolaren Linien als L_1, L_2, die der Schnittgeraden der beiden Bildebenen als SL und die des Schnittpunktes als SP, dann gilt:

$$\begin{aligned} L_1 &= E \cap BE_1 \\ L_2 &= E \cap BE_2 \\ SL &= BE_1 \cap BE_2 \end{aligned}$$

Der Schnitt der beiden epipolaren Linien ist nun definiert als

$$\begin{aligned} SP &= L_1 \cap L_2 \\ &= E \cap BE_1 \cap BE_2 \\ &= E \cap SL \end{aligned}$$

und liegt damit auf der Schnittgeraden SL der beiden Bildebenen. Voraussetzung für seine Existenz ist allerdings, daß die beiden Bildebenen nicht parallel sind ($BE_1 \cap BE_2 \neq \emptyset$) und daß die beiden epipolaren Linien und die Schnittgerade SL nicht jeweils parallel zueinander sind ($E \cap BE_1 \cap BE_2 \neq \emptyset$). In dem Falle, in dem die beiden Bildebenen identisch sind, ist der Schnitt der epipolaren Linien außerdem kein Punkt mehr
($BE_1 \cap BE_2 = BE_1 \Rightarrow L_1 = L_2$).

Aus dieser Tatsache lassen sich zwar keine Einschränkungen der möglichen Stereozuordnungen ableiten, sie verhilft aber zu einem tieferen Verständnis der allgemeinen Stereogeometrie und ermöglicht es z.B., bei bekannten Epipolen und bekannter Schnittgerade der Bildebenen, zu einem gegebenen Bildpunkt auf einfache Weise die beiden dazugehörigen korrespondierenden epipolaren Linien zu konstruieren.

Hiermit haben wir die allgemeine epipolare Geometrie bereits beschrieben. Welche Einschränkung ergibt sich hieraus nun für die möglichen Stereozuordnungen?

Wird ein Szenenpunkt beispielsweise auf einen Bildpunkt in der linken Bildebene abgebildet, so ist die epipolare Ebene, in der dieser Szenenpunkt liegt, durch den Projektionsstrahl im linken Bild (die Verbindungslinie optisches Zentrum – Bildpunkt) und durch die Basislinie eindeutig bestimmt. Der zu diesem Bildpunkt korrespondierende Bildpunkt des rechten Bildes kann nun aber nur auf der dazugehörigen epipolaren Linie der rechten Bildebene liegen.

Die fundamentale Einschränkung der möglichen Stereozuordnungen, die sich aus der epipolaren Geometrie ergibt, lautet also:

> Zwei Bildpunkte, die Abbildungen ein und desselben Szenenpunktes auf die linke bzw. rechte Bildebene sind, liegen auf den beiden korrespondierenden epipolaren Linien.
>
> Die Suche nach einer korrekten Stereozuordnung für einen gegebenen Bildpunkt reduziert sich damit von einem zweidimensionalen auf ein eindimensionales Problem.

Die Standard-Stereogeometrie Im vorigen Abschnitt haben wir die allgemeine Stereogeometrie beschrieben, bei der keine Annahmen über die relative Orientierung der beiden Kameras zueinander gemacht werden. Um von der geschilderten Einschränkung möglicher Stereozuordnungen Gebrauch machen zu können, ist für diesen allgemeinen Fall jedoch stets die aufwendige Berechnung der epipolaren Linien notwendig. Von besonderer Bedeutung ist daher eine spezielle Annordnung der Kamerasysteme, bei der die beide Bildebenen identisch sind ($BE_1 = BE_2$); die beiden Bilder sind lediglich zwei unterschiedliche Ausschnitte dieser Ebene. Diesen Fall erhalten wir, wenn beide Kamerasysteme so ausgerichtet werden, daß ihre Bildebenen parallel zur Basislinie sind und außerdem die Bildweiten b_l und b_r der beiden Kameras gleich sind. Hieraus folgt dann, daß zwei korrespondierende epipolaren Linien nur eine Gerade in der gemeinsamen Ebene bilden: sie sind identisch $(L_1 = L_2)$.

Richtet man die beiden Kamerasysteme zudem so aus, daß die Bildzeilen der bei der Aufnahme entstehenden digitalen Bilder parallel zur Basislinie liegen und definiert die Bildkoordinatensysteme so, daß ihr Ursprung jeweils an der gleichen Stelle der digitalen Bilder zu liegen kommt, so daß zueinander kollineare Bildzeilen der beiden Bilder jeweils die gleiche Zeilenkoordinate haben, dann sind die zu einem Pixel eines digitalen Bildes gehörenden epipolaren Linien in einfacher Weise durch seine Zeilenkoordinate gegeben.

Wegen der einfachen Ermittelbarkeit epipolarer Linien wird diese spezielle Stereogeometrie mit den beschriebenen weiteren Annahmen in vielen Stereosystemen explizit vorausgesetzt; wir wollen diese Anordnung daher *Standard-Stereogeometrie* nennen.

Diese Standard-Stereogeometrie hat laut Ayache and Faverjon, 1987 jedoch auch einige entscheidende Nachteile:

- Sie stellt extreme Anforderungen an die genaue Positionierbarkeit der Sensorflächen der bei einer Aufnahme verwendeten Kamerasysteme und wird aus diesem Grunde praktisch nicht realisierbar sein.
- Um bei der 3D-Rekonstruktion der beobachteten Szene eine große Genauigkeit zu erreichen, ist es wünschenswert, die Szene von möglichst unterschiedlichen Standpunkten aus zu beobachten; die Basislinie sollte also unter diesem Gesichtspunkt möglichst lang sein.[10] Damit die beiden Kamerasysteme dann überhaupt noch die gleichen Objekte im Blickwinkel haben, ist es erforderlich, sie gegeneinander zu drehen. Die Bildebenen sind dadurch jedoch nicht mehr parallel zueinander.
- In „künstlichen“, also von Menschenhand geschaffenen Umgebungen wie Städten, Gebäuden, Fabrikhallen etc. treten häufig horizontale Linien auf. Bilden die Basislinie sowie die beiden optischen Achsen des Stereokamerasystems in dieser Umgebung ebenfalls eine horizontale Linie, dann liegen alle Punkte einer horizontalen Linie der Szene auf einer epipolaren Ebene und damit in einem Bild auf derselben epipolaren Linie. Die Mehrdeutigkeiten bei der Stereozuordung von Bildmerkmalen sind hierdurch dann besonders groß.

In einigen anderen Stereosystemen (z.B. von Posch, 1988) werden bei der Aufnahme von Stereobildern ebenfalls beliebige Anordnungen der beiden Kamerasysteme zueinander zugelassen. Die dabei entstehenden Bilder werden dann jedoch nachträglich unter Zuhilfenahme der bekannten Translations- und Rotationsparameter so transformiert, daß sie die oben beschriebene Eigenschaft der Standard-Stereogeometrie besitzen: korrespondierende epipolare Linien haben in beiden Bildern die gleiche Zeilenkoordinate.

Einschränkung epipolarer Linien auf ein Intervall

Wie im vorigen Abschnitt ausführlich beschrieben, wird die Suche nach möglichen Stereozuordnungen zu einem detektierten Bildmerkmal durch die epipolare Geometrie auf eine Gerade im anderen Bild beschränkt.

[10] Unter dem Gesichtspunkt des Korrespondenzproblems ist natürlich eine kurze Basislinie wünschenswert, damit die beiden entstehenden Bilder einander möglichst ähnlich sind.

Dieser Suchbereich der theoretisch unendlichen, praktisch natürlich durch die Ränder eines Bildes beschränkten epipolaren Linie läßt sich durch die Kenntnis weiterer Eigenschaften des zugrunde gelegten Kameramodells jedoch noch weiter auf ein Intervall einschränken.

Verwendung des Lochkameramodells Das bereits beschriebene Modell einer Lochkamera realisiert eine perspektivische Abbildung der dreidimensionalen Szene auf die zweidimensionale Bildebene. Eine entscheidende Eigenschaft der perspektivischen Projektion ist jedoch, daß eine unendliche Gerade der dreidimensionalen Szene nicht auf eine unendliche Gerade in der Bildebene abgebildet wird, sondern auf der Bildebene in einem *Fluchtpunkt* endet. Der Fluchtpunkt ist dabei das Bild eines Punktes auf der Geraden in der Szene, der im Kamerakoordinatensystem die z-Koordinate „Unendlich“ hat.

Haben wir beispielsweise einen Punkt in der linken Bildebene gegeben, so können wir unter Kenntnis der relativen Orientierung der beiden Kamerasysteme zueinander für den Projektionsstrahl dieses Punktes den Fluchtpunkt in der rechten Bildebene berechnen. Dieser Fluchtpunkt liegt auf der zu dem gegebenen Punkt gehörenden epipolaren Line der rechten Bildebene und schränkt den Suchbereich nach einem korrespondierenden Punkt damit auf einen Strahl ein; ein zu dem gegebenen Punkt korrespondierender Punkt der rechten Bildebene kann auf der epipolaren Linie nicht jenseits des Fluchtpunktes liegen (Horn, 1986). Abbildung 1 zeigt den Fluchtpunkt als den Durchstoßpunkt eines Projektionsstrahls durch die rechte Bildebene; da es sich bei dem Fluchpunkt um die (gedachte) Abbildung eines Punktes im Unendlichen handelt, muß dieser Projektionstrahl eine Parallele zu dem Projektionsstrahl des Szenenpunktes P im linken Bild sein.

Neben der äußerst einfachen Ermittlung der zu einem gegebenen Bildpunkt gehörenden epipolaren Linien ermöglicht die oben beschriebene Standard-Stereogeometrie eine ebenso einfache Ermittlung des zu einem Projektionsstrahl eines Bildpunktes gehörenden Fluchtpunktes in der anderen Bildebene: Der Fluchtpunkt hat bei dieser speziellen Stereogeometrie die gleiche Zeilen- und Spaltenkoordinate wie der Bildpunkt selbst. Dies ist ein weiterer Grund für die häufige Verwendung der Standard-Stereogeometrie.

Der Fluchtpunkt definiert zu einem gegebenen Bildpunkt auf einer epipolaren Linie sozusagen einen Fernpunkt, also das Bild des am weitesten entfernten Szenenpunktes auf dem Projektionsstrahl des anderen Kamerasystems. Neben diesem Fernpunkt gibt es auf einer epipolaren Linie auch einen Nahpunkt, also das Bild des am dichtesten am anderen Kamerasystem auf dessen Projektionsstrahl liegenden Szenenpunktes, der noch auf dessen Bildebene abgebildet wird.

Bei dem Modell einer Lochkamera ist dieser Nahpunkt gegeben durch die Lochblende selbst, also durch das optische Zentrum des Kamerasystems. Man vergegenwärtige sich hierzu, daß bei einer realen Lochkamera die Bildebene hinter dem optischen Zentrum liegt. Bei dem von uns verwendeten Lochkameramodell haben wir die Bildebene lediglich der Bequemlichkeit halber vor das optische Zentrum gespiegelt, da dies mathematisch äquivalent ist. Das Bild des optischen Zentrums des einen Kamerasystems auf der Bildebene des anderen Kamerasystems ist gerade der Epipol dieser Bildebene. Bei Verwendung des Modells einer Lochkamera ist der Nahpunkt auf einer epipolaren Linie also definiert durch den Epipol der Bildebene. Hierdurch wird der Suchbereich auf der epipolaren Linie also ebenfalls auf einen Strahl eingeschränkt; ein zu einem gegebenen Punkt korrespondierender Punkt in der anderen Bildebene kann auf der epipolaren Linie nicht jenseits des Epipoles liegen (Horn, 1986).

Zusammenfassend erhalten wir hiermit die folgende Einschränkung für mögliche Stereozuordnungen:

> Fluchtpunkt und Nahpunkt schränken den Suchbereich nach korrespondierenden Punkten auf ein Intervall der epipolaren Linie ein.

In den meisten Fällen werden jedoch Fluchtpunkt und Nahpunkt nicht gemeinsam innerhalb des Ausschnittes der Bildebene liegen, der das tatsächlich aufgenommene Bild enthält. Bei der Standard-Stereogeometrie, bei der die optischen Achsen der beiden Kamerasysteme parallel zueinander sind, liegt der Fluchtpunkt zu einem gegebenen Projektionsstrahl immer innerhalb des Bildes. Der Nahpunkt liegt hierbei jedoch nicht innerhalb des Bildes; ein Epipol existiert in diesem Falle nicht einmal. Dreht man die beiden Kamerasysteme nun aufeinander zu, dann liegen nicht mehr zu allen Projektionsstrahlen eines Bildes die Fluchtpunkte innerhalb des anderen Bildes. Der Epipol einer Bildebene wandert beim Aufeinanderzudrehen der Kamerasysteme auf dieser Bildebene jedoch immer weiter an das tatsächliche Bild heran.

Hat man die beiden Kamerasysteme dann so weit aufeinander zugedreht, daß sich der Epipol einer Bildebene tatsächlich innerhalb des Bildes befindet, dann ist in diesem Bild natürlich auch die andere Kamera sichtbar. Dies ist in der Regel jedoch nicht wünschenswert. Die Einschränkung einer epipolaren Linie auf ein Intervall durch den Epipol als Nahpunkt ist daher praktisch ohne Bedeutung. Der Fluchtpunkt hat dagegen in vielen Fällen eine große Bedeutung; die Wahrscheinlichtkeit, daß er innerhalb eines Bildes liegt, ist bei kleinen Drehwinkeln der Kamerasysteme hoch und ergibt dann eine nützliche Einschränkung des Suchbereiches auf einer epipolaren Linie.

Verwendung des Linsenkameramodells: Wir wir gesehen haben, stellt der Epipol als Nahpunkt praktisch keine weitere Einschränkung des Suchbereiches auf einer epipolaren Linie dar. Es stellt sich jedoch die Frage, ob bei den tatsächlich bei der Aufnahme von Bildern verwendeten Videokameras das optische Zentrum ein physikalisch sinnvoller Nahpunkt ist. Bei einer Videokamera besteht das optische System natürlich nicht aus einer Lochblende sondern aus einem Linsensystem. Der geometrische Raumpunkt des optische Zentrums eines solchen Linsensystems kann durch dieses optische System jedoch nicht auf die Bildebene abgebildet werden. Das optische Zentrum ist hierbei als Nahpunkt also physikalisch unmöglich. Ein möglicher Nahpunkt muß zumindest außerhalb des Linsensystems liegen.

Um einen bei der Verwendung realer Videokameras physikalisch sinnvollen Nahpunkt konstruieren zu können, gehen wir jetzt über auf ein anderes Kameramodell, das Modell einer *Linsenkamera*. Hierbei besteht das optische System anstelle einer Lochblende aus einer idealen Linse mit einer *Brennweite* f. Unter einer idealen Linse verstehen wir hierbei eine Linse, die keinerlei Abbildungsfehler hat. Berücksichtigen wir die Abbildungsfehler einer realen Linse also, dann hat das Linsenkameramodell die gleichen geometrischen Eigenschaften wie unser Lochkameramodell. Der hier auszunutzende Unterschied zwischen den beiden Kameramodellen ist die Tatsache, daß eine Linse nur diejenigen Szenenpunkte scharf auf die Bildebene abbildet, die sich in einer bestimmten Entfernung auf einer Ebene, der sogenannten *Gegenstandsebene*, vor der Kamera befinden.

Der Abstand der Gegenstandsebene von der Kamera (z-Koordinate im Kamerakoordinatensystem) wird als *Gegenstandsweite* a bezeichnet und ist abhängig von der Brennweite sowie der Bildweite. Der Zusammenhang zwischen Brennweite, Bildweite und Gegenstandsweite ist dabei durch die wohlbekannte *Linsengleichung* der geometrischen Optik definiert:

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{b}$$

Alle Szenenpunkte, die sich vor oder hinter der Gegenstandsebene befinden, werden nicht auf einen Punkt abgebildet, sondern auf ein kleines Scheibchen mit einem Durchmesser s. Berücksichtig man jedoch die endliche Auflösung eines physikalischen Sensors, z.B. eines CCD-Chips, dann werden auch noch Szenenpunkte in einem Bereich vor und hinter der Gegenstandsebene scharf auf die Bildebene abgebildet, wenn ihr Scheibchendurchmesser s den Durchmesser eines Sensorelementes nicht überschreitet. Dieser Entfernungsbereich vor der Kamera, dessen Punkte gerade noch scharf auf die Bildebene abgebildet werden, wird als *Schärfentiefebereich* bezeichnet. Die Ausdehnung des Schärfentiefebereiches ist bei gegebenem Scheibchendurchmesser abhängig von Bildweite und Brennweite, vor allem jedoch vom *effektiven Linsendurchmesser* d, der bei einem realen Kameraobjektiv stets durch die sogenannte *Blendenzahl* n definiert ist: $n = f/d$.

Die beiden Extremwerte des Schärfentiefebereiches a_{min} und a_{max} können wie folgt berechnen werden, wobei alle erforderlichen Daten aus einer Kalibrierung des Kamerasystems und den Angaben über den Sensor zur Verfügung gestellt werden müssen:

$$\begin{aligned} a_{min} &= \frac{fb}{b - f + ns} \\ a_{max} &= \begin{cases} \frac{fb}{b-f-ns} & \text{wenn } b - f \geq ns \\ \infty & \text{sonst} \end{cases} \end{aligned}$$

Für eine praktische Anwendung ist der so berechnete Schärfentiefenbereich jedoch zu klein: Auch Szenenpunkte, die unschaft abgebildet werden, können noch zu detektierbaren Merkmalen im Bild führen. Eine sinnvolle Grenze für die zugelassene Unschärfe, und damit auch ein Schwellwert für den maximal zulässigen Scheibchendurchmesser s, kann aber durch die Forderung ermittelt werden, daß unscharf abgebildete Szenenpunkte im Bild überhaupt noch als Merkmale detektiert und auch hinreichend genau lokalisiert werden müssen. Hierfür ist eine genauere Analyse der eingesetzten Merkmalsdetektoren notwendig.

Die Bilder eines so berechneten Nah- und Fernpunktes für einen gegebenen Projektionsstrahl schränken den Suchbereich nach korrespondierenden Punkten auf einer epipolaren Linie, wie schon im vorigen Abschnitt bei Verwendung des Lochkameramodells, auf ein Intervall ein. Dieses Intervall ist bei Verwendung des Linsenkameramodells jedoch deutlich kürzer und die Wahrscheinlichkeit, daß bei einer bestimmten relativen Orientierung der beiden Kamerasysteme beide Extrempunkte des Intervalls innerhalb des Bildausschnittes liegen, erhöht sich damit. Wir erhalten durch eine bessere Modellierung des optischen Systems einer Videokamera also eine deutlich wirksamere Einschränkung der möglichen Stereozuordnungen.

Geometrische Ähnlichkeit von Merkmalen

Arnold and Binford, 1980 zeigen durch ein statistische Analyse, daß bei einer angenommenen Gleichverteilung der dreidimensionalen Orientierung von Kanten – hier also Geraden – in der Szene (Gleichverteilung auf der Gaußschen Kugel) die überwältigende Mehrkeit von korrespondierenden Kantensegmenten im Bild sowohl die gleiche Orientierung im Bild (z.B. Winkel zu den Bildzeilen) als auch die gleiche Länge haben, wenn man die oben beschriebene Standard-Stereogeometrie zugrunde legt und das Verhältnis von Basislinie B zu Entfernung von der Kamera a genügend klein ist.

Aus diesem Ergebnis kann dann eine Einschränkung für mögliche Stereozuordnungen von Kanten formuliert werden, die z.B. von Ayache and Faverjon, 1987 verwendet wird und von ihnen als *geometrische Ähnlichkeit* bezeichet wird:

Das Winkelkriterium:
: Zwei korrespondierende Geraden eines Stereobildpaares müssen eine ähnliche Orientierung in den beiden Bildern haben.

Das Längenkriterium:
: Zwei korrespondierenden Geraden eines Stereobildpaares müssen eine ähnliche Länge in den beiden Bildern habe.

Bei dieser Einschränkung handelt es sich jedoch lediglich um eine Heuristik. Liegt eine beobachtete Kante der Szene beispielsweise dicht am Kamerasystem, ist eine Kante stark von dem Beobachter weggeneigt (kleiner Winkel zur optischen Achse) oder sind die beiden Kamerasysteme sogar aufeinander zugedreht, dann sind die zugrundeliegenden Annahmen nicht mehr erfüllt, und korrekte Stereozuordnungen von Kanten können hierdurch dann teilweise unterbunden werden.

Ayache and Faverjon, 1987 verwenden diese Beschränkung nicht für alle Geraden eines Stereobildes. Um möglichen Fehlern bei der Detektion der Kanten im Bild begegnen zu können, wenden sie die Heuristik der geometrischen Ähnlichkeit nur an, wenn

- eine Gerade im Bild nicht zu lang ist, da sonst die Gefahr besteht, daß sie im jeweils anderen Bild unterbrochen ist,
- eine Gerade im Bild nicht zu kurz ist, da ihre Orientierung im Bild sonst nur schlecht geschätzt werden kann.

53.3 Die Ausnutzung von Objekteigenschaften

Kompatibilität von Merkmalen

Eine Einschränkung möglicher Zuordnungen von Bildmerkmalen, sowohl in der zeitlichen Folge als auch in den Stereobildern, geht auf die folgende von Marr, 1982 als *Kompatibilität* von Merkmalen bezeichnete Forderung zurück:

> Zwei Merkmale in verschiedenen Bildern können dann und nur dann korrespondieren, wenn sie die gleiche physikalische Ursache in der Szene haben können.

Diese Forderung ist offensichtlich: Zwei tatsächlich korrespondierende Merkmale sind Abbildungen ein und derselben Struktur in der Szene und haben daher dieselbe physikalische Ursache. Eine Einschränkung erhalten wir dann, wenn wir die Merkmale unserer Bilder nach ihrer physikalischen Ursache klassifizieren können: Nach einer solchen Klassifikation sind nur noch Zuordnungen zwischen Merkmalen der gleiche Klasse möglich.

Eine solche Klassifikation der Kanten in den Bildern, die zum Teil aufgrund von Farbinformationen vorgenommen werden kann (siehe hierzu die Arbeiten von Gershon et al., 1987 und Klinker et al., 1988), wäre z.B. die Unterscheidung in

Orientierungskanten: Dies sind Kanten, die durch eine Diskontinuität der Orientierung von Objektoberflächen entstehen.

Reflektanzkanten: Dies sind Kanten, die durch eine Diskontinuität der Reflektanzeigenschaft von Objektoberflächen entstehen (z.B. Änderung des Oberflächenmaterials).

Schattenkanten: Dies sind Kanten, die durch eine Diskontinuität in der Beleuchtung von Objektoberflächen entstehen.

Glanzlichtkanten: Dies sind die Begrenzungen von Glanzlichtern, die durch eine spezielle Orientierung von Lichtquelle, Objektoberfläche und Beobachter zustande kommen.

Tangentialkanten: Dies sind Kanten, bei denen der Beobachter eine glatte Objektoberfläche tangential sieht. Hierbei handelt es sich also nicht um eine physikalische Diskontinuität einer Oberfläche in der Szene, sondern lediglich um eine vom Beobachter aus betrachtete Abgrenzung eines Objektes gegen dessen Hintergrund. Jede Tangentialkante existiert nur aufgrund einer speziellen Sicht.

Eine solche Klassifikation der Kanten in den Bildern ist noch aus einem anderen Grunde erforderlich: Die Bildpunkte, aus denen die einander zuzuordnenden Mermale bestehen, müssen Abbilder ein und derselben Punkte auf den Oberflächen in der Szene sein – es muß sich also um *körperfeste* Merkmale handeln. Hieraus folgt, daß man nur bestimmte Klassen von Kanten bei der Zuordnung berücksichtigen darf, je nachdem, ob es sich um Stereozuordnungen oder zeitliche Zuordnungen handelt:

Bei Stereozuordnungen dürfen nur die folgenden Klassen berücksichtigt werden: Orientierungskanten, Reflektanzkanten und Schattenkanten.

Nicht berücksichtigt werden dürfen Glanzlichtkanten und Tangentialkanten, da diese von der Beobachterposition abhängen.

Bei zeitlichen Zuordnungen dürfen nur die folgenden Klassen berücksichtigt werden: Orientierungskanten und Reflektanzkanten.

Nicht berücksichtig werden dürfen im allgemeinen Schattenkanten, Glanzlichtkanten und Tangentialkanten.

Bei zeitlichen Zuordnungen dürfen also Schattenkanten zusätzlich nicht berücksichtigt werden, da deren Position auf einer Objektoberfläche von der Position der Lichtquelle abhängt. Es erscheint jedoch denkbar, daß man unter bestimmten Voraussetzungen aus der zeitlichen Zuordnung von Schattenkanten die Bewegung der Lichtquelle ermittelt.

Ein Problem bei der vollständigen Kantenklassifikation ist die Tatsache, daß sie sich nicht allein aufgrund von Informationen aus einem Bild durchführen läßt, sondern daß hierfür zumindest eine qualitative 3D-Rekonstruktion der beobachteten Szene erforderlich ist. Eine Möglichkeit für eine 3D-Rekonstruktion ist jedoch gerade die Zuordnung von Bildmerkmalen, die ihrerseits auf eine Kantenklassifikation angewiesen zu sein scheint. Dieses Problem haben wir zu Beginn das „Bootstrap-Problem“ genannt.

Das „Continuity Constraint“

Die in folgenden dargestellten Einschränkungen für mögliche Stereozuordnungen gehen von den folgenden Annahmen über Eigenschaften der Objekte unserer natürlichen Umwelt aus:

- Die Oberflächen von Objekten sind in der Regel *undurchsichtig*.
- Die Oberflächen von Objekten sind in der Regel *glatt* und *kontinuierlich*.
- Die Distanz der einzelnen Oberflächenpunkte vom Beobachter variiert *kontinuierlich*.

Diesen einschränkenden Annahmen über die Eigenschaften einer beobachteten Szene werden von Marr und Poggio als „continuity constraint“ bezeichnet (Marr and Poggio, 1976; Marr, 1982). Wir zeigen im folgenden, wie sich aus diesen Annahmen Einschränkungen für mögliche Korrespondenzen ergeben.

„Disparity Limit“ Marr and Poggio, 1976 folgern hieraus, daß die *Disparität*[11] von korrespondierenden Bildpunkten in einem Stereobildpaar kontinuierlich variiert und fordern für mögliche Stereozuordnungen dann, daß die Disparität zweier korrespondierender Punkte eine Schwelle nicht überschreitet. Die Verwendung dieses sogenannten „Disparity Limit“ wird häufig auch durch die Ergebnisse psychophysischer Experimente am menschlichen Sehsystem motiviert (siehe hierzu z.B. (Burt and Julesz, 1980; Marr, 1982; Poggio and Poggio, 1984) und die Diskussion um „Panum's Fusional Area“).

Die Forderung eines „Disparity Limit“ bedeutet jedoch, daß die Objekte der Szene eine durch diese Schwelle definierte Mindestdistanz vom Beobachter (der Kamera) nicht unterschreiten dürfen. Diese Forderung erscheint physikalisch jedoch nicht plausibel.

„Disparity Gradient Limit“ Burt and Julesz, 1980 folgern aus den Ergebnissen psychophysischer Experimente, daß die Frage, ob zwei abgebildete Punkte gleichzeitig durch das menschliche Sehsystem fusioniert und damit scharf gesehen werden können, nicht durch ein „Disparity Limit“, sondern durch ein „Disparity Gradient Limit“ bestimmt wird.

Trivedi and Lloyd, 1985 definieren den Disparitätsgradienten wie folgt:

$$\nabla D = \frac{\|\mathbf{r}_L - \mathbf{r}_R\|}{\frac{1}{2}\|\mathbf{r}_L + \mathbf{r}_R\|}$$

mit $\mathbf{r}_L$ als dem Differenzvektor der beiden Punkte im linken Bild und $\mathbf{r}_R$ als dem Differenzvektor der beiden hierzu korrespondierenden Punkte im rechten Bild.

[11] Die Disparität zweier korrespondierender Punkte L und R ist bei zugrunde gelegter Standard-Stereogeometrie, die wir weiter oben beschrieben haben, definiert als die Differenz der Spaltenkoordinaten beider Bildpunkte $D = column(L) - column(R)$, wobei durch die Standard-Stereogemetrie außerdem gilt: $column(L) \geq column(R)$. Die Disparität sagt damit etwas über die Entfernung des beobachteten Szenenpunktes vom Stereokamerasystem aus.

Wird an diesen Disparitätsgradienten zweier korrespondierender Punktpaare eine Schwelle angelegt, so bedeutet dies, daß hiermit eine maximal zulässige Neigung von Objektoberflächen der Szene gegenüber dem Beobachter festlegt wird. Trivedi and Lloyd, 1985 zeigen, daß das Anlegen eines „Disparity Gradient Limit“ von weniger als 2 sicherstellt, daß keine Stereozuordnungen zwischen zwei Punktpaaren durchgeführt werden, die in der Szene auf sich selbst verdeckenden Oberflächen liegen würden. Der Grenzwert von 2 entspricht gerade der Situation, bei der ein Punkt des einen Bildes zwei Punkten des anderen Bildes zugeordnet würde, also einer Selbstverdeckung (vergleiche hierzu die Diskussion der Eindeutig von Merkmalszuordungen als Konsequenz des Generalitätsprinzips auf Seite 639).

Ein „Disparity Gradient Limit“ wird in einigen Stereosystemen verwendet, um die Menge möglicher Stereozuordnungen einzuschränken (siehe z.B. (Pollard et al., 1985a; Pollard et al., 1985b; Ayache and Faverjon, 1987)). Pollard et al., 1985b geben an, daß mit einer Schwelle aus dem Intervall $[0.5 \ldots 1.0]$ die Menge der möglichen Stereozuordnungen drastisch eingeschränkt wird, ohne dadurch eine nennenswerte Anzahl von korrekten Stereozuordnungen auszuschließen. Sie zeigen außerdem durch eine statistische Analyse (analog zu den Untersuchungen zur geometrischen Ähnlichkeit von Arnold and Binford, 1980), daß die Wahrscheinlichkeit für einen gegebenen Disparitätsgradienten ∇D im Bild bei angenommener Gleichverteilung der Oberflächenorientierungen in der Szene (Gleichverteilung auf einer Gaußschen Kugel) schon für kleine Werte von ∇D sehr groß ist: beispielsweise 90% für $\nabla D = 0.5$ (bei einer Basislinie von $B = 6.5\mathrm{cm}$ und einer Mindestdistanz der Oberflächen vom Beobachter von $a = 26\mathrm{cm}$).

„Ordering Constraint“ Fordert man einen Disparitätsgradienten ≤ 2 für die Zuordnungen von Punktpaaren, die auf einer gemeinsamen epipolaren Linie liegen, dann erhält man das bereits von Baker, 1982 definierte „Ordering Constraint“. Dieses beschreibt die Annahme, daß Punkte, die auf derselben epipolaren Linie liegen, im anderen Bild in der gleichen Reihenfolge auf die korrespondierende epipolare Linie abgebildet werden. Das „Ordering Constraint“ macht damit eine weniger einschränkende Annahme an die Eigenschaften der beobachteten Szene als dies bei der Forderung eines „Disparity Gradient Limit“ der Fall ist. Es wurde neben Baker unter anderen auch von Ohta and Kanade, 1985 in ihrem Stereosystem verwendet.

„Generalized Ordering Constraint“ Weit einschränkendere Annahmen über die Eigenschaften der beobachteten Szene als bei der Forderung eines „Disparity Gradient Limit“ liegen dem von Yuille and Poggio, 1984 formulierten „Generalized Ordering Constraint“ zugrunde: Hierbei wird angenommen, daß die Punkte eines Bildes in der gleichen Reihenfolge von links nach rechts auf das andere Bild eines Stereobildpaares abgebildet werden. Die Bedingung des „Ordering Constraints“, daß hierfür jeweils nur Punkte betrachtet werden, die auf zwei korrespondierenden epipolaren Linien liegen, wurde hierbei fallen gelassen. Dies impliziert jedoch, daß zum Beispiel ein vertikal angeordnetes Punktpaar, das auf einer nur leicht horizontal zum Beobachter geneigten Oberfläche liegt, dieser Einschränkung nicht genügt, und daß deren Abbilder in den beiden Bildern daher nicht gemeinsam eineinder zugeordnet werden könnten. Pollard et al., 1985b lehnen die Verwendung dieses „Generalized Ordering Constraint“ aus diesem Grunde auch ab.

Die hier vorgestellten Einschränkungen für mögliche Stereozuordnungen lassen sich nicht aus physikalischen Gesetzen ableiten, es werden vielmehr die eingangs genannten einschränkende Annahmen über die beobachtete Szene gemacht, die dann die hier besprochenen Einschränkungen möglicher Stereozuordnungen implizieren. Es kann vorkommen, daß korrekte Zuordnungen diesen Einschränkungen nicht genügen und hierdurch dann aus der Menge möglicher Zuordnungen ausgeschlossen werden. Insbesondere bei Objekten, deren Ausdehnung klein ist gegenüber den Abständen zwischen den Objekten, und bei kurzen Distanzen der Objekte zum Beobachter kann häufig das „Ordering Constraint“ verletzt sein.

Ausnutzung dynamischer Objekteigenschaften

In diesem Abschnitt sollen aus der Kenntnis physikalisch möglicher Bewegungen massebehafteter Objekte im dreidimensionalen Raum Einschränkungen für mögliche zeitliche Zuordnungen formuliert werden. Hierbei betrachten wir jedoch Zuordnungen von bereits aus Stereozuordnungen rekonstruierten Punkten im dreidimensionalen Raum der Szene, da hierdurch weit strengere Einschränkungen formuliert werden können. Dies bedeutet jedoch nicht, daß die dazu notwendigen Stereozuordnungen bereits alle korrekt sein müssen, es erlaubt vielmehr die Überprüfung der Stereozuordnungen auf Korrektheit.

Maximale Geschwindigkeit Eine oft verwendete Einschränkung ergibt sich durch die Forderung, daß die Bewegung eines Objektes in der Szene eine maximale Geschwindigkeit nicht überschreiten darf (siehe z.B. (Barnard and Thompson, 1980; Dreschler and Nagel, 1982; Jenkin, 1984; Jenkin and Tsotsos, 1986)). Ein hierfür festgelegter Schwellwert definiert dann eine Strecke, um die sich ein Punkt der Szene von einem Zeitpunkt zum nächsten bewegt haben kann. Die Richtung der Bewegung wird hierbei jedoch nicht betrachtet, so daß sich für die möglichen zeitlichen Zuordnungen eines Punktes der dreidimensionalen Szene ein kugelförmiger Einschränkungsbereich mit dem Punkt als Mittelpunkt ergibt, innerhalb dessen ein korrespondierender Punkt zum nächsten Zeitpunkt dann liegen muß.

Hierbei wird bereits deutlich, daß die Einschränkung strenger ist, wenn sie im dreidimensionalen Raum der Szene anstatt im zweidimensionalen Raum des Bildes betrachtet wird: Die Forderung einer maximalen Geschwindigkeit der Merkmale im Bild ergäbe einen kreisförmigen Einschränkungsbereich im Bild, der in der Szene einem kegelförmigen Einschränkungsbereich entspräche. Innerhalb dieses Kegels könnten Objekte aber immer noch beliebige Geschwindigkeiten aufweisen, ohne die Einschränkung zu verletzten.

Die hier beschriebene Forderung einer maximalen Geschwindigkeit im Raum läßt sich nicht aus physikalischen Gesetzmäßigkeiten ableiten, da es für beliebige Objekte zumindest praktisch keine obere Grenze für ihre Geschwindigkeit gibt. Für bestimmte Objektklassen existiert jedoch eine obere Grenze ihrer Geschwindigkeit: eine Person wird sich beispielsweise kaum schneller als mit einer Geschwindigkeit von $10\mathrm{m/s}$ bewegen und auch aus der Kenntnis der Art der beobachteten Szene lassen sich maximal mögliche Geschwindigkeiten ableiten: in einer Straßenverkehrsszene mit parkenden Autos und schmaler Fahrbahn kann sich ein PKW kaum schneller als mit einer Geschwindigkeit von $80\mathrm{km/h}$ bewegen.

Zur Formulierung eines sinnvollen Schwellwertes für eine maximale Geschwindigkeit bewegter Objekte der dreidimensionalen Szene ist also szenenspezifisches Wisses vonnöten, das in einem allgemein einsetzbaren System nicht vorausgesetzt werden sollte. Zudem kann es wie beschrieben für unterschiedliche Objektklassen unterschiedliche Schwellwerte geben, deren Verwendung aber eine bereits durchgeführte Klassifikation der Bildpunkte in ihre Zugehörigkeit zu bestimmten Objektklassen voraussetzt.

Aus rein pragmatischen Gründen ergibt sich letztlich doch noch eine Grenze für die zugelassene Geschwindigkeit von Objekten: Ein Szenenpunkt sollte mindesten in zwei aufeinanderfolgenden Bildern sichtbar sein, da sonst keinerlei Aussagen über ihn gemacht werden können; er könnte dann lediglich als Rauschen betrachtet werden. Außerdem darf er sich nur so schnell bewegen, daß in den Bildern keine allzugroße Bewegungunschärfe entsteht. Andernfalls wäre er dann nicht mehr exakt detektierbar und lokalisierbar.

Glatte Bewegung im Raum Wir wollen nun aus den physikalischen Eigenschaften, die uns von unserer natürlichen Umgebung bekannt sind, eine Einschränkung der möglichen Bewegungen der Objekte unserer Umwelt ableiten, aus der wir dann eine Einschränkung für mögliche zeitliche Zuordnungen von Punkten im dreidimensionalen Raum der Szene erhalten.

Die Bahnkurve eines sich durch die Szene bewegenden Punktes, also seine Orts-Zeit-Funktion, ist durch die folgende Gleichung definiert:

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \int_t^{t+1} \left(\mathbf{v}_t + \int_t^{t+1} \mathbf{a}\, dt \right) dt$$

Hierbei sind $\mathbf{s}$ der Ortsvektor, $\mathbf{v}$ der Geschwindigkeitsvektor und $\mathbf{a}$ der Beschleunigungsvektor.

Lassen wir beliebige Beschleunigungen zu, dann gibt es für die Bahnkurve eines Punktes keinerlei Einschränkungen. Objekte unserer natürlichen Umgebung sind jedoch massebehaftet und unterliegen daher dem Trägheitsgesetz. Um eine Änderung der Geschwindigkeit hervorzurufen, ist das Einwirken einer Kraft vonnöten, die in unserer natürlichen Umgebung endlich sein wird. Damit ist aber auch die Beschleunigung endlich, sie ist eine beschränkte Funktion der Zeit.

Gehen wir weiter davon aus, daß sich die auftretenden Kräfte in einem endlichen Zeitintervall nicht beliebig oft sprunghaft ändern können, dann ist die Beschleunigung stückweise stetig. Das Integral der Beschleunigung, dies ist die Geschwindigkeitsfunktion, existiert also und ist stetig, mit der Folge, daß das Integral der Geschwindigkeit, also die Bahnkurve eines Punktes, stetig differenzierbar ist. Diese Eigenschaft der Bahnkurve wollen wir als *Glattheit* der Bewegung definieren:

Definition: 125

Sei die Bahnkurve eines Punktes im dreidimensionalen Raum stetig differenzierbar, dann nennen wir dies eine glatte Bewegung.

Satz: 126

Eine glatte Bewegung folgt aus der Beschränktkeit und stückweisen Stetigkeit der Beschleunigung.

Nach dieser Definition handelt es sich also bei allen Bewegungen in unserer natürlichen Umwelt um glatte Bewegungen. Typische Bewegungsklassen sind hier z.B.: Freier Fall, ballistische Kurven, Bewegungen unter Zwangskräften, wie z.B. das Auftrumpfen eines Balles, Bewegungen von Körpern mit eigenem Antrieb, wie z.B. Autos, oder Kreisbewegungen.

Diese Aussagen beziehen sich alle auf kontinuierliche Orts-Zeit-Funktionen. Bei einer zu verarbeitenden Bildfolge sind die einzelnen Stereobilder, und damit die daraus rekonstruierten Punkte der dreidimensionalen Szene, nur zu diskreten Zeitpunkten vorhanden. Wir haben also keine kontinuierliche Orts-Zeit-Funktion zur Verfügung, sondern müssen aus den diskreten Bahnkurven, die zu jedem Zeitpunkt aus bereits durchgeführten zeitlichen Zuordnungen rekonstruiert worden sind, lokal Werte für die Geschwindigkeiten und Beschleunigungen im Raume schätzen. Wir verwenden hierfür einen Punkt zu jeweils drei verschiedenen Zeitpunkten $\mathbf{s}_{t-1}$, $\mathbf{s}_t$ sowie $\mathbf{s}_{t+1}$. Die Geschwindigkeit $\mathbf{v}_t$ zum Zeitpunkt t schätzen wir aus dem Differenzenquotienten

$$\mathbf{v}_t = \frac{\mathbf{s}_t - \mathbf{s}_{t-1}}{\Delta t}$$

und die Beschleunigung $\mathbf{a}_t$ zum Zeitpunkt t schätzen wir aus dem Differenzenquotenten

$$\mathbf{a}_t = \frac{\mathbf{v}_{t+1} - \mathbf{v}_t}{\Delta t} = \frac{(\mathbf{s}_{t+1} - \mathbf{s}_t) - (\mathbf{s}_t - \mathbf{s}_{t-1})}{\Delta t^2}$$

Der Bequemlichkeit halber können wir Δt gleich 1 setzen, da der zeitliche Abstand von Bild zu Bild einer Folge konstant ist.

Gehen wir davon aus, daß wir die zeitliche Zuordnung zwischen den Punkten s_{t-1} und s_t schon ermittelt haben, dann können wir hiermit nun prüfen, ob der Punkt s_{t+1} zu einer glatten Bahnkurve führt und so als korrespondierender Punkt in Frage kommt. Hierzu muß an die geschätzte Beschleunigung eine Schwelle angelegt werden, die bei zeitlichen Zuordnungen von rekonstruierten Raumpunkten dann nicht überschritten werden darf. Im Gegensatz zu den oben angeführten Betrachtungen im Kontinuierlichen, können dabei dann jedoch nichtglatte Raumkurven entstehen. Trotz dieser Schwierigkeiten bei diskreten Bildfolgen stellt die Beschränkung der Beschleunigungsbeträge jedoch eine wertvolle Einschränkung der Menge der möglichen zeilichen Zuordnungen dar.

Für die Beschränkung der Beschleunigung existieren mehrere Möglichkeiten, die für die oben angesprochenen Bewegungsklassen jeweils mehr oder weniger geeignet sein werden. Die wichtigste hiervon ist die Beschränkung des Betrages des Beschleunigungsvektors. Jenkin, 1984 sowie Jenkin and Tsotsos, 1986 haben dies, neben einer Beschränkung der Geschwindigkeit, bereits erfolgreich eingesetzt.

Es sei noch einmal darauf hingewiesen, daß wir die Glattheit der Bewegung von Objekten im Raume zwar aus physikalischen Eigenschaften unserer natürlichen Umwelt abgeleitet haben, daß die hier vorgeschlagene Beschränkung des Beschleunigungsbetrages aber trotzdem eine einschränkende Annahme über die Eigenschaften unserer beobachteten Szene darstellt.

53.4 Zusammenfassung

In diesem Unterkapitel haben wir das Problem der geometrischen Szenenrekonstruktion beschrieben. Auf die merkmalsbasierte Korrespondenzanalyse als einen trigonometrischen Ansatz wurde dann näher eingegangen, und wir haben eine Übersicht über einschränkende Randbedingungen gegeben, die bei der Korrespondenzanalyse verwendet werden können, um die dort auftretenden Mehrdeutigkeiten zu reduzieren.

Bei den formulierten Einschränkungen, die wir zum einen aus der Kenntnis der Bildentstehung und zum anderen aus der Kenntnis von Eigenschaften der Objekte unserer natürlichen Umwelt erhalten haben, handelt es sich einerseits um strenge physikalische Gesetze, zum anderen jedoch um einschränkende Annahmen über die beobachtete Szene und um Heuristiken. Bei einigen der einschränkenden Annahmen, wie beispielsweise der Beschränkung der Beschleunigung der Objekte einer beobachteten Szene, ist es nicht möglich, sie auf ihre Gültigkeit hin zu überprüfen. Für diese Einschränkungen sowie solche, die sich aus physikalischen Gesetzen ergeben, bietet es sich daher an, sie in prozeduraler Form zu fassen und die Menge der dadurch noch möglichen Zuordnungen von Merkmalen einfach „auszurechnen“. Für die Heuristiken, die sich aus dem Generalitätsprinzip ergeben, kann deren zugrundeliegende Annahme – das Prinzip des allgemeinen Betrachterstandortes – in einer Folgen von Bildern jedoch überprüft werden.

Diese Gesetze und Heuristiken reichen zusammengenommen noch nicht aus, um die auftretenden Mehrdeutigkeiten bei der Szenenrekonstruktion zu beseitigen. Werden alle möglichen Zuordnungen von Merkmalen erzeugt, so bleibt als Problem festzustellen, welche dieser Zuordnungen noch konsistent zueinander sind, um dann aus der Menge konsistenter Zuordnungen die beste Lösung auszuwählen. Dieses wird erst in einem weiteren Schritt der höheren Bilddeutung möglich sein, bei dem versucht wird, der geometrisch rekonstruierten Szene eine „Bedeutung“ beizumessen.

IVC Interactive Visual Computing: Computer Vision

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 9:

Bildverstehen

Universität Hamburg
Fachbereich Informatik

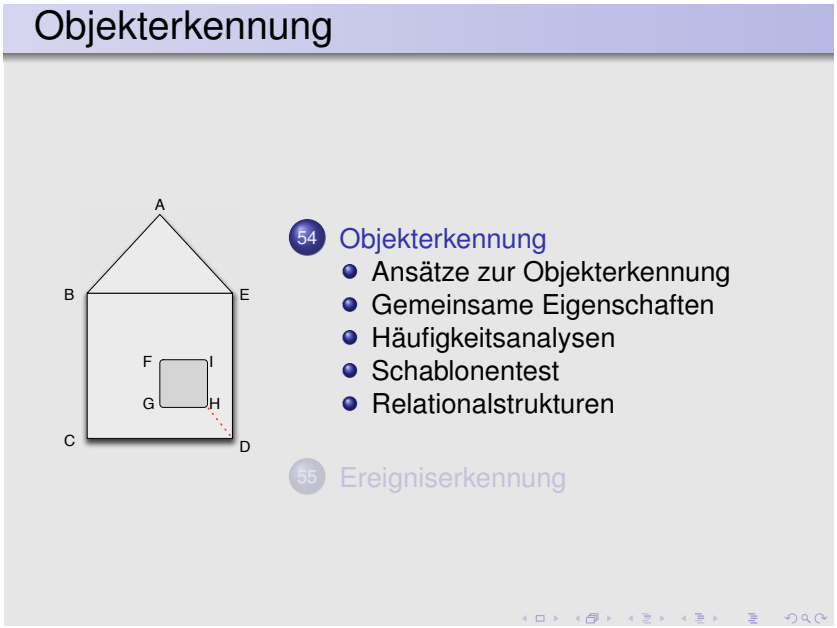
WS 2014/2015

Teil XXI

Bildverstehen

54 Objekterkennung

54.1 Ansätze zur Objekterkennung



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

Überlegen Sie: wie würden Sie einen Stuhl beschreiben, so daß mit Ihrer Beschreibung jeder Stuhl in diesem Bild als Stuhl erkannt wird?

Ansätze zur Objekterkennung

Gemeinsame Eigenschaften: Merkmalsvergleich.

Häufungsgebiete: Abstandsmessung, Ähnlichkeit (siehe Bildverarbeitung 1 und 2).

Prototypen: Schablonentest.

Relationalstrukturen: Graphzuordnung.

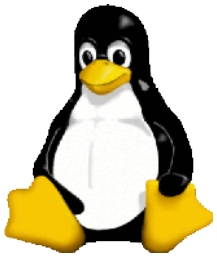
Lineare Maschinen: Perceptrons, Neuronale Netze (siehe Bildverarbeitung 1).

Modellbasiert: 2-D, 3-D Modellvergleich.

54.2 Gemeinsame Eigenschaften

Klassifikation über gemeinsame Merkmale:

Gegeben:
Drei Klassen von Objekten: *Pinguin, Hund, Goldfisch.*

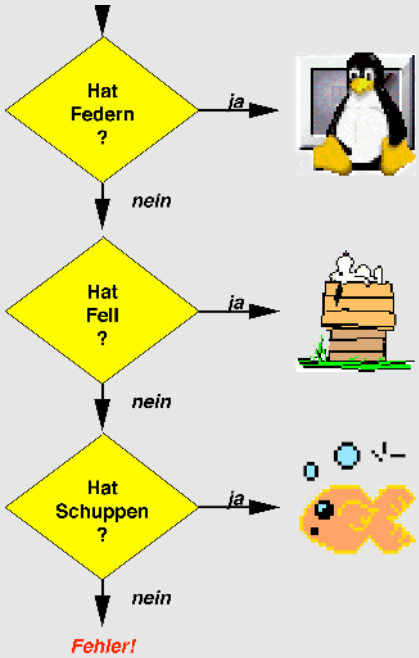


Gesucht:

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

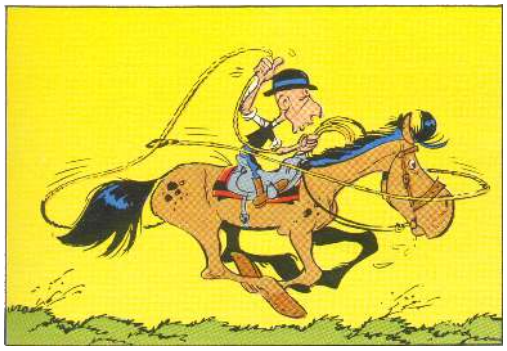
Solche Entscheidungsverfahren sind nur begrenzt anwendbar.

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.

54.3 Häufigkeitsanalysen

Gemeinsame Merkmale: Häufigkeitsanalysen



Gemeinsame Merkmale: Häufigkeitsanalysen

Problem: 127 (Problem:)
Wie können wir erkennen, welchen Sport eine Person ausübt, beispielsweise Rugby-Spieler von Jockeys unterscheiden?

Wir beobachten:

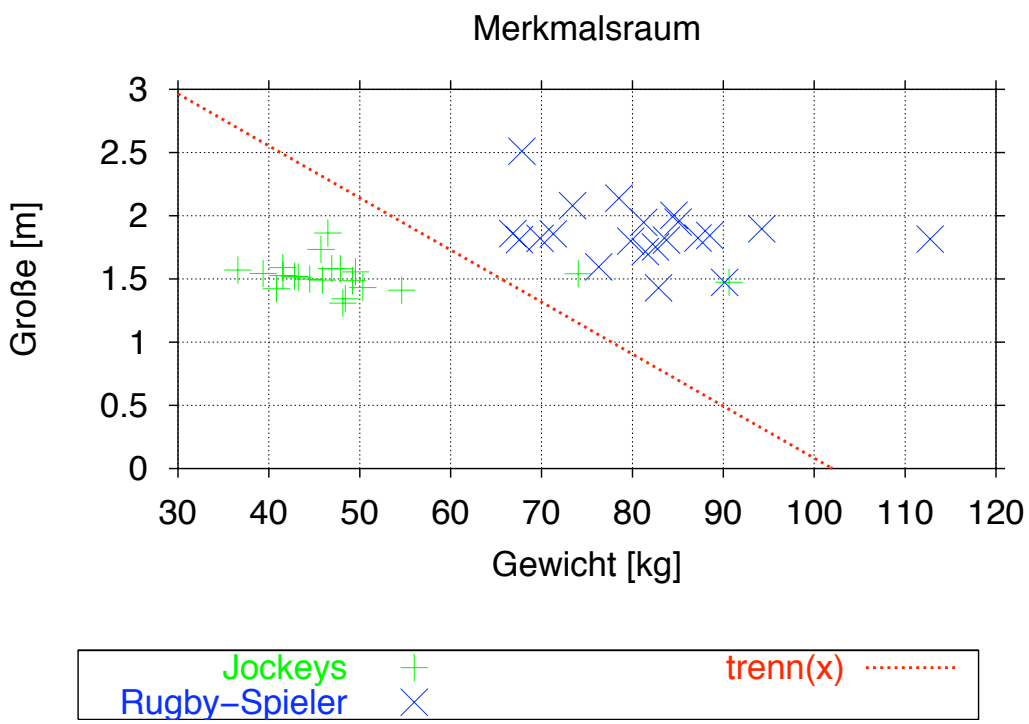
Jockeys sind meistens klein und leicht.

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.*



Wir erheben zwei Stichproben von Jockeys und Rugby-Spielern, wobei wir für jeden Sportler einen Merkmalsvektor $\vec{x} = (x_{Gewicht}, x_{Groesse})$ messen. Diese Merkmalsvektoren sind Elemente eines zweidimensionalen Merkmalsraums, den wir graphisch darstellen können. Wenn unsere Merkmale *Körpergröße, Gewicht* wirklich aussagekräftig sind, dann werden wir feststellen, daß die Elemente der beiden Klassen *Rugby-Spieler, Jockeys* im Merkmalsraum Ballungen bilden und gut voneinander getrennt sind. Wenn wir günstige Merkmale gefunden haben, dann können wir eine Trennlinie(fläche) durch den Merkmalsraum legen und klassifizieren alle Vektoren auf der einen Seite dieser Linie als Jockeys und alle Vektoren auf der anderen Seite als Rugby-Spieler. Diesen Ansatz werden wir noch mathematisch genauer untersuchen.

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?

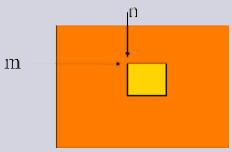
Lernstichprobe: Wie groß soll die Lernstichprobe gewählt werden?

54.4 Schablonentest

Bilderkennung (engl. *image recognition*) ist die Anwendung von Methoden der → *Mustererkennung* zur Erkennung von Objekten in Bildern. Die folgende Folie zeigt ein Beispiel für Bilderkennung. Hier wurden die Abbildungen von Flugzeugen in einer Aufnahme des Nürnberger Flughafens mit der Methode des Schablonenvergleichs gesucht. Von einem Flugzeug-Prototypen wurde eine Abbildung aufgenommen und mit einem Kantenoperator (→ Bildverstehen) gefiltert. Diese so erzeugte Schablone wurde mittels Korrelation mit dem in gleicher Weise gefilterten Flughafenbild verglichen. Punkte hoher Korrelation, d.h. hoher Übereinstimmung zwischen Schablone und Bild, sind in der Abbildung durch einen Kreis markiert, der sich jeweils auf die rechte untere Ecke der Schablone bezieht. Das Beispiel macht deutlich, daß so einfache Verfahren nur dann erfolgreich sind, wenn die Schablone genau paßt, insbesondere müssen auch die Größe und die Orientierung stimmen. Der Schablonenvergleich wird beispielsweise in der Robotik eingesetzt, um Werkstücke zu erkennen, in der Photogrammetrie zur Erkennung von Paßpunkten oder bei der Überwachung von Sicherheitsräumen zur Messung von Geschwindigkeiten (Verkehrsüberwachung, Zielverfolgung).

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

$$E(m,n) = \sqrt{\sum_i \sum_j [g(i,j) - t(i-m, j-n)]^2}$$

$$\text{mit} \quad (i-m, j-n) \in D$$

$$
\begin{aligned}
E(m,n) &= \sqrt{\sum_i \sum_j [g(i,j) - t(i-m, j-n)]^2} \\
\text{mit} \quad & (i-m, j-n) \in D
\end{aligned}
$$

$$
= \sqrt{\sum_i \sum_j [g(i,j)^2 - 2g(i,j)t(i-m,j-n) + t^2(i-m,j-n)]}
$$

$$
\begin{aligned}
E^2(m,n) &= \sum_i \sum_j g(i,j)^2 \\
&\quad - \sum_i \sum_j 2g(i,j)t(i-m,j-n) \\
&\quad + \sum_i \sum_j t^2(i-m,j-n)
\end{aligned}
$$

$$
\begin{aligned}
E^2(m,n) &= \underbrace{\sum_i \sum_j g(i,j)^2}_{\text{Autokorrelation des Bildausschnitts}} \\
&\quad - \underbrace{\sum_i \sum_j 2g(i,j)t(i-m,j-n)}_{\text{Korrelation zwischen Bild und Schablone}} \\
&\quad + \underbrace{\sum_i \sum_j t^2(i-m,j-n)}_{\text{Autokorrelation der Schablone, konstant}} \\
R_{gt}(m,n) &= \sum_i \sum_j g(i,j)t(i-m,j-n) \quad \text{zu maximieren}
\end{aligned}
$$

Wir suchen das Minimum des Fehlers – daher können wir das Quadrieren des Fehlermaßes ignorieren.

Wir vernachlässigen weiterhin den additiven Term der Autokorrelation der Schablone, da dieser konstant ist. Ebenso vernachlässigen wir den den additiven Term der Autokorrelation des Bildausschnitts, unter der Annahme, daß dieser Wert nicht stark variiert. Es bleibt nur noch der negative Term, der die Korrelation zwischen Bild und Schablone beschreibt. Diesen werden wir jetzt halbieren, negieren und dann maximieren:

$$R_{gt}(m,n) = \sum_i \sum_j g(i,j)t(i-m, jn)$$

Anschließend muß noch der Fehler korrigiert werden, der durch die Annahme einer konstanten Autokorelation des Bildes entstehen kann.

Normalisierung

Korrektur für variierende Bildenergie

$$\begin{aligned}
R_{gt}(m,n) &= \sum_i \sum_j g(i,j)t(i-m, j-n) \quad \text{zu maximieren} \\
N_{gt}(m,n) &= \frac{R_{gt}(m,n)}{\sqrt{g(i,j)^2}} \\
N_{gt}(m,n) &<= \underbrace{\sqrt{t^2(i-m, j-n)} = R_{tt}(m,n)}_{\text{Cauchy-Schwarz'sche Ungleichung}} \\
\frac{N_{gt}(m,n)}{R_{tt}(m,n)} &\leqslant 1 \quad \text{Normierte Kreuzkorrelation}
\end{aligned}$$

Normierte Kreuzkorrelation

Definition: 128 (***Normierte Kreuzkorrelation***)
Das Verhältnis $\frac{N_{gt}(m,n)}{R_{tt}(m,n)}$ *heißt normierte Kreuzkorrelation zwischen Bild* g *und Schablone* t.

Satz: 129
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.

✎ **Anmerkung:** Eine andere Sicht auf die Kreuzkorrelation:
Der Schablonentest kann als Klassifikation mittels Prototypen verstanden werden: Die Pixel des Bildausschnitts und die Pixel der Schablone werden hierbei als Komponenten eines Merkmalsvektors betrachtet. Die Korrelation ist dann ein Abstandsmaß für den Merkmalsvektor des Bildausschnitts vom Merkmalsvektor des Prototypen (der Schablone).

- Die „Bildenergie“ $R_{tt}(m, n)$ der Schablone ist dann der Betrag des Merkmalsvektors der Schablone.
- Die „Bildenergie“ $R_{gg}(m, n)$ des Bildausschnitts ist dann der Betrag des Merkmalsvektors des Bildausschnitts.
- Die normierte Kreuzkorrelation ist der Cosinus des Winkels zwischen den Merkmalsvektoren der Schablone und des betrachteten Bildausschnitts.

Die normierte Kreuzkorrelation ist daher unabhängig von der Bildenergie.

Probleme

- Nebenmaxima der Korrelation
- Nur für echte Translation der Schablone geeignet, keine Drehung oder perspektivische Verzerrung zulässig.
- Aufwand proportional zu N*M*I*J
- Rechenzeit sparen durch Faltung im Frequenzraum (Faltungstheorem)

Resultat für das Flughafenbild

Die Kreise markieren lokale Maxima der Korrelation (rechte untere Ecke der Schablone).

Objektverfolgung durch Grauwertkorrelation

Problem: 130
Die Aufgabe: Ein Objekt soll in einer Bildfolge (Film) von Bild zu Bild verfolgt werden.

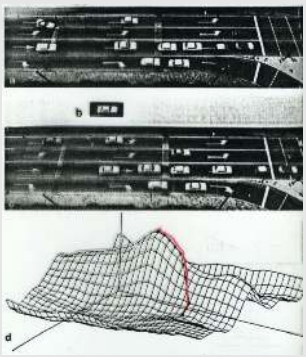
Lösung: 131
Vorgehensweise:

- *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



FIM-Tracker

a-c) Drei Ausschnitte einer Straßenszene, zeitlicher Abstand je eine Sekunde.
b) Ausschnitt als Schablone
d) Verlauf der normierten Kreuzkorrelation und Gradientenaufstieg

Anwendungsbeispiel: FIM-Tracker

Beispiel: 132
FIM-Tracker Der FIM-Tracker war eines der ersten Systeme zur Echtzeitverfolgung von Objekten.

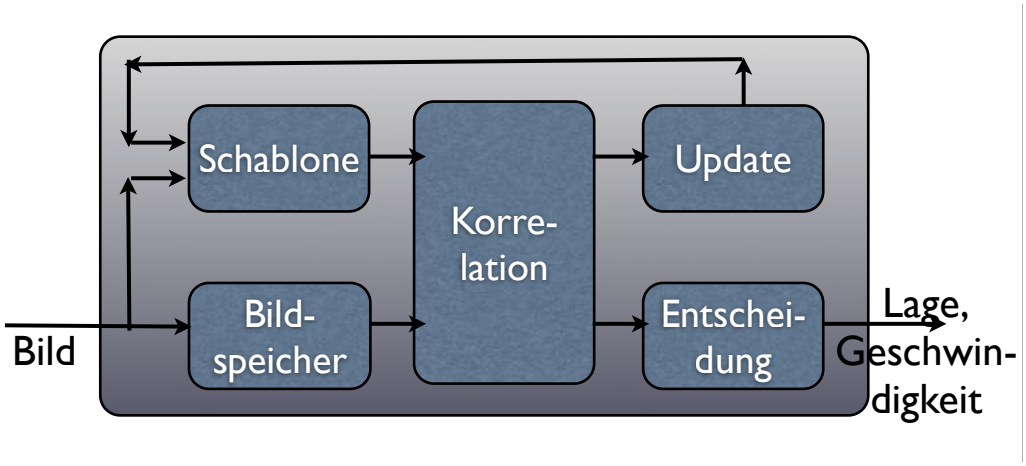
Probleme bei Realwelt-Vorlagen:

- Maßstabsänderung
- Verdeckung
- Drehung
- Echtzeitverarbeitung

Lösung: 133
Laufende Anpassung der Schablone

FIM = Forschungsinstitut Mustererkennung, Ettlingen.

Systemaufbau



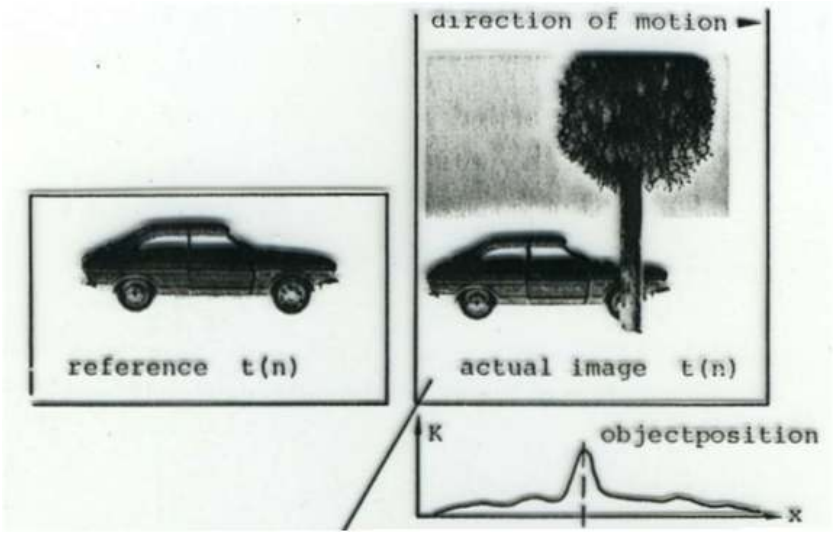
Aktualisierung der Schablone

Die Schablone wird in jedem Schritt ausgetauscht und durch das aktuelle Bild des gefundenen Objektes ersetzt.

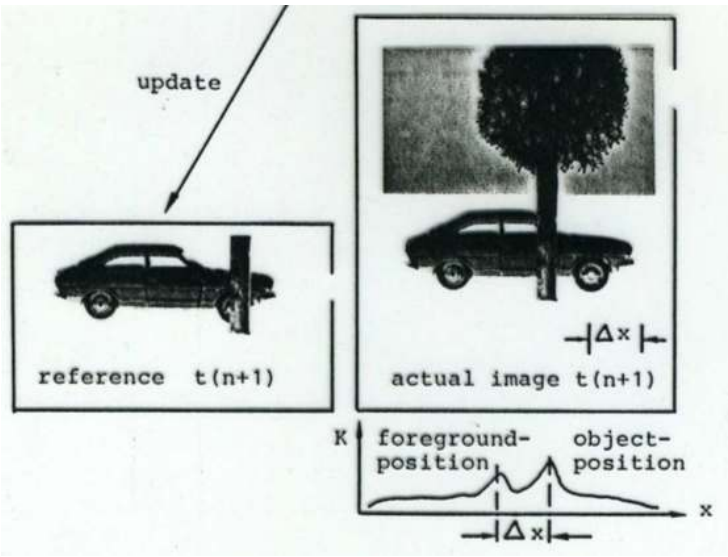
Damit paßt sich die Schablone laufend

- 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.

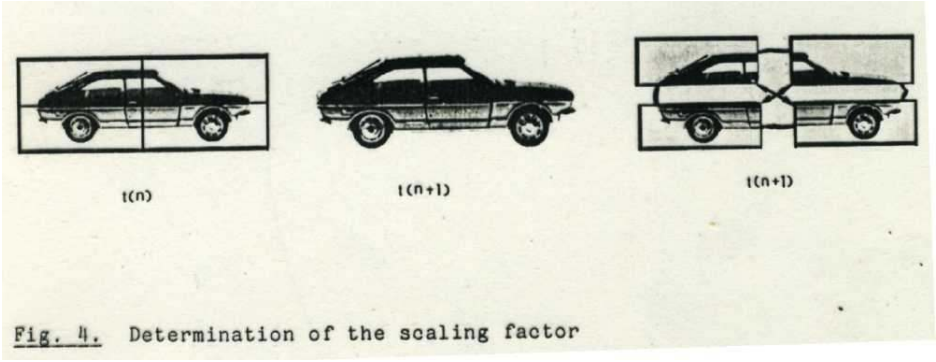
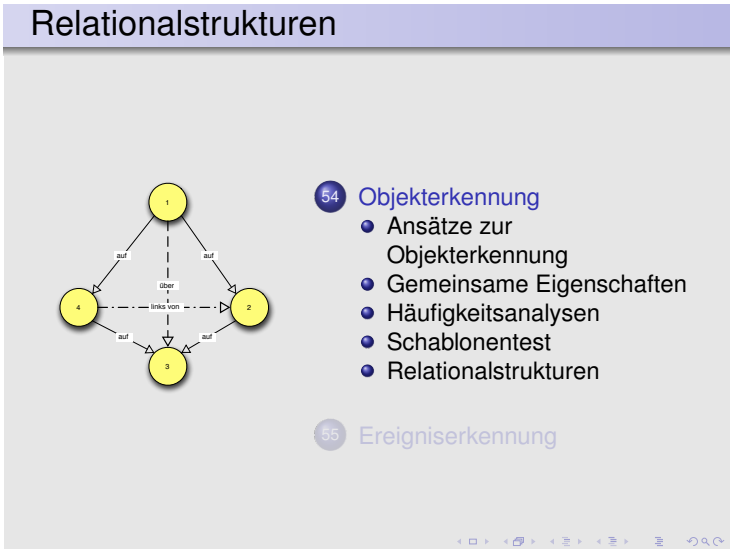


Fig. 4. Determination of the scaling factor

Die Kreuzkorrelation ist im kontinuierlichen Fall eine Faltung des Bildes mit der Schablone. Dieses kann sehr effizient im Frequenzraum berechnet werden, siehe Abschnitt ??.

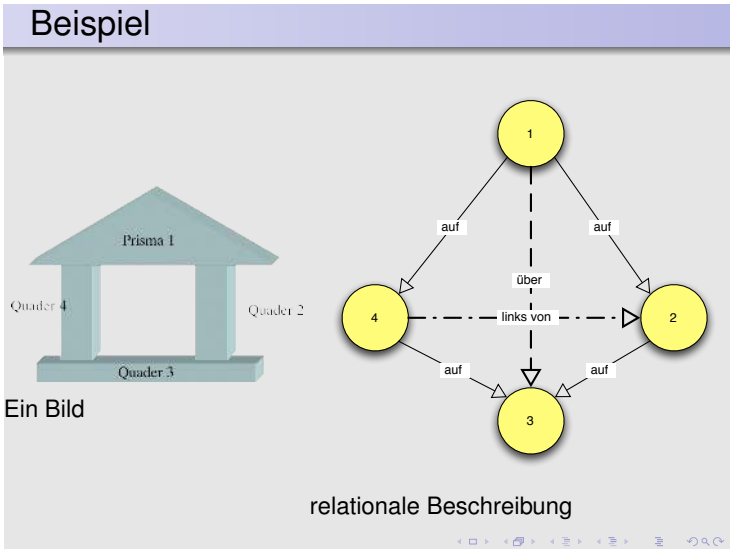
54.5 Relationalstrukturen



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.



Anforderung an eine relationale Beschreibung

Ein Graph, der die Objekte eines Bildes (einer Szene) und ihre Anordnung beschreiben soll, muß folgende Eigenschaften haben :

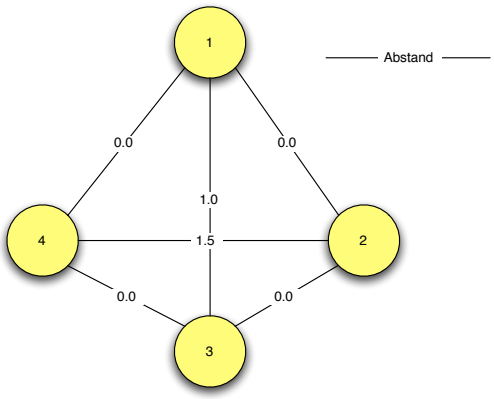
- Die Knoten des Graphen müssen benannt werden können.
- Die Kanten des Graphen müssen benannt werden können.
- Die Knoten und Kanten müssen Attribute beinhalten konnen.
- Die Kanten des Graphen müssen gerichtet sein.

✎ **Anmerkung:** Anforderungen:

- Die Eigenschaft Attribute tragen zu können, ist nicht zwingend notwendig, doch vorteilhaft, um die repräsentierten Objekte genauer zu beschreiben. Das gleiche gilt auch für die Kanten.
- Die Kanten des Graphen müssen nicht unbedingt gerichtet sein. Eine Richtung ist dann notwendig, wenn die repräsentierten Beziehungen gerichtet sind.

Beispiel: Abstandsrelation

Eine alternative relationale Repräsentation des Blockbildes: Abstandsrelation.



Definition: Bildbeschreibungs-Graph

Definition: 134 (***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

$\alpha : N \to 2^A$: eine Funktion, die einem Knoten eine Menge von Attributen zuordnet.

$\beta : E \to 2^A$: 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: 135 (*Relationenqebilde*)
Ein Relationenqebilde (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 konnen.*

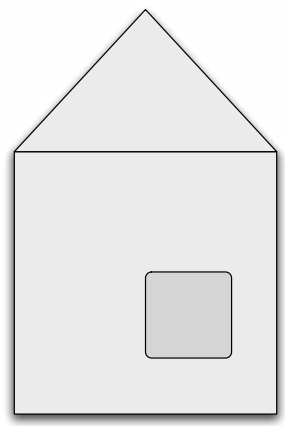
R**:** $R = (R_1, \ldots, R_n)$ *ist ein n-Tupel von Relationen über* C*. Jede Relation* R_i *ist eine Menge von* k_i*-Tupeln über der Trägermenge* C*. Die Relation* R_i *hat dann also* k_i *Stellen.*

R**:** R *heißt Relationalstruktur über (auf)* C*.*

✎ **Anmerkung:** Eine Relation kann durch Aufzählung der Relationstupel oder durch eine Konstruktionsbeschreibung bestimmt werden.

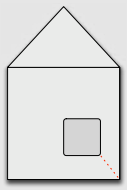
Ein Beispiel

Beispiel: 136 (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:

  $(< KantenID > \quad < Punkt > \quad < Punkt > \quad < Laenge >)$

- Die Zusammenhangsrelation:

  $(< WinkelID > \quad < Kante > \quad < Kante > \quad < Punkt > \quad < Winkel >)$

- Relative Position:

  $(< Pos.ID > \quad < virt.Kante > \quad < Winkel > \quad < Winkel >)$

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.

Die Trägermenge C ist die Vereinigungsmenge der Mengen der

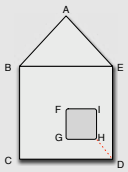
- Kantenbezeichner,
- Punktbezeichner,
- Winkelbezeichner,
- der auf das Intervall von 0 bis 1 normierten Längenangaben
- und der Winkelangaben im Bogenmaß oder Grad.

Das Relationentupel R besteht aus der Kantenrelation R_1, der Winkelrelation R_2 und der Positionsrelation R_3.

Das Relationengebilde

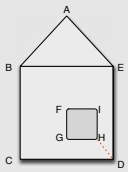
$$
\begin{aligned}
C &= Kantenbezeichner \\
&\cup Punktbezeichner \\
&\cup Winkelbezeichner \\
&\cup (0,1) \cup (-180°, 180°) \\
R &= (Kantenrelation, Winkelrelation, Positionsrelation)
\end{aligned}
$$

Die Kantenrelation



$$
\left\{
\begin{array}{cccc}
(Kante1 & A & B & 0.7) \\
(Kante2 & B & C & 1.0) \\
(Kante3 & C & D & 1.0) \\
(Kante4 & D & E & 1.0) \\
(Kante5 & E & A & 0.7) \\
(Kante6 & E & B & 1.0) \\
(Kante7 & F & 6 & 0.4) \\
(Kante8 & G & H & 0.4) \\
(Kante9 & H & I & 0.4) \\
(Kante10 & I & F & 0.4) \\
(VKante1 & H & D & 0.37)
\end{array}
\right\}
$$

Die Winkelrelation



$$\left\{\begin{array}{lllll} (\mathit{Winkel}1 & \mathit{Kante}1 & \mathit{Kante}2 & B & 135^\circ) \\ & & \vdots & & \\ (\mathit{Winkel}13 & \mathit{Kante}7 & \mathit{Kante}10 & F & 90^\circ) \\ (\mathit{Winkel}14 & \mathit{VKante}1 & \mathit{Kante}3 & D & 60^\circ) \\ (\mathit{Winkel}15 & \mathit{VKante}1 & \mathit{Kante}9 & H & 120^\circ) \end{array}\right\}$$

$$\left\{ (\mathit{Position}1 \quad \mathit{VKante}1 \quad \mathit{Winkel}14 \quad \mathit{Winkel}15 \quad) \right\}$$

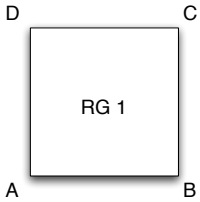
Vergleich von Objektmodell und Bild

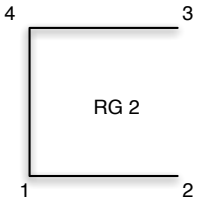
Gegeben seien zwei Relationengebilde RG^1 und RG^2, die ein Objektmodell und Bildstrukturen beschreiben.

$$\begin{aligned} RG^1 &= (C^1, R^1) \\ RG^2 &= (C^2, R^2) \end{aligned}$$

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	$1.0)$				
$(Winkel1$	$Kante1$	$Kante2$	$-90^\circ)$	$Angle1$	$Edge3$	$Edge2$	-90°
$(Winkel2$	$Kante2$	$Kante3$	$-90^\circ)$	$Angle2$	$Edge2$	$Edge1$	-90°
$(Winkel3$	$Kante3$	$Kante4$	$-90^\circ)$				
$(Winkel4$	$Kante4$	$Kante1$	$-90^\circ)$				

Abbildung von Relationengebilden

Definition: 137 (***Abbildung***)
Eine Abbildung ist eine Menge von Paaren $\{(r_1 r_2)\} \subseteq RG_1 \times RG_2$.

Jedes Paar beschreibt eine Zuordnunq 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.

Daraus folgt sofort, daB eine Abbildung zwischen Relationengebilden deren Relationalstruktur nicht homolog ist, d.h. deren Relationalstrukturen nicht die gleiche Stellenverteilung haben, nicht sinnvoll ist.

Damit kann man eine Abbildung übrigens als Relationalstruktur über $RG_1 \times RG_2$ verstehen.

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:

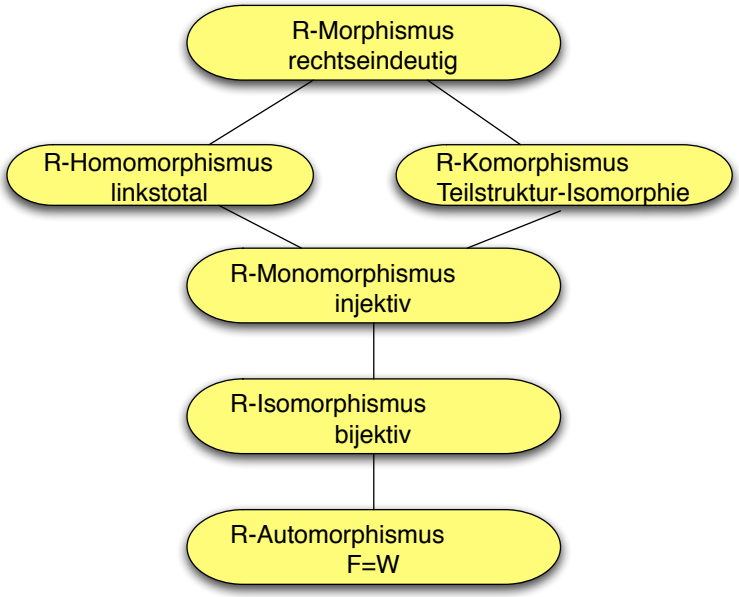
1. Für die Trägermengen:

$$((m_1^1, m_2^1, m_3^1, \dots)(m_1^2, m_2^2, m_3^2, \dots) \quad \Rightarrow \quad \forall_j : (m_j^1, m_j^2) \in f_0$$

2. Jede Abbildung f_i ist rechtseindeutig (und ist damit eine Funktion).

Definition: 138 (*R-Morphismus*)
Eine Abbildung, die diesen Eigenschaften genügt, heißt R-Morphismus

Taxonomie von R-Morphismen



Konstruktion von R-Morphismen

Die einfachste Konstruktionsvorschrift lautet:

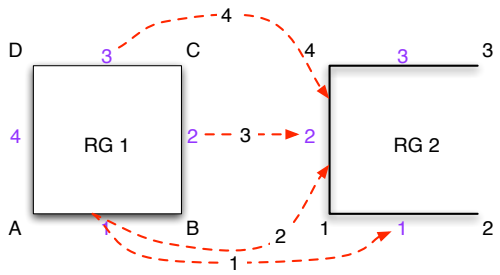
- Betrachte alle möglichen Zuordnungen
- und streiche alle die weg, die nicht zu einem R-Morphismus gehören.

Problem:
Wann gehört eine Zuordnung nicht zu einem R-Morphismus?

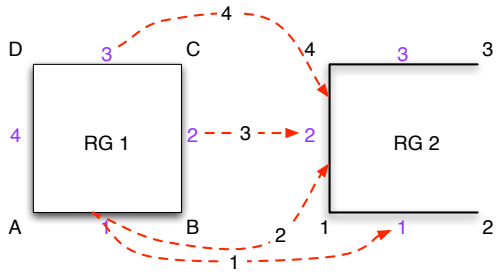
Nicht-kompatible Zuordnungen

- 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 Zuordnunqen paarweise kompatibel sind.

Problem:
Wann sind zwei Zuordnungen nicht kompatibel?



$$\begin{array}{llll} (Kante1 & A & B & 1.0) \\ (Kante2 & B & C & 1.0) \\ (Kante3 & C & D & 1.0) \\ (Kante4 & D & A & 1.0) \end{array} \quad \Big| \quad \begin{array}{llll} (Edge1 & 2 & 1 & 1.0) \\ (Edge2 & 1 & 4 & 1.0) \\ (Edge3 & 4 & 3 & 1.0) \end{array}$$



$$\begin{aligned} z_1 &= \{(Kante1 \quad A \quad B \quad 1.0) \quad (Edge1 \quad 2 \quad 1 \quad 1.0)\} \\ z_2 &= \{(Kante1 \quad A \quad B \quad 1.0) \quad (Edge2 \quad 1 \quad 4 \quad 1.0)\} \\ z_3 &= \{(Kante2 \quad B \quad C \quad 1.0) \quad (Edge2 \quad 1 \quad 4 \quad 1.0)\} \\ z_4 &= \{(Kante3 \quad C \quad D \quad 1.0) \quad (Edge2 \quad 1 \quad 4 \quad 1.0)\} \end{aligned}$$

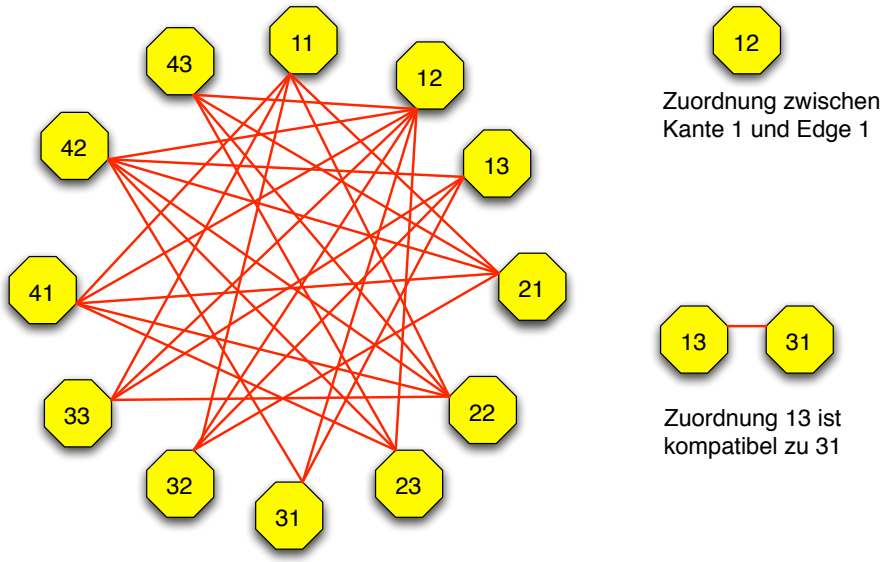
z_1, z_2: nicht kompatibel,*nicht linkseindeutig*

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



R-Morphismen und Cliquen in einem Graph

Satz: 139
In einem R-Morphismus sind alle Zuordnungen paarweise kompatibel.

Der Umkehrschluß ist ebenfalls richtig :

Satz: 140
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)*.

Problem: 141
<2->[]
Wie findet man alle Cliquen in einem (Kompatibilitäts-)Graphen?

Lösung: 142
<3->[Cliquensuche:] Rekursives Zerlegen der Menge der Zuordnungen

✎ **Anmerkung:** Eigentlich sind nur die maximialen, maximal zusammenhängenden Teilgraphen interessant, da sie die maximalen R-Morphismen reprasentieren.

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.

IVC Interactive Visual Computing: Computer Vision

Leonie Dreschler-Fischer

Prüfungsunterlagen Teil 10:

Übungen: Werkzeuge, Handbücher und Beispiele

Universität Hamburg
Fachbereich Informatik

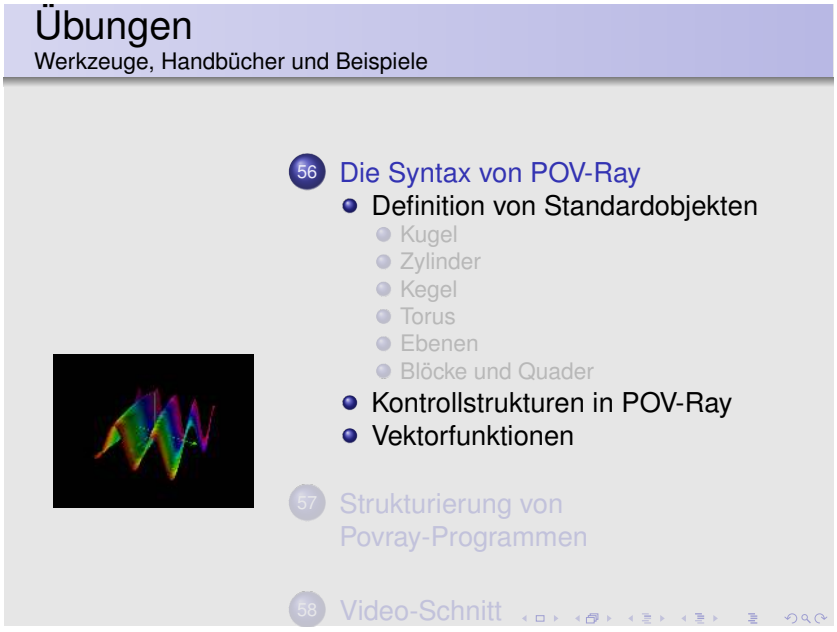
WS 2014/2015

Teil XXII

Übungen: Werkzeuge, Handbücher und Beispiele

55 Die Syntax von POV-Ray

55.1 Definition von Standardobjekten



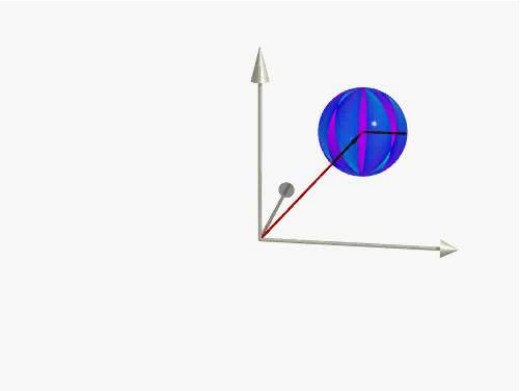
Für die Übungsaufgabe und Anforderungen an den zu erstellenden Film siehe: Unterabschnitt 3.1, Seite 40.

55.1.1 Kugel

Definition einer Kugel

Die Kugel wird beschrieben durch:

- Vektor zum Mittelpunkt
- Radius
- Eigenschaften der Oberfläche



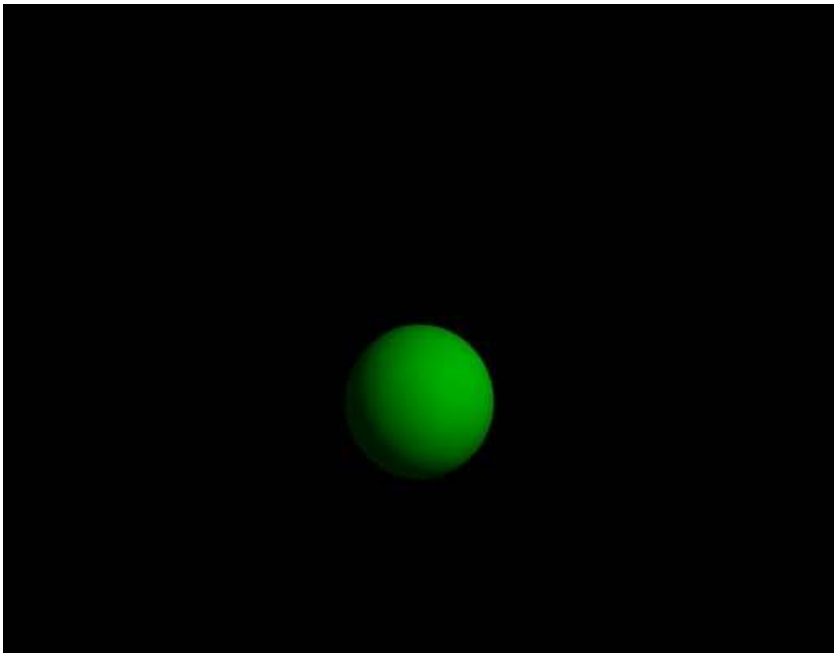
Eine Kugel im Ursprung

Beispiel: 143 (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

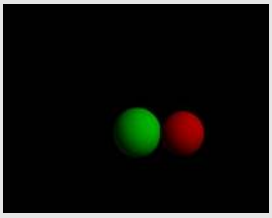
Beisp. 143



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: 144
Der Radius der zweiten Kugel „radiusB“ wird abhängig vom Radius der Kugel „A“ definiert.

Variable und arithmetische Ausdrücke

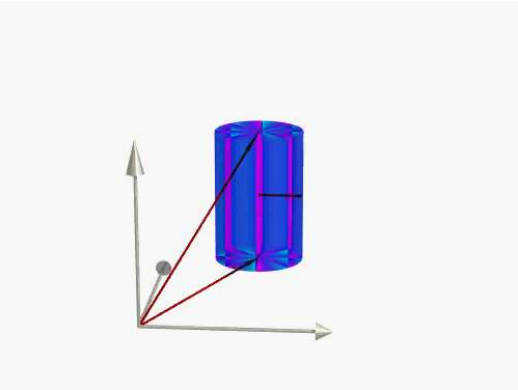
```
#declare radiusA=1;
#declare radiusB=radiusA*0.9;
#declare centerA=0;
#declare centerB=centerA+x*(radiusA+radiusB);

sphere {
  centerA, // center
  radiusA  // radius
  pigment { Green }
}
sphere {
  centerB, // center
  radiusB  // radius
  pigment { Red }
}
```

55.1.2 Zylinder

Definition eines Zylinders

Ein Zylinder wird beschrieben durch:

- Vektor zum Mittelpunkt des Deckels
- Vektor zum Mittelpunkt des Bodens
- Radius
- Eigenschaften der Oberfläche



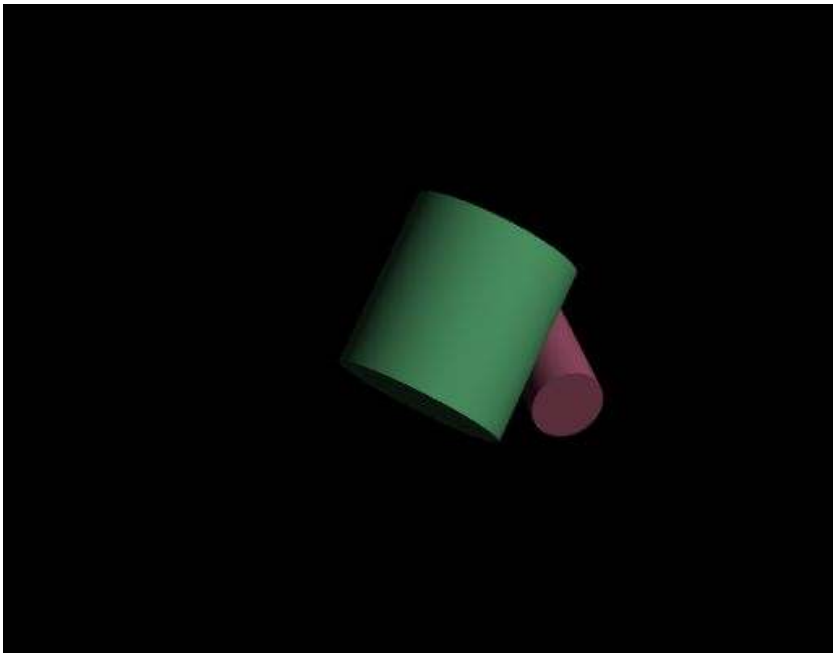
Zwei Zylinder in POV-Ray

Beispiel: 145
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. 145

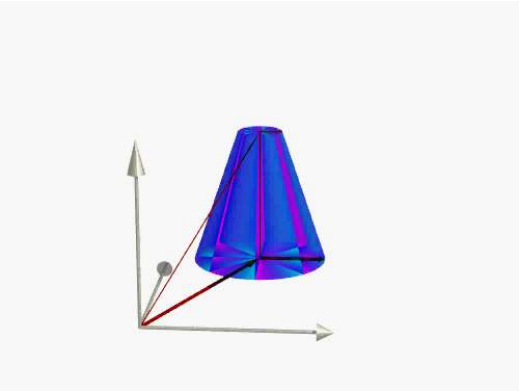


55.1.3 Kegel

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



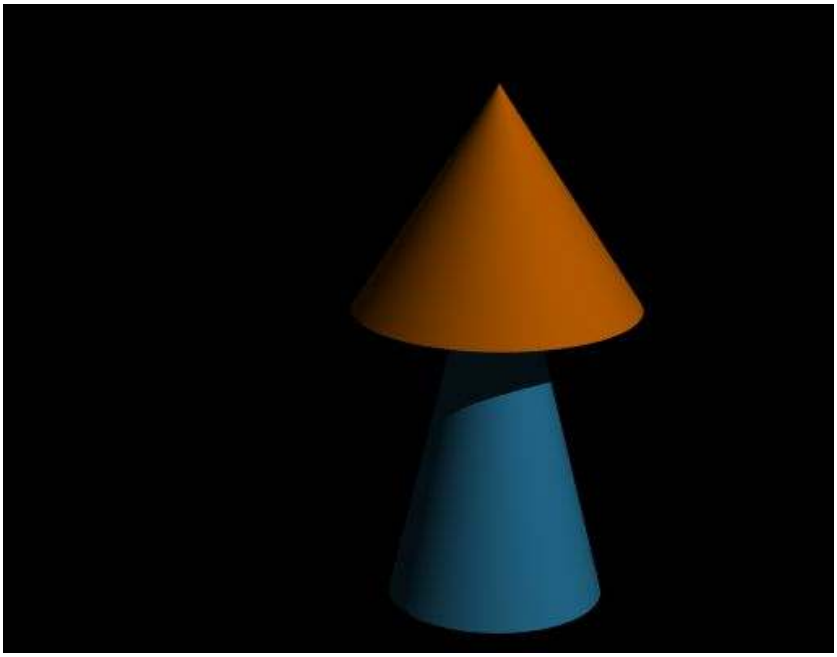
Zwei Kegel in POV-Ray

Beispiel: 146
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. 146

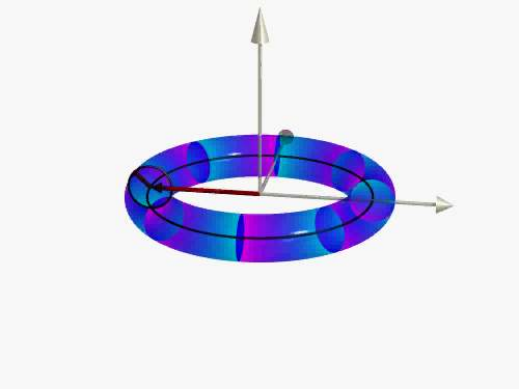


55.1.4 Torus

Definition eines Torus

Ein Torus wird beschrieben durch:

- Hauptradius (major radius)
- Nebenradius (minor radius)
- Lage (nicht in POV-Ray)
- Eigenschaften der Oberfläche



✎ **Anmerkung:** In POV-Ray liegt ein Torus in der xz-Ebene, der Mittelpunkt ist im Ursprung.

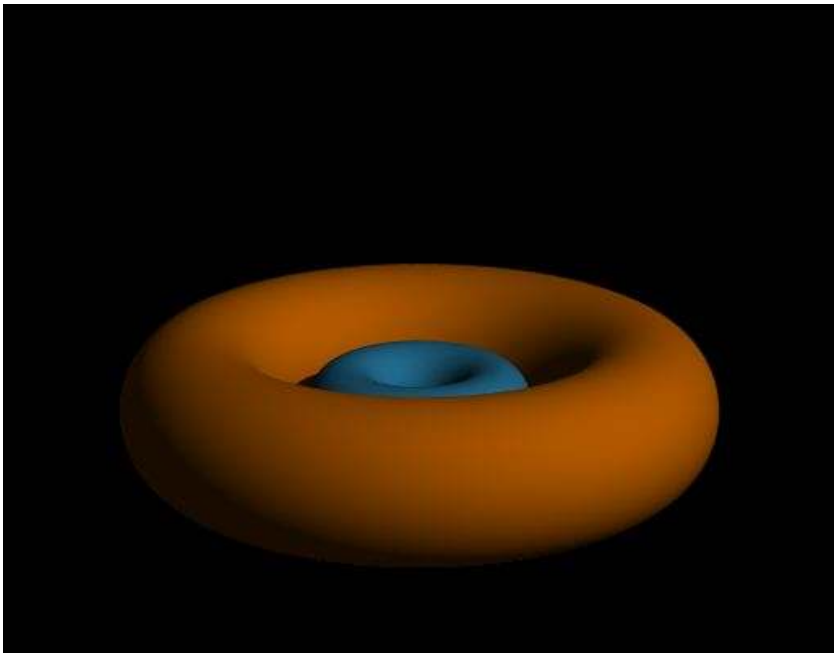
Ein großer und ein kleiner Torus

Beispiel: 147
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. 147

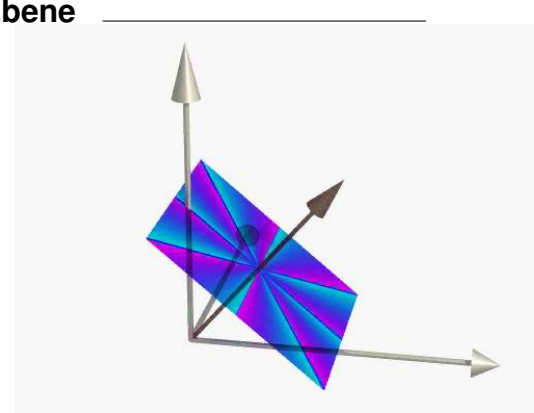


55.1.5 Ebenen

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)



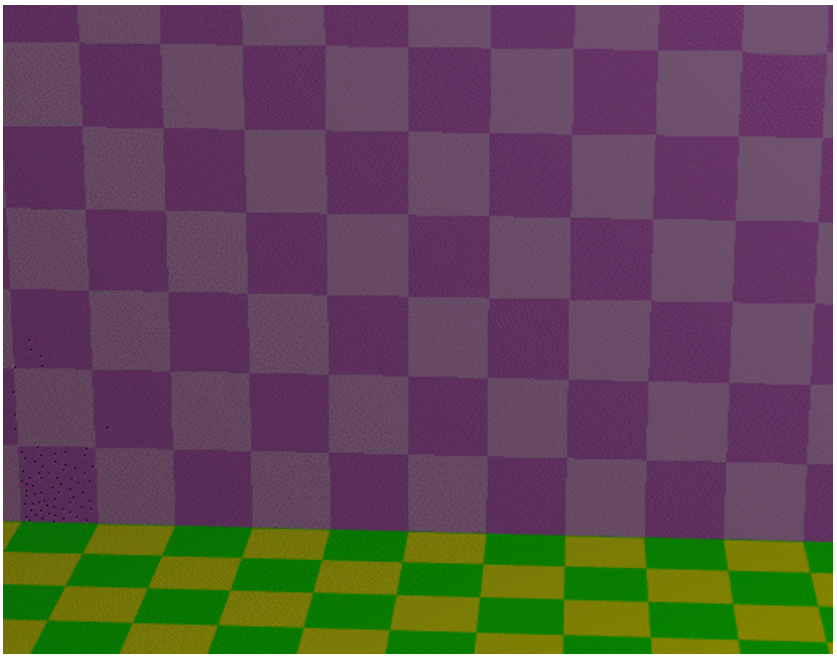
Zwei Ebenen senkrecht zueinander

Beispiel: 148
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. 148

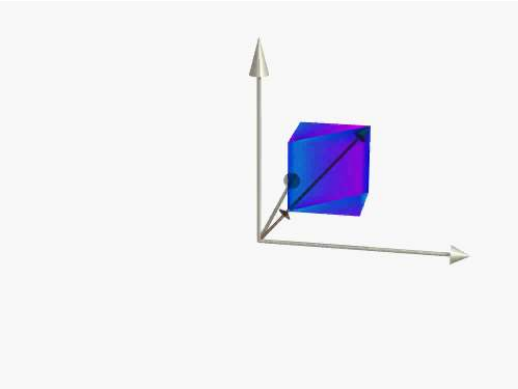


55.1.6 Blöcke und Quader

Definition eines Quaders

Beispiel: 149
Definition eines animierten Quaders (box)

- Ein Quader wird beschrieben durch zwei gegenüberliegende Eckpunkte:
- Die Seitenflächen des Quaders sind parallel zu den Grundebenen des Koordinatensystems.



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}}
}
```

✎ **Anmerkung:** clock-Funktion In diesem Beispiel variierieren die Farbe der Lichtquelle sowie die Position der zweiten Ecke des Quaders in Abhängigkeit von der Animationsuhr clock. Der Wert der clock-Funktion wird für die Dauer einer Animation linear zwischen 0 und 1.0 interpoliert.

55.2 Kontrollstrukturen in POV-Ray

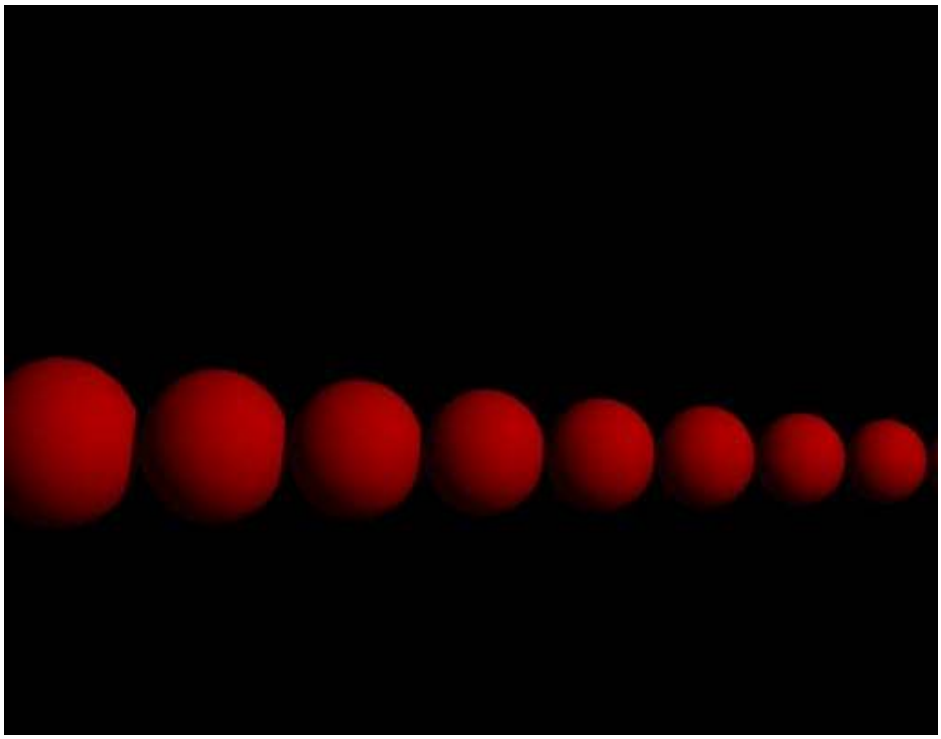
55.2.1 Wiederholung: While-Schleife

Wiederholung: While-Schleife

Beispiel: 150
Eine lange Reihe von Kugeln: In jedem Iterarationschritt wird die Kugel um den Faktor 0.9 kleiner und nach rechts verschoben.

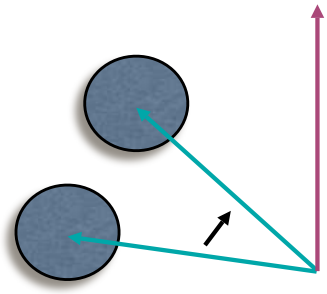
```
#declare Radius=1.0;
#declare Center=<-5,0,0>;
#declare smaller=0.9;
#while (Radius>0.3)
  sphere {Center, Radius
          pigment { Red }}
  #declare Center=Center
                  +(1+smaller)*<Radius,0,0>;
  #declare Radius=Radius*smaller;
#end
```

Das erzeugte Bild zu Beispiel 150



Anordnung der Kugeln im Kreis

Beispiel: 151
Um die Kugeln im Kreis anzuordnen, wird der Radiusvektor des Kreises bei jedem Iterationsschritt um einen festen Winkel weitergedreht.

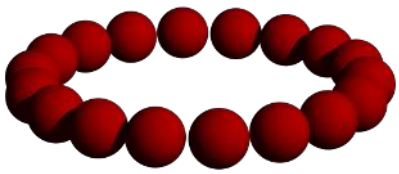


Anordnung im Kreis durch Rotation

```
#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. 151



55.3 Vektorfunktionen

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

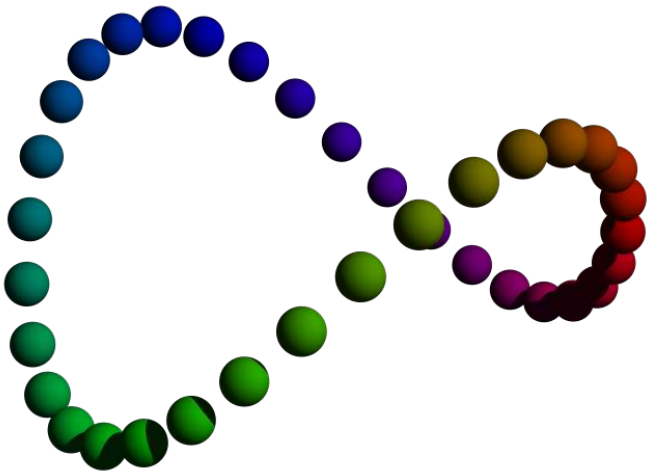
Beispiel: 152 (Rotiere den Farbvektor im Farbkreis)

```
#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
```

Dieses Programm erzeugt eine Kette von Kugeln, die im Kreis angeordnet sind. Die Farbe der Kugel wird in Abhängigkeit von der Position berechnet: Die erste Kugel hat das Pigment der Farbe <1,0,0>, die Farbe der weiteren Kugel entsteht durch Rotation des Farbvektors um den Diagonalenvektor <1,1,1>.

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
```

Die Kugeln führen eine harmonische Schwingung durch, wobei die Phasenlage in Abhängigkeit von der „clock“ berechnet wird.

56 Strukturierung von Povray-Programmen

Modularisierung des Programms

- Makros, Funktionen
- Include-Files:
  - für vordefinierte Objekte,
  - für gemeinsam benutzte Parameter
  - und als Algorithmenbibliothek.

56.1 Bibliotheken

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

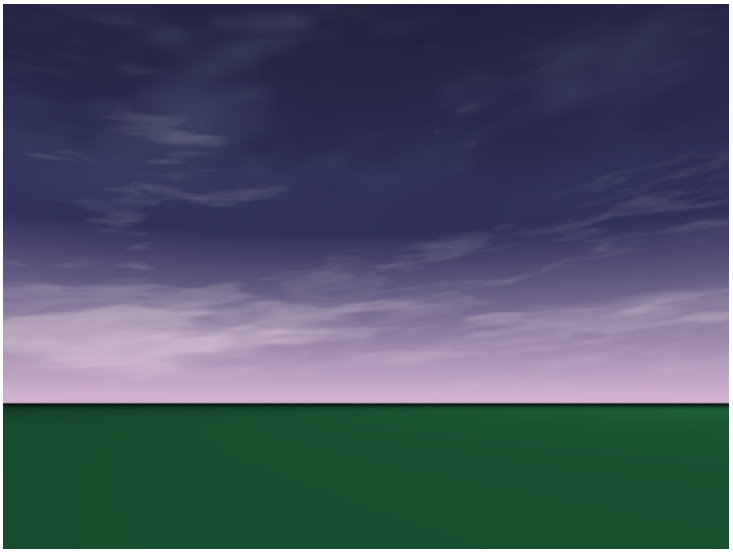
skies.inc: Wolkentexturen

stars.inc: Sternenhimmel

woods.inc: Holzmaserungen

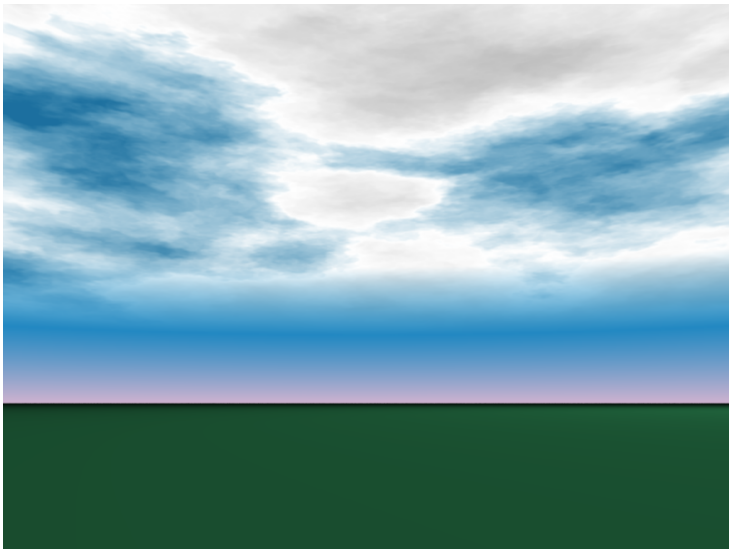
Beispiel: 153 (Simulierte Cirrostratus-Wolken)

```
#include{skies.inc}
sky_sphere{S_cloud1}
```



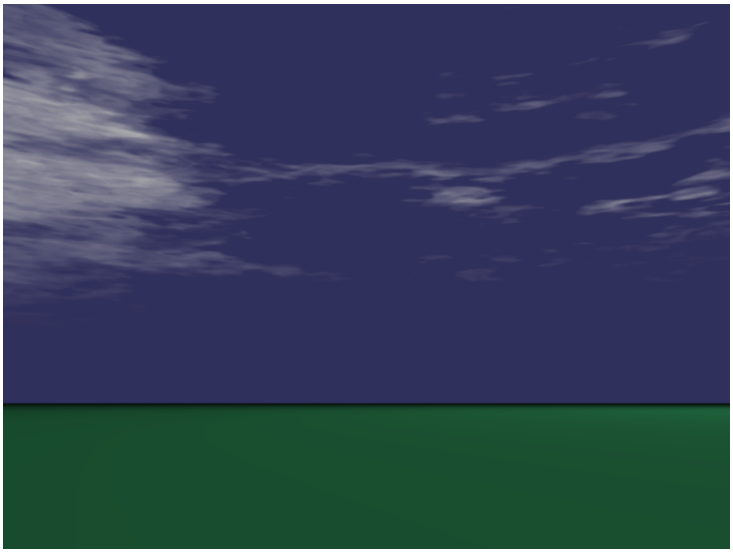
Beispiel: 154 (Blauer Himmel mit Schäfchenwolken)

```
#include{skies.inc}
sky_sphere{S_cloud2}
```



Beispiel: 155 (Geringe Cirrusbewölkung)

```
#include{skies.inc}
sky_sphere{S_cloud3}
```

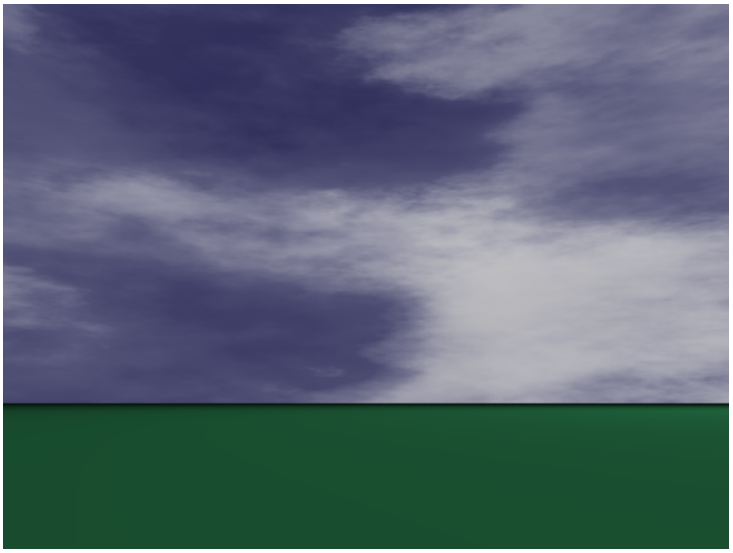


Beispiel: 156 (Regenwolken)

```
#include{skies.inc}
sky_sphere{S_cloud4}
```

Beispiel: 157 (Dichte Cirruswolken)

```
#include{skies.inc}
sky_sphere{S_cloud5}
```



Beispiel: 158 (Ein flammendes Inferno)

```
#include{skies.inc}
sky_sphere{pigment{Apocalypse}}
```

Himmelsobjekte

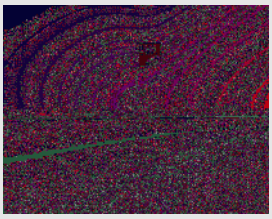
Himmelsobjekte haben besondere Eigenschaften:

- 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}
```

56.1.1 Definition von Objektmodellen

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} }
```

Diese Beispiel zeigt, wie eine Szene mit schattenlosen Lichtquellen aufgehellt werden kann, so daß keine störenden Schlagschatten entstehen. Außerdem sehen wir, wie durch Variablendefinitionen die relativen Positionen von Objektkomponenten ausgerechnet werden können.

Modellierung der Flagge

"UHH_ROT.png"

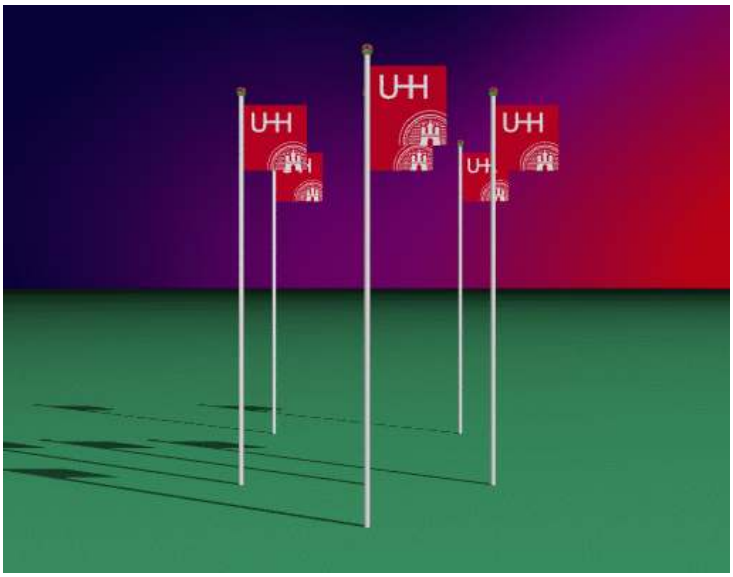
- Das Flaggentuch wird als Polygon modelliert.
- Das Bild auf der Flagge wird als **image_map**-Textur auf das Polygon projiziert.
- Das Bild hat in der X, Y-Ebene die Eckpunkte $(0, 0)$ und $(1, 1)$.

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. 159)



Erzeugen mehrerer Instanzen des Modells

Beispiel: 159 (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

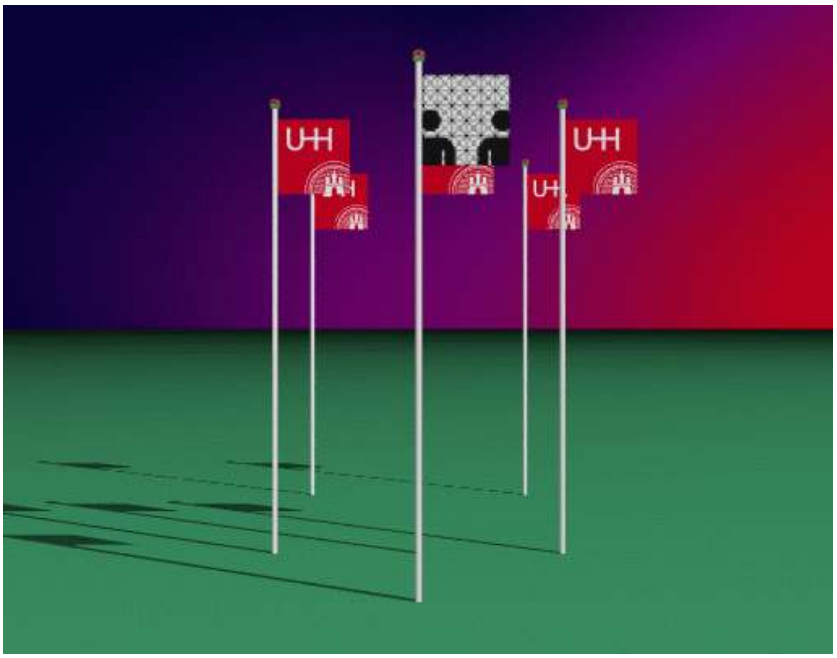
Beispiel: 160 („fbilogo“:)
Eine zweite Flagge mit anderem Muster:

"fbilogo.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 "fbilogo.png"}
                   scale flagsize}
                translate 0.8*topofflagpole+x*radiusofpole}
}
```

Flaggenwald mit Departmentflagge

Beisp. 160

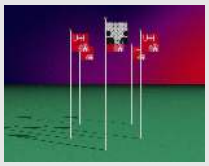


Flaggenwald mit Departmentflagge

```
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}
```

56.2 Macros und Funktionen

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: 161 (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: 162 (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 ⟶ 3*xx+xx ⟶ 3*3+3 ⟶ 12

Definition von Funktionen

Funktionen werden mittels des Schlüsselworts *function* definiert:

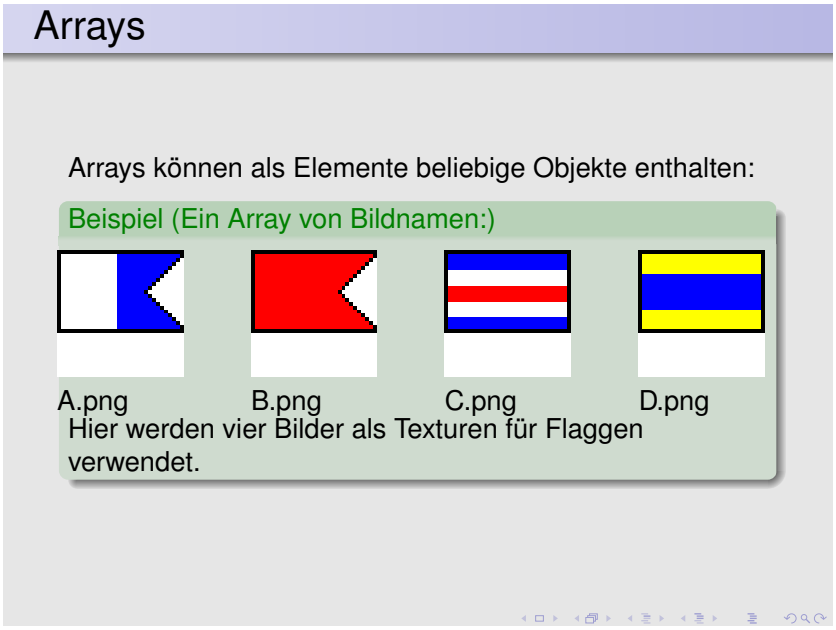
Beispiel: 163 (Der Mittelwert zweier Zahlen)
Z.B.:

```
#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.

56.3 Datenstrukturen

56.3.1 Arrays

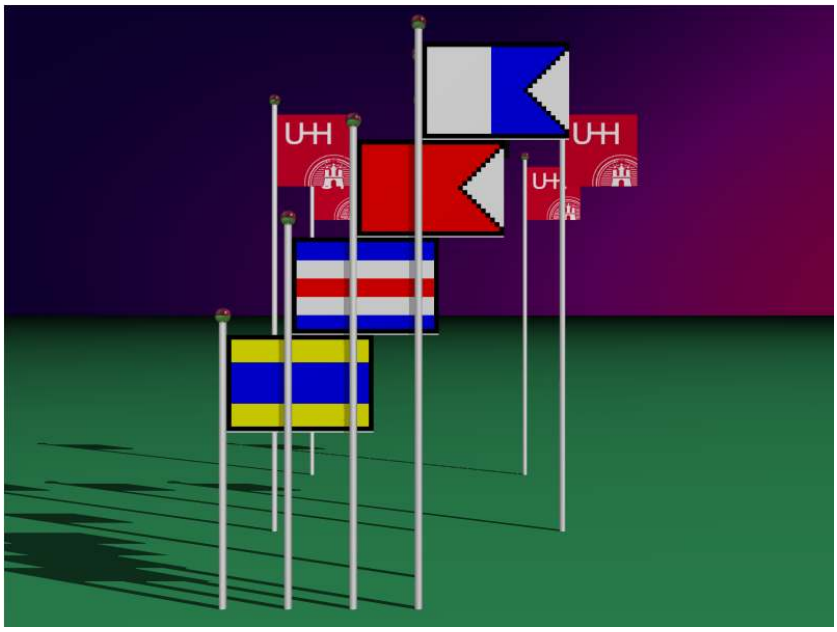


Eine Schleife über die Elemente eines Arrays

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. ??)



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: 164 (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
```

Schneeflocken in zufälliger Lage und Größe

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.

56.4 Fallunterscheidungen

Fallunterscheidungen

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: 165 (#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 if the
                      // identifier "User_Thing" was
                      // 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
#end
```

Ist ein Array-Element definiert?

Beispiel: 166 (#ifdef:)
*Mit #**ifdef** und #**ifndef** kann geprüft werden, ob ein Array-Element initialisiert ist.*

```
#declare MyArray = array[10];
#ifdef(MyArray[0])
  #debug "first element is assigned\n"
#else
  #debug "first element is not assigned\n"
#end
```

#debug druckt 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 ... [#break]
```

Zwei Werte werden als gleich gewertet, wenn der Unterschied kleiner als 1e-10 ist.

Die *switch clauses* werden der Reihe nach geprüft. Wenn eine Klausel erfüllt ist, werden die zugehörigen *tokens* berechnet. Dann wird geprüft, ob weitere Klauseln erfüllt sind, sofern keine Break-directive folgt.

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 <= HIGH_1
#break //Third case ends
#range (LOW_2,HIGH_2)
// This section is parsed if LOW_2 <= VALUE<=HIGH_2
#break //Fourth case ends
#else // This section is parsed if no other case or
    // 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

56.4.1 Synchronisation einer periodischen Bewegung mit der Musik

Das folgende Beispiel zeigt, wie eine periodische Bewegung mit der Musik synchronisiert werden kann:

Beispiel: 167
Im Takt zur Musik: Es werden zur Begleitmusik drei Angaben benötigt:

Das Tempo: *Wieviele Taktschläge (beats per minute, bpm) sind es pro Minute?*

Taktart: *Wieviele Taktschläge pro Takt?*

Auftakt: *Wieviele Taktschläge als Auftakt vor dem ersten Takt?*

Das Macro initSceneClock initialisiert dann die globale Variable taktnummer, so daß wir die Animation auf die aktuelle Phase der Bewegung abstimmen können.

```
#version unofficial MegaPov 1.21;

#declare FpS = 12; // frames per second

// Indizes in der Datentabelle:
// der Parameter für das Macro "initSceneClock"
#declare nordseewellen = 0;
#declare leiserieselt = 1;
#declare roundroundround = 2;
#declare winterwonderland = 3;
#declare letitsnow = 4;
#declare wolgawolga = 5;
#declare finale = 6;

#declare anzahlTakte = array[finale+1] {
        48.0,/*nordseewellen = 0;*/
        23.75, /*leiserieselt = 1;*/
        0.0, /*roundroundround = 2;*/
        31.75,/* winterwonderland = 3;*/
        0.0,  /* letitsnow = 4;*/
        0.0, /*wolgawolga = 5; */
        0.0 /* finale = 6;*/
```

```
}
#declare auftakte = array[finale+1] {
        1.0,/*nordseewellen = 0;*/
        0.0, /*leiserieselt = 1;*/
        0.0, /*roundroundround = 2;*/
        1.0,/* winterwonderland = 3;*/
        0.0,  /* letitsnow = 4;*/
        0.0, /*wolgawolga = 5; */
        0.0 /* finale = 6;*/
}

#declare bpms = array[finale+1] { // Taktschlaege pro Minute
        180.0,/*nordseewellen = 0;*/
        120.0, /*leiserieselt = 1;*/
        0.0, /*roundroundround = 2;*/
        124.0,/* winterwonderland = 3;*/
        0.0,  /* letitsnow = 4;*/
        0.0, /*wolgawolga = 5; */
        0.0 /* finale = 6;*/
}

#declare viertelproTakt= array[finale+1] {
        3,/*nordseewellen = 0;*/
        2, /*leiserieselt = 1;*/
        0.0, /*roundroundround = 2;*/ // ab hier noch undefiniert
        4.0,/* winterwonderland = 3;*/
        0.0,  /* letitsnow = 4;*/
        0.0, /*wolgawolga = 5; */
        0.0 /* finale = 6;*/
}

#macro taktDauer (dieSzene)
        //  die Dauer eines Taktes in  Sekunden
        (60.0/bpms[dieSzene]* viertelproTakt[dieSzene])
#end

#macro sDauer (dieSzen)
        // die Laenge der Szene in Sekunden
        (taktDauer(dieSzen)* anzahlTakte[dieSzen])
#end
```

```
#macro dieTaktnummer (dieSze, dieUhr)
        // der Takt als Funktion der clock
        (dieUhr * anzahlTakte[dieSze]
        + 2.0 - auftakte[dieSze])
#end

#declare anfangszeiten = array[finale+1]
   // Zeit in Sekunden fuer den Anfang der Szene
#declare anfangszeiten[0] = 0.0;
#local sz = 1;
#while (sz <= finale)
        #declare anfangszeiten[sz]
        = anfangszeiten[sz-1] + sDauer (sz-1);
        #set sz = sz+1;
#end

#macro pvec (vec, mes)
// drucke vektor
        #warning concat("\n", mes,
        str(vec.x,10,3),
        str(vec.y,10,3),
        str(vec.z,10,3), "\n" )
#end

#macro initSceneClock (szene)
// szene: hier den Index der Szene in den Tabellen uebergeben
        #declare aktuelleSzene = szene;
        #declare taktnummer = dieTaktnummer (szene, clock);
        #declare taktBild1 = dieTaktnummer (szene, 0.0);
        #declare taktletztesBild = dieTaktnummer (szene, 1.0);
        #local  szenendauer = sDauer (szene);
        #declare skyclock = anfangszeiten[szene] +
                            clock*szenendauer;
        // Drucke aktuellen Takt und aktuelle Bildnummer
        #warning concat("\n␣␣␣Takt␣:␣␣␣␣␣␣",
                        str(taktnummer,10,3))
        #warning concat("\n␣␣␣Dauer␣[s]:␣",
                        str(szenendauer,10,3))
        #warning concat("\n␣␣␣Frames:␣␣␣␣",
                        str(szenendauer*FpS,10,3))
        #warning concat("\n␣␣␣Bild:␣␣␣␣␣␣",
```

```
                                str(frame_number,10,3))
#end

#macro einstellungszeit (anfangstakt, endtakt, takt)
        #declare sceneclock
           = (takt-anfangstakt)/(endtakt-anfangstakt);
        #warning concat("\nsceneclock:      ",
                                str(sceneclock ,10,7))
#end
```

57 Video-Schnitt

Für den Videoschnitt stehen Ihnen auf den iMacs die Programme iMovie und Final cut pro zur Verfügung. Hiermit können sie Titel, Überblendungen und Effekte und Toneffekte in Ihren Film einfügen. eine gute Einführung in den Videoschnitt finden Sie in ?.

57.1 Eine Folge von Einzelbildern zum Quicktime Movie verbinden

Mit ffmpeg: Öffnen sie ein Terminalfenster uund geben Sie folgendes Kommando ein:

```
/opt/local/bin/ffmpeg -f image2 -i PATH_TO_IMAGES/%d.png -b 400000
```

Dabei wird aus den Bildern, die in PATH_TO_IMAGES/%d.**png** liegen, ein MPEG-Film movie.mpgmit einer Bitrate von 4000kBit/s erzeugt. Der Parameter %d im Pfadnamen gibt an, wieviele Ziffern die Nummerierung der Bilder hat. Die Bitrate kann man mit dem Parameter -b gegebenenfalls auch ändern. Interessant ist noch, wie die Bilder übergeben werden, hierzu ein paar Beispiele:

1. Die Bilder liegen in PATH_TO_IMAGES/: Dateinamen 001.**png** – 999.**png**

   Zu übergebenes Argument: PATH_TO_IMAGES/%03d.**png**

2. Die Bilder liegen in PATH_TO_IMAGES/: Dateinamen szene1−001.**png** – szene1−999.**png**

Zu übergebenes Argument: PATH_TO_IMAGES/szene1−%03d.**png**

3. Die Bilder liegen in PATH_TO_IMAGES/: Daeinamen szene1−1.**png** - szene1−999.**png**

   Zu übergebenes Argument: PATH_TO_IMAGES/szene1−%d.**png**

Mit Quicktime 7: Mit Quicktime 7 können sie die Folge der Einzelbilder, die POV-Ray ausgegeben hat, zu einem Quicktime-Movie verbinden. Die Einzelbilder müssen forlaufend nummerierte Namen haben, z.B. bild01.png, bild02.png usw. Wählen Sie im Menue „File/Open Image Sequence“ das erste Bild Ihrer Folge und setzen Sie Abspielgeschwindigkeit als die Zahl der Bilder pro Sekunde.

Mit GraphicConverter: Die Einzelbilder müssen ebenfalls fortlaufend nummerierte Namen haben, z.B. bild01.png, bild02.png usw. und sich in einen separaten Ordner befinden. Wählen Sie im Menue „File/Export Slideshow to Movie“.

Achtung! In der neuen Version von GraphicConverter geht das nur noch für eine Bildrate von höchstens 1 Bild/Sekunde.

57.2 iMovie

Am einfachsten ist der Schnitt mit iMovie. iMovie steht auf allen iMacs zur Verfügung. Als besondere Effekte sind „green screen“ und „Bild in Bild“ möglich.

57.3 Final cut pro

Final Cut Pro ist eine sehr aufwendige Schnittumgebung, die nur an zwei Arbeitsplätzen zur Verfügung steht. Zum Final Cut Pro Paket gehören auch die Werkzeuge: Soundtrack Pro (Vertonen), DVD Studio Pro (Ausgabe auf DVD) und LiveType (animierte Titel)

57.4 Quartz composer

Quartz composer finden Sie im Verzeichnis „Entwickler: Programme“. Hier können Sie selbst Filter definieren, mit denen Ihre Videos kombiniert werden sollen. Beispiele fuer Blue Screen mit Quartz Composer: http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/StellingenUFO.html

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/HandUFOMovie.html

http://kogs-www.informatik.uni-hamburg.de/~dreschle/Animationen/IVC_Movies/Pages/HandMaennchenMovie.html

58 Vertonen

58.1 Garageband

Am flexibelsten läßt sich eine Tonspur mit Garageband zusammenstellen. Hier sind viele vordefinierte Toneffekte, Audio Loops und virtuelle Musikinstrumente vorhanden. Sie können auch Sprache aufnehmen oder E-Guitarren oder Keybords anschliessen. Garageband finden Sie auf allen iMAcs.

58.2 Finale

Nur an wenigen Arbeitsplätzen sind Soundtrack Pro und Finale vorhanden. Mit Finale können Sie Notenblätter setzen oder scannen und so Midi-Files für die musikalische Untermalung erzeugen. Diese Midi-Files können Sie dann in Garageband mit vordefinierten Audio-Loops zur Begleitung kombinieren.

59 Bildverarbeitung

Als Bildverarbeitungsprogramme finden Sie auf allen iMacs Photoshop, GIMP und GraphicConverter. Mit allen Programmen können sie die hier besprochenen ikonischen Bildoperationen durchführen.

60 Zur Prüfungsvorbereitung: Verständnisfragen

1. **Koordinaten und Transformationen:**

   (a) Wie unterscheiden sich das linkshändige und das rechtshändige Koordinatensystem?

   (b) Geben Sie die Matrizen für grundlegende Transformationen im 2D und 3D an.

   (c) Welche Transformationen sind kommutativ, welche nicht?

   (d) Welche Transformationen benötigen Sie, um den Radius einer Kugel zu verdoppeln, deren Mittelpunkt die Koordinaten (1,1,2) hat? Geben Sie die Matrizen an.

   (e) Gegeben sei ein achsenparalleler Einheitswürfel zentriert um den Ursprung und direkt daneben, zentriert um die positive X-Achse, eine Einheitkugel. Welche Transformation ist nötig, um die Kugel direkt mittig über den Würfel zu setzen?

   (f) Welche Fixpunkte haben Translation, Rotation und Scherung?

   (g) Gegeben seien die Punkte $P_1 = (1, 2, 3, 4)$, $P_2 = (2, 4, 6, 8)$ und $P_3 = (0.75, 1, 1.5, 2)$ in homogenen Koordinaten. Welche Punkte sind identisch?

2. **Projektionen:**

   (a) Erkären Sie den Unterschied zwischen orthogonaler und schiefer Parallelprojektion.

   (b) Nennen Sie Vor- und Nachteile und wichtige Anwendungen der einzelnen Projektionen.

   (c) Was sind Hauptfluchtpunkte und wo liegen diese im Bild?

   (d) Gegeben sei eine Kamera, die auf der Y-Achse über der XZ-Ebene steht und leicht nach unten geneigt ist.

Wie verändert sich die Lage der Hauptfluchtpunkte, wenn Sie die Kamera etwas stärker nach unten neigen? Wie verändert sich die Lage der Hauptfluchtpunkte, wenn Sie die die Brennweite der Kamera verdoppeln?

(e) Was sind konjugierte Fluchtpunkte? Wo liegt der konjugierte Fluchtpunkt zu einem Fluchtpunkt auf der Horizontlinie?

(f) Ist die Zentralprojektion invertierbar?

(g) Geben Sie die Projektionsmatrix für die Zentralprojektion und die inverse Zentralprojektion an.

(h) Was ist die perspektivische Verkürzung?

(i) Erklären Sie die Illusion des Ames-Raumes.

(j) Wie unterscheidet sich die Abbildung durch eine Lochkamera von der Abbildung durch eine Linsenkamera?

(k) Was sind Nahpunkt und Fernpunkt und wovon hängt deren Lage ab?

3. **Licht und Farbe:**

(a) Wie hängen Wellenlänge, Frequenz, Geschwindigkeit und Farbe des Lichtes zusammen?

(b) Wie hängt die Farbe einer Lichtquelle mit der Temperatur der Lichtquelle zusammen?

(c) Nennen Sie einige Farbmodelle.

(d) Wie können Farben aus Grundfarben gemischt werden?

(e) Was sind Komplementärfarben?

(f) Nennen Sie die Komplementärfarbe zu „rot“ im RGB- und im CMY-System.

(g) Welche Farbe erhalten Sie bei der additiven Mischung von „rot“ und„blau“ und welche bei der subtraktiven Mischung?

(h) Was ist die chromatische Aberration? Welche Konsequenzen hat sie für das Farbensehen?

(i) Welche Farbe können wir am schlechtesten wahrnehmen? Warum?

(j) Was ist Dispersion?

4. **Fraktale und Texturen:**

   (a) Nennen Sie wichtige Eigenschaften der quadratischen Kochkurve.

   (b) Wie können Fraktale eingesetzt werden, um Oberflächen zu modellieren?

   (c) Was sind Drachenkurven?

   (d) Warum sind Texturen für die photorealistische Computergrafik wichtig?

   (e) Warum sind Texturen für die räumliche Wahrnehmung wichtig?

5. **Ikonische Bildverarbeitung:**

   (a) Welche Teilaufgaben sind zu lösen, um aus einem kontinuierlichen Bildsignal ein Digitalbild zu machen?

   (b) Wie kann der Kontrast eines Bildes erhöht werden?

   (c) Was ist beim Verkleinern eines Bildes zu beachten?

   (d) Was ist Aliasing und wann tritt dieser Effekt auf?

   (e) Wie kann Aliasing vermieden werden?

   (f) Wodurch kann die Schärfe eine Bildes erhöht werden?

   (g) Wodurch können systematische Bildstörungen mit einer typischen Frequenz beseitigt werden?

   (h) Was bewirkt ein Tiefpaßfilter?

6. **Segmentieren von Bildern:**

   (a) Nennen Sie einige Kantenfinder und beschreiben Sie die Algorithmen.

   (b) Welche Rolle spielen Gradienten beim Kantenfinden?

   (c) Wie können mit dem Laplace-Operator Kanten gefunden werden?

   (d) Welcher Kanten-Operator ist besonders rauschempfindlich umd warum?

Index

Barber pole illusion, 609
Bernsteins Ungleichung, 396
Fourier-Reihe
  Dreieck-Funktion, 381
  Koeffizienten, 380
  Periodische Funktion
    Herleitung, 378
  Rechteck-Funktion, 380
  Sägezahn-Funktion, 381
Fourier-Transformation
  eindimensional, 382
Gaußfilter, 495
Grenzfrequenz
  Beschränkung der ersten Ableitung, 396
Kompatibilitätsgraph, 693
Mandelbrot-Menge
  Definition, **339**
Moving Light Displays MLD, 607
Opt. Illusionen:
  Alte und junge Frau, 547
  Ames-Raum, 133
  Barber Pole Illusion, 609
  Bewegungsfeld: Drei rotierende Kugeln, 605
  Blendeneffekt, 608
  Broken wheel illusion, 403
  Chromatische Aberration der Linse: Chromatek-Verfahren, 274
  Farbenfehlsichtigkeit, 272
  Gegenfarbe, 270
  Gestaltsillusion Dreiecke, 272
  Kontext: Die Rolle des Kontextes einer Strecke, 547
  Metamerie, 269
  Moving Light Displays MLD, 607
  Napoleon an seinem Grab, 273
  Necker-Würfel
    Parallelprojektion, 584
    Zentralprojektion, 585
  Schachbrettillusion, 265
  Streckenillusion, 131
    Quader, 132
  Unmögliche Objekte, 134
    Escher, M.C., 136
    Hogarth, William, 135
Optischer Fluß
  Definition, **606**
Weißabgleich, 206
Autoren:
  Agin, 544
  Arnold, R. David, 653
  Avinash C. Kak, Avinash C., 26
  Ayache, Nicholas, 648, 653
  Baker, 658
  Ballard, Dana H., 382, 394
  Barnard, 660
  Barrow, H. G., 536, 539
  Binford, Thomas O., 574, 580, 639
  Binford, Thomas O. Binford, 653
  Boyle, Roger, 43, 610
  Brown, Christopher M., 382, 394
  Burt, 657
  Canny, J., 504
  Clocksin, 590
  Crowley, James L., 641
  Davies, E. R., 43
  Dreschler, 660

Dreschler-Fischer, Leonie, 575, 585, 640
Duda, Richard O., 43, 119, 396
Erb, C., 261
Escher, M.C., 135
Fahle, M., 261
Faverjon, Bernard, 648, 653
Fellner, W. D., 277
Fellner, W.D, 25
Fellner, W.D., 16, 42, 49
For human vision to be explained by a computational theory, 524
Gershon, 654
Glassner, 550
Glassner, Andrew S., 42, 141, 191, 192, 261
Goethe, Johann Wolfgang von, 214
Gonzales, Rafael C., 26, 43
Haberäcker, P., 42
Hanson, 450
Hart, Peter E., 119, 396
Hart, Peter, E., 43
Helmholtz, H. von, 247, 264
Hildreth, 499
Hildreth, Ellen, 499
Hlavac, Vaclav, 43, 610
Hogarth, William, 135
Horn, 590
Horn, Berthold Klaus Paul, 43, 523, 549, 572, 649, 650
Horowitz, S. L., 478
Hough, 510
Ikeuchi, K., 549
Jenkin, 660
Julesz, 657
Kak, Avinash C., 365
Kanade, 658
Klinker, 654
Lloyd, 657
Marr, 654, 656
Marr, David, 498, 499, 524, 626
Maxwell, J. C., 247, 264
Nagel, 660
Nagel, H.-H., 640
Neumann, Bernd, 524, 576
Nevatia, 590
Nischwitz, A., 42
Nishihara, H. Keith, 524
Nyquist, 386
Ohta, 658
Parzen, 512
Pavlidis, T., 478
Peitgen, H.-O., 336
Peterson, Ivars, 173, 336
Poggio, 656
Poggio, Tomaso, 626
Posch, Stefan, 648
Richter, P. H., 336
Riseman, 450
Rosenfeld, Azriel, 365
Shannon, 386
Siegel, David, 397
Sonka, Milan, 43, 610
Sprengel, Rainer, 640
Steinmüller, Johannes, 43
Stork, David G., 43
Tenenbaum, M. G., 536, 539
Trivedi, 657
Tsotsos, 660
Ullman, 590
Ullman, Shimon, 640
Weigert, A., 177
Wood, Richard E., 26, 43
Woodham, R. G., 549, 565
Young, T., 247, 264
Zimmermann, H., 177

Beispiele:

Absorbierendes Medium, 238
Rauch, 240
Arithmetische Ausdrücke, 699

Array als Tabelle von Bildnamen, 724
Bedingter Ausdruck, 726
Bodennebel, 232
Brechung mit Dispersion (Prisma), 194
Definition einer Ebene, 705
Definition einer Kugel, 698
Definition eines animierten Quaders, 706
Definition eines Torus, 703
Definition von Objekten: Stardot, 64
Definition zweier Kegel, 702
Definition zweier Zylinder, 701
Diamanten mit Dispersion, 197
Ein Kristallglass-Material, 280
Ein virtuelles Treibhaus, 310
Emittierendes Medium, 236
Farbige Kugeln in weißem Licht, 213
FIM-Tracker, 679
Finish, 562
Flagge
  mit Unilogo, 717
Funktion
  Mitte, 722
Geschichteter Nebel: layered fog, 234
Histogramm Schärfen, 432
Kamera, 147
  Fischauge, 151
  Omnimax-Projektion, 153
  Zentralprojektion, 149
Kamera für Zentralprojektion, 147
Lichtquelle
  looks_like, 223
Makro: flagB, 720
Nebel
  einfach, 229
object
  Instanzen von Objektmodellen, 718
Parzenfensterschätzung für drei Geraden, 514
Parzenfensterschätzung für Notenbild, 515
Quaternionen-Julia-Fraktal, 348
Regenbogen, 187
Rekursive Objekte: Pyramide, 92
Relationengebilde für Hausbild, 685
Rotation eines Farbvektors, 710
Scherung einer Kugel, 75
Schneemann, 82
sky_sphere, 713
  Apocalypse, 715
Streuendes Medium, 239
  Fenster, 242
Synchronisation mit der Hintergrundmusik, 729
Test, ob ein Array-Element initialisiert ist., 727
Tropische Insel
  Texturkarten für Haie, 301
Tropische Insel und Meer mit Wellentextur, 295
Vektoroperationen: vaxis_rotate, 708
Visualisierung einer Helix, 165
While-Schleife: Eine Folge von Kugeln, 707
Zufällige Kugeln, 725
Definitionen:
Abbildung, 689
Affine Transformation, 59
Bewegungsfeld, 599
Bild
  Bildfolgen, 400
  Multispektralbild, 359

Volumenbild, 360
Bildbeschreibungs-Graph, 684
Binarisieren, 453
Chromatische Aberration, 203
Computergrafik, 17
Dispersion, 185
Empirischer Mittelwert, 423
Expansionspunkt, 602
Faltung, 385
Farbe, 207
Komplementärfarbe, 249
Farbtemperatur, 206
Farbton, 246
Fernpunkt, 143
Filter
Übertragungsfunktion, 410
Fluchtpunkt
Hauptfluchtpunkt, 120
Konjugierte Fluchtpunkte, 125
Grafische Datenverarbeitung, 17
Grauwerthistogram, 431
Grauwertverschiebungen, 606
Grenzfrequenz, 386
Hausdorff-Dimension, 317
Histogram
maskiert, 432
Homologe Punkte, 593
Hyperkomplexe Zahlen, 346
Julia-Menge, 337
Kante, 483
Korrespondenzproblem, 606
Kreuzkorrelation
normiert, 675
Mandelbrot-Menge, 339
Metamerie, 269
Nahpunkt, 143
Objektiv
Relative Öffnung, 555
Vignetierung, 555
Optischer Fluß, 606
Ortsraum, 360
Pixel, 364
Projektion, 97
Orthogonale Projektion, 102
Schiefe Parallelprojektion, 102
Zentralprojektion, 100
Quaternion, 345
R-Morphismus, 690
Reflexionsfunktion BDRF, 559
Regularisierung, 570
Relationenqebilde, 685
Rotation, 52
Schwarze Strahlung, 204
Schwarzer Körper, 204
Segmentierung, 448
Skalierung, 50
Sättigung, 246
Tiefenschärfe, 143
Translation, 49
Zusammenhang, 468
POV-Ray:
#break, 727
#case, 727
#debug, 727
#declare, 699
#end, 707
#if, 726
#ifdef, 727
#ifndef, 727
#include, 64
colors.inc, 64, 257
golds.inc, 64
metals.inc, 64
textures.inc, 64
#macro, 720
#range, 727
#switch, 727
#while, 707, 724
0, 709
1, 709
Animation

clock, 706
array, 724
dimension_size, 724
box, 706
camera, 147, 148
angle, 149
aperture, 155
fisheye, 151
focal_point, 155
location, 149
look_at, 149
omnimax, 153
orthographic, 150
perspective, 147, 149
clock, 77
color, 257
cone, 702
cylinder, 700, 701, 718
filter, 183
finish
Lambert, 562
mirror, 562
mixed, 562
fog, 229
distance, 231
fog_alt, 231
fog_offset, 231
fog_type, 231
ground fog, 231
layered fog, 234
function, 722
granite, 344
image_map, 718
include, 698
interior, 183
caustics, 194
dispersion, 194
dispersion_samples, 194
fade_distance, 194
fade_power, 194
ior, 183
light_source, 216
fade_distance, 218
shadowless, 717
spotlight, 220
lightsource, 698
looks_like, 223
mandel, 344
material, 280
interior, 280
texture, 280
matrix, 74
media
absorption, 236, 238, 240
density, 236, 238–240
emission, 236
scale, 240
scattering, 236, 239, 242
turbulence, 240
warp, 240
normal
quilted, 293
waves, 292
object, 64
pigment, 698
image_map, 301
brick, 283
gradient, 284
granite, 285
hexagon, 285
quilted, 293
radial, 285
spiral1, 283
spiral2, 284
turbulence, 287
warp, 288
plane, 705
polygon, 718
rainbow, 91, 187
rand, 725
rotate, 72
scale, 68

seed, 725
sky_sphere, 713
Apocalypse, 715
S_cloud1, 713
sphere, 697, 698, 718
Standardobjekte
Kugel, 697
torus, 703
translate, 64
union, 718
vaxis_rotate, 709, 710
vcross, 709
Vektor
0, 709
1, 709
x, 709
y, 709
z, 709
vnormalize, 709, 710
x, 709
y, 709
z, 709
Zitate:
For human vision to be explained by a computational theory
„Nishihara, H. Keith“, 524
shape from
contour, 523, 536
shading, 523, 569
stereo, 523
texture, 523
Öffnung
effektiv
Definition, **555**
relativ
Definition, **555**
4D-Kugel
Kreise auf der Oberfläche, 170
4D-Würfel, 167
Abbildung, 689
Definition, **689**
Abtasttheorem, 386
Abtastung
Faltung mit Kammfunktion, 394
Rekonstruktion, 390
Affine Transformation, 59
Agin, 544
Algorithmus
Definition, **400**
Aliasing, 391
Halo, 397
Allgemeinheitsannahme, 574
Anaglyphendarstellung, 614
Antialiasing, 396, 397
Strukturierter Hintergrund, 398
Arnold, R. David, 653
Attraktor, 335
Auflosung
geometrisch, 652
Aufnahmebedingungen
Segmentierung, 466
Aufösung, 375
Auge
Anatomie, 260
Blauschwäche, 262
chromatische Aberration, 262
Gegenfarbentheorie, 270
Netzhaut
Stäbe, 266
Zapfen, 266
Rezeptoren
Verteilung, 266
Transmission, 262
Zonentheorie, 270
Autostereogramm
Tapetentechnik, 632
Avinash C. Kak, Avinash C., 26
Ayache, Nicholas, 648, 653

Baker, 658

Ballard, Dana H., 382, 394
Bandbreite, 386
Bandsperre, 418
barber pole illusion, 608
Barnard, 660
Barrow, H. G., 536, 539
BDRF, 558
Beleuchtung, 525
Bereichs-Nachbarschafts-Graph, 478
Bestrahlungsstärke, 551
Bewegungsfeld, 599
    Definition, **599**
Bewegungsstereo, 593, 594
Bifurkationsdiagramm, 335
Bild
    Aussagekraft, 16
    Bildfolgen, 400
    Bildfunktion, 355
    Grautonbild, 355
    Grauwertgebirge, 356
    Multispektralbild, 359
    Volumenbild, 360
Bildbeschreibungs-Graph, 684
    Definition, **684**
Bilddeutung, 524
Bildentstehung, 525, 574, 641
    Bildintensität, 553
    photometrisch, 549
Bildfolgen, 400, 575
Bildformat
    Gif, 373
    Jpeg, 373
    Tiff, 373
    Xpm, 373
Bildrestauration, 418
Bildstrukturen, 586
Bildverarbeitung
    ikonisch, 421
Bildverstehen, 524
Binarisieren, 453
    Definition, **453**
Binford, Thomas O., 574, 580, 639
Binford, Thomas O. Binford, 653
Blendeneffekt, 608
Bootstrap-Problem, 586
Boyle, Roger, 43, 610
Brechungsgesetz, 182
Brechzahl, 184
Brennweite, 651
Brown, Christopher M., 382, 394
Burt, 657

CAD/CAM, 24
Canny, J., 504
Cantor-Balken, 322
Chromatische Aberration, 203
Clique, 693
Clocksin, 590
colors.inc, 258
Computergrafik, 17
    Definition, **17**
constructive solid geometry
    union, 718
Crowley, James L., 641

Davies, E. R., 43
Definitionspunkt, 49
Digitalbild
    Speicherbedarf, 372
Dimension
    Einheitswürfel, 316
    Hausdorff-Dimension, 316
    Multidimensionale Skalierung, 164
Diskretisierung
    Abtasten, 361
    Mosaik, 364
    Quantelung, 361
    Rasterung, 361
disparity limit, 657
Dispersion, 184, 185
    Cauchy Formel, 192

Definition, **185**
Regenbogen, 186
Sellmeier Formel, 191
Dithering, 373
Dreschler, 660
Dreschler-Fischer, Leonie, 575, 585, 640
Duda, Richard O., 43, 119, 396

edgel, *siehe* Kantenelement
Empirischer Mittelwert, 423
epipolare Geometrie, 642–644
epipolare Linien, 644
Erb, C., 261
Escher, M.C., 135
Expansionspunkt, 600, 602
Definition, **602**

Fahle, M., 261
Faltung, 385
Definition, **385**
Farbe, 207, 654
additive Mischung, 247
CMY-Einheitswürfel, 254
Definition, **207**
Komplementärfarbe, 249
Mischen
additiv, 248
subtraktive Mischung, 247
Trichromatische Theorie, 247
Farben
darstellbar, 251
Farbenfehlsichtigkeit, 271
Farbenkreis, 246
Farbparameter
Farbton, 246
Intensität, 246
Sättigung, 246
Farbsystem
CIE-System, 250
uniform, 252
CMY, 255
CNS, 256
HSV, 255, 256
Pyramide, 255
NTSC, 253
PAL, 253
RGB-System, 252
YIQ, 253
Farbtemperatur, 206
Definition, **206**
Farbton, 246
Definition, **246**
Faverjon, Bernard, 648, 653
Fehlerminimierung, 440
Fellner, W. D., 277
Fellner, W.D, 25
Fellner, W.D., 16, 42, 49
Fernpunkt, 143, 650
Definition, **143**
Filter
Übertragungsfunktion H(F)
Definition, **410**
Audio-Filter, 420
Bandsperre, 418
Diskret
FFT, 682
Faltung, 411
Hochpaß
Exponential-Hochpaß, 416
Impulsantwort, 411
Linearität, 410
Optimaler Tiefpaß, 413
Tiefpaß, 412
Exponential-Tiefpaß, 415
Wiener Filterung, 419
Übertragungsfunktion, 410
Filterung, 309
Fluchtpunkt, 649
Hauptfluchtpunkt, 120
konjugiert, **125**
Konjugierte Fluchtpunkte, 125

Fluchtpunkte, 637
fog, 229
For human vision to be explained by a computational theory, 524
Fourier-Transformation
  2-dimensional Bezeichnungen, 406
  2-dimensional Eigenschaften, 406
  Dreieck-Funktion, 383
  Kamm-Funktion, 385
  Kosinus-Funktion, 384
  Rechteck-Funktion, 382
  Sinus-Funktion, 384
  eindimensional, 382
  zweidimensional, 404
Fraktale
  Koch'sche Schneeflocke, 314
fraktale Objekte
  Dimension, 318
  Gebirge, 321
  Menger-Schwamm, 315
  Sierpinski-gasket, 318
  Sierpinski-Teppich, 315
frame, 80
Fraunhofer-Linien, 184

Generalitatsprinzip, 574, 642
geometrische Ahnlichkeit, 653
Geraden
  im Raum und im Bild, 581
Geradengleichung
  Polarform, 518
Gershon, 654
Glassner, 550
Glassner, Andrew S., 42, 141, 191, 192, 261
Glattheitsforderung, 661
Goethe, Johann Wolfgang von, 214
Gonzales, Rafael C., 26, 43
Gradient
  Digitalbild, 485
Gradientenraum, 563
Grafische Datenverarbeitung, 17
Grafische Datenverarbeitung
  Definition, **17**
Grauwertgebirge, 356
Grauwerthistogram, 431
  Definition, **431**
  Histogramausgleich, 433
  maskiert
    schärfen, 432
  Modifikation, 433
Grauwerthistogramme
  Anwendungen, 433
Grauwertinterpolation, 441
  Bikubisch, 442
  Bilinear, 442
  Nächster Nachbar, 441
Grauwertverschiebungen, 606, 640
  Definition, **606**
Grenzfrequenz, 386
  Definition, **386**

Haberäcker, P., 42
Hanson, 450
Hart, Peter E., 119, 396
Hart, Peter, E., 43
Hauptfluchtpunkt
  siehe Projektion, 120
Hausdorff-Dimension, 317
  Überdeckungsmenge, 317
  Definition, **317**
Helmholtz, H. von, 247, 264
Hildreth, 499
Hildreth, Ellen, 499
Histogram, *siehe* Gauwerthistogram431
  maskiert, 432
Hlavac, Vaclav, 43, 610
Hogarth, William, 135
Homologe Punkte, 593

Definition, **593**
Hopf-Karte, 174
Hopf-Kreise, 174
Horizont, 124
Horn, 590
Horn, Berthold Klaus Paul, 43, 523, 549, 572, 649, 650
Horowitz, S. L., 478
Hough, 510
Hough-Transformation, 510
hue, 246
Hyperkomplexe Zahlen, 346
Definition, **346**
hyperkomplexe Zahlen, 345

Ikeuchi, K., 549
Ikonische Bildverarbeitung, 421
Illusion
Broken wheel illusion, 403
Illusionen
Barber pole illusion, 609
interior, 183
ior, 183

Jaggies, 397
Jenkin, 660
Julesz, 657
Julia-Menge, 337
4-dimensional, 346
Cantor-Menge, 337
Cantor-Staub, 337
Definition, **337**
Drachenkurve, 337
Fatou-Staub, 337
Julia-Kurve, 337

Kak, Avinash C., 365
Kalibrierung, 157
Kamerahauptpunkt, 643
Kamerakoordinatensystem, 643
Kameramodell, 643
Kammfunktion, 393
Kanade, 658
Kante, 483
Beleuchtungskante, 537
Definition, **483**
Kantenelement, 483
Oberflächendiskontinuität, 537
Reflektanzkante, 537
Tangentialkante, 537
Kanten
Glanzlichtkanten, 655
Klassifikation von, 654
Orientierungskanten, 654
Reflektanzkanten, 655
Schattenkanten, 655
Tangentialkanten, 655
Kantenelement, 510
Kantenfinden
Hough-Transformation, 510
Kantenfinder
Canny-Operator, 505
Marr-Hildreth, 499
Mexican-Hat, 495
Roberts-Cross-Operator, 488
Sobel-Operator, 489
ableitungsorientiert, 483
DoG-Operator, 498
Laplace-Operator, 492
Klinker, 654
Koch-Kurve
quadratisch, 314
Koinzidenz
Schnittpunkte, 638
Kollage
Ahorn, 323
Ahornblatt, 323
Farnwedel, 326
Fliederstrauß, 324
Schneeflocke, 324
Kollagetechnik
Datenkompression, 325
Komplementärfarbe

Definition, **249**
Kontrastumkehr, 483
Kontrastveränderung, 370
Koordinaten
Gerätekoordinatensystem, 47
homogen
2-dimensional, 55
homogene Koordinaten, 55
kartesische Koordinaten, 47
normalisiertes Koordinatensystem, 47
Polarkoordinaten, 47
Weltkoordinatensystem, 47
Koordinatensystem
Händigkeit, 61
Händigkeit, 62
Rotation, 71
Koordinatentransformation, 79
Frames, 79
Koordinatentransformationen
Bild
affin, 439
geometrisch, 438
polynomial, 439
Korrelation
FFT
Faltung, 682
Korrespondenz
physikalisch, 640
visuell, 640
Korrespondenzanalyse, 640, 641
wissensbasiert, 642
Korrespondenzproblem, 525, 606, 639
Homologe Punkte, 619
Merkmalsdetektion, 640
Korrospondenzproblem
Definition, **606**
Kreuzkorrelation
Definition, **675**
normiert, 675
Licht, 176
Brechung, 182
Brechungsgesetz, 181
Brechzahl, 180
Dispersion, 181
Geschwindigkeit, 176
Intensität, 177
Interferenz, 180
Polarsation, 177
Reflexion, 180
Spektrum, 176
Streuung, 180
Lichtquelle
ausgedehnte Lichtquelle, 221
Punktlichtquelle, 215
Scheinwerfer, 219
Lindenmayer-Systeme, 327
Linien
parallel, 637
Linien, gerade, 636
Linse
Chromatische Aberration
Definition, **203**
chromatische Aberration, 184
Aberration, 203
chromatische Aberration, 181
Einstellebene, 140
Fernpunktsebene, 143
Gravitationslinse, 146
Nahpunktsebene, 143
Tiefenschärfe, 143
Zerstreuungskreis, 141
Linsenkamera, 651
Lloyd, 657
Lochkamera, 98, 643, 649

Makro
Fallstricke, 721
Mandelbrot-Menge, 339
Berechnung, 342
Mandelbrotmenge

als Wegweiser, 343
Marr, 654, 656
Marr, David, 498, 499, 524, 626
material, 280
maximale Geschwindigkeit, 659
Maxwell, J. C., 247, 264
Merkmalsraum, 670
Merkmalsvektor, 670
Metamerie, 269
Metrik, 435
Absolutwert-Metrik, 435
Absolutwertmetrik, 436
Digitalbild, 435
Einheitskreis, 436
Maximalwert-Metrik, 435
Maximalwertmetrik, 436
Zusammenhang, 436
Meßkammer, 157
Mitteln, *siehe* Rauschfilter
Mittelwert
Berechnung
inkrementell, 423
empirisch
Definition, **423**
Multispektralbild
Definition, **359**

Nagel, 660
Nagel, H.-H., 640
Nahpunkt, 143, 650
Definition, **143**
Necker-Würfel, 584
Netzhaut
geometrische Auflösung, 267
Neumann, Bernd, 524, 576
Nevatia, 590
Nischwitz, A., 42
Nishihara, H. Keith, 524
normal
quilted, 293
waves, 292
Normalentexturen, 289
Nyquist, 386

Objektansicht
untypisch, 578
Objekterkennung
Häufigkeitsanalysen, 670
Kreukorrelation
normiert, 675
gemeinsame Merkmale, 668
Objektiv
Relative Öffnung, 555
Vignetierung, 555
Ohta, 658
Optischer Fluß, 606
ordering constraint, 658
Ortsraum, 360
Definition, **360**

Parzen, 512
Parzen-Fenster, 512
Pavlidis, T., 478
Peitgen, H.-O., 336
Penrose triangle, 136
Peterson, Ivars, 173, 336
Phasenwinkel, 559
Photometrisches Stereo, 567
Photonen, 242
pigment
brick, 283
gradient, 284
granite, 285
hexagon, 285
quilted, 293
radial, 285
spiral1, 283
spiral2, 284
turbulence, 287
warp, 288
Pixel, 364
Definition, **364**

Poggio, 656
Poggio, Tomaso, 626
Posch, Stefan, 648
POV-Ray
  Animation
    clock, 706
  camera, 147
    Fischauge, 151
    Fokussierungsunschärfe, 155
    Omnimax-Projektion, 153
    Orthogonale Parallelprojektion, 150
    Sphärische Projektion, 155
    Zentralprojektion, 149
  Kontrollstrukturen
    while, 707
  Objektdefinition, 64
  Standardobjekte
    Ebene, 704, 705
    Kegel, 701, 702
    Kugel, 697, 698
    Quader, 706
    Torus, 703
    Zylinder, 700, 701
  translation, 64
Prinzip der kleinsten Verpflichtung, 642
Projektion, 97
  nicht planar, 97
  Augpunkt, 100
  Orthogonale Projektion, 102
  Parallelprojektion, 100, 101
    Orthogonale Projektion, 102
    Schiefe Parallelprojektion, 102
    Aufriß, 103
    Grundriß, 103
    normalaxonometrisch, 104
    Riß, 103
    schiefe Parallelprojektion, 106
    Schrägriß, 106
    Seitenriß, 103
  Perspektivisch, 637
  planar, 97
  Projektionsebene, 100
  Projektionsstrahl, 100
  Projektionszentrum, 100
  Schiefe Parallelprojektion, 102
  Spaltenprofil, 434
  stereografisch, 569
  Zeilenprofil, 434
  Zentralprojektion, 99, 100
    Dreipunktprojektion, 120
    Einpunktprojektion, 120
    Hauptfluchtpunkt, 120
    Hauptfluchtpunkt der Vertikalen, 122
    Inverse Zentralprojektion, 115
    Perspektivische Verkürzung, 109
    Transformationsgleichung, 113
    Zweipunktprojektion, 120

Quantelung, 367
Quantisierung
  Farbpaletten, 372
  numerische Lösung, 369
  optimal, 368
Quaternion, 345
  Definition, **345**
Quaternionen, 345
Querschnitt, 434

R-Morphismus, 690
  Definition, **690**
  Taxonomie, 691
Rangordnungsfilter, 430
Rastergrafik, 16
Rasterung
  Hexagonal, 365
  Matrix-Rasterung, 364
Raumkurve, 539
  Krümmung, 540

Schmiegeebene, 540
begleitendes Dreibein, 539
Raumwinkel, 551
Rauschen, 424
weißes Rauschen, 424
Signal/Rauschverhältnis , 425
Rauschfilter
Bimodales Mitteln, 429
Glätten eines Bildes, 428
Medianfilter, 430
Rangordnungsfilter, 430
rotierende Maske, 429
Reflektanzkarte, 565
Reflexionsfunktion, 558
Lambert'scher Strahler, 561
idealer Spiegel, 560
Definition, **559**
Reflexionsfunktion BDRF, 559
Regularisierung, 570
Definition, **570**
Glattheitsforderung, 570
Relationengebilde, 684
Relationenqebilde, 685
Definition, **685**
Relaxation, 587
RGB-Einheitswürfel, 252
Richter, P. H., 336
Riseman, 450
Rosenfeld, Azriel, 365
Rotation, 52

Scharfzeichnungsfilter, 444
Scheinfrequenzen, 391
Scherung
Beispiel, 75
Schwarze Strahlung, 204
Schwarzer Körper, 204
Schwellenwert
verrauschte Bilder, 464
Scharfentiefe, 652
Segmentierung, 448, 641
Multi-Skalen-Ansatz, 506
blob coloring, 470
Ballungsanalyse, 479
Binarisieren, 454, 456
Kriteriumsfunktion, 458
Referenzpositionen, 458
clustering, 479
Definition, **448**
Histogramdekomposition, 477
Schwellenwertverfahren, 453
Split-and-Merge, 478
Sehsystem
menschliches, 524
Shannon, 386
shape-from, 586, 588
Siegel, David, 397
Simultankontrast, 270
Sinusfunktion, 377
Skalierung, 50
smoothing, *siehe* Rauschfilter
Sonka, Milan, 43, 610
Spektralfarben, 246
Spektrum
elektromagnetisch, 178
Spielfilm, 33
Sprengel, Rainer, 640
Stanford Cart, 592
Steinmüller, Johannes, 43
Stereogramm
Zufallspunktstereogramm, 625
Stereorekonstruktion
Genauigkeit, 618
Stereoskopie, 525, 610
Geschichte, 611
Stork, David G., 43
Strahldichte, 552
Strahlung
Meßgrößen, 550
photometrisches System, 550
radiometrisches System, 550
spektrometrisches System, 550

schwarz, 204
Strahlungsgesetze
Plancksches Strahlungsgesetz, 204
Stephan-Boltzmann-Gesetz, 204
Streuung, 239
Aerosolstreuung, 226
Kontrastminderung, 226
Molekülstreuung, 224, 225
Rayleigh-Streuung, 224
Suchprobleme, 587
Szenenbeschreibung
geometrische, 524
Szenenrekonstruktion, 525, 576
geometrische, 524
Szenenrekonstruktion, photometrisch, 523
Sättigung, 246
Definition, **246**

T-Vertex, 639
Tenenbaum, M. G., 536, 539
texture, 280
Texturgradient, 533
Tiefenhinweise, 527
Monokulare
Objektmodelle, 528
Schattenwurf, 530
Schattierung, 530
statisch, 528
Verdeckung, 530
motorisch, 528
Akkomodation, 528
Vergenz, 528
Tiefenschärfe, 143
Topologische Dynamik, 331
Transformation
affin, 59
POV-Ray, 74
dreidimensional, 67
Roll, pitch, yaw, 69
Rotation, 69
Rotationsrichtung, 70
Skalierung, 68
Translation, 68
Fixpunkt, 50
Hough-Transformation, 509–511
Geradenbüschel, 510
Koordinatentransformation, 78
lokale Operatoren, 443
Rechenregeln, 57
Rotation
Händigkeit, 70
Scherung, 59
Skalierung, 50
uniform, 50
Translation, 49
Translation, 49
Transparenz, 309
Trivedi, 657
Tsotsos, 660

Ullman, 590
Ullman, Shimon, 640

Vektorgrafik, 16
Vignetierung
Definition, **555**
Visualisierung
Skalierung
Multidimensional, 166
Volumenbild
Definition, **360**

Weigert, A., 177
Weißabgleich, 371
Weißabgleich, 210
Wood, Richard E., 26, 43
Woodham, R. G., 549, 565
Würfel
mehrdimensional, 316

Young, T., 247, 264

Zimmermann, H., 177
Zufallszahlen, 724
Zusammenhang, 468
    Definition, **468**

Übungsaufgabe, 40

Glossary

Abtastheorem Eine Funktion mit Bandbreite W kann exakt aus Abtastwerten zurückgewonnen werden, wenn diese höchstens $\frac{1}{2W}$ auseinanderliegen. (Nyquist?Shannon sampling theorem). 376, 386

Bewegungsfeld Das Vektorfeld der Verschiebungen vom Zeitpunkt einer Bildaufnahme zum Zeitpunkt der nächsten Aufnahme (Verschiebungsvektoren) heißt „Bewegungsfeld“ (motion field).. 599

Bild Ein Bild ist ein ebenes Objekt mit variabler Helligkeit oder Farbe.. 16

Bild: Multispektralbild Die Bildfunktion eines Multispektralbildes bildet die Punkte der Bildebene auf einen Vektor von spektralen Messungen in mehreren spektralen Kanälen des elektromagnetischen Spektrums ab.. 359

Bild: Volumenbild 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).. 360

Bildfunktion: Die grundlegende Abstraktion eines Bildes. Sie beschreibt die Intensität als Funktion des Ortes und/oder der Zeit. 22, 354

Bildfunktion: Frequenzraum Im Frequenzraum werden die Bildfunktionen als Funktion der Ortsfrequenzen definiert.

. 360, 421

Bildfunktion: Ortsraum Die Bildebene $X \times Y$ eine 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. Der Ortsraum kann als weitere Dimension um die Zeit t erweitert werden.. 360, 421

Bildverarbeitung, ikonisch Bildverarbeitung im engeren Sinne bezeichnet alle Verarbeitungsverfahren, bei denen Bilder transformiert werden (Bildrestauration, Registrierung, Falschfarbendarstellung usw.) oder elementare Merkmale der Bilder abgeleitet werden.. 19

Bildverarbeitung, 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.. 19

Bildverstehen Bildverstehen hat die Aufgabe, die Bedeutung einer abgebildeten Szene zu erfassen, d.h. die Situation zu verstehen, in der sich die abgebildeten Personen oder Objekte befinden.. 19

Blendeneffekt Unser Sehsystem kann den optischen Fluß nur dort messen, wo eine Struktur in mindestens 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.. 608, 763

Brechungsgesetz Das Brechungsgesetz beschreibt die Richtungsänderung einer Welle beim Übergang von einem Medium mit der Ausbreitungsgeschwindigkeit c_1 in ein Medium mit der Ausbreitungsgeschwindigkeit c_2.

Sei α der Einfallswinkel und β der Ausfallswinkel, dann gilt für die Richtungen c_1 und c_2:

$$\frac{\sin\alpha}{\sin\beta} = \frac{c_1}{c_2}$$

. 181, 757

Brechzahl n Sei c_0 die Lichtgeschwindigkeit im Vakuum und c die Lichtgeschwindigkeit in einem bestimmten Stoff. Das Verhältnis beider Geschwindigkeiten $n = \frac{c_0}{c}$ wird die *Brechzahl* oder der Brechungsindex (index of refraction) genannt und ist für jeden Stoff eine charakterische Konstante. Die Größe der Richtungsänderung wird durch das Brechungsgesetz gegeben. 183, 758

Chromatische Aberration Licht unterschiedlicher Wellenlängen wird unterschiedlich stark gebrochen, siehe Dispersion. Dadurch entsteht die chromatische Aberration, ein Abbildungsfehler von Linsen, der bei schlecht korrigierten Objektiven zu unscharfen Bildern und farbigen Rändern führen kann.. 203, 262

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.

. 17

Definitionspunkte Definitionspunkte sind Punkte, die die Lage, Größe oder Form eines Objektes festlegen: Eckpunkte eines Rechtecks, Mittelpunkt eines Kreises usw.. 49

Digitalisierungsmodelle Digitalisierungsmodelle beschreiben den Prozeß der Diskretisierung von analogen Bildfunktionen.. 22

Diskretisierung Zur Verarbeitung im Digitalrechner müssen analoge Bilder diskretisiert werden: sie werden im Definitionsbereich gerastert und im Wertebereich quantisiert. Ein Bild wird so durch eine endliche Zahl von diskreten Abtaststellen repräsentiert, die wiederum nur eine endliche Zahl von Farben oder Helligkeiten haben. Das entspricht der Wiedergabe als ein Mosaik — auch hier wird das Bild aus einem endlichen Vorrat von Steinchen zusammengesetzt.. 361

Dispersion Unter Dispersion versteht man (in der Optik) die Wellenlängenabhängigkeit der Fortpflanzungsgeschwindigkeit einer Wellenbewegung, und damit auch der Brechzahl n 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**:** $\nu = \frac{1}{\omega}$.

. 184, 198, 757

Einstellebene Die *Einstellebene* ist diejenige Ebene senkrecht zur optischen Achse, deren Punkte bei einer gegebenen Bildweite s' auf der Bildebene scharf abgebildet werden.. 140, 763, 766

Expansionspunkt Der Expansionspunkt (focus of expansion)FOE ist der Bildpunkt des scheinbaren Ausgangspunktes der Bewegung zum Zeitpunkt $t = -\infty$.

$$FOE = (\frac{u}{w}, \frac{v}{w})$$

. 602

Farbe Farbe ist eine Empfindung, die durch sichtbare Strahlung über das Sinnesorgan „Auge“ ausgelöst wird.. 207

Farbe: 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.. 246

Farbmischung Farben lassen sich eindeutig angeben, indem sie als Mischung (Linearkombination) ausgewählter, definierter Grundfarben beschrieben werden.. 247

Farbmischung: additiv Die additive Farbmischung ist eine gleichzeitige oder rasch periodisch wechselnde Beleuchtung derselben Netzhautstelle durch verschiede Farbreize. Es wird das Licht verschiedener Lichtquellen vereinigt (addiert), z.B. auf einem Bildschirm.. 247, 355

Farbmischung: subtraktiv [Subtraktive (substantielle) Farbmischung:] Eine Folge von spektralen Veränderungen am ursprünglichen Farbreiz; es werden absorbierende Pigmente kombiniert, z.B. beim Druck.. 247, 355

Farbmodelle Farbmodelle beschreiben die Beziehung zwischen den Farben im Bild und spektralen Messungen.. 22

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.. 206

Farbtemperatur: Lichtquelle Die Farbtemperatur einer Lichtquelle ist die Temperatur des schwarzen Körpers, bei der dieser ein Licht gleicherFarbe 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 ähnlichste Farbtemperatur zugeordnet. (DIN 5033, Bl.1, 1953). 206

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.. 245

Fluchtpunkt Bei der Zentralprojektion enden die Abbildungen paralleler Geraden in einem gemeinsamen Punkt, dem Fluchtpunkt.. 108, 110, 760

Fluchtpunkt: 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.. 120

Fluchtpunkte: Konjugierte Fluchtpunkte Die Fluchtpunkte von zwei zueinander senkrechten Geraden $g_1 \perp g_2$ parallel zur Grundebene heißen konjugierte Fluchtpunkte.. 125

Fraunhoferlinien Die Absorptionslinien im Sonnenspektrum werden Fraunhoferlinien genannt. Sie wurden 1802 von Wollaston entdeckt und unabhängig davon 1804 von Fraunhofer entdeckt. `http://de.wikipedia.org/wiki/Fraunhoferlinie`. 184, 203, 758

Geometrische Modelle Geometrische Modelle dienen zur Modellierung der Form und Anordnung von Objekten, sowie der Abbildung von dreidimensionalen Szenen auf zweidimensionale Bildräume.. 22

Gerätekoordinaten Die Koordinaten, die von einem spezifischen Ausgabegerät verwendet werden, nennt man *Gerätekoordinaten*. 47, 762

grafische Datenverarbeitung Die grafische Datenverarbeitung umfaßt Verfahren zur Eingabe und Manipulation von Bildbeschreibungen sowie deren Umwandlung in grafische Darstellungen.. 17

Grauer Körper Der schwarze Körper ist ein idealisiertes Modell. Reale Körper geben meistens nicht soviel Strahlungenergie ab, wie für einen schwarzen Körper gleicher Temperatur zu erwarten wäre. Eine gute Näherung ist meistens der *graue Körper*, dessen spektrale Strahldichte (spectral radiance) relativ zum schwarzen Körper gleicher Temperatur angegeben wird, siehe spektraler Emissionsgrad. Die Sonne kann näherungsweise als *grauer Strahler* mit einer Emissivität ϵ von 0.99 modelliert werden.. 765

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.. 606

Grenzfrequenz Eine Funktion $g(t)$ ist bandbeschränkt mit Grenzfrequenz W, wenn die Fouriertransformierte $G(f) = 0$ für $|f| > W$.. 386

Homologe Punkte Homologe Punkte (auch korrespondierende Punkte genannt) sind Punkte in verschiedenen Bildern, die Abbildungen desselben Objektpunktes sind.. 593, 606, 624, 762

Kante Eine starke lokale Variation der Helligkeit (oder Farbe) im Bild heißt Kante.. 483

Kantenfinder Es gibt hunderte von Ansätzen zum Kantenfinden; im Wesentlichen drei Gruppen von Vorgehensweisen:

- Approximation des Gradienten oder des Laplace-Operators (Roberts, Sobel, Marr, ..)
- Diskrete Näherung, Reihenentwicklung,
- Schablonentest, Anpassung von parametrisierbaren Kantenmodellen
- Hypothesentest

. 483

Komplementärfarben Komplementärfarben sind Paare von Farben, deren additive Mischung weiß ergibt, bzw. schwarz bei subtraktiver Mischung. Komplementärfarben (Gegenfarben) liegen im Farbdiagramm auf einer Geraden durch den Weißpunkt oder Schwarzpunkt.. 249

Koordinaten: homogene Koordinaten Homogene Koordinaten repräsentieren Punkte eines n-dimensionalen Raumes durch eine äquivalente Gerade in einem $N+1$-dimensionalen Raum. In homogenen Koordinaten können affine Transformationen als Multiplikation eines Vektors mit einer Transformationsmatrix dargestellt werden.. 55

Koordinaten: Weltkoordinatensystem Weltkkordinatensysteme sind anwendungsspezifische Koordinatensysteme, in denen die darzustellende Welt modelliert wird. Diese Koordinatensysteme sind unabhängig von den Gerätekoordinaten des Ausgabegerätes.. 47, 760

Korrespondenzproblem Das Problem in einer Menge von Bildern *homologe Punkte* zu finden, heißt Korrospondenzproblem. Das Korrespondenzproblem tritt auf beim Berechnen des Bewegungsfeldes, bei der Stereoskopie und beim Registrieren von Bildern.

Das Korrespondenzproblem ist nicht immer eindeutig zu lösen, siehe Blendeneffekt.. 606, 761

Metrik eine Metrik ist ein Abstandsmaß $d(x, y)$ für zei Punkte x und y mit folgenden Eigenschaften:

Positiv:

$$\begin{aligned} d(x,y) &\geqslant 0 \\ d(x,y) = 0 &\rightarrow x = y \end{aligned}$$

Symmetrisch:

$$d(x,y) = d(y,x)$$

Dreiecksungleichung:

$$d(x,y) + d(y,z) \geqslant d(x,z)$$

. 435

Nahpunkt und Fernpunkt Die Ebenen, die den Bereich begrenzen, in dem bei einer gegeben 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.. 143, 766

Optischer Fluß Das Vektorfeld der vom visuellen System wahrgenommenen Verschiebungen heißt *optischer Fluß* (optical flow). 573, 606

Pixel Die diskreten Rasterelemente eines Digitalbildes heißen Pixel.. 364

Plancksche Strahlungsgesetz Die thermische Ausstrahlung eines schwarzen Körpers ist nur von der Temperatur und der Wellenlänge der Strahlung abhängig.. 204, 765

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 $m < n$.. 97, 763

Projektion: Aufriß Normalprojektion in die xy-Ebene.. 103

Projektion: Dreipunktprojektion Eine Zentralprojektion mit *drei* Hauptfluchtpunkten.. 120

Projektion: Einpunktprojektion Eine Zentralprojektion mit *einem* Hauptfluchtpunkt.. 120

Projektion: Grundriß Normalprojektion in die xz-Ebene.

. 103

Projektion: Orthogonale Projektion Eine Parallelprojektion ist eine *orthogonale Projektion* (engl. orthographic projection), auch *Normalprojektion* genannt, wenn die Projektionsstrahlen senkrecht zur Bildebene stehen.. 102, 764

Projektion: 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.. 100, 102

Projektion: Projektionsstrahl Projektionsstrahlen sind Geraden durch den Augpunkt und die abzubildenden Punkte.. 102, 764

Projektion: Riß Orthogonale Parallelprojektionen (Normalprojektionen) auf die Grundebenen des Koordinatensystems heißen *Risse.*. 103

Projektion: Schrägriß, schiefe Parallelprojektion Eine Parallelprojektion ist eine *schiefe Parallelprojektion* (engl. oblique projection), wenn sie keine Normalprojektion ist, also wenn die Projektionsstrahlen schräg auf die Bildebene treffen.. 102

Projektion: Seitenriß, Kreuzriß Normalprojektion in die yz-Ebene.. 103

Projektion: Zentralprojektion Eine Zentralprojektion wird festgelegt durch die Angabe eines *Projektionszentrums (Augpunkt)* und einer *Projektionsebene*. 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*.. 100

Projektion: Zweipunktprojektion Eine Zentralprojektion mit *zwei* Hauptfluchtpunkten.. 120

Radiometrische Modelle Radiometrische Modelle beschreiben, wie die Anordnung von Licht, Objektoberflächen und Bildwandlern die Helligkeit und Farbe im Bild beeinflußt.. 22

Raumwinkel Der Raumwinkel Ω wird durch das Flächensegment A_1 der Einheitskugel angegeben, das von dem Winkel Ω aus der Oberfläche der Kugel ausgeschnitten wird.. 550

Schwarze Strahlung Schwarze Strahlung ist ein idealisierter Grenzfall der Strahlung eines Körpers, wobei die Energieverteilung durch das Plancksche Strahlungsgesetz beschrieben wird.. 204

schwarzer Körper Ein schwarzer Körper ist ein Körper, der eine schwarze Strahlung aussendet.. 204, 206, 761, 763, 765

Segmentierung 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.. 448

Spektraler Emissionsgrad, Emissivität Der spektrale Emissionsgrad ϵ_λ ist eine Materialkonstante. Er ist definiert als das Verhältnis der spektralen Strahldichte eines grauen Körpers zur spektralen Strahldichte eines schwarzen Körper gleicher Temperatur.. 761

Stefan Boltzmann Strahlungsgesetz Die gesamte Abstrahlung $L_{sk}(T)$ eines schwarzen Körpers über alle Wellenlängen integriert ist proportional zur vierten Potenz der Temperatur T.. 204

Strahldichte L (radiance) Die Strahldichte gibt die Leistung P pro Raumwinkelelement Ω. und pro Flächenelement A *senkrecht* zum Strahl an. Sie berücksichtigt auch die Orientierung der strahlenden Fläche A zum Raumwinkel Ω.. 553

Tiefenschärfe Die Tiefenschärfe ist der Abstand $t_F - t_N$ zwischen Fernpunkt und Nahpunkt. Die Tiefenschärfe hängt bei gegebener Einstellebene und gegebener Auflösung nur von der Blendenöffnung ab und ist umso größer, je kleiner die Blende ist.. 144

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

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

. 59, 766

Transformation: Pitch Drehung im Ursprung um die Y-Achse χ.. 69

Transformation: Roll Kippung im Ursprung um die Z-Achse ψ.. 69

Transformation: 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.. 49

Transformation: Scherung Eine Scherung eine bestimmte affine Abbildung, bei der der Flächeninhalt jeder ebenen Figur (Scherung der Ebene) oder allgemeiner das Volumen jeder Figur (Scherung eines mindestens 3-dimensionalen Raumes) erhalten bleibt.. 49

Transformation: 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.. 49

Transformation: Translation Unter Translation versteht man eine geradlinige Verschiebung eines Objektes oder eines Definitionspunktes.. 49, 766

Transformation: Yaw Neigung im Ursprung um die X-Achse ϕ. 69

Visualisierung Visualisierung hat die Aufgabe, eine gegebene Bedeutung in einem geeigneten Bild darzustellen.. 19

Wiensches Verschiebungsgesetz Je heißer ein schwarzer Körper ist, desto mehr verschiebt sich die Wellenlänge λ_{max} des Maximums der Strahlung zu den kürzeren Wellenlängen. Sei T die Temperatur des schwarzen Körpers, dann gilt: $\lambda_{max}T = const = 0.2897cm \cdot K$. 206, 759

Literatur

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*. Prentice-Hall, Englewood Cliffs, New Jersey.

Barnard, S. and Thompson, W. (1980). Disparity analysis of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2:333–340.

Barrow, H. G. and Tenenbaum, J. M. (1981). Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116.

Binford, T. (1981). Inferring surfaces from images. *Artificial Intelligence*, 17:205–244.

Bovik, A. (2009). *The Essential Guide to Image Processing*. Elsevier.

Brauer, M. (2008). L-systeme zur generierung geometrischer modelle mit besonderer berücksichtigung von pflanzen. Bachelor-arbeit, Department Informatik, MIN-Fakultät, Universität Hamburg, Hamburg.

Burt, P. and Julesz, B. (1980). A disparity gradient limit for binocular fusion. *Science*, 208:615–617.

Canny, J. (1986). A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):679–698.

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.

Crowley, J. L. (1984). A computational paradigm for three dimensional scene analysis. Technical Report CMU-RI-TR-84-11, The Robotics Institute, Carnegie-Mellon University, Pittburgh, PA.

Cyganek, B. and Siebert, J. P. (2009). *An Introduction to 3D-Computer Vision Techniques and Algorithm*. Wiley.

Davies, E. R. (1997). *Machine Vision*. Academic Press, 2nd edition.

Davison, A. (2005). *Killer game programming in Java*. O'Reilly, Beijing – Cambridge – Farnham.

Dreschler, L. S. and Nagel, H.-H. (1982). Volumetric model and 3d-trajectory of a moving car derived from monocular tv frame sequences of a street scene. *Computer Graphics and Image Processing*, 20:199–228.

Dreschler-Fischer, L. S. (1986). A knowledge based approach to the correspondence problem in sequences of stereo images. In Jorrand, P. and Sgurev, V., editors, *2nd Int. Conf. on Artificial Intelligence, Methodology, Systems, Applications AIMSA-86*, Varna, Bulgaria. NORTHHOLLAND++ 1987.

Dreschler-Fischer, L. S. (1987). Das "Bootstrap-Problem" bei der geometrischen Szenenrekonstruktion — eine Übersicht. In Morik, K., editor, *11. Fachtagung über Künstliche Intelligenz GWAI-87*, volume 152 of *Informatik Fachberichte*, pages 1–15, Schloß Eringerfeld bei Geseke. SPRINGER++ 1987.

dtv Lexikon (1970). *dtv-Lexikon der Physik*. Deutscher Taschenbuch Verlag.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, New York, 2nd edition.

Erb, C. and Fahle, M. (2006). Farbensehen und erworbene farbsinnstörungen. *Der Ophthalmologe*, 4:349–361.

Fellner, W. (1992). *Computergrafik*, volume 58 of *Reihe Informatik*. BI Wissenschaftsverlag, Mannheim – Wien – Zürich, 2. edition.

Fellner, W. (2004). *Computergrafik*, volume 58 of *Reihe Informatik*. BI Wissenschaftsverlag, Mannheim – Wien – Zürich, 3. edition.

Foley, J. D. and van Dam, A. (1984). *Fundamentals of Interactive Computer Graphics*. Addison-Wesley Systems Programming Series. Addison-Wesley, 2 edition.

Gershon, R., Jepson, A. D., and Tsotsos, J. K. (1987). Highlight identification using chromatic information. In *International Joint Conference on Computer Vision ICCV-87*, pages 161–170, London.

Glassner, A. S. (1995). *Principles of Digital Image Synthesis*, volume 2 Volumes of *The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publ. Inc., Los Altos/CA.

Gonzales, R. C. and Wood, R. E. (2008). *Digital Image Processing*. Pearson Prentice Hall, New Jersey, 3rd edition.

Görz, G., editor (2005). *Einführung in die Künstliche Intelligenz*. Addison-Wesley, Reading, Mass., 5th edition.

Hanson, A. R. and Riseman, E. M., editors (1978). *Computer Vision Systems*. Academic Press, Orlando.

Horn, B. K. P. (1986). *Robot Vision*. The MIT Press, Cambridge, Mass.McGraw-Hill, New York.

Hough, P. V. C. (1962). A method and means for recognizing complex patterns. US Patent 3,069,654 1962.

Ikeuchi, K. and Horn, B. K. P. (1981). Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184.

Jenkin, M. (1984). The stereopsis of time-varying images. Technical Report RBCV-TR-84-3, Department of Computer Science, University of Toronto.

Jenkin, M. and Tsotsos, J. K. (1986). Applying temporal constraints to the dynamic stereo problem. *Computer Vision, Graphics, and Image Processing*, 33:16–32.

Klinker, G. J., Shafer, S. A., and Kanade, T. (1988). The measurement of highlights in color images. *International Journal of Computer Vision*, 2:7–32.

Marr, D. (1976). Early processing of visual information. *Phil. Trans. R. Soc. (London)*, B275:483–524.

Marr, D. (1982). *Vision*. Freeman & Co., San Fransisco.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proc. R. Soc. (London)*, B207:187–217.

Marr, D. and Nishihara, H. K. (1978). Visual information processing: Artificial intelligence and the sensorium of light. *Technology Review*, pages 28–49.

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, pages 283–287.

Maxwell, J. C. (1860). On the theory of compound colours and the relations of the colours of the spectrum. *Phil. Trans. R. Soc. (London)*, 150.

Nagel, H.-H. (1986). Image sequences — ten (octal) years — from phenomenology towards a theoretical foundation. In Haton, editor, *8th Int. Joint Conf. on Pattern Recognition*, pages 1174–1185, Paris.

Neumann, B. (1982). Knowledge sources for understanding and describing image sequences. In Wahlster, W., editor, 6th *German Workshop on Artificial Intelligence "GWAI-82"*, number 58 in Informatik Fachberichte, pages 1–21, Bad Honnef. SPRINGER.

Nevatia, R. (1976). Depth measurement by motion stereo. *Computer Graphics and Image Processing*, 5(2):203–214.

Nischwitz, A. and Haberäcker, P. (2004). *Masterkurs Computergrafik und Bildverarbeitung*. Vieweg.

Ohta, Y. and Kanade, T. (1985). Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7:139–154.

Peitgen, H.-O., Jürgens, H., and Saupe, D. (1992). *Bausteine des Chaos: Fraktale*. Springer Verlag und Klett-Cotta, Stuttgart.

Peitgen, H.-O. and Richter, P. (1986). *The Beauty of Fractals: Images of Complex Dynamical Systems*. Springer-Verlag, Berlin – Heidelberg – New York,.

Peterson, I. (1988). *The Mathematical Tourist*. Freeman & Co., San Fransisco.

Poggio, G. F. and Poggio, T. A. (1984). The analysis of stereopsis. *Annual Review of Neuroscience*, 7:379–412.

Pollard, S. B., Mayhew, J. E., and Frisby, J. P. (1985a). Pmf: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470.

Pollard, S. B., Porrill, J., Mayhew, J. E., and Frisby, J. P. (1985b). Disparity gradient, lipschitz continuity, and computing binocular correspondences. Technical Report AIVRU Memo 010, Artificial Intelligence Vision Research Unit, University of Sheffield. auch in Proc. 3. Int. Symp. Robotics Research, 1985, 19–26.

Posch, S. (1988). Hierarchische linienbasierte tiefenbestimmung in einem stereobild. In *GWAI-88*, pages 275–285.

Reinhard, E., Ward, G., Pattanaik, S., DEbevec, P., Heidrich, W., and Myszkowski, K. (2010). *High dynamic range imaging: acquisition, display, and image based lighting*. Morgan Kaufmann und Elsevier.

Reiser, M. and Wirth, N. (1994). *Programming in Oberon — Steps Beyond Pascal and Modula*. ACM Press New York.

Rosenfeld, A. and Kak, A. C. (1982). *Digital Picture Processing*, volume I + II of *Computer Science and Applied Mathematics*. Academic Press, Orlando, 2 edition.

Siegel, D. (1996). *Creating Killer Web Sites*. Hayden Books.

Sonka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis, and Machine Vision*. PWSPublishing.

Sprengel, R. (1988). Untersuchung diffrentieller Ansätze zur Schätzung des optischen Flusses in Grauwert- und Farbbildfolgen. Bericht FBI-HH-B-137/88, Fachbereich Informatik, Universität Hamburg, Hamburg.

Steinmüller, J. (2008). *Bildanalyse: Von der Bildverarbeitung zur räumlichen Interpretation von Bildern*. eXamen.press. Springer, Berlin Heidelberg. ISBN 978-3-540-79742-5.

Stereogramm-1994 (1994). Stereogramm. Ars Edition.

Trivedi, H. P. and Lloyd, S. A. (1985). The role of disparity gradient in stereo vision. *Perception*, 14:685–690.

Ullman, S. (1979). *The Interpretation of Visual Motion*. The MIT Press, Cambridge, Mass.

von Helmholtz, H. (1852). Über die Theorie der zusammengesetzten Farben. *Annales Physique*, 887:45–66.

Weigert, A. and Zimmermann, H. (1971). *ABC Astronomie*. Verlag Werner Dausin, Hanau/Main.

Woodham, R. G. (1981). Analysing images of curved surfaces. *Artificial Intelligence*, 17:117–140.

Young, T. (1802). The bakerian lecture on the theory of light and colours. *Phil. Trans. R. Soc. (London)*, 92:12–48.

Yuille, A. L. and Poggio, T. (1984). A generalized ordering constraint for stereo correspondence. Technical Report AI-Memo 777, Artificial Intelligence Laboratory, MIT.