

COMPUTER GRAPHIK

5. Teils

RAY TRACING

Prof. Dr. Elke Hergenröther

Wichtig! Inhalte der Veranstaltung sind urheberrechtlich geschützt. Weder die Folien noch das Vorlesungsvideo dürfen an unbeteiligte Dritte weitergegeben werden.

Der Unterschied zwischen Reflexions- und Beleuchtungsmodell

- Reflexion wird an einem Punkt im Modell berechnet. Zur Reflexionsberechnung wird nur der Lichteinfallsvektor, der Vektor in Richtung des Betrachters, die Normale und das Material berücksichtigt.
- Das Phong'sche Reflexionsmodell, BRDF oder BSSRDF sind Reflexionsmodelle.
- Im Beleuchtungsmodell wird die Geometrie der gesamten Szene berücksichtigt. Wichtig ist die räumliche Anordnung der Geometrien, die Verdeckungen, Abschattungen, Spiegelungen und transparente Durchblicke verursachen. Ray Tracing ist beispielsweise eine Beleuchtungsmodell.
- Man kann auch definieren, dass man im Beleuchtungsmodell die Umgebung nicht berücksichtigen möchte, dann spricht man vom lokalen Beleuchtungsmodell. Wird die Umgebung einberechnet, dann ist es, um den Gegensatz deutlich zu machen, ein globales Beleuchtungsmodell.

Beleuchtungsverfahren: Ray Tracing



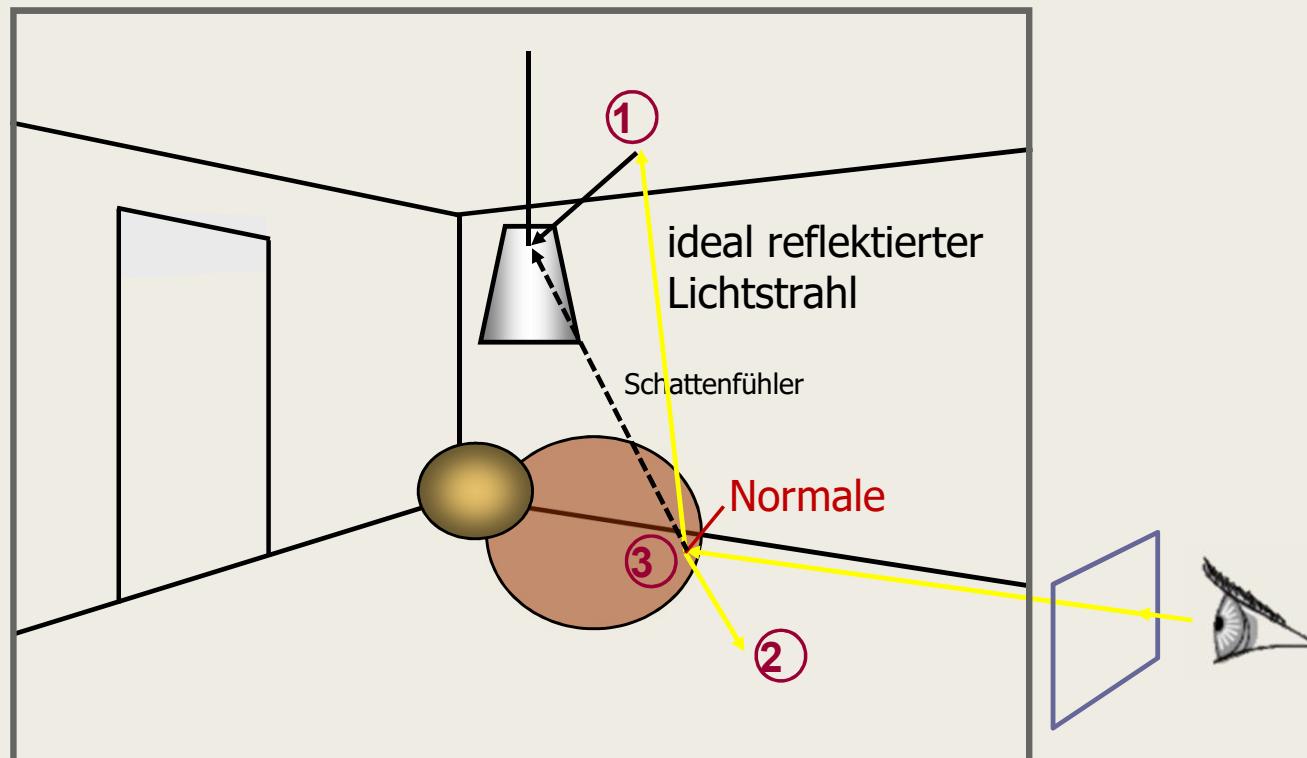
Spiegelungen, Schatten und bei komplexeren Ray Tracingverfahren kann man auch Lichtbündelungen – sogenannte Kaustics sichtbar machen.

Bild: Tobias Geis

Ray Tracing Grundidee:

Verwendung von Schattenfühler, um überprüfen zu können, ob der Punkt, der auf den Pixel abgebildet wird, beleuchtet ist.

Berechnung der Reflexionen an Position
1 (diffus), 2 (spiegelnd transparent) und **3 (spiegelnd opak + diffus + 2 + 1)**. Die Kugel ist also teilweise diffus, spiegelnd und transparent. Die Decke und der Boden aber sind nur diffus reflektierend:



Einschränkungen des Ray Tracing Verfahrens

- keine weichen Schatten (Halbschatten)
- Keine indirekte diffuse Beleuchtung
- Keine “Kaustiks“ (engl. Caustics)

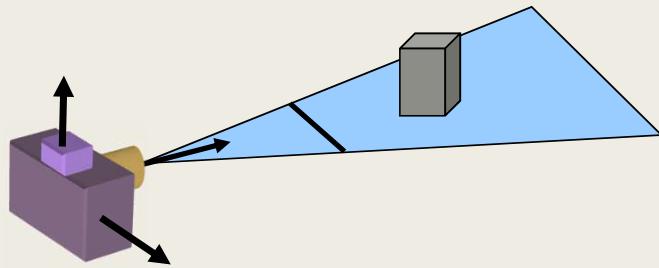


- Szene wird immer bezüglich eines Blickpunkts berechnet

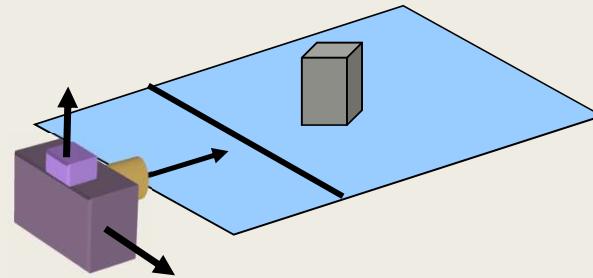
Wie erzeugt man eine orthographische Projektion im Ray Tracer?

Kurze Wiederholung des Unterschieds zwischen der perspektivischen und orthographischen Projektion.

Modell der virtuellen Kamera

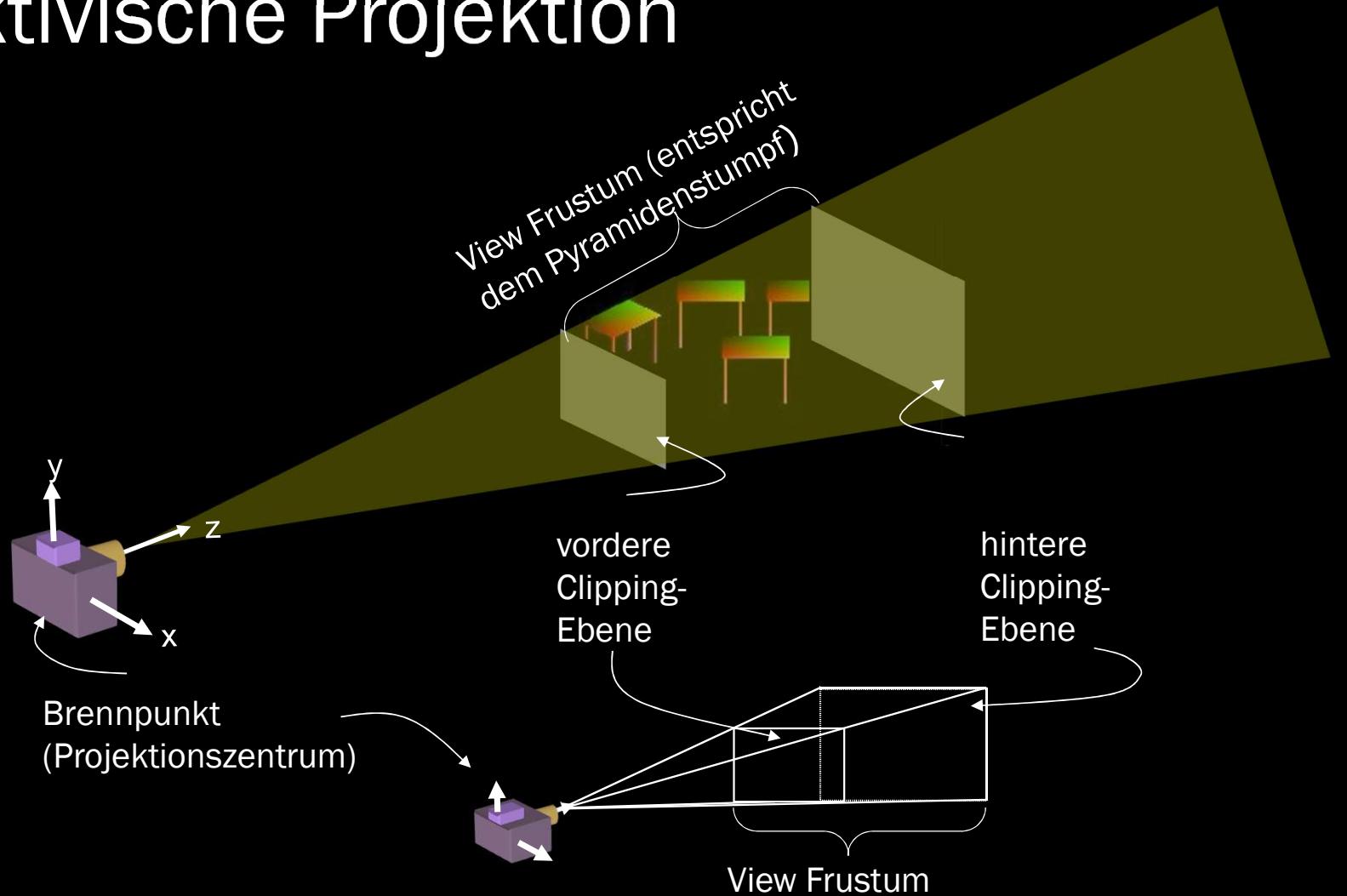


Perspektivische Projektion



Parallel- bzw. orthographische
Projektion

Perspektivische Projektion





Darstellung der perspektivischen Projektion

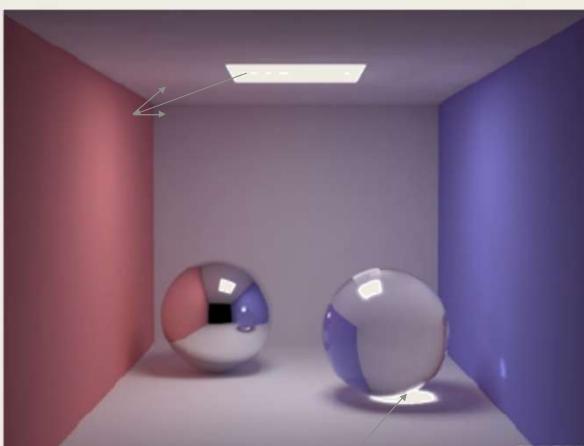
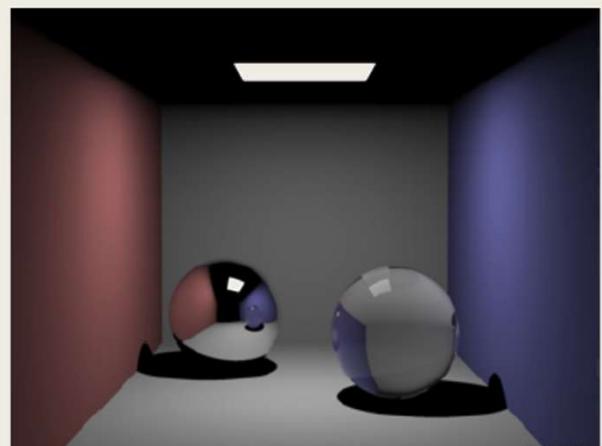
- Natürlichste Darstellungsart
- Sehstrahlen laufen auf einen Fluchtpunkt zu, so dass vordere Objekte größer dargestellt werden, als gleich große Objekte, die im hinteren Teil des Bildes zu finden sind

Wie erzeugt man eine orthographische Projektion im Ray Tracer?

- a) Indem man die Sehstrahlen von einem Projektionszentrum aus durch die Pixel der Bildebene schickt.
- b) Indem man die Sehstrahlen von den einzelnen Pixeln der Bildebene aus startet und sie dann entlang der Normalen der Bildebene in die Szene schickt.
- c) Weder a) noch b) sind richtig. Es muss eine andere Methode geben, um parallele Sehstrahlen berechnen zu können.

Wie erzeugt man eine orthographische Projektion im Ray Tracer?

- a) Indem man die Sehstrahlen von einem Projektionszentrum aus durch die Pixel der Bildebene schickt.
- b) Indem man die Sehstrahlen von den einzelnen Pixeln der Bildebene aus startet und sie dann entlang der Normalen der Bildebene in die Szene schickt.
- c) Weder a) noch b) sind richtig. Es muss eine andere Methode geben, um parallele Sehstrahlen berechnen zu können.



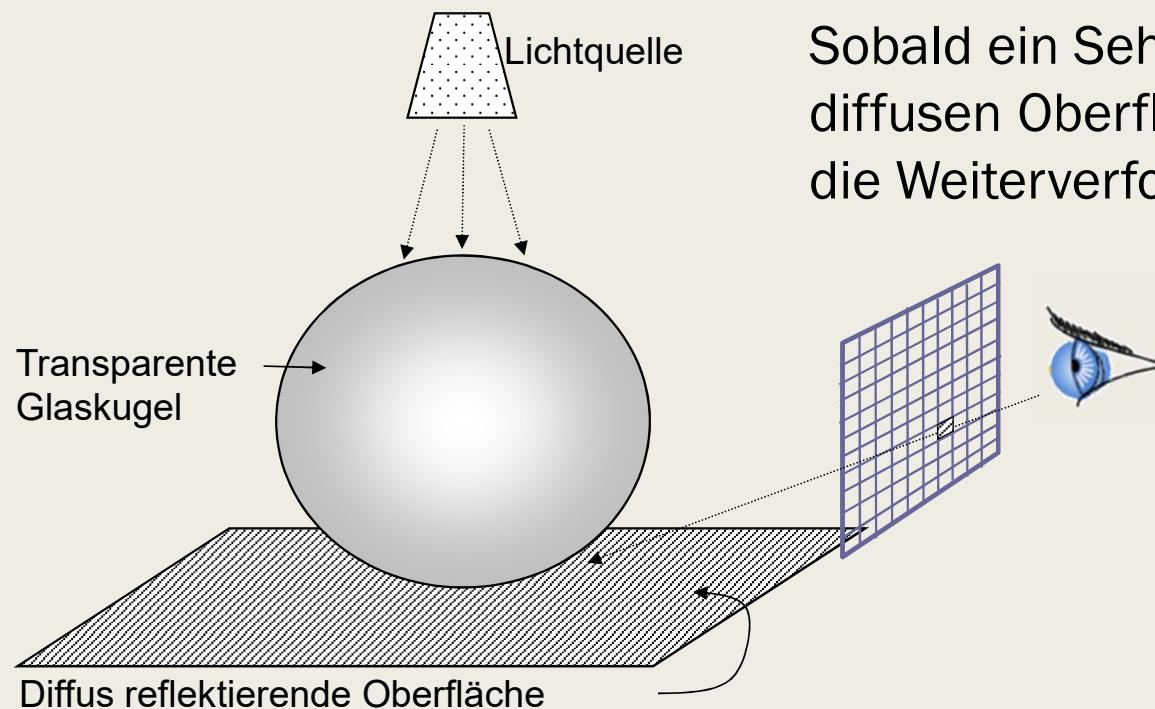
Kaustiks (= Lichtbündelungen) entstehen,
wenn die Lichtstrahlen durch die
Lichtbrechung in einem transparenten
Objekt konzentriert werden.

Unser Ziel:

Folgende Bildeigenschaften zu erzeugen...

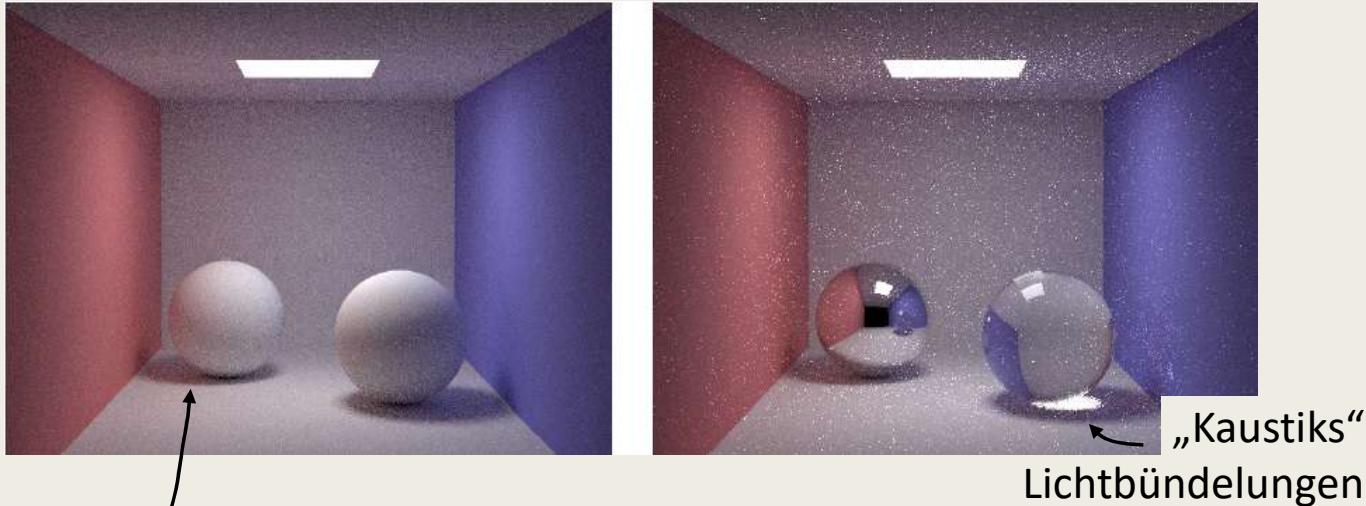
- weichen Schatten (Halbschatten)
- indirekte diffuse Beleuchtung
- Kaustiks (*engl. Caustics*)

Grundproblematik, die indirekte diffuse Beleuchtung und Kaustiks verhindert:



Sobald ein Sehstrahl auf einer diffusen Oberfläche auftrifft wird die Weiterverfolgung gestoppt.

Monte Carlo Path Tracing



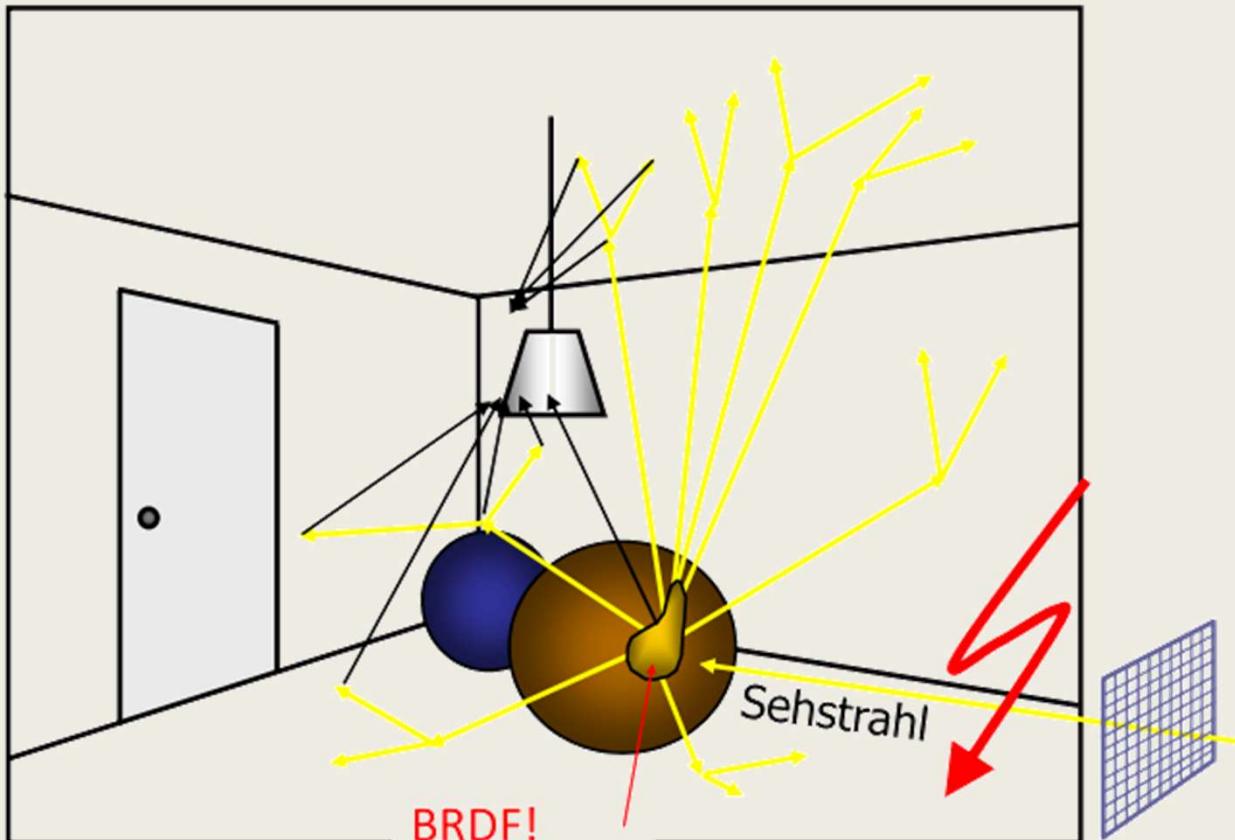
Vorteil: Auch diffuse indirekte Reflektionen können berechnet werden.

Nachteil: Starke Neigung zum Verrauschen

„Kaustiks“ und Spiegelungen verstärken das Rauschen.

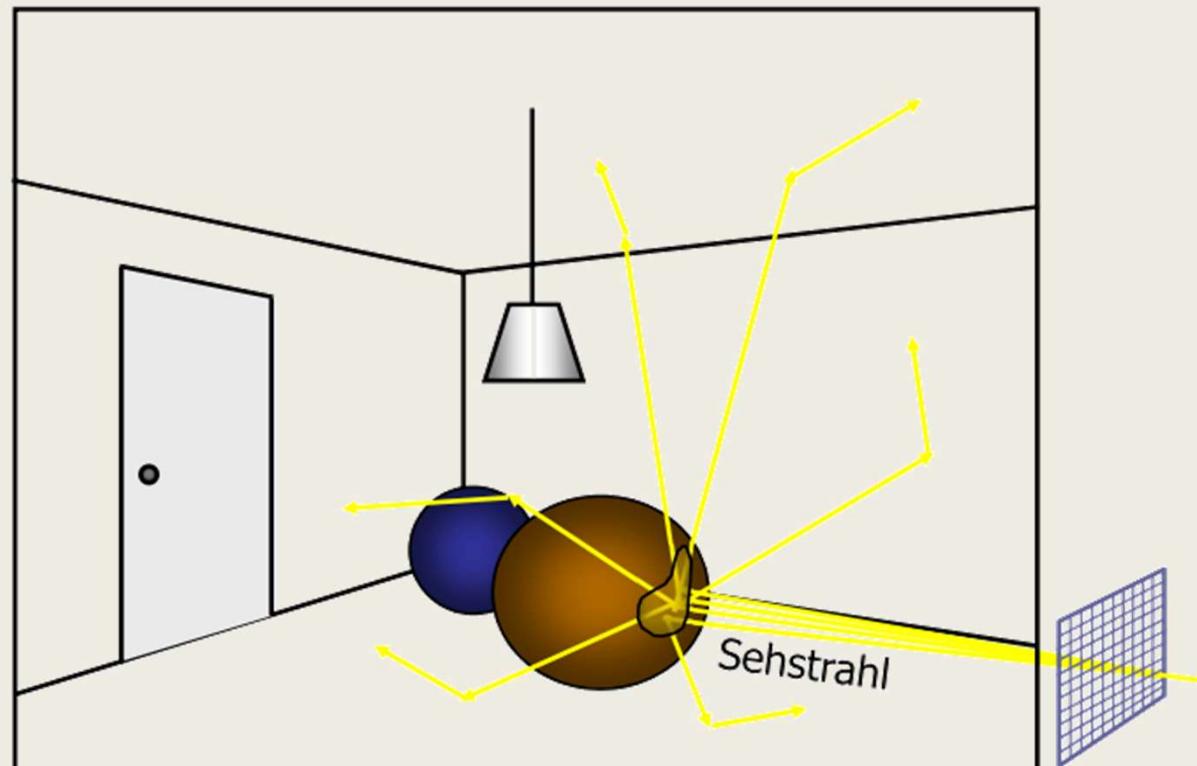
Monte Carlo Path Tracing

- Das Ziel ist, dass auch diffuse Materialoberflächen zu einer Weiterverfolgung des Sehstrahls führen!
- Würde man nach jeder Aufspaltung mehrere Sehstrahlen verfolgen, hätte man exponentielles Wachstum



Monte Carlo Path Tracing

- Da man kein exponentielles Wachstum will, wird jeder Sehstrahl nur einmal, in eine zufällig ausgewählte Richtung weitergeleitet.
 - Farbe des Pixels entsteht indem die Reflexionen aller durch den Pixel gehenden Sehstrahlen werden gemittelt.
 - Problem des Verfahrens: Starke Farbschwankungen bei benachbarten Pixel ist als Rauschen sichtbar.
- Warum ist das so?**
- Eine Methode der Reduktion: Untere Grenze an Sehstrahlen: 1.000 bis 10.000 Sehstrahlen pro Pixel, je nach Szene



Monte Carlo Path Tracing

Die Wahrscheinlichkeit in welche Richtung der Sehstrahl, der auf einer diffusen Oberfläche auftrifft, weitergeleitet wird, wird durch die BRDF festgelegt. So kann man bspw. wählen, dass eine Weiterleitung in die Richtung der Spiegelung wahrscheinlicher ist.

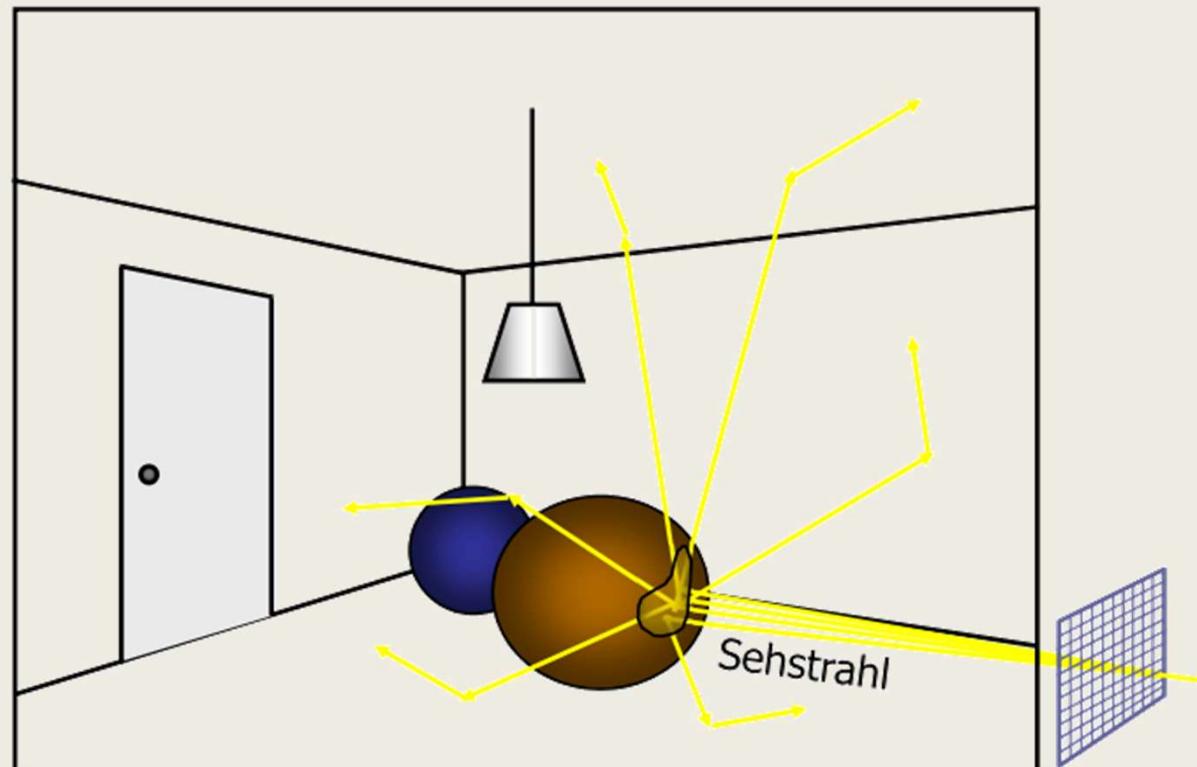
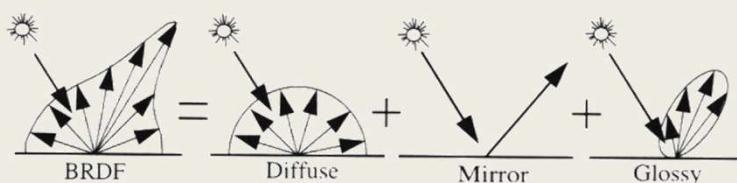


Bild links Hendrik Lenzsch aus: <http://resources.mpi-inf.mpg.de/departments/d4/teaching/ws200708/cg/slides/CG07-Brdf+Texture.pdf>



SPP: 100

SPP = Samples per Pixel

Prof. Dr. Elke Hergenröther

Bild aus: https://chunky.llbit.se/path_tracing.html

Qualität des Ergebnisses ist abhängig von den berechneten Farbwerten pro Pixel (äquivalent zu SSP)

Warum heißt das Verfahren *Monte Carlo* Path Tracing

- a. Weil das Verfahren in Monte Carlo entwickelt wurde.
- b. Weil es von Carlo Monte entwickelt wurde.
- c. Keine Ahnung, aber es ist ein zufallsbasiertes Verfahren, das mit einer nach oben beschränkten Wahrscheinlichkeit auch falsche Ergebnisse zulässt. Vielleicht soll der Name an die Spielbanken in Monte Carlo erinnern.
- d. Im Gegensatz zum Las-Vegas-Algorithmus ist das Monte Carlo Verfahren kein stochastisches Verfahren. Das wollte man mit einer klassischen europäischen Namensgebung unterstützen.

Warum heißt das Verfahren *Monte Carlo* Path Tracing

- a. Weil das Verfahren in Monte Carlo entwickelt wurde.
- b. Weil es von Carlo Monte entwickelt wurde.
- c. Keine Ahnung, aber es ist ein zufallsbasiertes Verfahren, das mit einer nach oben beschränkten Wahrscheinlichkeit auch falsche Ergebnisse zulässt. Vielleicht soll der Name an die Spielbanken in Monte Carlo erinnern.
- d. Im Gegensatz zum Las-Vegas-Algorithmus ist das Monte Carlo Verfahren kein stochastisches Verfahren. Das wollte man mit einer klassischen europäischen Namensgebung unterstützen.

Evolution der Ray Tracing Verfahren

- Rekursives Ray Tracing ✓
- Monte Carlo Path Tracing ✓
- Bidirectional Path Tracing
- Metropolis Light Transport
- Photon Mapping

Bidirektional Path Tracing war dann der nächste Entwicklungs- schritt



(a) Bidirectional path tracing with 25 samples per pixel



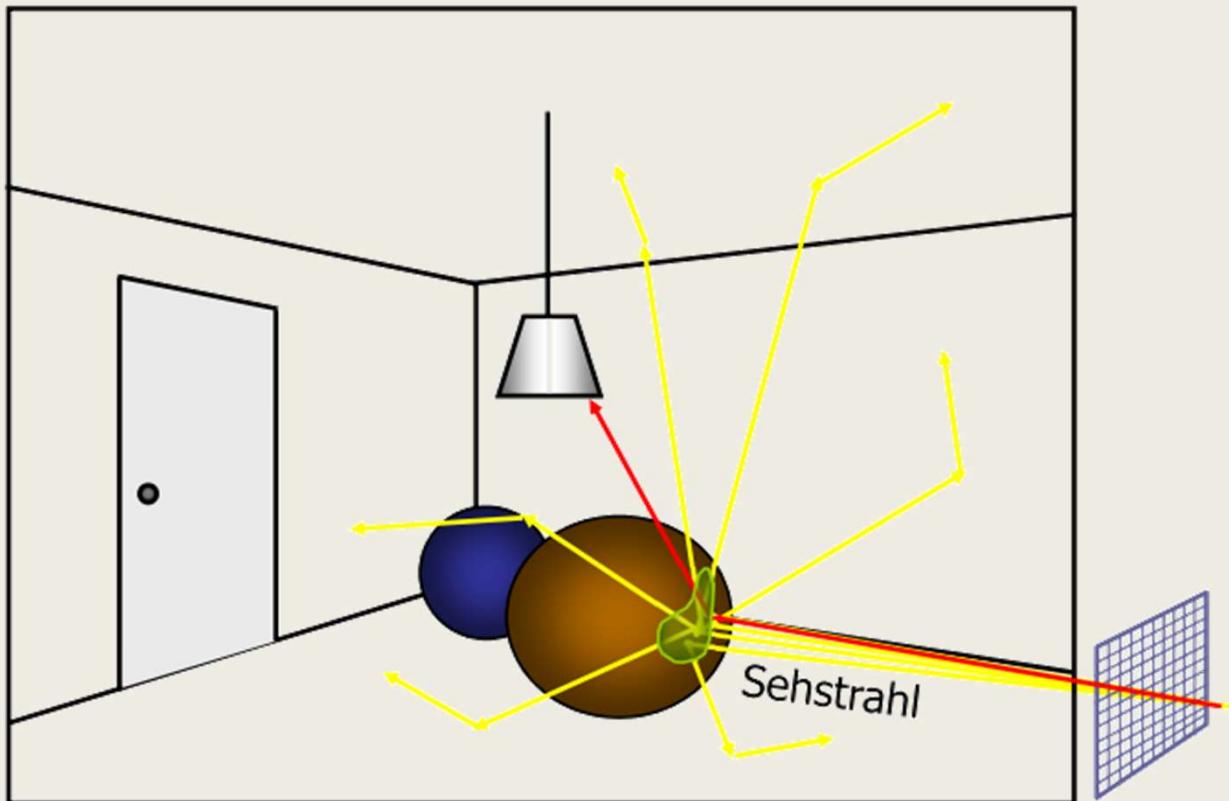
(b) Standard path tracing with 56 samples per pixel (the same computation time as (a))

Bilder von Wei-Feng Wayne Huang aus (2016): <http://rendering-memo.blogspot.com/2016/03/bidirectional-path-tracing-9-some.html>

Motivation: Bidirectional Path Tracing

Um Kaustiks besser erfassen zu können und, um das Rauschen zu unterdrücken, müssen beim Monte Carlo Path Tracing sehr, sehr viele Sehstrahlen verschickt werden.

Die Reduktion der Sehstrahlen bei Reduzierung des Rauschens und besserer Erfassung der Kaustiks ist das Ziel.



Ziel: Anzahl der Sehstrahlen reduzieren, Kaustiks besser darstellen, sowie das Rauschen unterdrücken.

Idee: Nicht nur den Sehstrahlen verfolgen sondern auch die Lichtstrahlen verfolgen.

Warum?

Weil beim alleinigen Verfolgen der Sehstrahlen (Path Tracing) ein großer Anteil der indirekten Beleuchtung nicht berücksichtigt wird. Den auch zu erfassen ist das Ziel.

Gestrichelte Linien sind die Schattenfühler

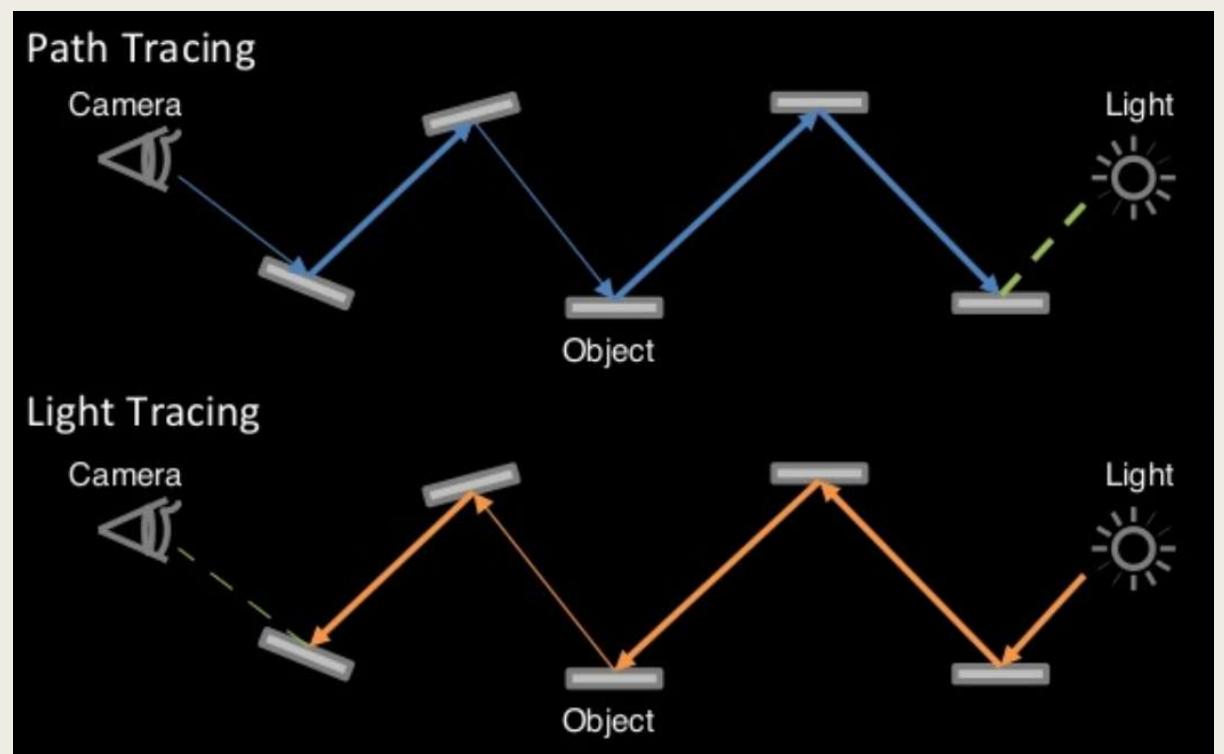
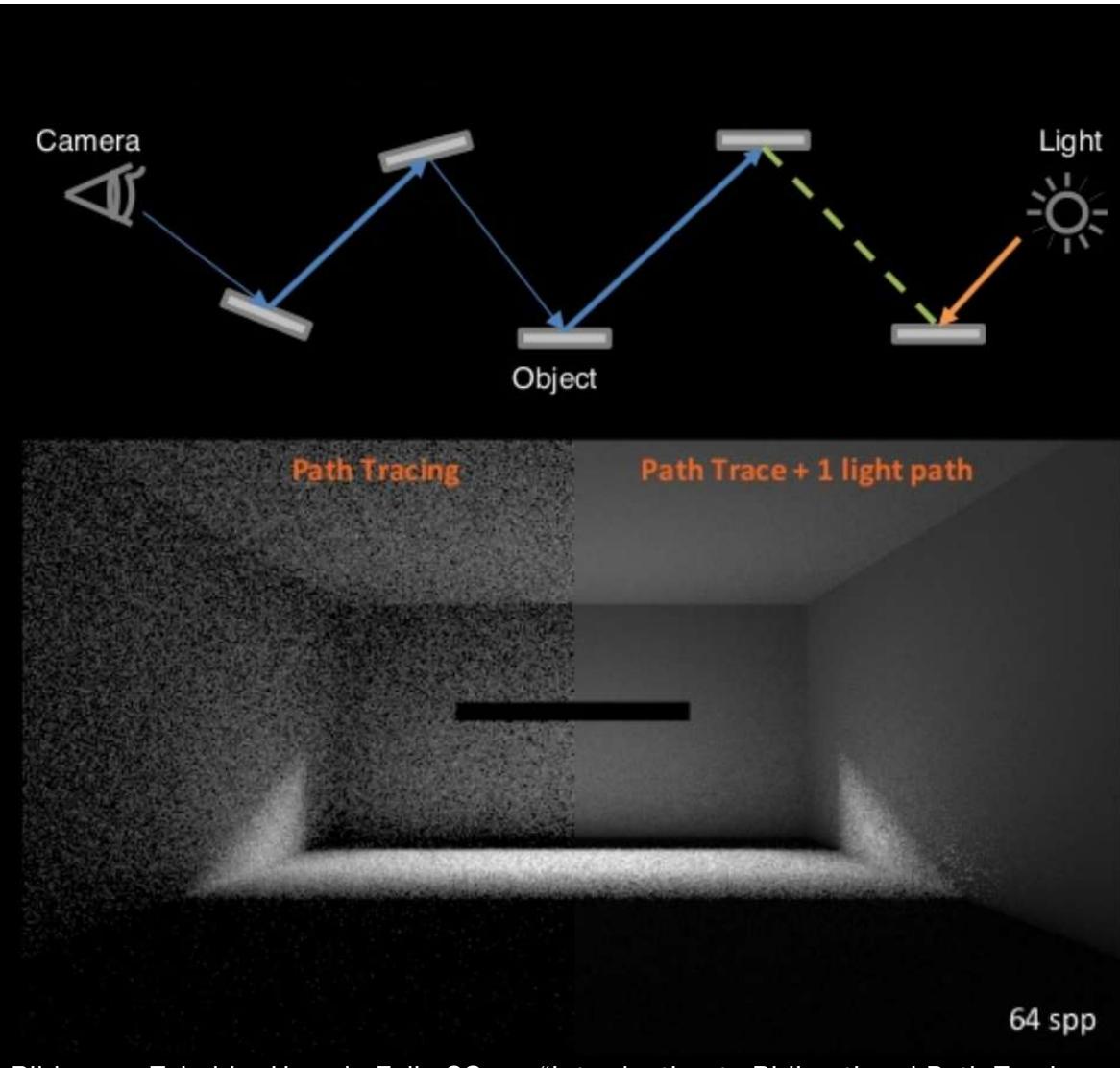


Bild von Takahiro Harada Folie 36 aus "Introduction to Bidirectional Path Tracing (BDPT) & Implementation using OpenCL (CEDEC 2015)"
<https://www.slideshare.net/takahiroharada/introduction-to-bidirectional-path-tracing-bdpt-implementation-using-opencl-cedec-2015>

Path Tracing + 1-Light Path

- Die Verfolgung von Sehstrahlen (Path Tracing) und die Verfolgung von Lichtstrahlen (Light Path Tracing) mischen.
- Im Beispiel wird jeweils ein ein-schrittweiter Light Path mit dem „Path Trace“, also dem weitergeleiteten Sehstrahl, verknüpft.
- Das Verknüpfen von Light Path mit dem Path Tracing bezeichnet man als „Bidirektional Path Tracing“.
- Bereits durch die Verknüpfung mit ein-schrittweiten Light Paths kann das Ergebnis verbessert werden. Siehe rechts. Funktioniert aber nicht immer.
- Wie kann man mehrere schrittweiten Light Path mit unterschiedlich weiten Path Tracings verknüpfen?



Bilder von Takahiro Harada Folie 38 aus "Introduction to Bidirectional Path Tracing (BDPT) & Implementation using OpenCL (CEDEC 2015)"
<https://www.slideshare.net/takahiroharada/introduction-to-bidirectional-path-tracing-bdpt-implementation-using-opencl-cedec-2015>

Bidirektional Path Tracing

Prinzip der Verknüpfung von m-Path Tracing Paths mit n-Light Paths, um die Reflexion (I) eines Sampel per Pixels (SPP) zu berechnen:

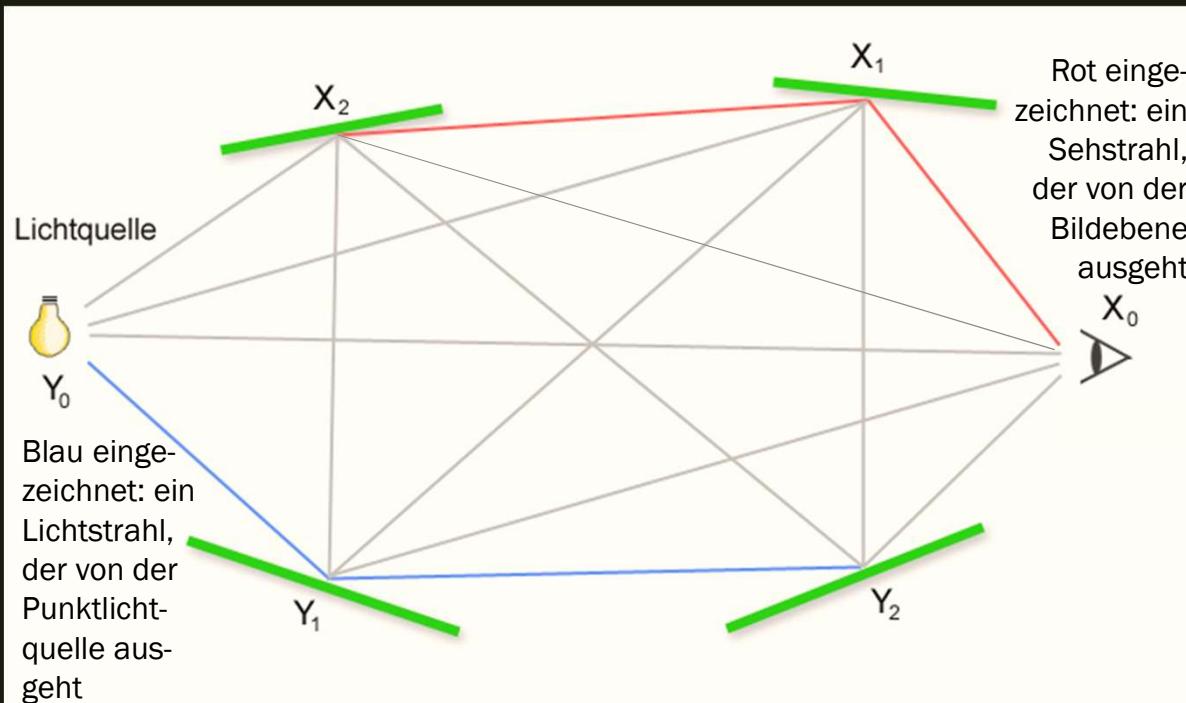
$$I = w_0 \cdot \left[\begin{array}{c} x_0 \\ \text{Speaker} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ x_k \\ \text{Sun} \end{array} \right] + w_1 \cdot \left[\begin{array}{c} x_0 \\ \text{Speaker} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ x_k \\ \text{Sun} \end{array} \right] + \dots + w_k \cdot \left[\begin{array}{c} x_0 \\ \text{Speaker} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \\ x_k \\ \text{Sun} \end{array} \right]$$

ω_i = Gewichtung einer Path Tracing-Light Path Kombination

Bilder von Takahiro Harada Folie 44 aus "Introduction to Bidirectional Path Tracing (BDPT) & Implementation using OpenCL (CEDEC 2015)"
<https://www.slideshare.net/takahiroharada/introduction-to-bidirectional-path-tracing-bdpt-implementation-using-opencl-cedec-2015>

Wie legt man die Rekursionstiefe von Light-Path und Sehstrahl fest?

Bidirectional Path Tracing

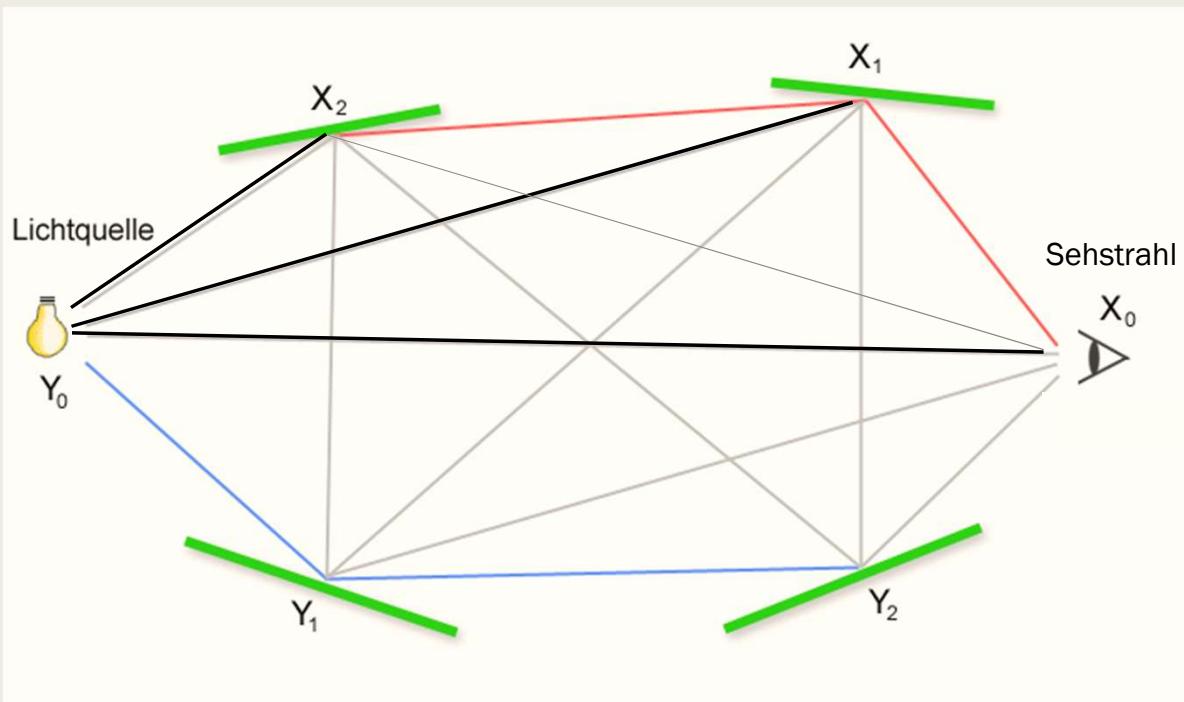


X_1 und x_2 sind die Schnittpunkte des Sehstrahls auf dem Weg zur Lichtquelle.
 Y_1 und y_2 sind die Schnittpunkte des Lichtstrahls auf dem Weg zum Betrachter.

Schema der Verknüpfung der Sehstrahlen mit den Lichtstrahlen:

- Ziel: Kontenpunkte der Sehstrahlen x_i mit Kontenpunkten der Lichtstrahlen y_j verknüpfen.
- Von **jedem** Sehstrahlknotenpunkt werden Verknüpfungen, die die Funktion eines Schattenfühlers übernehmen, zu **jedem** Lichtstrahlpunkt ausgesendet. Beobachterposition wird auch als Sehstrahlknotenpunkt betrachtet.

Bidirectional Path Tracing

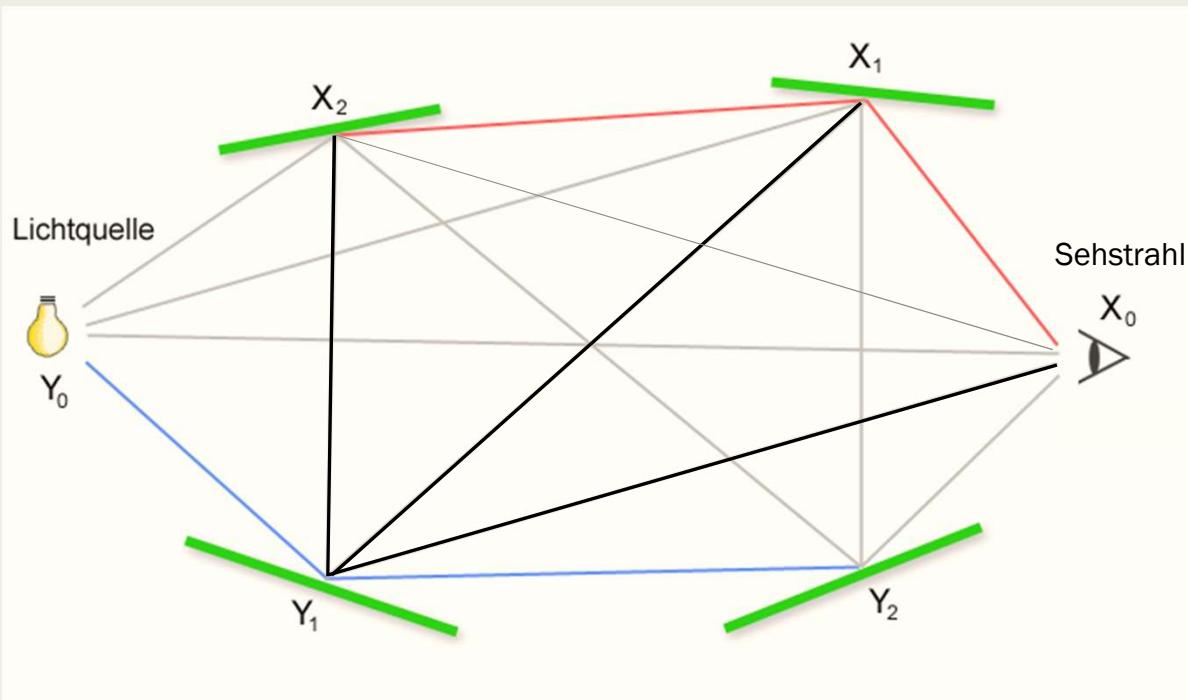


Schwarze Linien sind die Sehstrahlen

Schema der Verknüpfung der Sehstrahlen mit den Lichtstrahlen:

- Ziel: Kontenpunkte der Sehstrahlen x_i mit Kontenpunkten der Lichtstrahlen y_j verknüpfen.
- Von jedem Sehstrahlknotenpunkt werden Verknüpfungen, die die Funktion eines Schattenfühlers übernehmen, zu jedem Lichtstrahlpunkt ausgesendet. Beobachterposition wird auch als Sehstrahlknotenpunkt betrachtet.

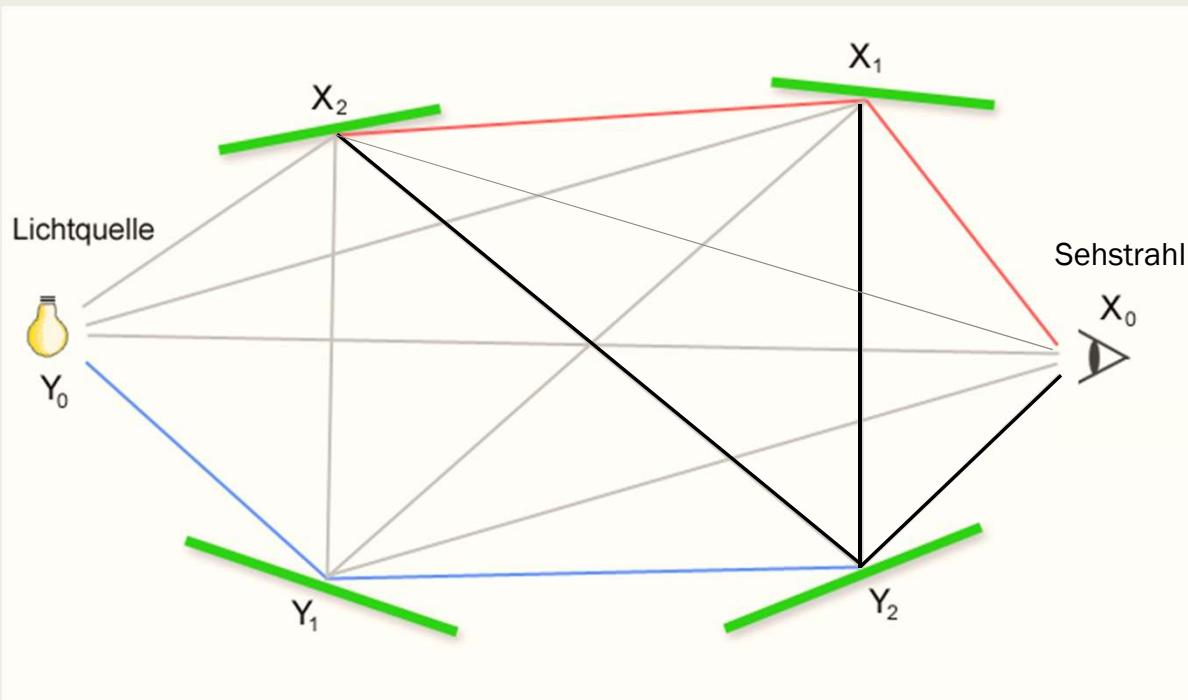
Bidirectional Path Tracing



Schema der Verknüpfung der Sehstrahlen mit den Lichtstrahlen:

- Ziel: Kontenpunkte der Sehstrahlen x_i mit Kontenpunkten der Lichtstrahlen y_j verknüpfen.
- Von **jedem** Sehstrahlknotenpunkt werden Verknüpfungen, die die Funktion eines Schattenfühlers übernehmen, zu **jedem** Lichtstrahlpunkt ausgesendet. Beobachterposition wird auch als Sehstrahlknotenpunkt betrachtet.

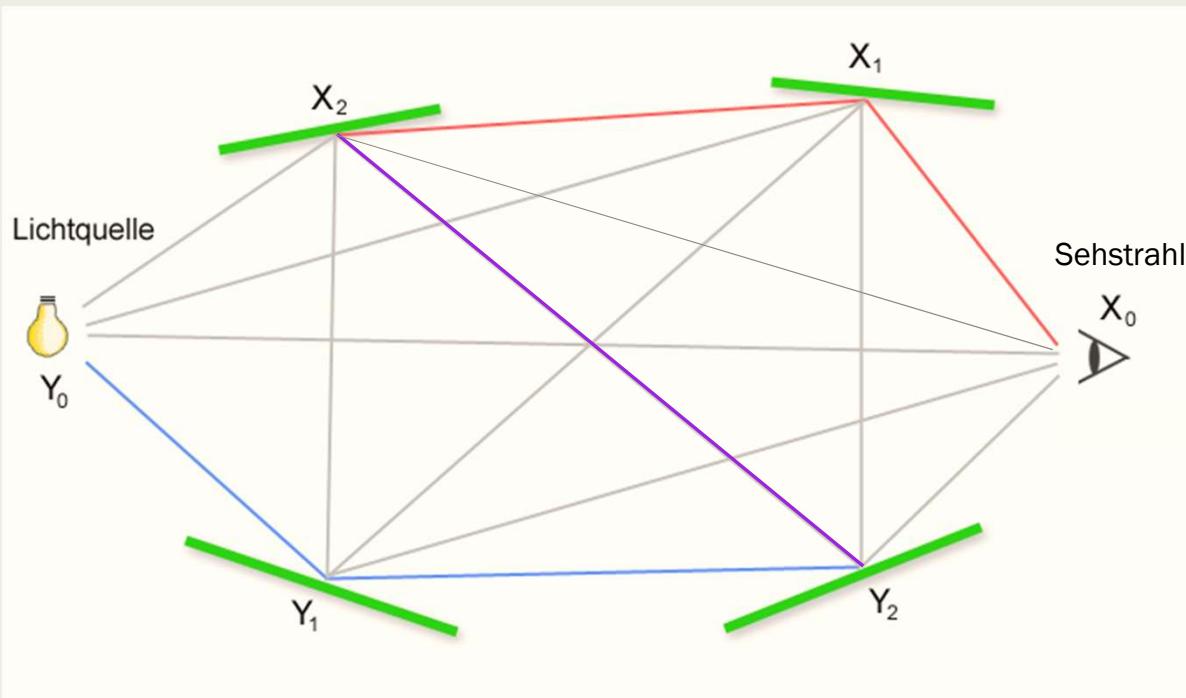
Bidirectional Path Tracing



Schema der Verknüpfung der Sehstrahlen mit den Lichtstrahlen:

- Ziel: Kontenpunkte der Sehstrahlen x_i mit Kontenpunkten der Lichtstrahlen y_j verknüpfen.
- Von jedem Sehstrahlknotenpunkt werden Verknüpfungen, die die Funktion eines Schattenfühlers übernehmen, zu jedem Lichtstrahlpunkt ausgesendet. Beobachterposition wird auch als Sehstrahlknotenpunkt betrachtet.

Bidirectional Path Tracing



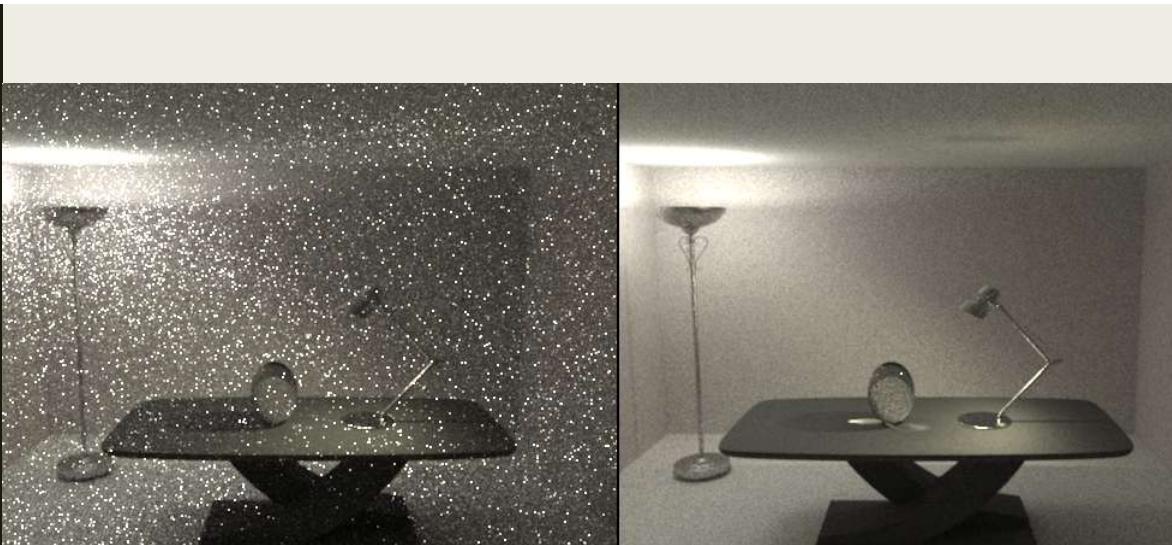
Die Reflexionsberechnungen dieses einen bunt markierten Pfads trägt in unserem Fall 1/9 zur Sample per Pixel Berechnung bei.

Schema der Verknüpfung der Sehstrahlen mit den Lichtstrahlen:

- Ziel: Kontenpunkte der Sehstrahlen x_i mit Kontenpunkten der Lichtstrahlen y_j verknüpfen.
- Von jedem Sehstrahlknotenpunkt werden Verknüpfungen, die die Funktion eines Schattenfühlers übernehmen, zu jedem Lichtstrahlpunkt ausgesendet. Beobachterposition wird auch als Sehstrahlknotenpunkt betrachtet.

Vergleich des Bidirektional Path-Tracing mit dem Path-Tracing

Bilder von Wei-Feng Wayne Huang aus (2016): <http://rendering-memo.blogspot.com/2016/03/bidirectional-path-tracing-9-some.html>



left: path tracing with 36 spp (sample per pixel)

right: bidirectional path tracing with 25 spp (around the same rendering time)



left: path tracing with 10000 spp

right: bidirectional path tracing with 6400 spp (around the same rendering time)

Zusammenfassung des Bidirectional Path Tracing

Motivation:

Kaustiks besser visualisiert zu können und eine Unterdrückung des Rauschens zu erreichen.

Prinzip:

Erweiterung der Sehstrahlverfolgung des Monte Carlo Path Tracings, um die Lichtstrahlverfolgung.

Von jedem Pixel der Bildebene ausgehend werden Sehstrahlen verschickt.

Zusätzlich werden von der Punktlichtquelle ausgehend Lichtstrahlen verschickt.

Licht und Sehstrahlen müssen miteinander verknüpft werden.

Zwei visuellen Eigenschaft machten die Erweiterung des Path Tracings, um das Light Tracing notwendig. Eine nicht – welche ist das?

- a. Um Kaustiks besser erfassen zu können
- b. Um spiegelnde Reflexionen darstellen zu können
- c. Um das Rauschen zu unterdrücken

Zwei visuellen Eigenschaft machten die Erweiterung des Path Tracings, um das Light Tracing notwendig. Eine nicht – welche ist das?

- a. Um Kaustiks besser erfassen zu können
- b. Um spiegelnde Reflexionen darstellen zu können
- c. Um das Rauschen zu unterdrücken

Warum kann man ohne Light Tracing Kaustiks schlecht darstellen?

- a. Weil man mit Path Tracing genau wie mit Ray Tracing keine Kaustiks berechnen kann.
- b. Weil ohne Light Tracing zu wenig Sehstrahlen durch die transparenten Objekte gehen, um Kaustiks erfassen zu können.
- c. Mit Bidirektional Path Tracing kann man keine Kaustiks berechnen.

Warum kann man ohne Light Tracing Kaustiks schlecht darstellen?

- a. Weil man mit Path Tracing genau wie mit Ray Tracing keine Kaustiks berechnen kann.
- b. Weil ohne Light Tracing zu wenig Sehstrahlen durch die transparenten Objekte gehen, um Kaustiks erfassen zu können.
- c. Mit Bidirektional Path Tracing kann man keine Kaustiks berechnen.

Bild aus:

A. S. Kaplanyan , J. Hanika C. Dachsbacher: „The Natural-Constraint Representation of the Path Space for Efficient Light Transport Simulation“
ACM TOG 33(4) (Proc. of SIGGRAPH 2014).



MLT berechnet in 60 Stunden

METROPOLIS LIGHT TRANSPORT (MLT)

MLT ist eine Weiterentwicklung des
Bidirectional Path Tracings

Metropolis Light Transport (MLT)

- Weiterentwicklung des Bidirectional Path Tracing.
- Anordnung der Seh- und Lichtstrahlen wird kontinuierlich optimiert.
Hierfür sind spezielle „Mutationsstrategien“ der Lichtpfade notwendig
- Abtaststrahlen steigen in Bereichen hoher Ausleuchtung an.
- Konvergiert schlecht, wenn man viele beleuchtete Durchbrüche hat.
- Kaum Vorteile in Szenen mit gleichmäßiger Ausleuchtung.
- Kaustiks und Spiegelungen können (wenn genügend Zeit und Rechenleistung vorhanden) optimal mit guter Rauschunterdrückung berechnet werden.

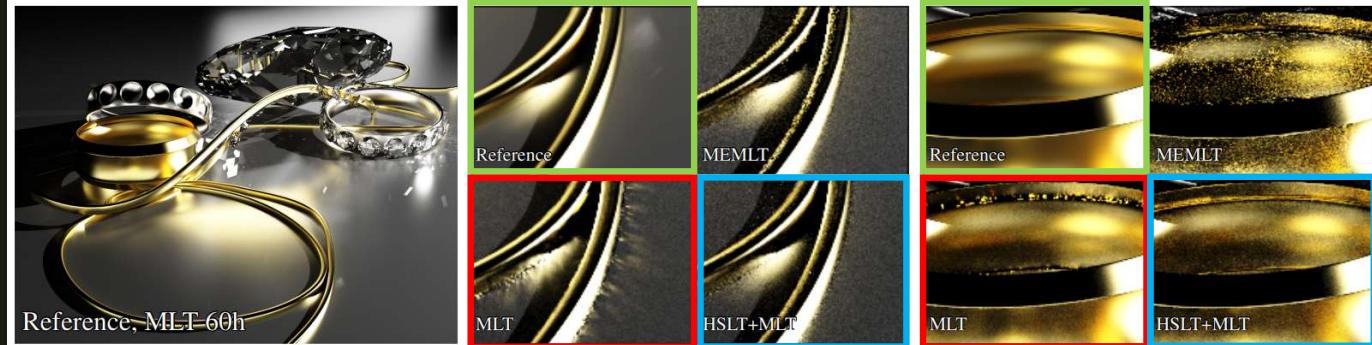


Figure 1: Equal-time comparison of the JEWELRY scene with difficult glossy paths. Left: Reference computed with the original Metropolis Light Transport (MLT) in 60 hours. The two closeups show 10 minute renders using original MLT, Manifold Exploration (MEMLT) and MLT with our proposed mutation (HSLT+MLT). Note that BDPT cannot efficiently explore indirect caustics even at high sample counts. All MLT variants used 1 million initial samples, MEMLT used $\lambda = 50$. There was no special handling of direct illumination. The average number of samples per pixel: MEMLT 621, MLT 1424, HSLT+MLT 612. Our new mutation estimates the optimal sampling density of difficult specular and glossy transport and excels at capturing hard features like thin highlights along highly curved rims.

Bild aus:

A. S. Kaplanyan , J. Hanika C. Dachsbacher: „The Natural-Constraint Representation of the Path Space for Efficient Light Transport Simulation“ ACM TOG 33(4) (Proc. of SIGGRAPH 2014).

ZIEL IST EINE ERHÖHUNG DER BERECHNUNGSGE-SCHWINDIGKEIT

Statt zuvor 60 Stunden werden nun nur noch 10 Minuten gebraucht. Allerdings müssen Qualitätseinbußen in Kauf genommen werden.

Referenzbild mit MLT mit 60 Stunden Berechnungszeit: grün

MLT allerdings nur 10 min Berechnungszeit: rot

Optimierung der Autoren mit Berechnungszeit 10 min: blau



Bild aus: https://www.ci.i.u-tokyo.ac.jp/~hachisuka/starpm2013a/Combining_PM_and_BPT_annotated.pdf

WIE GEHT ES WEITER IN DER ENT- WICKLUNG?

Das Augenmerk liegt auf den Kaustiks, der möglichst realistischen Ausleuchtung von Detail und der Rauschunterdrückung.

Links ein gerendertes Bild welches mit einer Kombination von Bidirectional Path Tracing und Photon Mapping 2013 erzeugt wurde.



VERGLEICH VON METROPOLIS LIGHT TRANSPORT & PHOTON MAPPING

Aus:

<http://forum.jotero.com/viewtopic.php?t=36&sid=d4fa321282c6b46f27d0c46b006b28e7>

Evolution der Ray Tracing Verfahren

- Ray Tracing ✓
- Monte Carlo Path Tracing ✓
- Bidirectional Path Tracing ✓
- Metropolis Light Transport ✓
- Photon Mapping

Und wie geht die Entwicklung weiter: mit Glitzer und Glitter....



Aus: „Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints“ von T. Zeltner, I. Georgiev, W. Jacob, <http://rgl.s3.eu-central-1.amazonaws.com/media/papers/Zeltner2020Specular.pdf>

COMPUTER GRAPHIK

6. Teil

RAY TRACING

Prof. Dr. Elke Hergenröther

Wichtig! Inhalte der Veranstaltung sind urheberrechtlich geschützt. Weder die Folien noch das Vorlesungsvideo dürfen an unbeteiligte Dritte weitergegeben werden.

Evolution der Ray Tracing Verfahren

- Ray Tracing ✓
- Monte Carlo Path Tracing ✓
- Bidirectional Path Tracing ✓
- Metropolis Light Transport ✓
- Photon Mapping

Photon Mapping

Motivation:

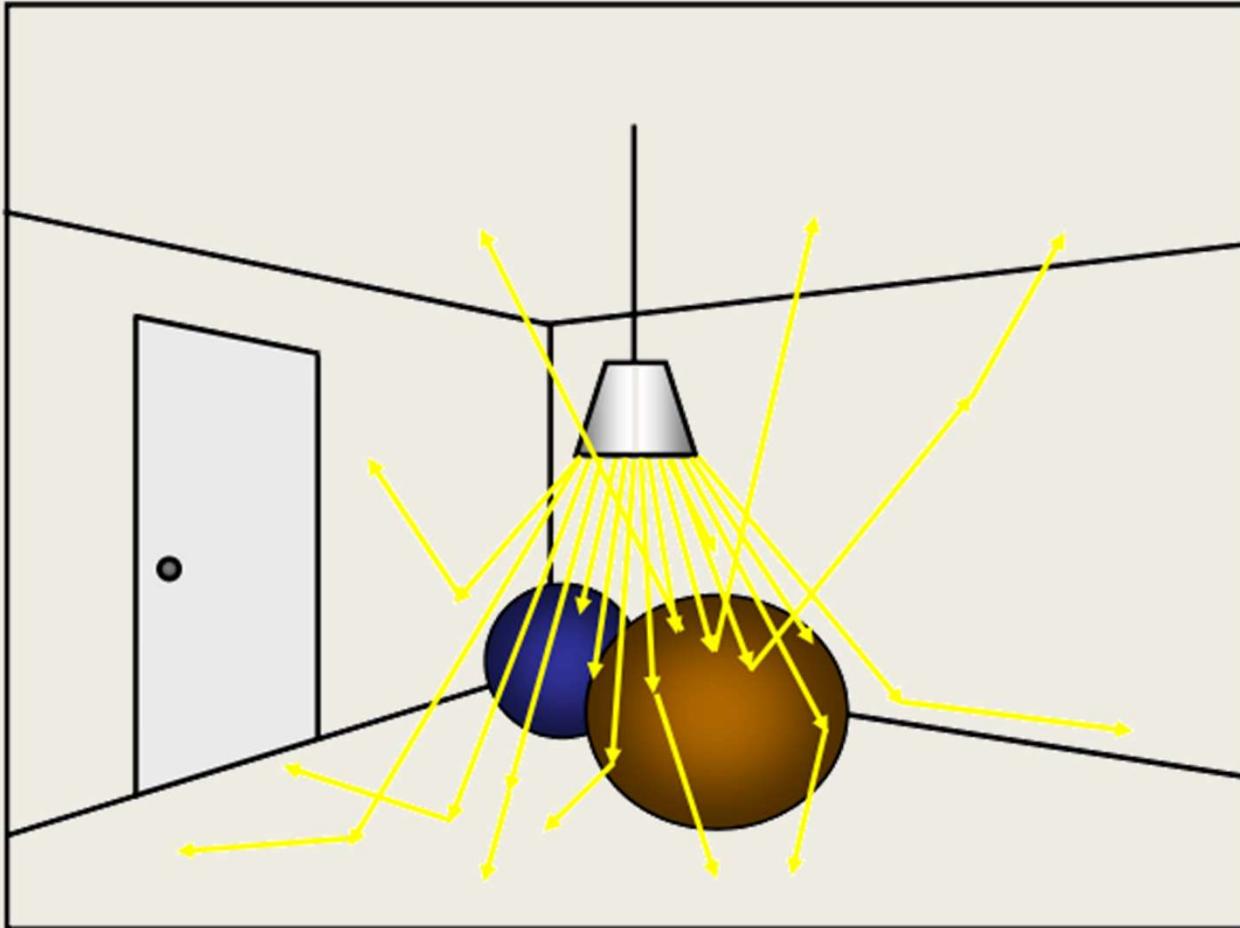
- Unterdrückung des Farbrauschens und Visualisierung der Kaustik
- Die Vorteile der Monte Carlo Path Tracing Verfahren sollen erhalten bleiben.

Prinzip:

Light Path (Photon) Tracing und Rekursives Ray Tracing arbeiten unabhängig voneinander!

Verfahren spaltete sich in zwei Teilverfahren auf:

1. **Photon Tracing:** Verfolgung der Lichtstrahlen
2. **Rekursives Ray Tracing:** Verfolgung der Sehstrahlen

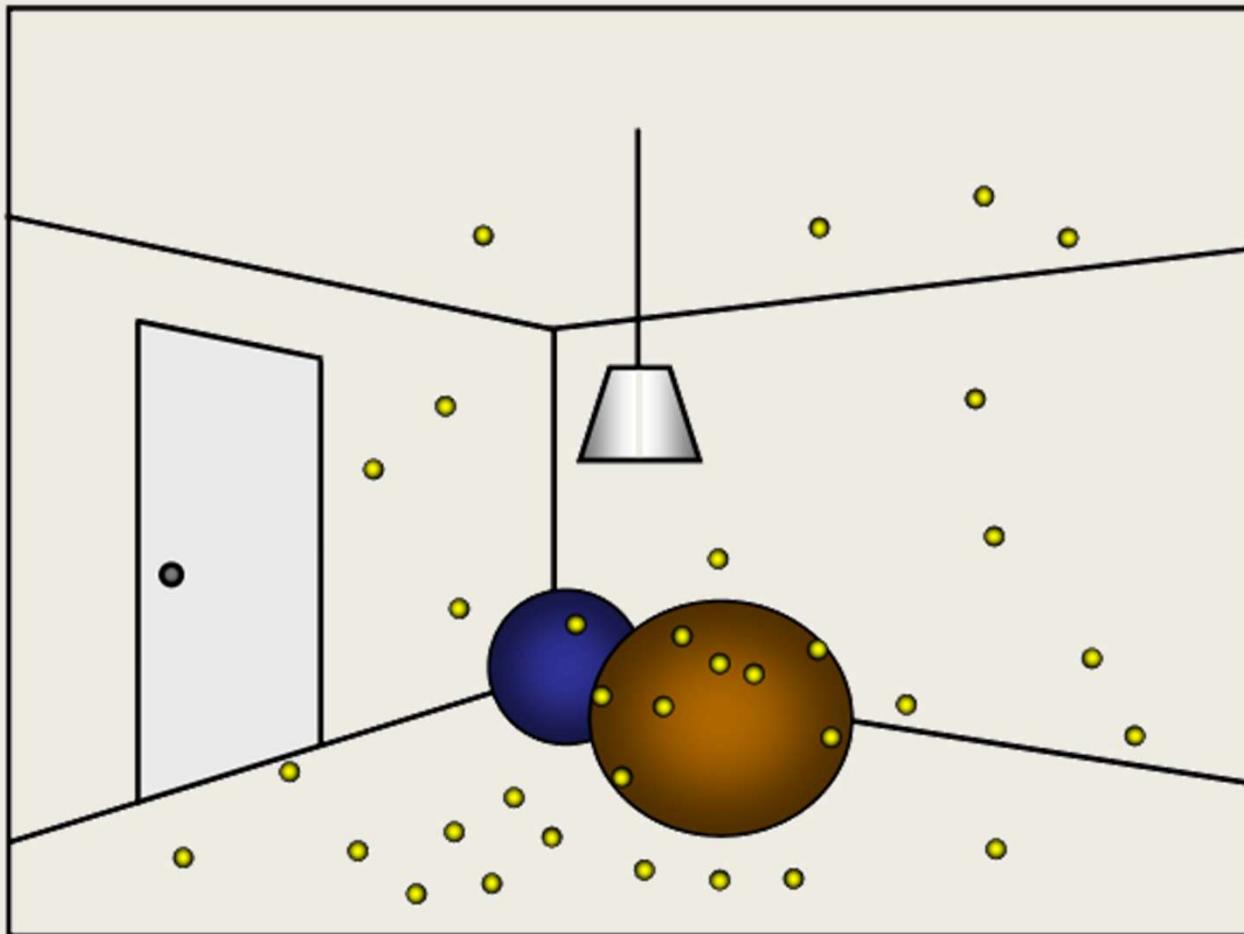


1. Teil des Photon Map Verfahrens: **Photon Tracing**

Photonen werden ausgehend von den Lichtquellen durch die Szene geschickt.

In Abhängigkeit von den Wahrscheinlichkeiten der Materialeigenschaft (BRDF) wird der Lichtstrahl zufällig:

- reflektiert,
- transmittiert oder
- diffus gestreut



Ergebnis des Photon Tracing: Die Photon Map

Photonen die auf spiegelnd reflektierenden Oberflächen auftreffen werden nicht gespeichert sondern nur weitergleitet.

Nur Photonen die auf diffuse Materialen auftreffen werden in der Photon Map gespeichert.

Photon Map
bestehend
aus 80.000
Photonen

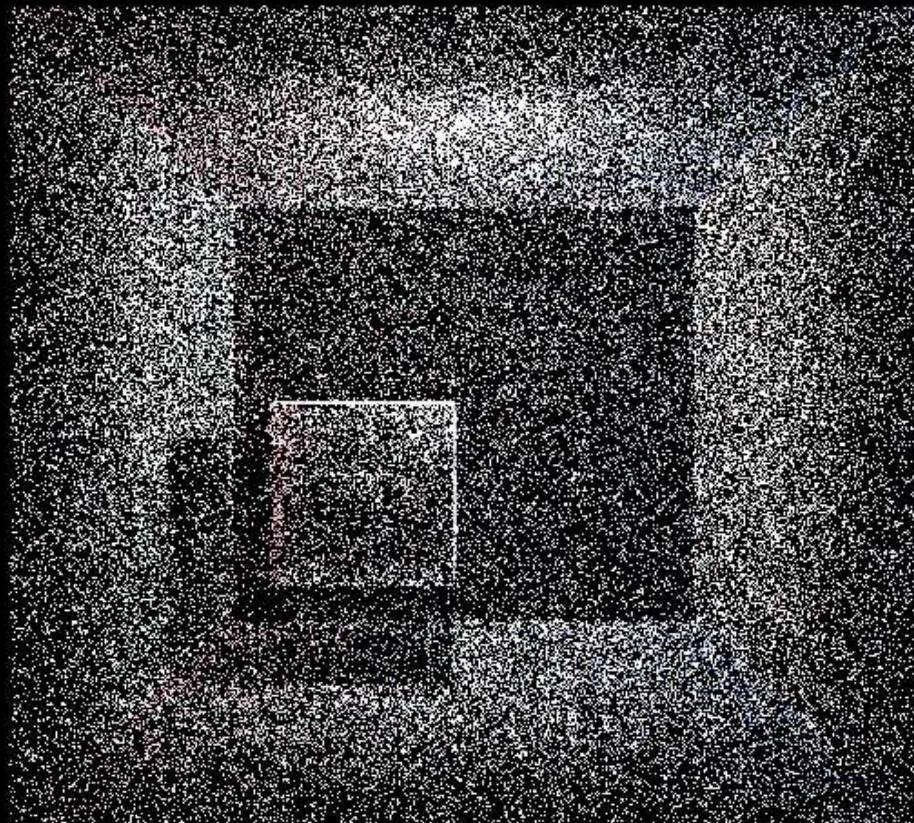


Bild: Tobias Geis

Photon Map mit
80.000
Photonen mit
geometrischem
Modell im
Hintergrund

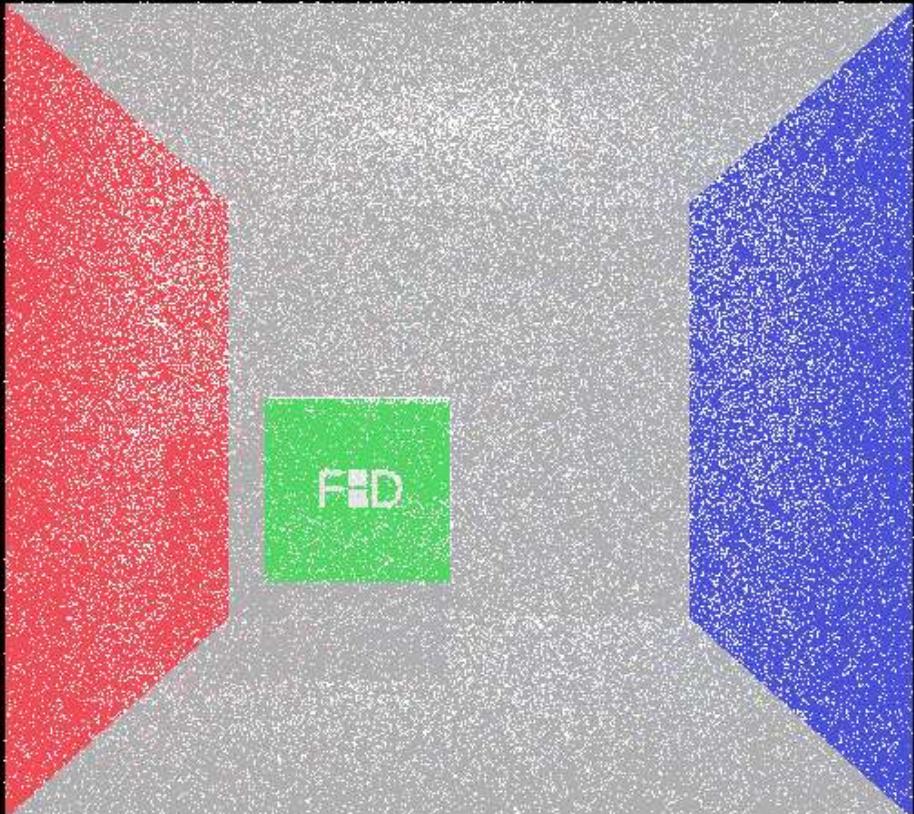
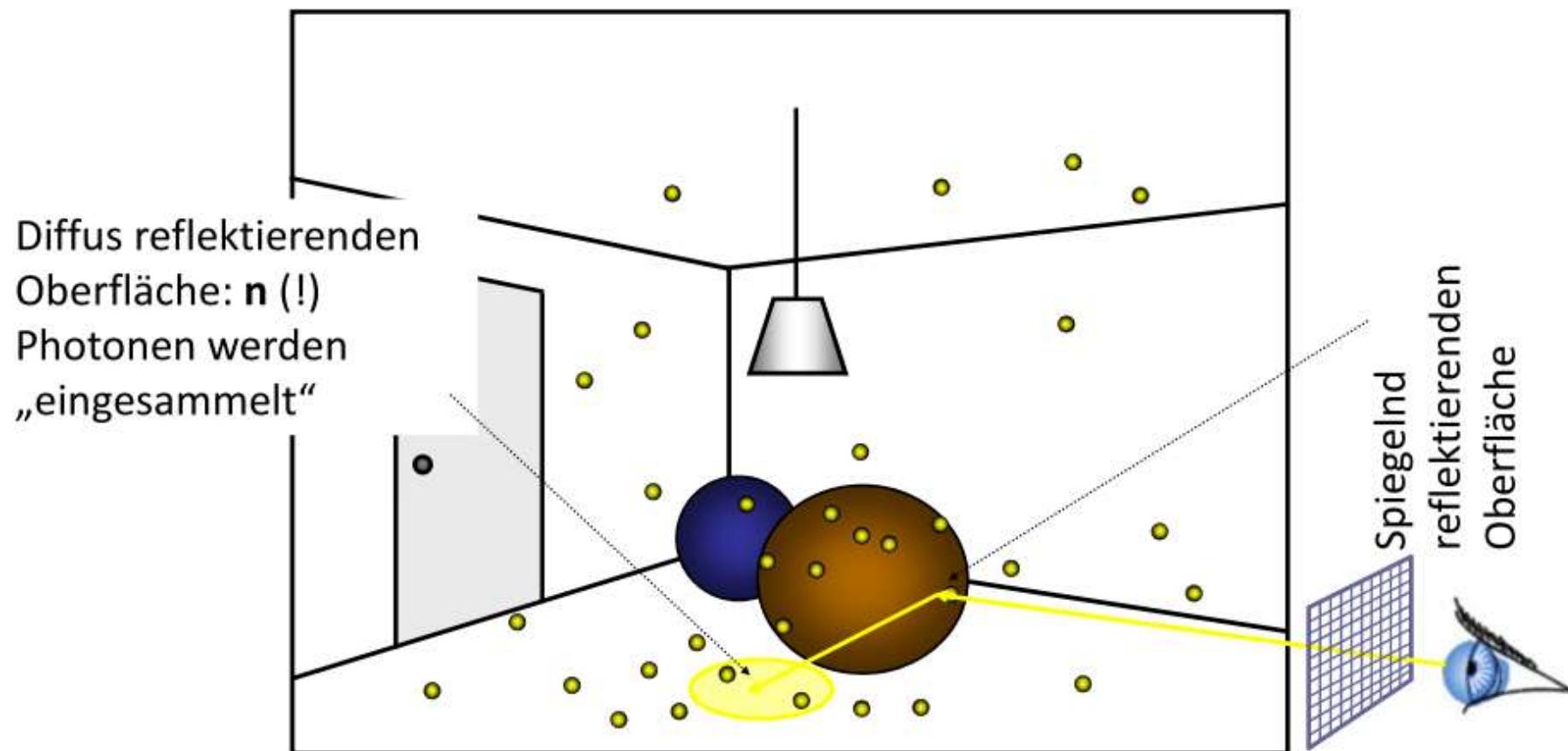


Bild: Tobias Geis

2. Teil des Photon Map Verfahrens: Ray Tracing





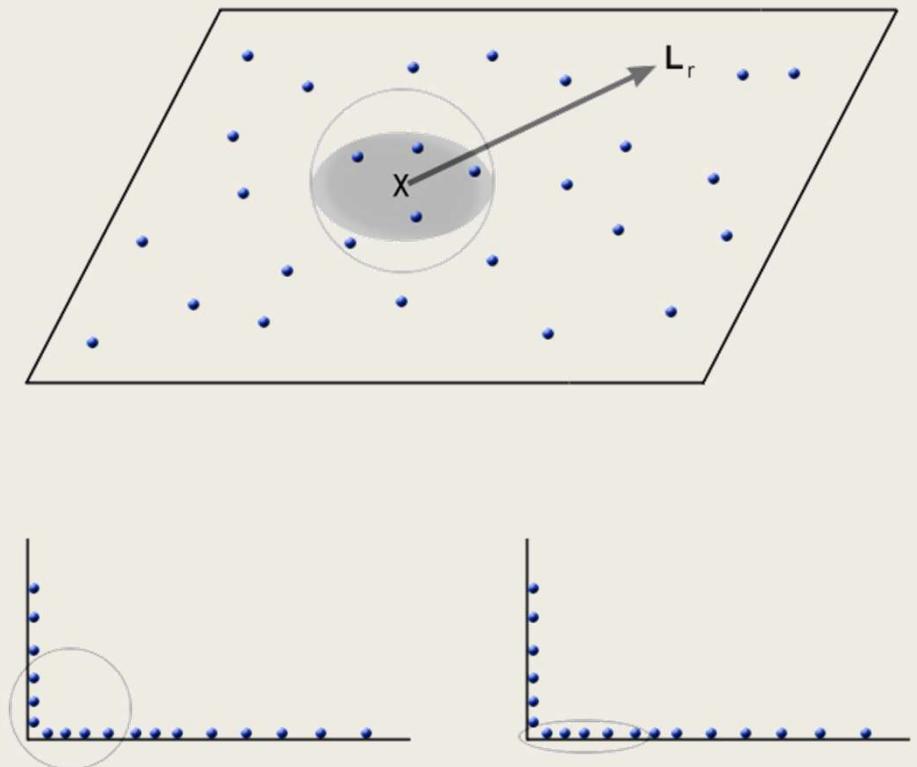
Integration der Photonen mit einer Kugel

Prof. Dr. Elke Hergenröther

Integration der Photonen

Detailansicht: Eine expandierende Kugel am Punkt in dem der Sehstrahl auf die Objektoberfläche auftrifft. Die Kugel ist dazu da, um eine festgelegte Photonenzahl auf zu sammeln.

Durch die Kugelform entstehen Fehler im Randbereich. Den Fehler reduziert man durch Verwendung eines Ellipsoiden statt einer Kugel.



Bilder: „Photon Mapping“ von Hendrik van Jensen

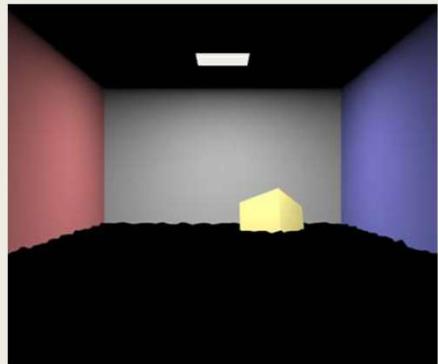


Integration der Photonen mit einem Ellipsoiden

Prof. Dr. Elke Hergenröther

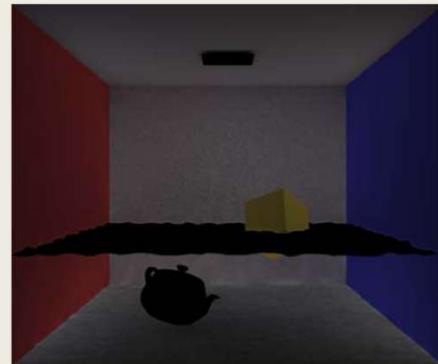
Ergebnis Photon Mapping

... nutzt die Additivität des RGB-Farbsystems aus:



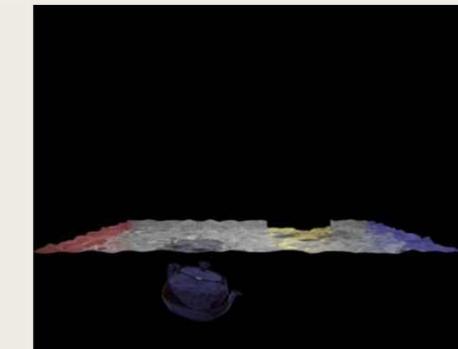
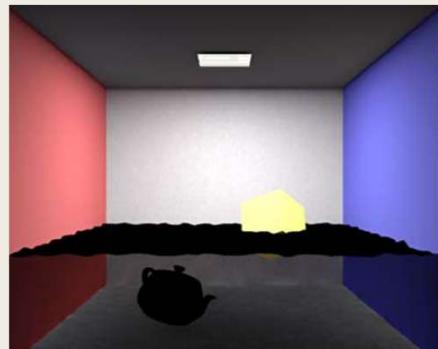
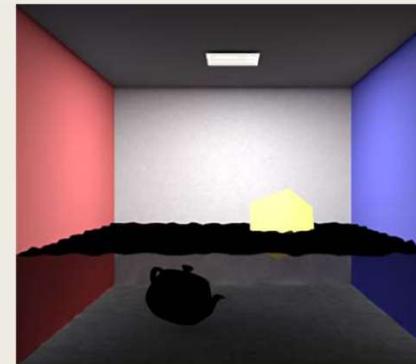
direkte diffuse Beleuchtung (RT)

+

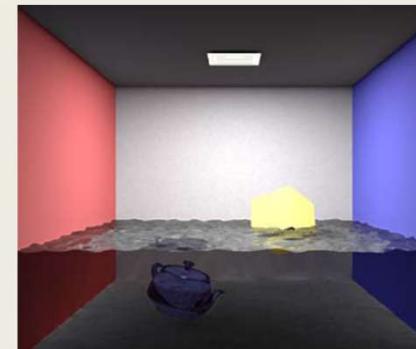


indirekte Beleuchtung (Photon Map)

=

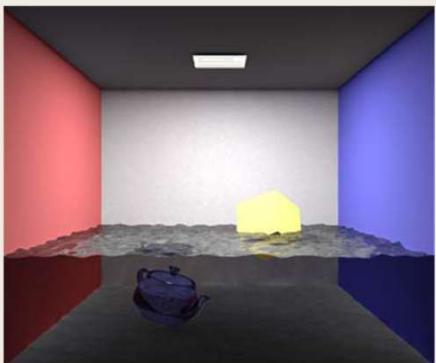


+

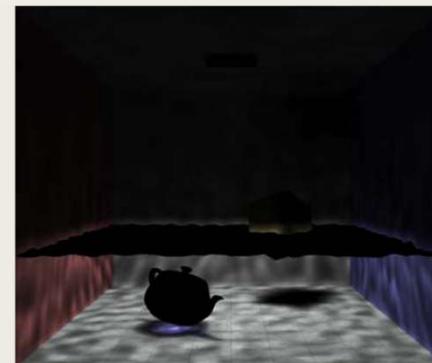


spiegelnde Reflexion (RT)

Ergebnis Photon Mapping

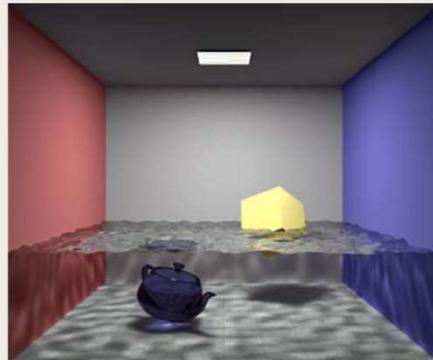


+

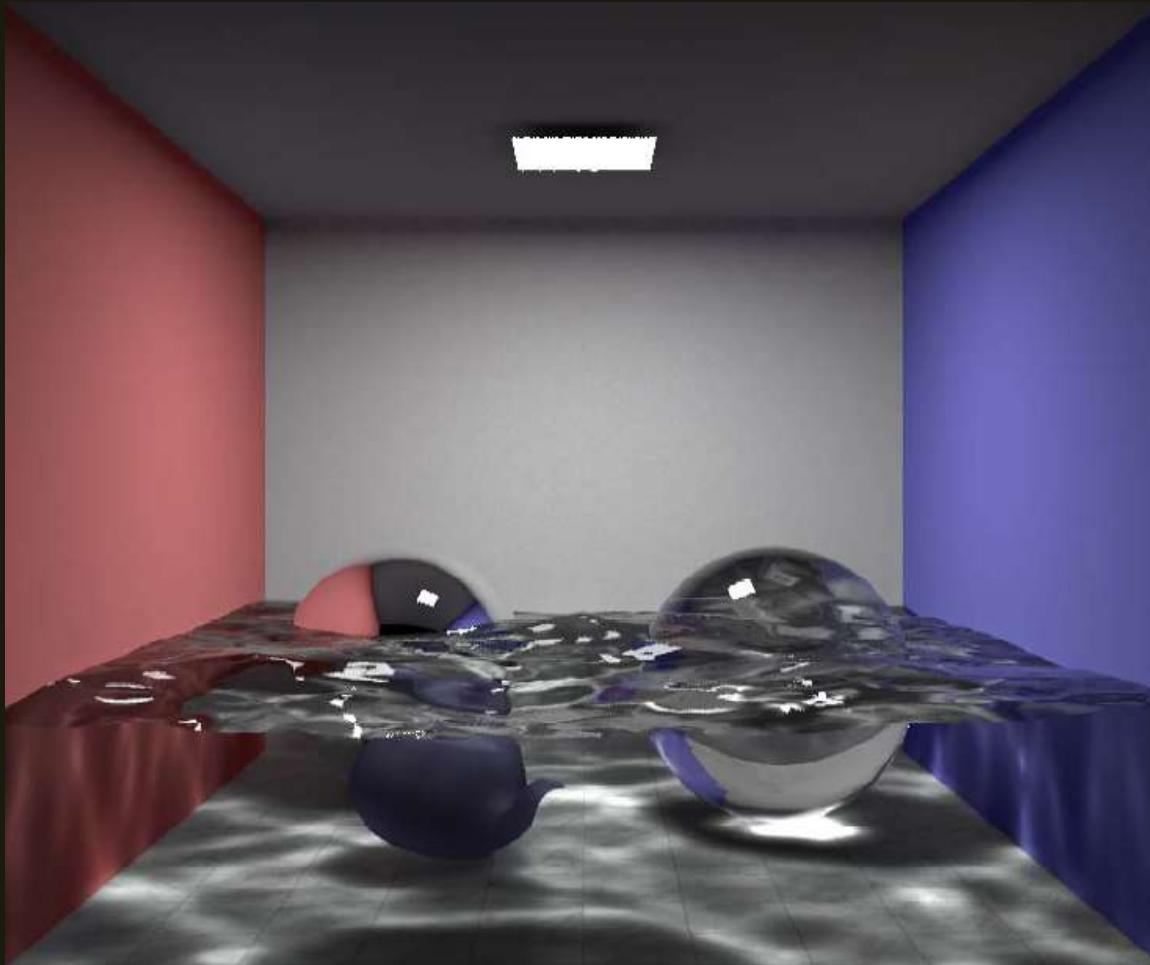


Kaustiks (extra Photon Map für Lichtbündelungen)

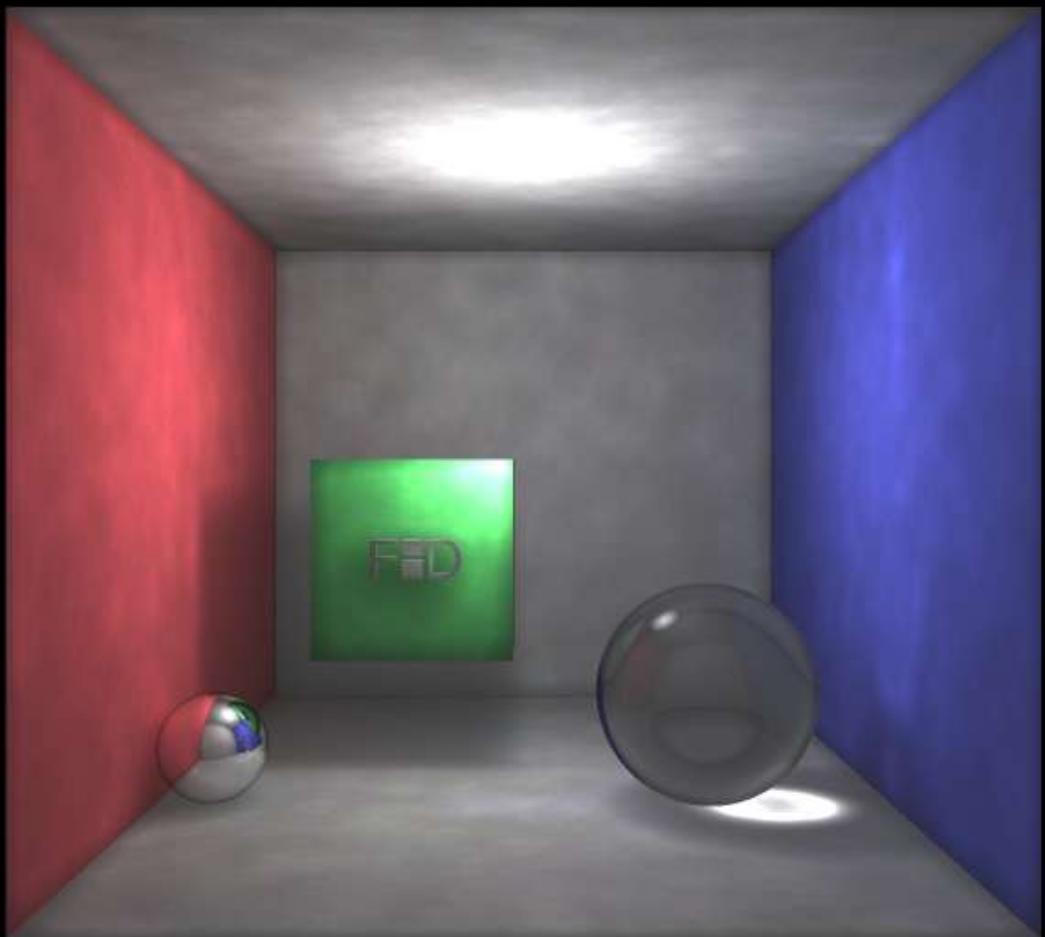
=



Ergebnis Photon Mapping



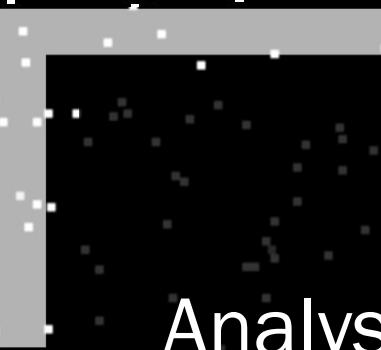
vom Erfinder des Photon
Mappings Hendrik von Jensen



Photon Mapp mit 500.000 Photonen, Bild von Tobias Geis

Und bei Tobias
Geis nach all
der Mühe ...
immer noch
Wolken.

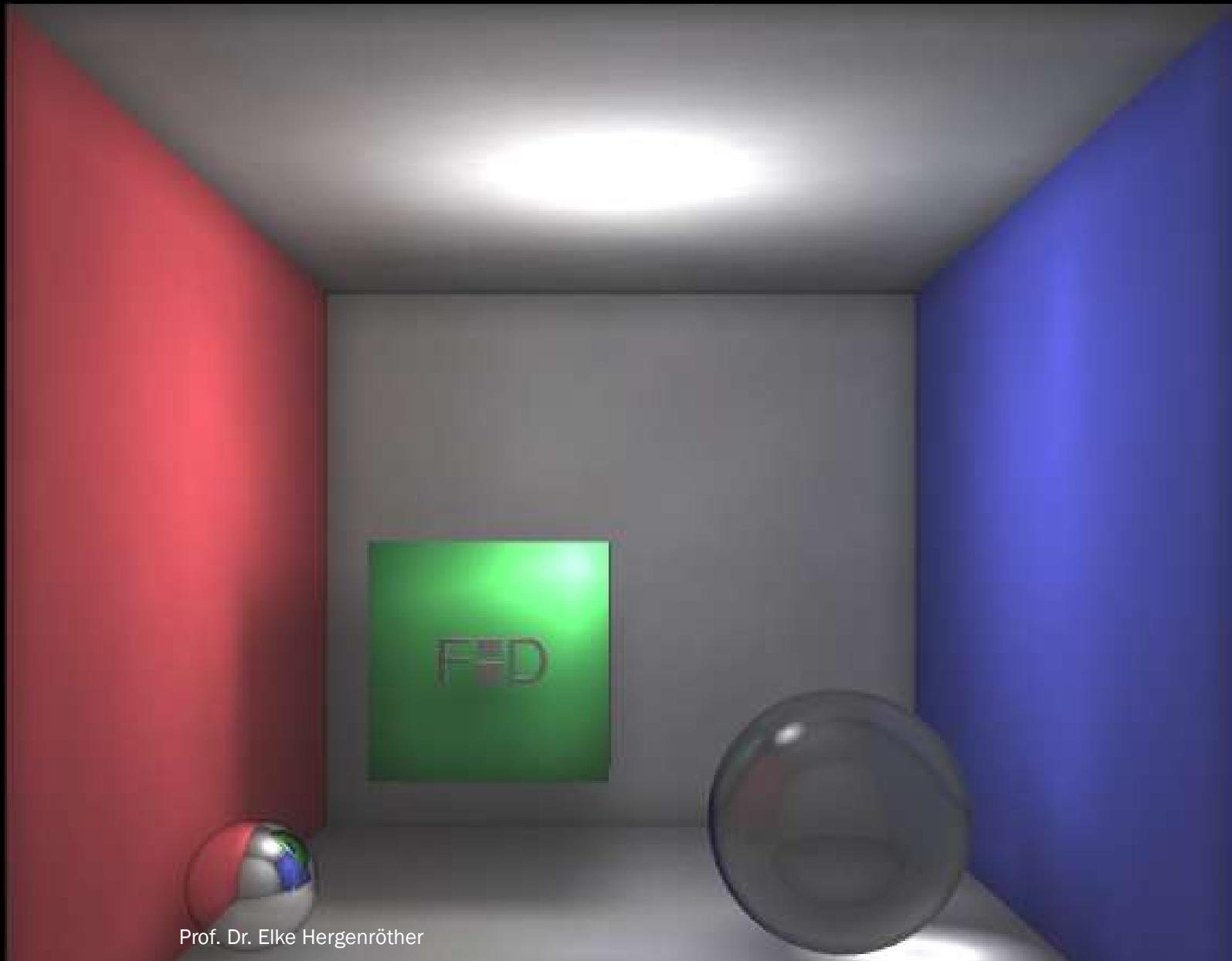
Woran liegt es?



Analyse der Photon Map:
Photonen sind nicht
gleichmäßig verteilt!

Nach
Anwendung
des
Korrektur-
verfahrens

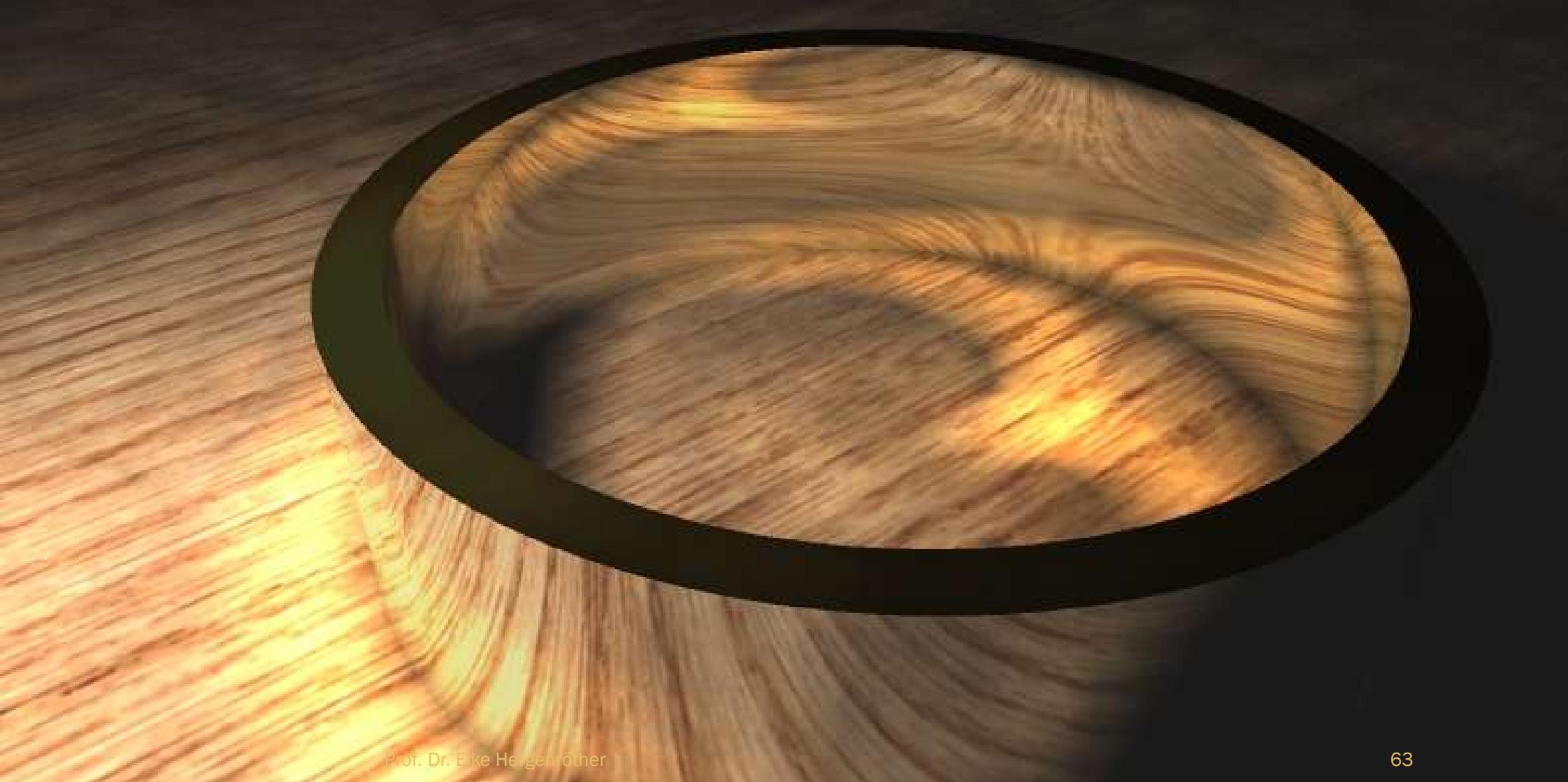
Photon Map
enthält
500.000
Photonen



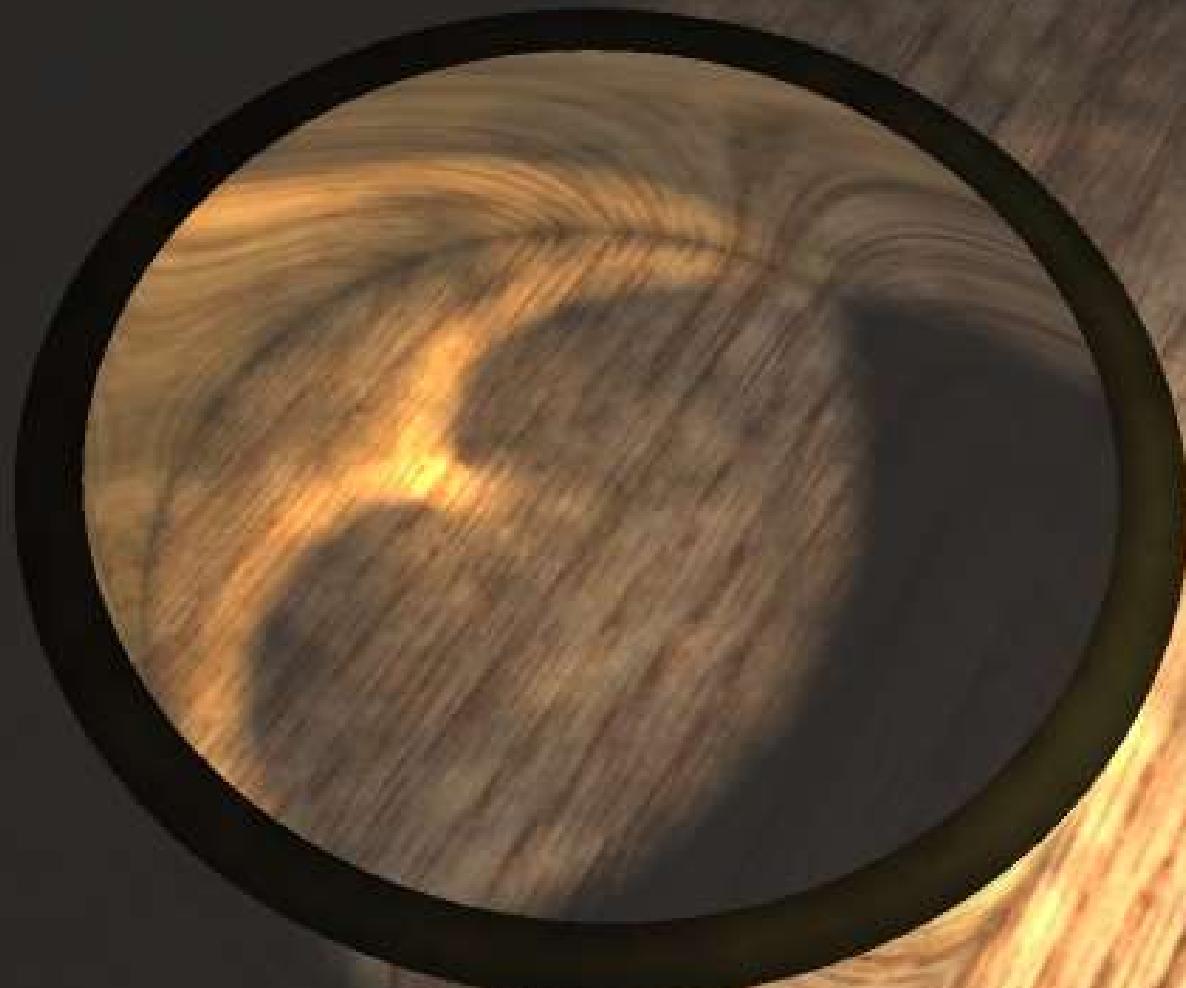
Ergebnis Photon Mapping von Hendrik van Jensen



Ergebnis Photon Mapping von Tobias Geis



Ergebnis Photon Mapping von Tobias Geis



Und wie geht die Entwicklung weiter: mit Glitzer und Glitter....



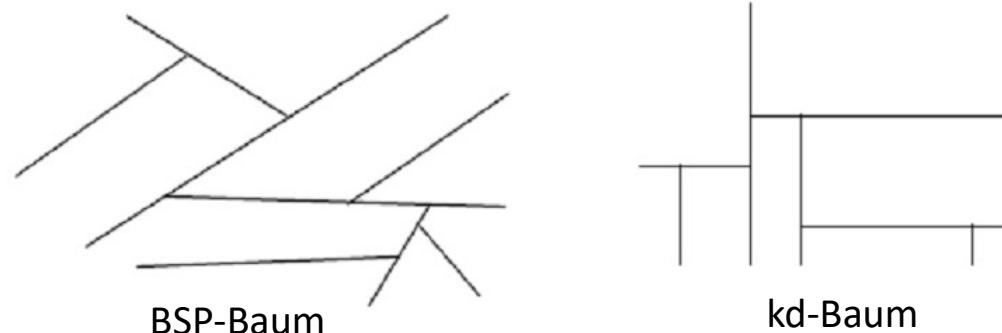
Aus: „Specular Manifold Sampling for Rendering High-Frequency Caustics and Glints“ von T. Zeltner, I. Georgiev, W. Jacob, <http://rgl.s3.eu-central-1.amazonaws.com/media/papers/Zeltner2020Specular.pdf>

Weitere Themen zu Ray Tracing

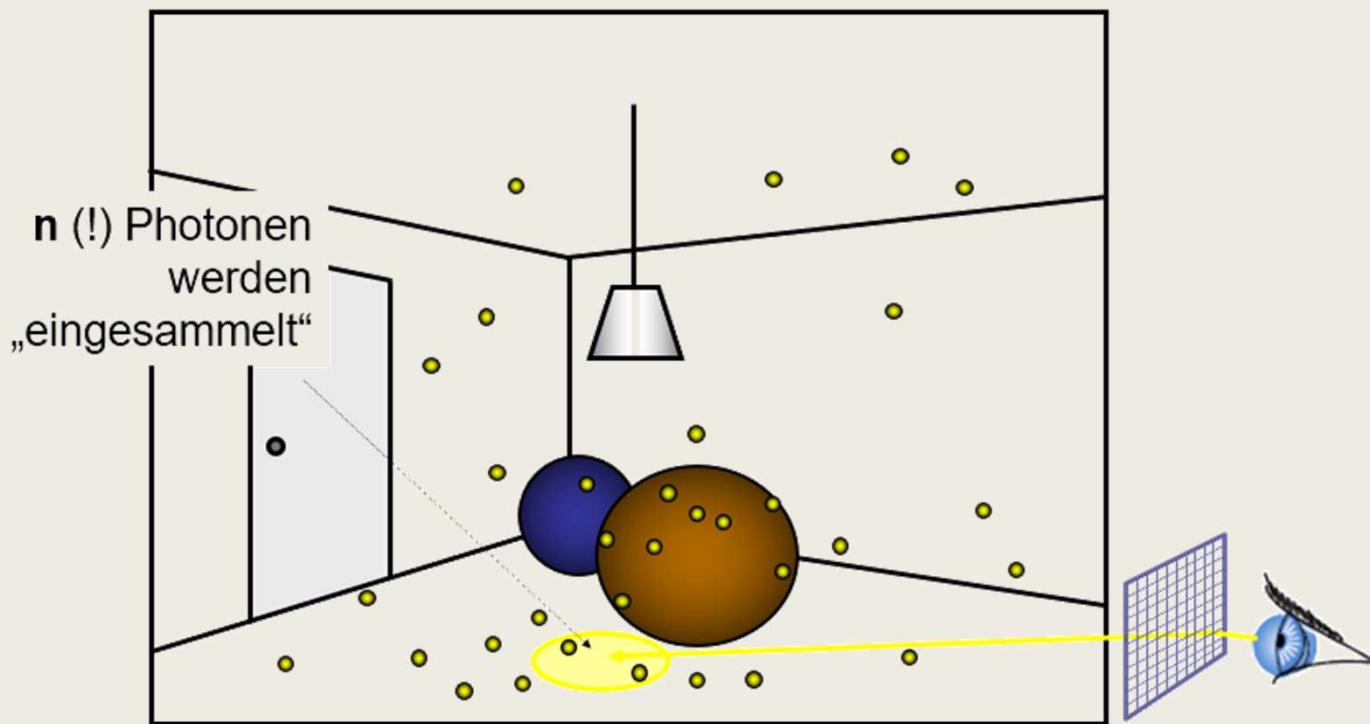
- Optimierungsstrukturen (**kd-Baum**, Octree, etc.)
- Super-Resolution – ein Verfahren der KI
- Deep Photon Mapping, 2020 von Hendrik van Jensen
- Material Definition Language (MDL) Standard von NVIDIA zur Beschreibung von Materialoberflächen – geeignet für Rasterisierung und Ray Tracing
- NVIDIA OptiX™ Ry Tracing Engine (<https://developer.nvidia.com/optix>)

Optimierung: BSP- Baum und kd-Baum

- Verfahren zur Verwaltung räumlicher Daten
- Binäre hierarchische Unterteilung
 - Szene wird rekursiv in zwei Teilräume geteilt
 - Teilung erfolgt durch eine orientierte Ebene
 - *BSP-Baum (Binary Space Partition):*
 - Teilung erfolgt durch eine beliebige Ebene im Raum
 - *kd-Baum (k dimensional)*
 - Teilung erfolgt durch achsenparallel Ebenen im Raum



Motivation für den kd-Baum:



kd-Baum

Aus: Alexander Polansky: „Elektronisches Interaktives Skriptum“, Diplomarbeit am Institut für Computergrafik Technische Universität Wien, 1996

Original Literatur: Bentley u.a. (1975) in den Communications of the ACM.

Aufgabe: Es sollen Punkte P_i des k -dimensionalen Raums gespeichert werden.
Jeder Punkt ist durch seine Koordinaten x_1, x_2, \dots, x_k gegeben.
Suchaufgaben sollen effizient durchgeführt werden können.

Der Baum: kd-Baum ist ein Suchbaum und kein Blattsuchbaum (siehe Ottmann und Widmayer „Algorithmen und Datenstrukturen“). D.h. Informationen sind in den Knoten und nicht in den Blättern gespeichert.

Jeder Knoten des Baums enthält

- einen Punkt P (Koordinaten)
- eine Dimensionsangabe j
- 2 (oder mehr) Zeiger auf Nachfolger

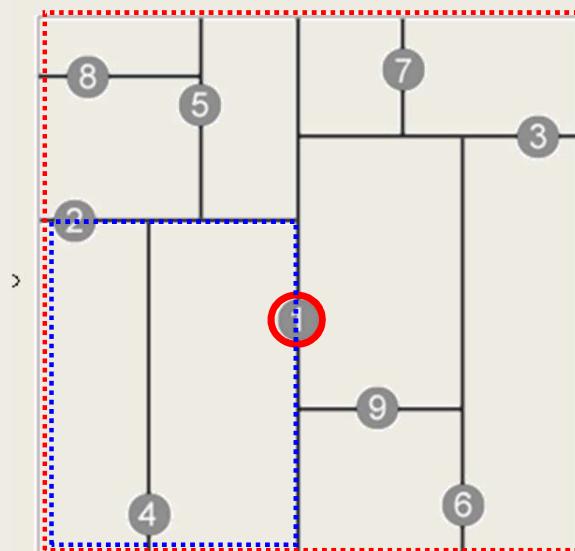
Aufbau des kd-Baum

Der Beispiel-Knoten enthält die folgenden Informationen:

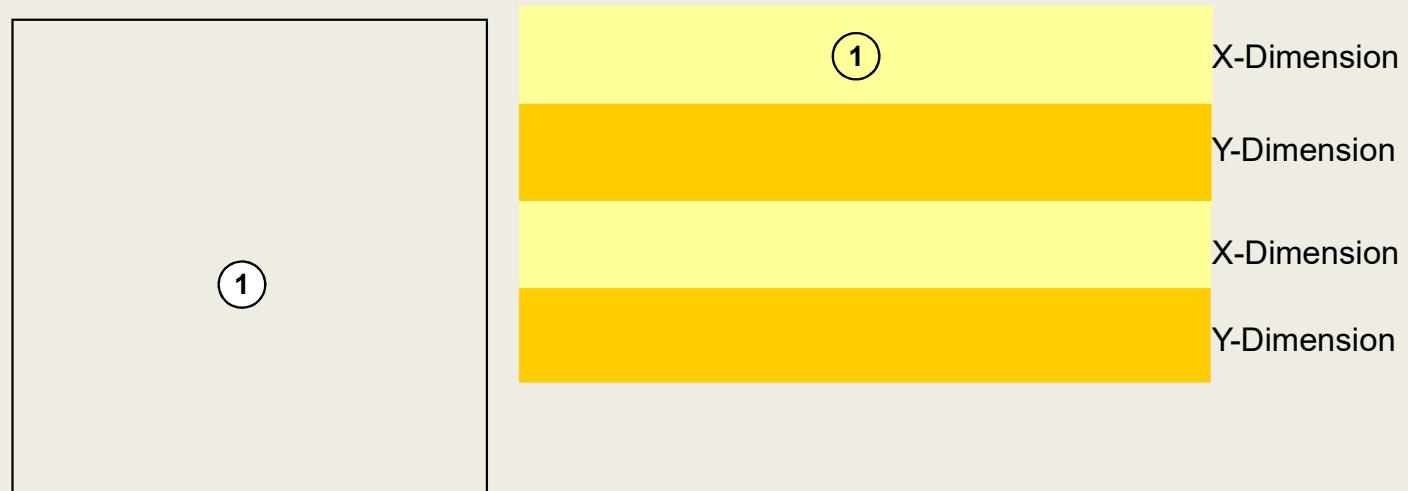
- Position des Punktes P:
 $P.x, P.y, P.z$
- Dimensionsangabe: In 2D ist dies bspw. x oder y
- Wenn die Dimension X ist, dann enthält der linke Unterbaum alle Punkte deren x-Komponente $< P.x$ ist und der rechte Unterbaum alle anderen.

Bound für Punkt 4
Bound für die Wurzel

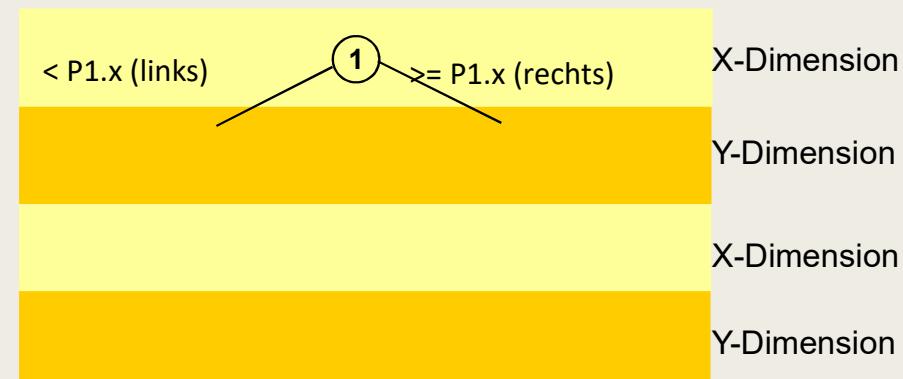
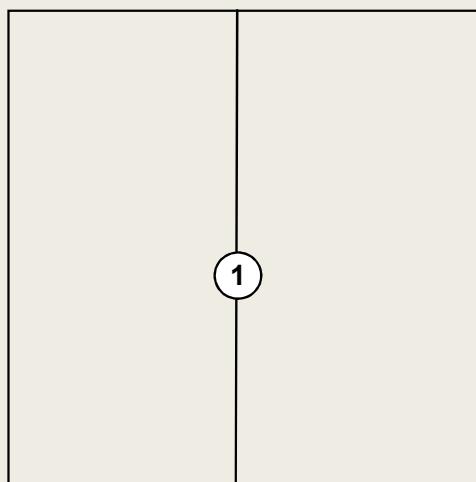
Jeder Knoten des kd-Baums ist also die Wurzel eines Teilbaums, der alle Punkte eines bestimmten Quaders enthält: **bound (Grenze)**.



Erstellen eines kd-Baums



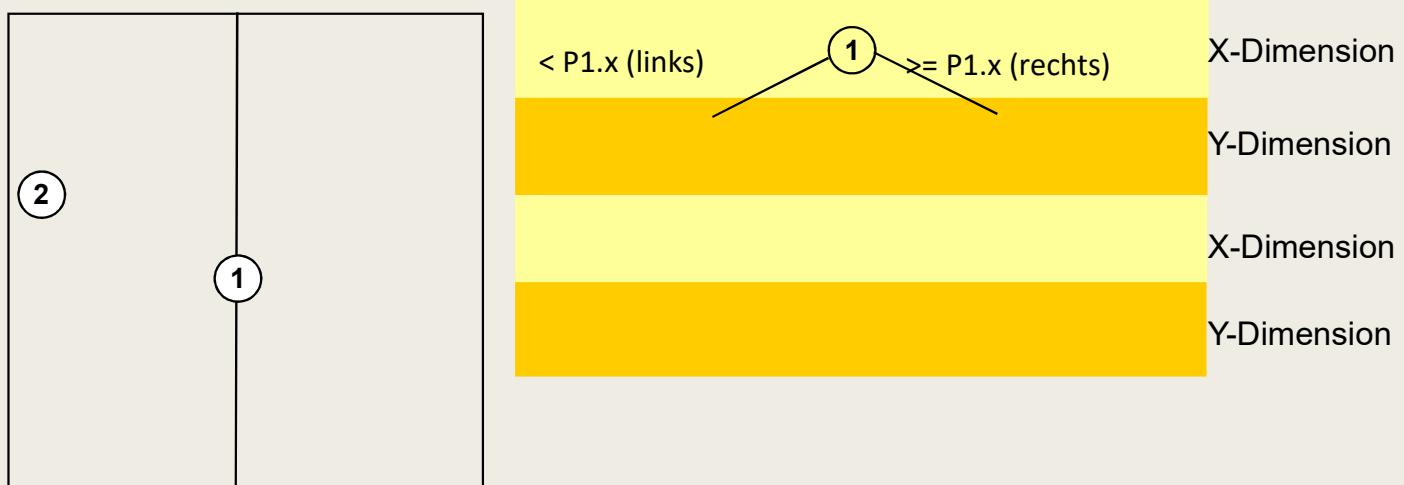
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

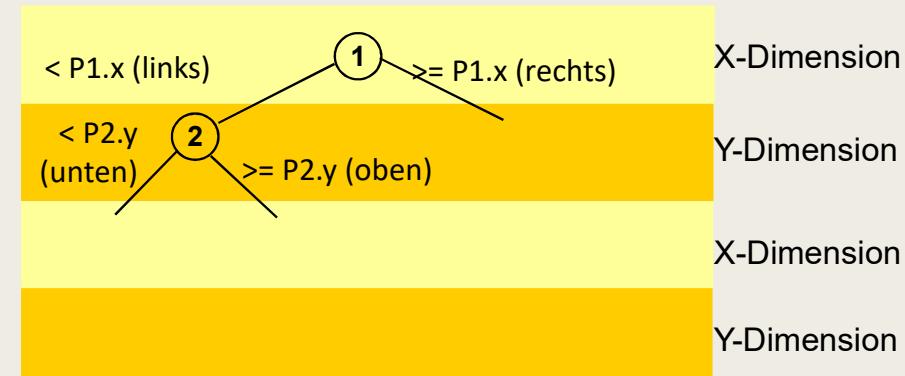
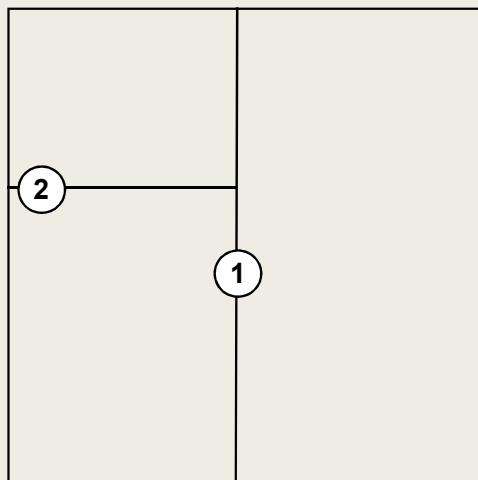
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

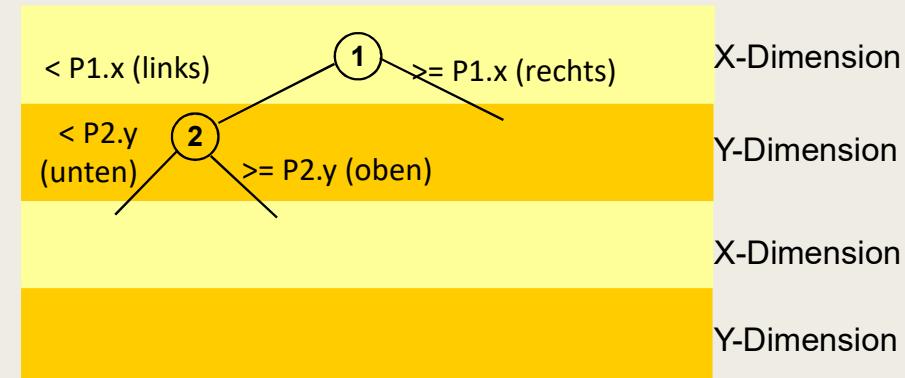
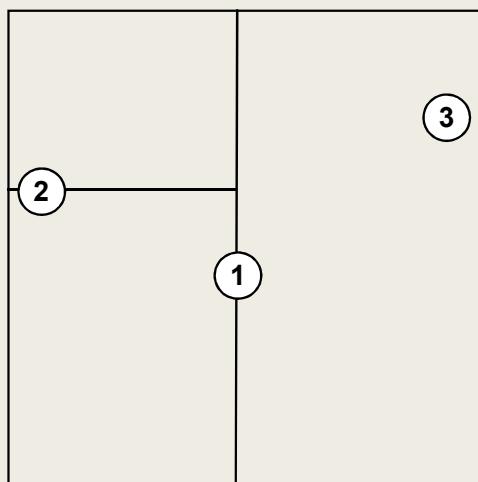
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

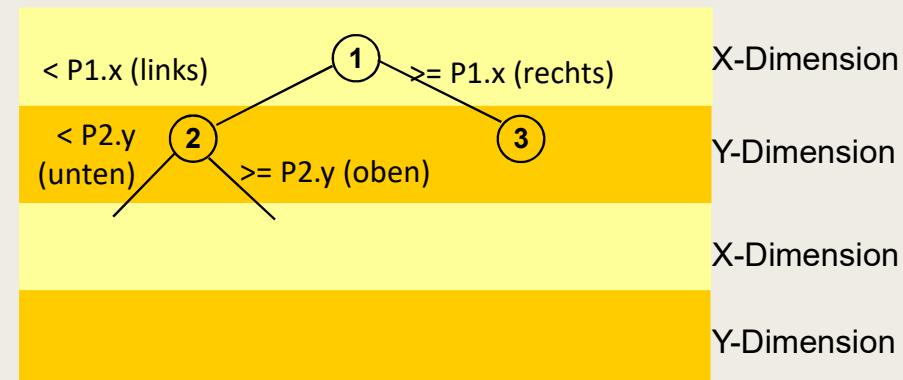
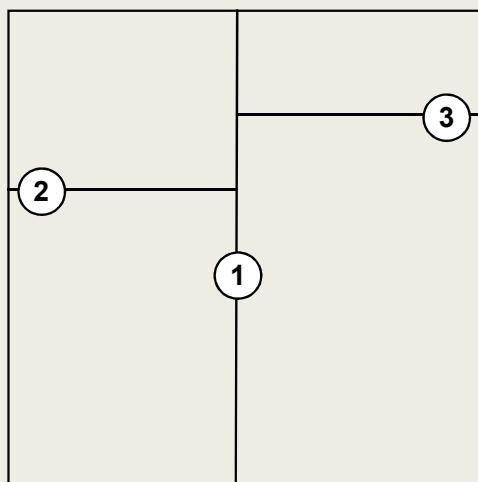
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

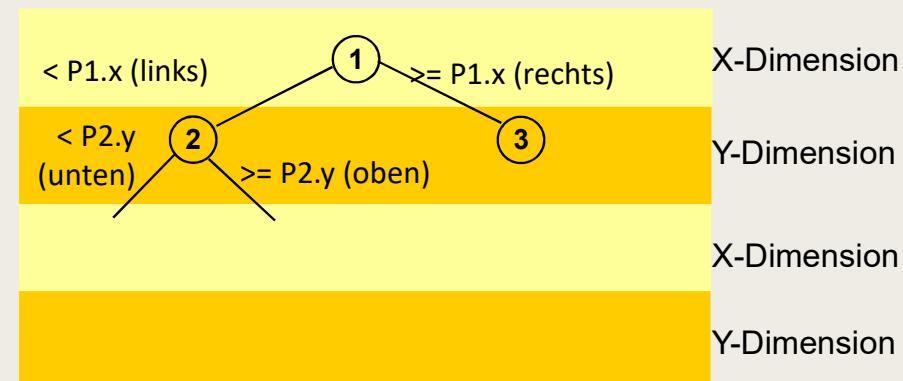
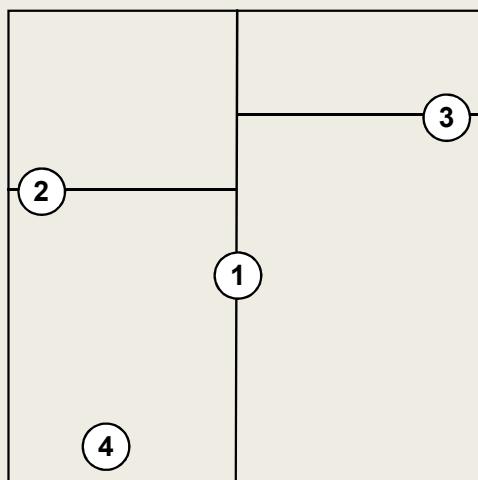
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

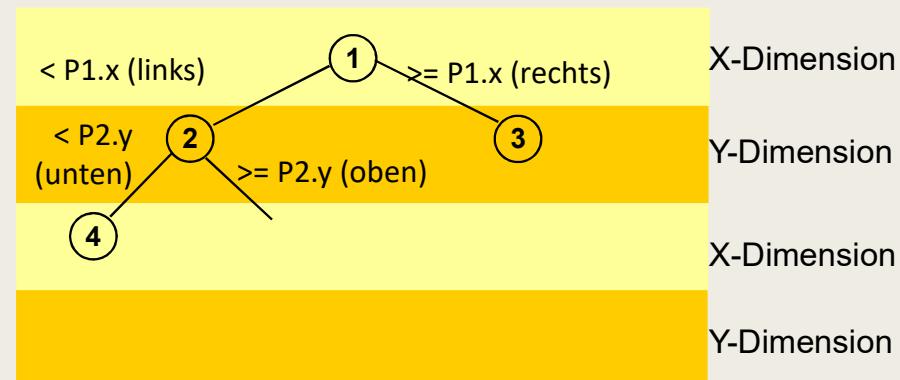
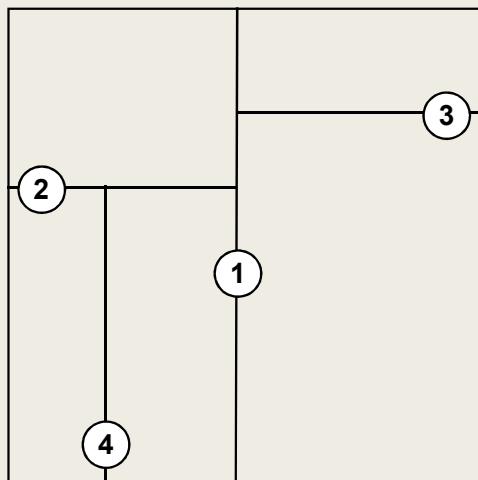
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

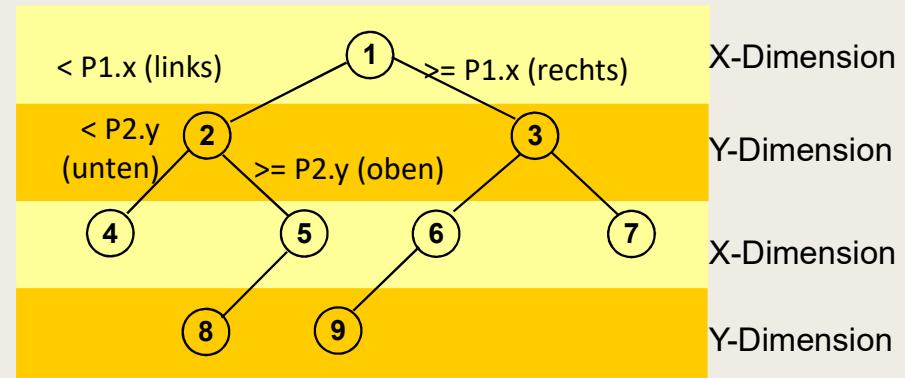
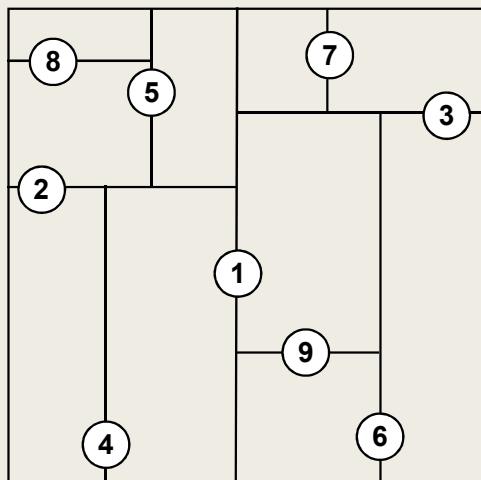
Erstellen eines kd-Baums



Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

Erstellen eines kd-Baums



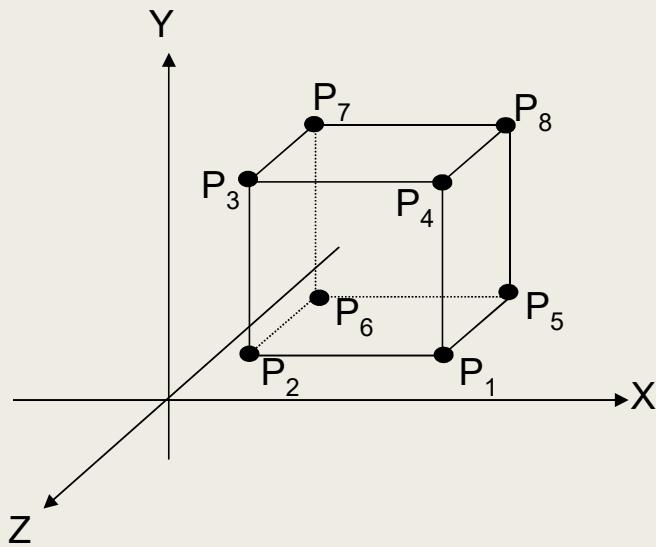
Fläche auf der die einzusortierenden Punkte liegen

kd-Baum

Übungsaufgabe:

Erstellen Sie anhand der unten vorgegebenen Punkte P_1 bis P_8 einen kd-Baum im 3-dimensionalen Raum. Die gegebenen Punkte sind Eckpunkte eines zum Koordinatensystem achsenparallelen ausgerichteten Würfels. Die Reihenfolge, in der die Punkte bearbeitet werden sollen, ist durch die Indices festgelegt.

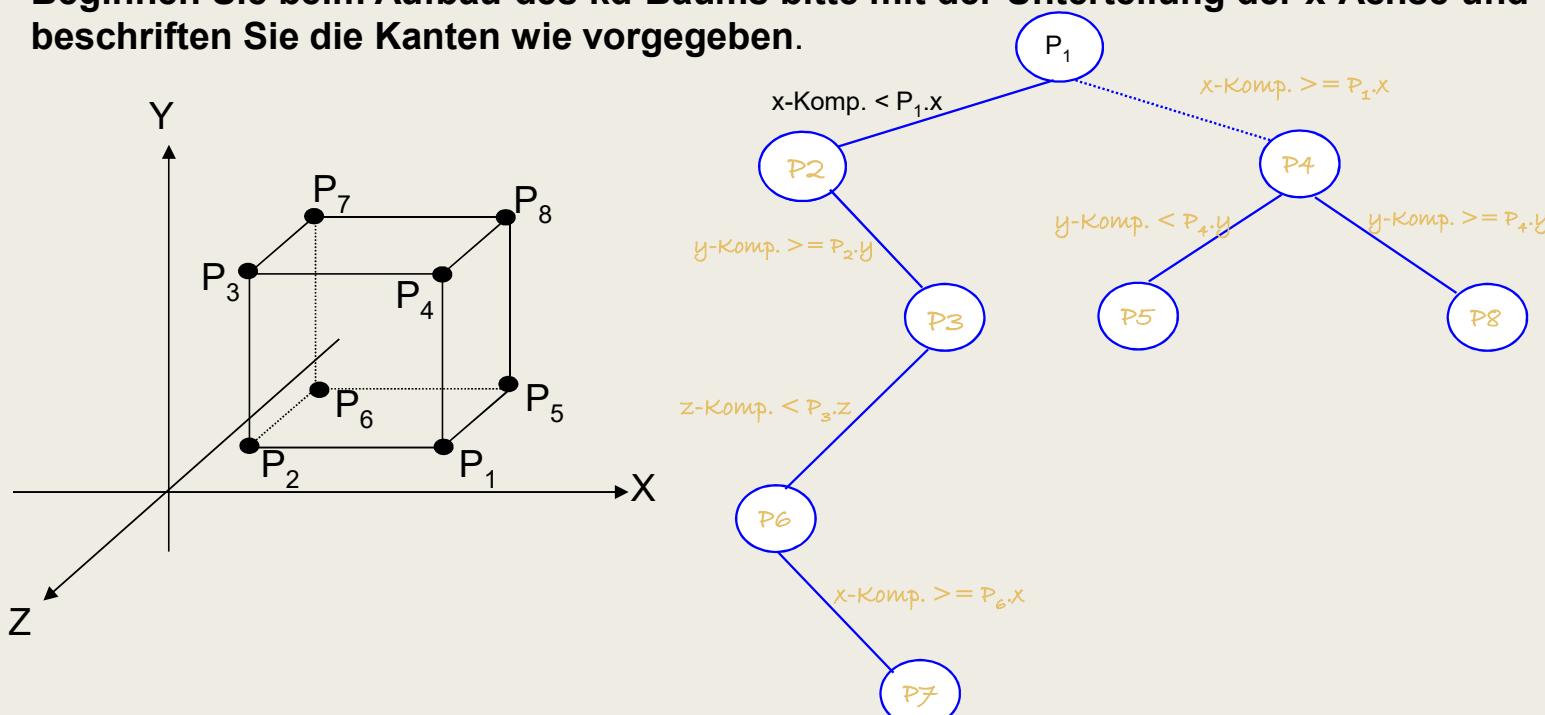
Beginnen Sie beim Aufbau des kd-Baums bitte mit der Unterteilung der x-Achse und beschriften Sie die Kanten wie vorgegeben.



Übungsaufgabe:

Erstellen Sie anhand der unten vorgegebenen Punkte P_1 bis P_8 einen kd-Baum im 3-dimensionalen Raum. Die gegebenen Punkte sind Eckpunkte eines zum Koordinatensystem achsenparallelen ausgerichteten Würfels. Die Reihenfolge, in der die Punkte bearbeitet werden sollen, ist durch die Indices festgelegt.

Beginnen Sie beim Aufbau des kd-Baums bitte mit der Unterteilung der x-Achse und beschriften Sie die Kanten wie vorgegeben.



kd-Baum

Erstellen eines kd-Baums:

- Suchen der Stelle, an dem der neu einzufügende Punkt liegen müsste.
- An dieser Stelle wird der Punkt dann auch eingefügt.

Wie sucht man einen Punkt P im kd-Baum?

- Entspricht dem normalen Suchen im binären Baum unter Berücksichtigung der Dimensionsangabe;
- die Dimensionsangabe j des gerade erreichten Knotens P_i gibt an, welche Komponenten von P und P_i man vergleichen muss, um zu erfahren, in welchem Unterbaum weiter zu suchen ist.

kd-Baum

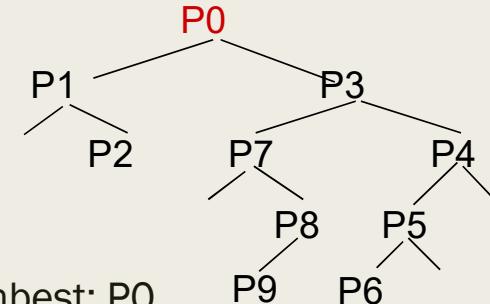
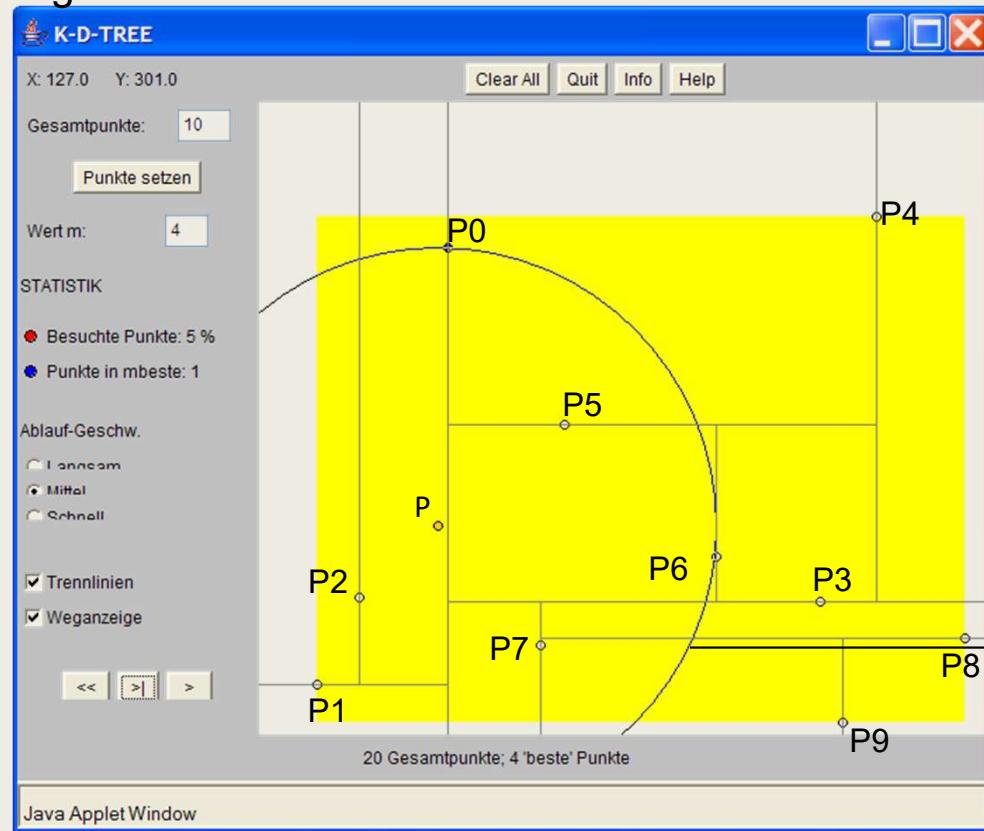
Ganz wichtig für Photon Mapping:

Suche die m nächsten Punkte zu einem gegebenen P:

- **$mbest$:** Hilfsliste, enthält die m am nächsten zu P liegenden Punkte (Photonen), der bisher besuchten Punkte des Baums (am Anfang sind das weniger als m).
- **$mdist$:** Ein Feld, dass die Distanz von P zu dem am weitesten entfernten Punkt der Liste $mbest$ enthält (MAX-Wert, wenn es noch weniger als m sind).

kd-Baum

Gesucht werden die m ($m=4$) nächsten zu Punkt P liegenden Photonen.

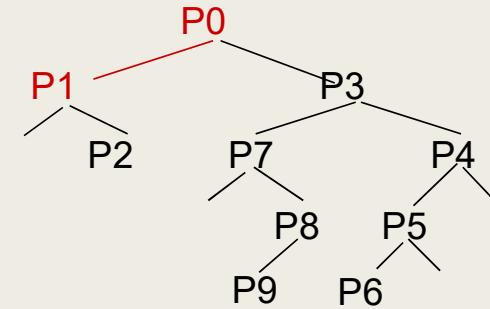
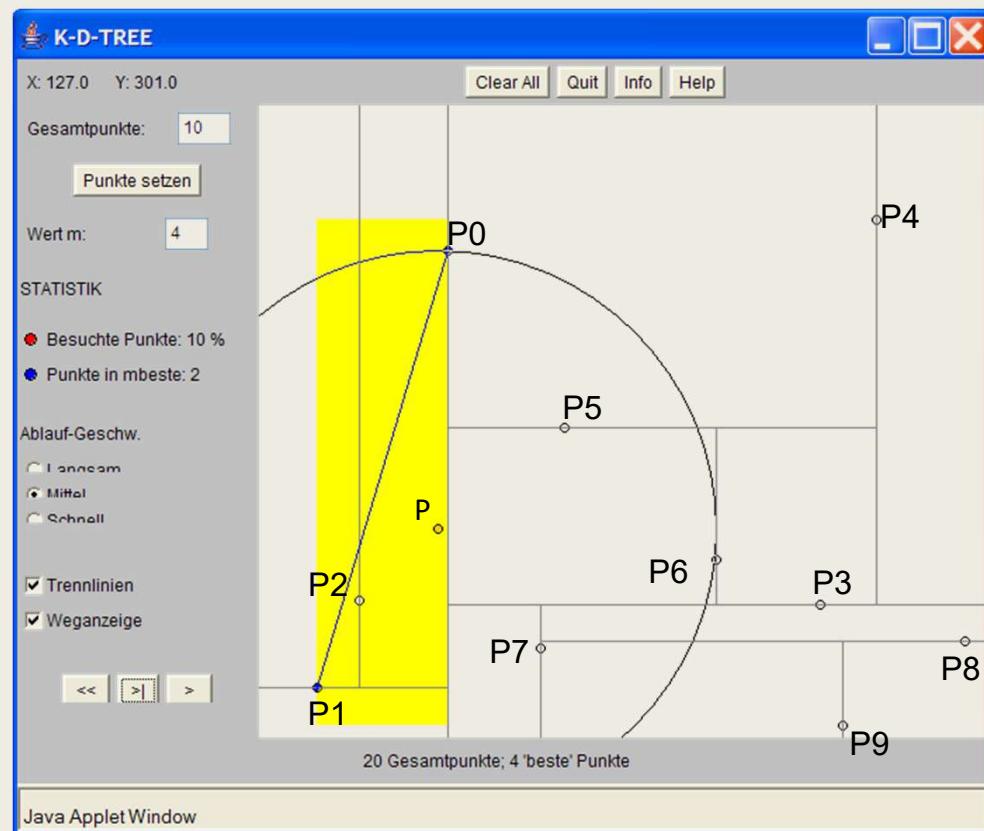


An P0 wird der Raum aufgrund der X-Koord. aufgeteilt und P liegt auf der linken Seite (es gilt $P.x < P0.x$) daher wird im Baum „links“ gesucht.

$$||P-P1|| < mdist$$

Radius des Umkreises = mdist
Bzw. wenn $mdist == \text{MAX-Wert}$: max. Distanz zu den Punkten in mbest

kd-Baum



mbest: P1, P0

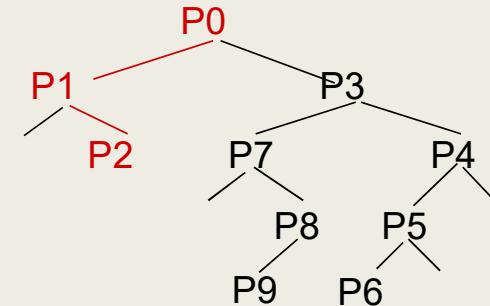
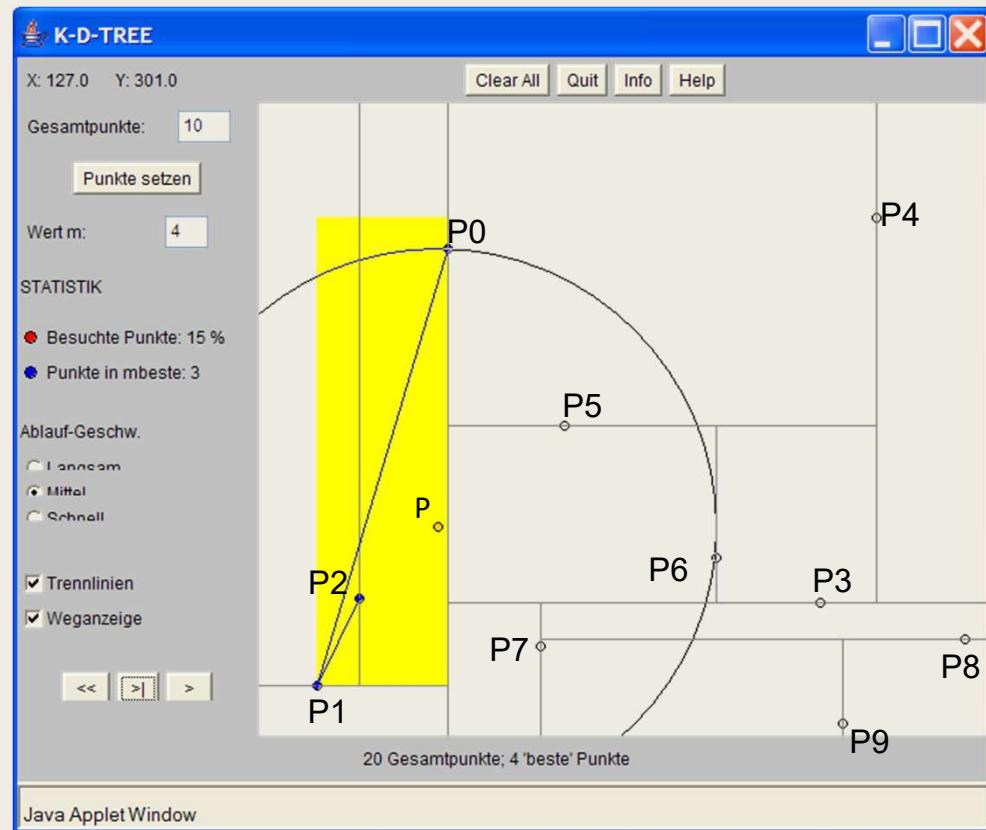
mdist: MAX-Wert

Aufgrund der P.y-Koord. der Baum erst „oben“ (Null-Pointer) und dann „unten“ durchsucht.

Nächster: P2

$|P - P2| < mdist$

kd-Baum



mbest: P2, P1, P0

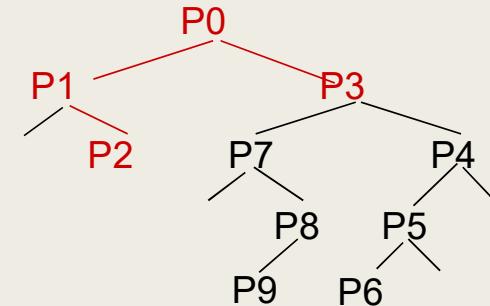
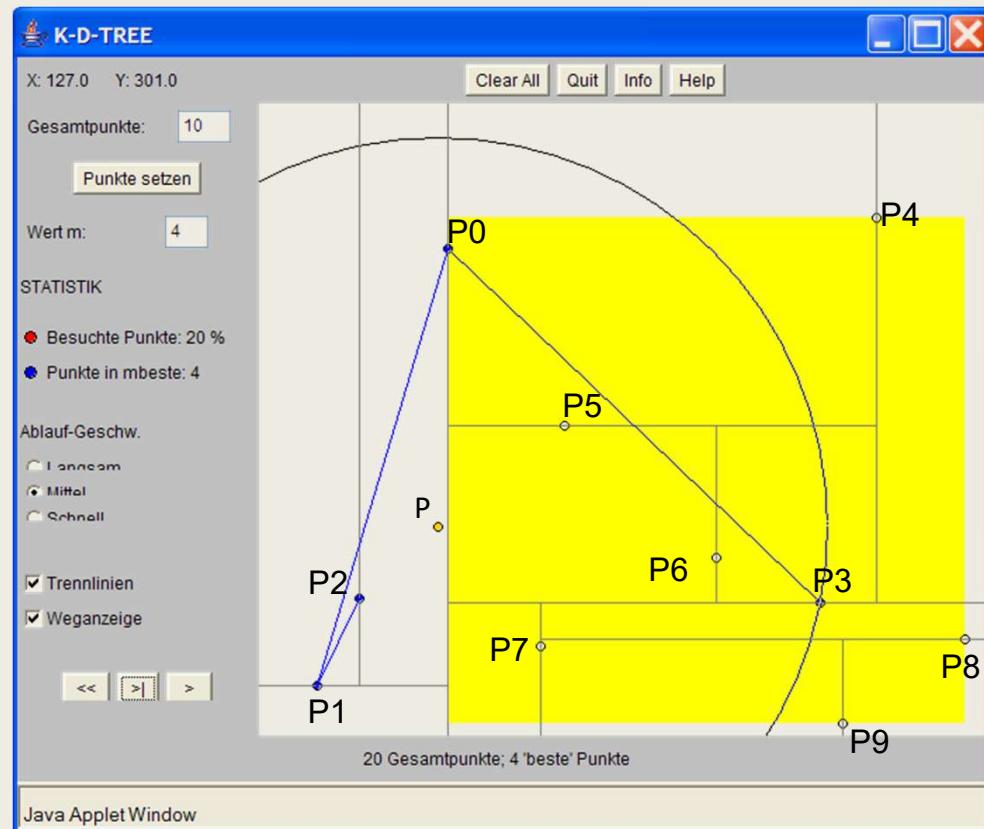
mdist: MAX-Wert

Teilbaum links vollständig durchsucht.

Nächster Punkt: P3

$\|P - P3\| < \text{mdist}$

kd-Baum



mbest: P3, P2, P1, P0

(Array ist voll!)

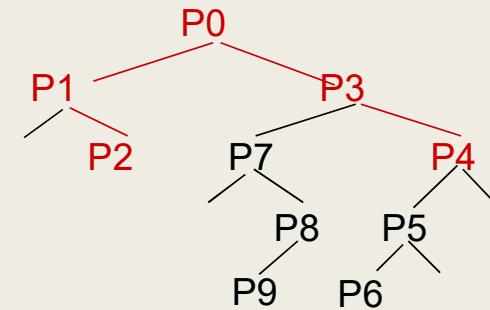
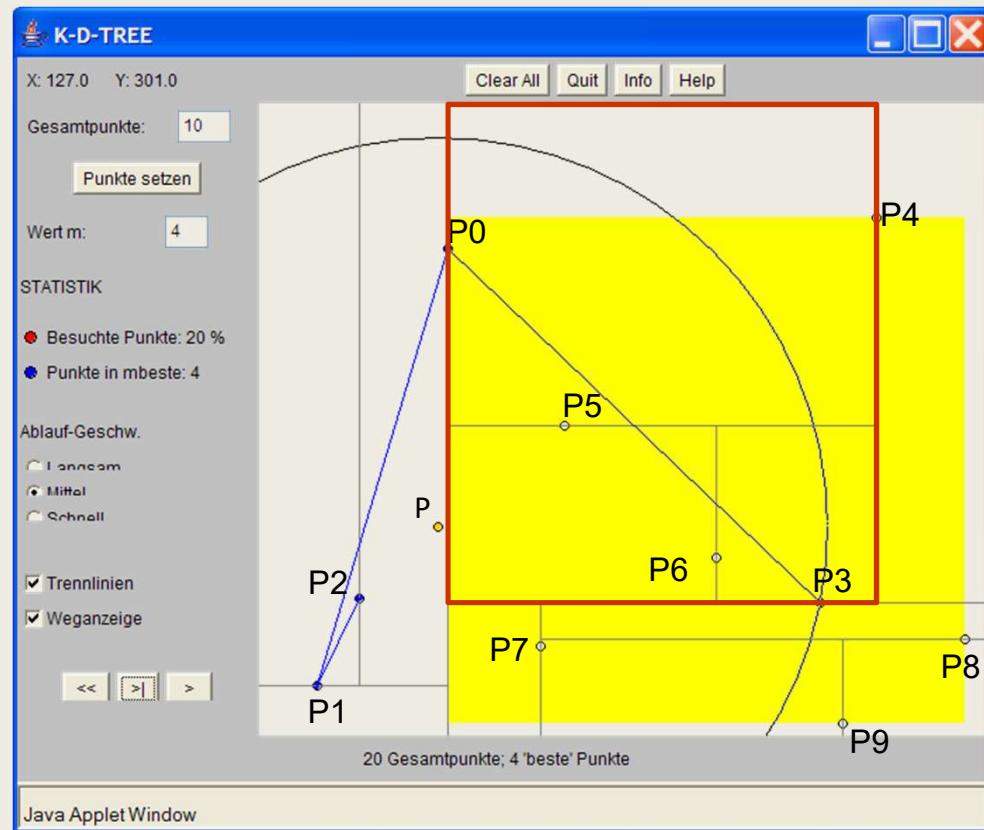
mdist: ||P-P3||

Nächster: P4

An P3 wird der Raum aufgrund der Y-Koord. aufgeteilt und P liegt über P3 (es gilt P.y >= P3.y).

||P-P4|| >= mdist !

kd-Baum



mbest: P3, P2, P1, P0

(Array ist voll!)

mdist: ||P-P3||

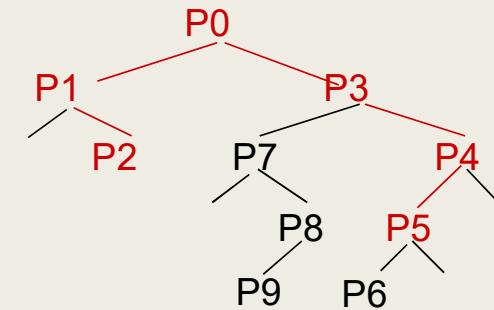
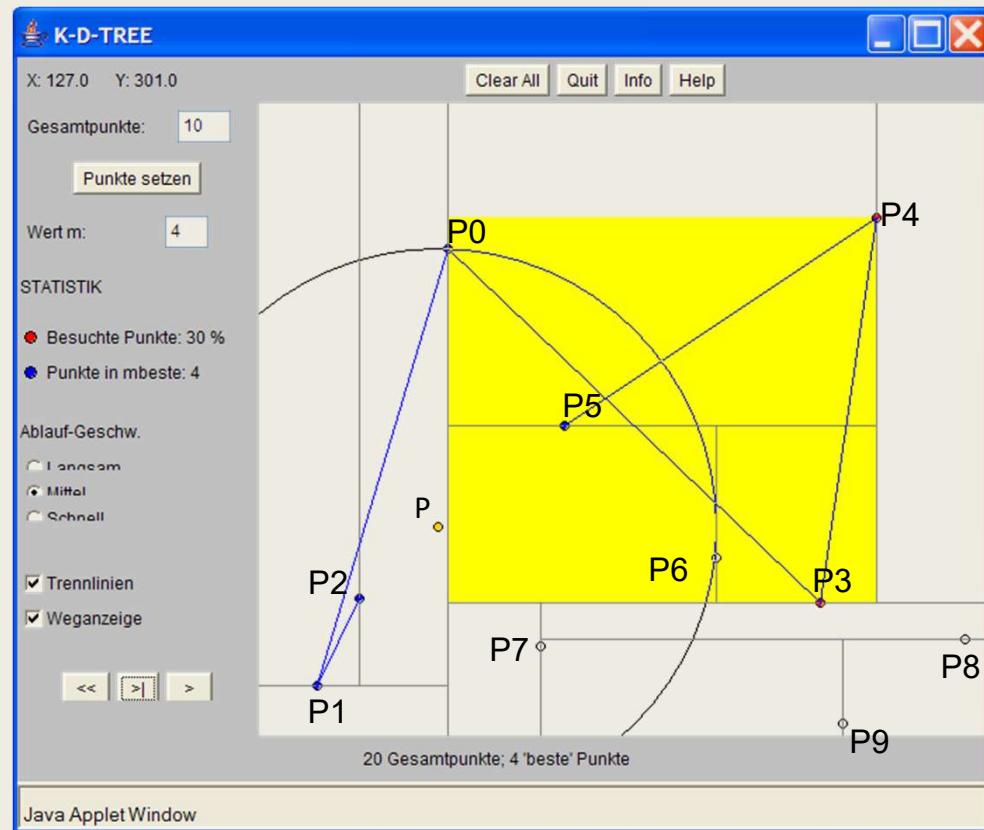
Aufgrund der X-Koord. von P

Nächster Punkt: P5

$||P-P5|| < mdist !$

Solange der Bound eines Knotens bspw. P5 (rotes Quadrat) mit dem Umkreis von P eine Schnittmenge hat, muss der Teilbaum des Knotens durchsucht werden.

kd-Baum



mbest: P5, P2, P1, P0

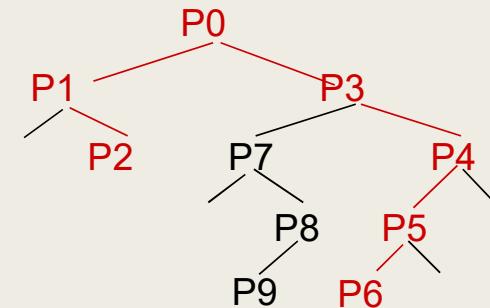
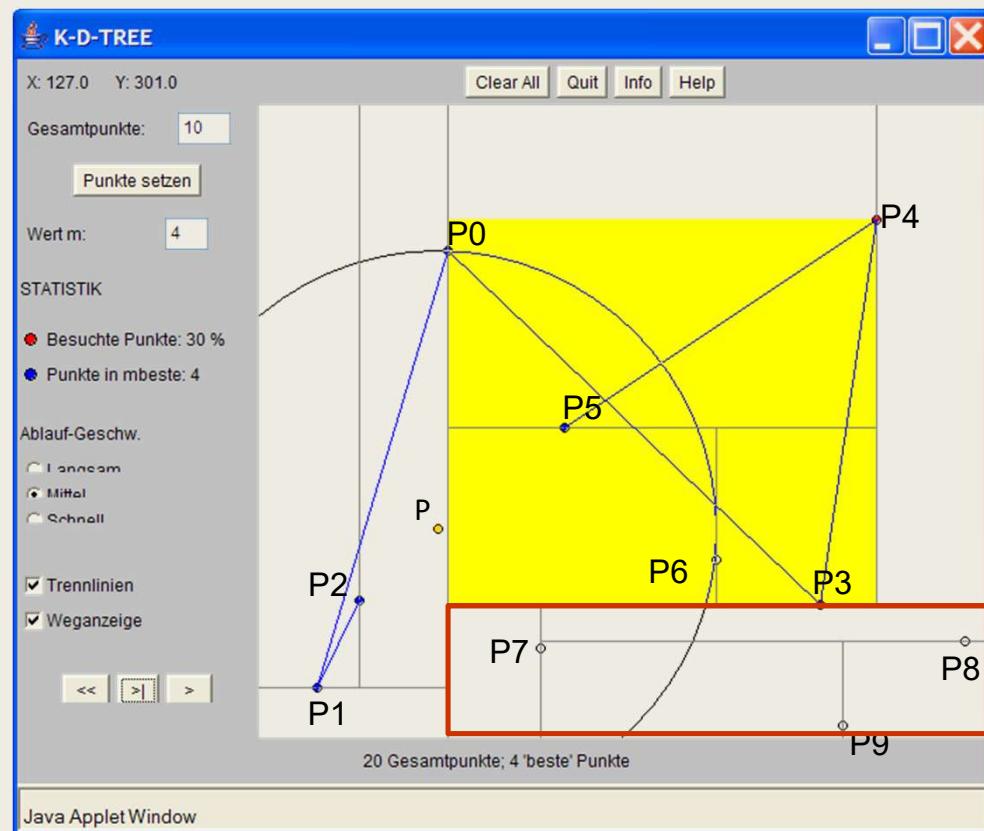
(Array ist voll!)

mdist: ||P-P0|| !!!

Nächster: P6

||P-P6|| >= mdist !

kd-Baum



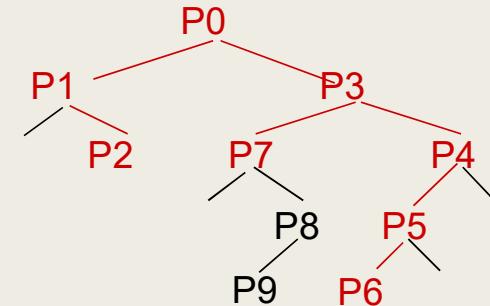
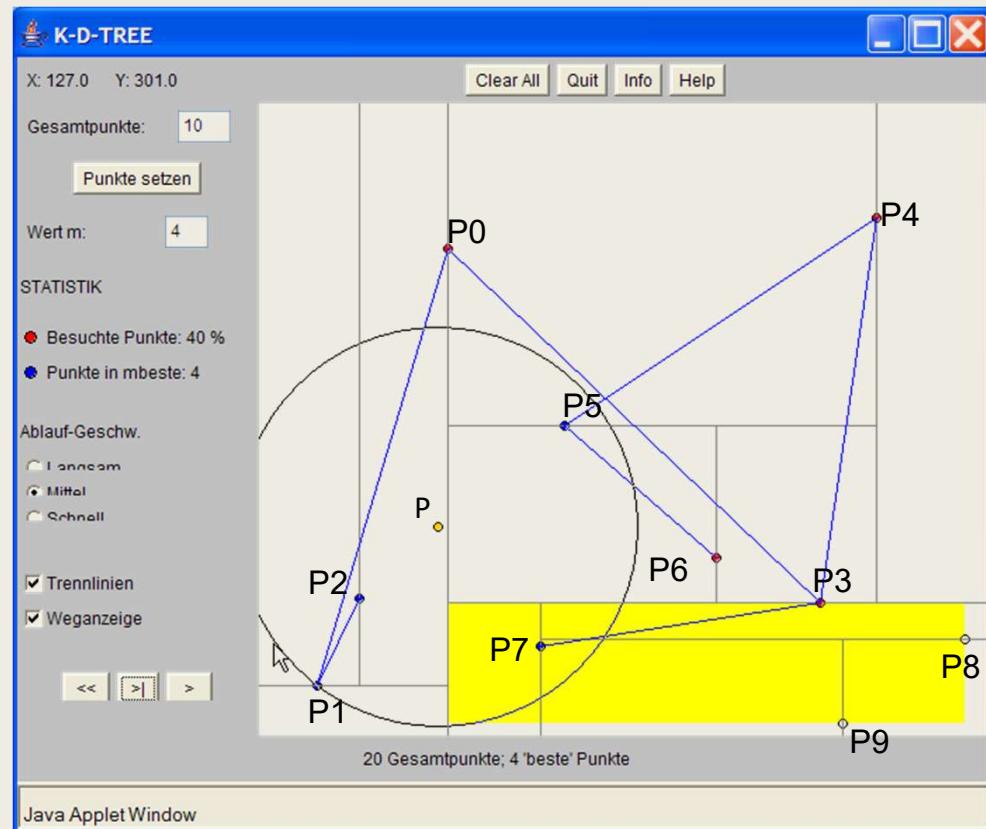
mbest: P5, P2, P1, P0

(Array ist voll!)

mdist: ||P-P0|| !!!

Nächster: P7
||P-P7|| < mdist

kd-Baum



mbest: P7, P5, P2, P1

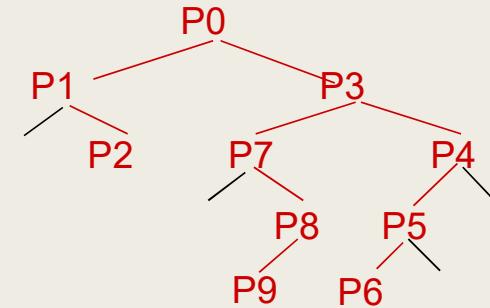
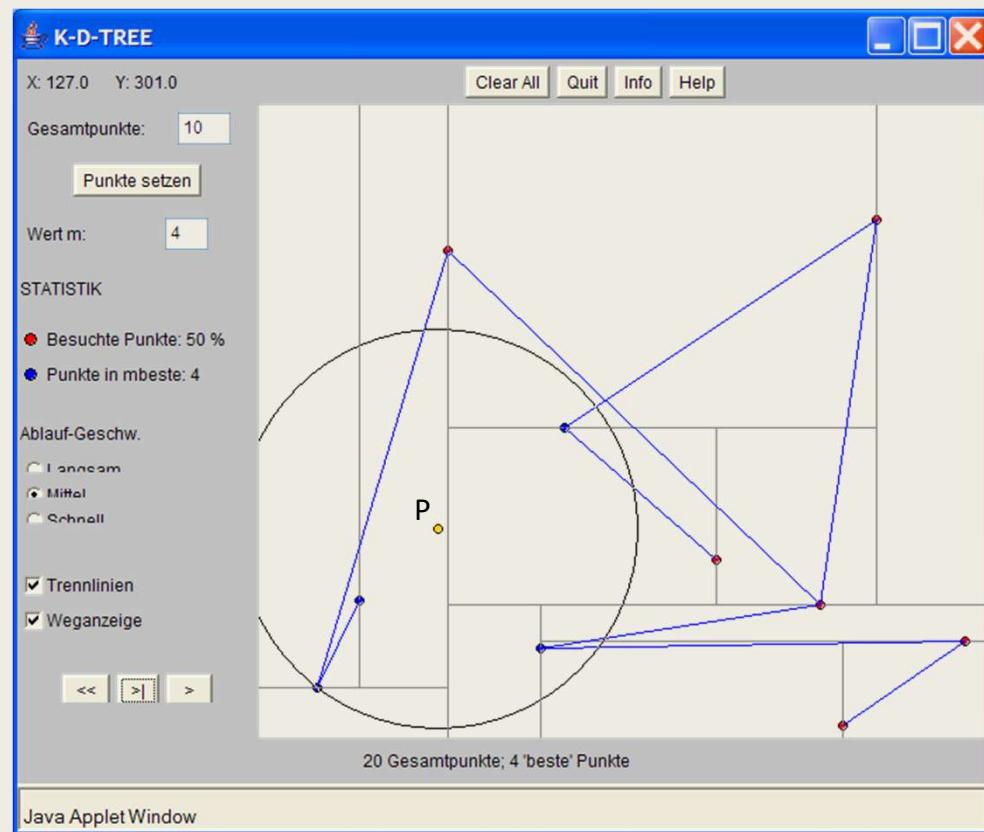
(Array ist voll!)

mdist: ||P-P1||

Fertig? Eigentlich nicht:

Bound für P8 müsste noch getestet werden, da er vom Umkreis um P mit Radius mdist geschnitten wird.

kd-Baum



Abbruchkriterium:

Die Suche ist beendet, wenn kein undurchsuchter *bound* mehr vorhanden ist, der ein Schnittvolumen mit dem Umkreis (Radius mdist) um P besitzt.

Kd-Baum in Photon Mapping:

- kd-Baum speichert die Photon Map.
- D.h. es wird nur noch darin gesucht und nichts neues mehr eingefügt.
- Daher sollte der Baum ausbalanciert werden (AVL-Baum)
- Laufzeit beträgt dann : $O(\log_2 n)$

Literatur:

1. M. Pharr, G. Humpherys: „*Physically Based Rendering*“, Morgan Kaufmann Verlag, 2004
2. J. Foley, A. van Dam, St. Feiner and J. Hughes: „*Computer Graphics: Principles and Practice*“, 2nd Ed., Addison-Wesley, 1990
3. P. Dutre, K. Bala, P. Bekaert: „*Advanced Global Illumination*“, 2nd Ed., A K Peters, 2006
4. H. W. Jensen: „*Realistic Image Synthesis Using Photon Mapping*“, A K Peters, 2001
5. P. Tipler: „*Physik*“, Spektrum Akademischer Verlag, 1998
6. R. L. Cook and K. E. Torrance „*A Reflectance Model for Computer Graphics*“, 1981