

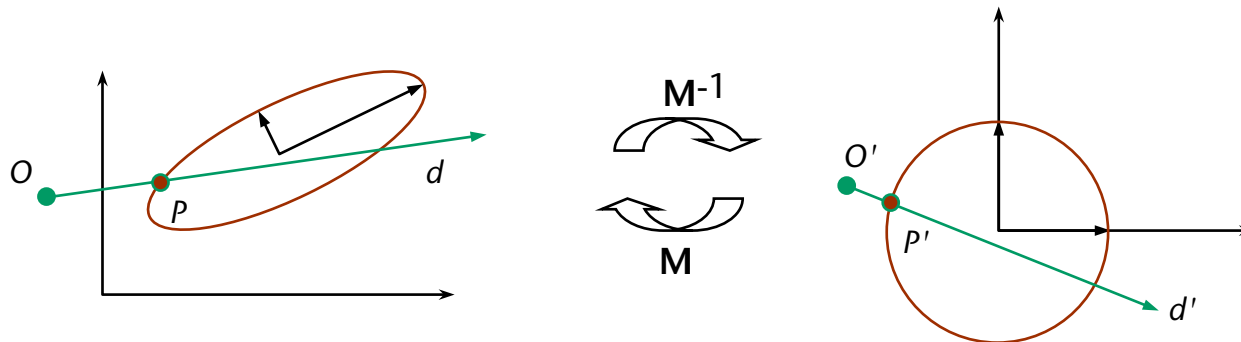


# Instancing / Strahltransformation



- Kompliziertere (transformierte) Formen lassen sich oft reduzieren auf einfache (kanonische) Formen
- Idee: transformiere Strahl; führe Schnitt mit der einfachen Form durch; transformiere Schnittpunkt und -normale zurück
- Beispiel  
Ellipsoid:

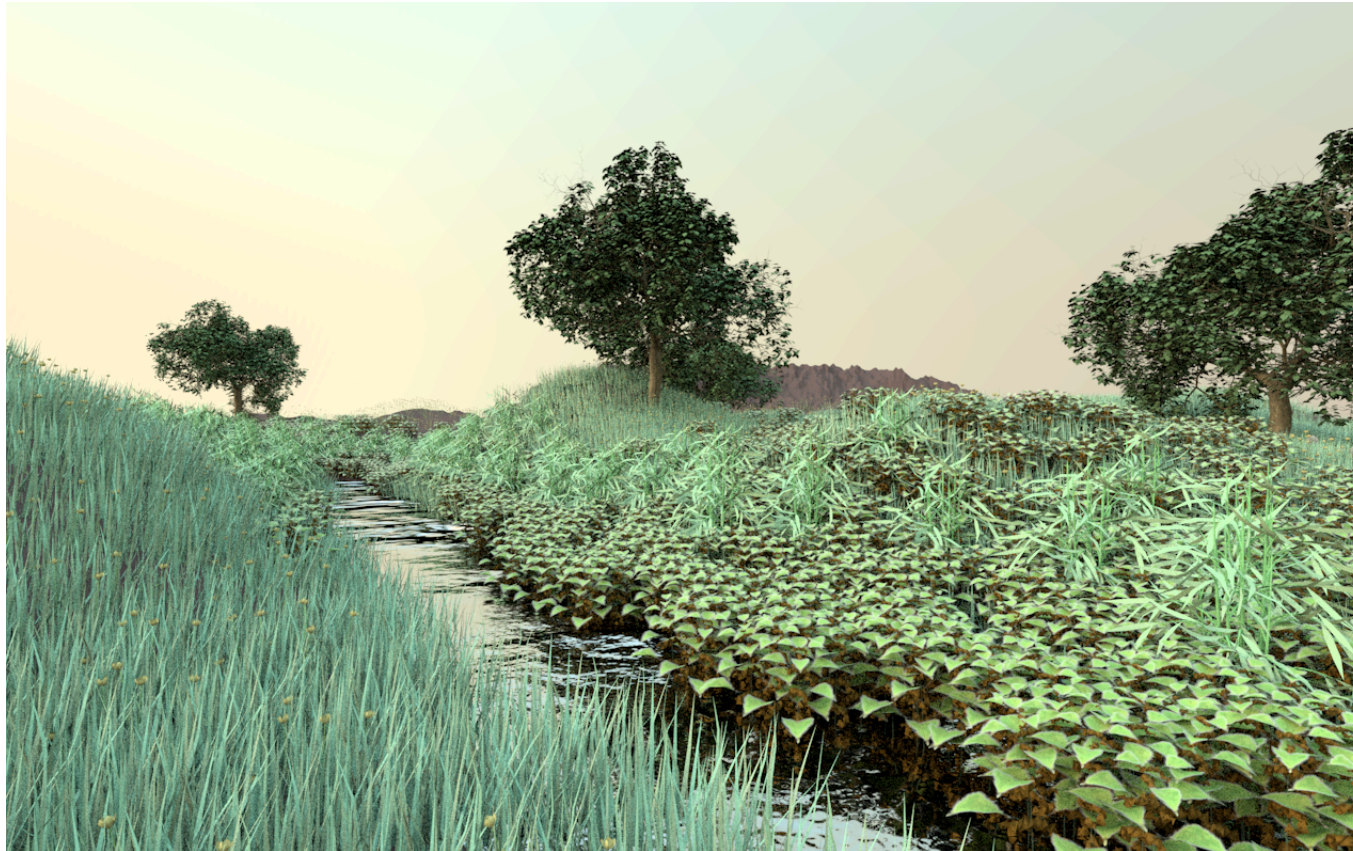
berechne  $P'(t) = \mathbf{M}^{-1}O + t\mathbf{M}^{-1}d$   
schneide  $P'(t)$  mit Einheitskugel  $\rightarrow P', \mathbf{n}', t'$   
 $P := \mathbf{M} \cdot P'$  ;  $\mathbf{n} := (\mathbf{M}^{-1})^T \cdot \mathbf{n}'$  ;  $t := ?$





## Weiterer Grund für *Instancing*

- Speichereinsparung: nur mittels Instancing passen solch riesige Szenen komplett in den Speicher



61 unique plant models, 1.1M unique triangles, 300MB —  
4000 plants in the scene, 19.5M triangles



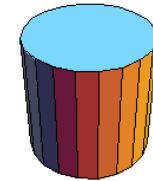
# Weitere Quadriken



- Mit diesen Techniken kann man viele weitere Objekte testen
  - Parametrische Liniengleichung in implizite Form des Objektes einsetzen
  - Ergibt Polynom in t vom Grad 2, 4, ...
  - Nach t lösen (analytisch, Newton, ...)

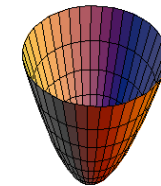
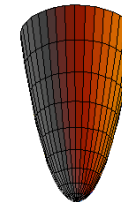
- Unendlicher Zylinder:

$$x^2 + y^2 = 1$$



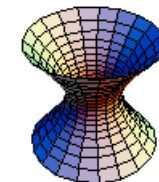
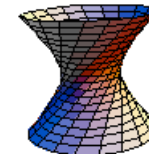
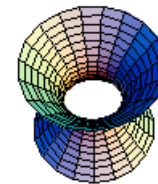
- Paraboloid:

$$x^2 + y^2 - z = 0$$



- Hyperboloid (*one sheet*):

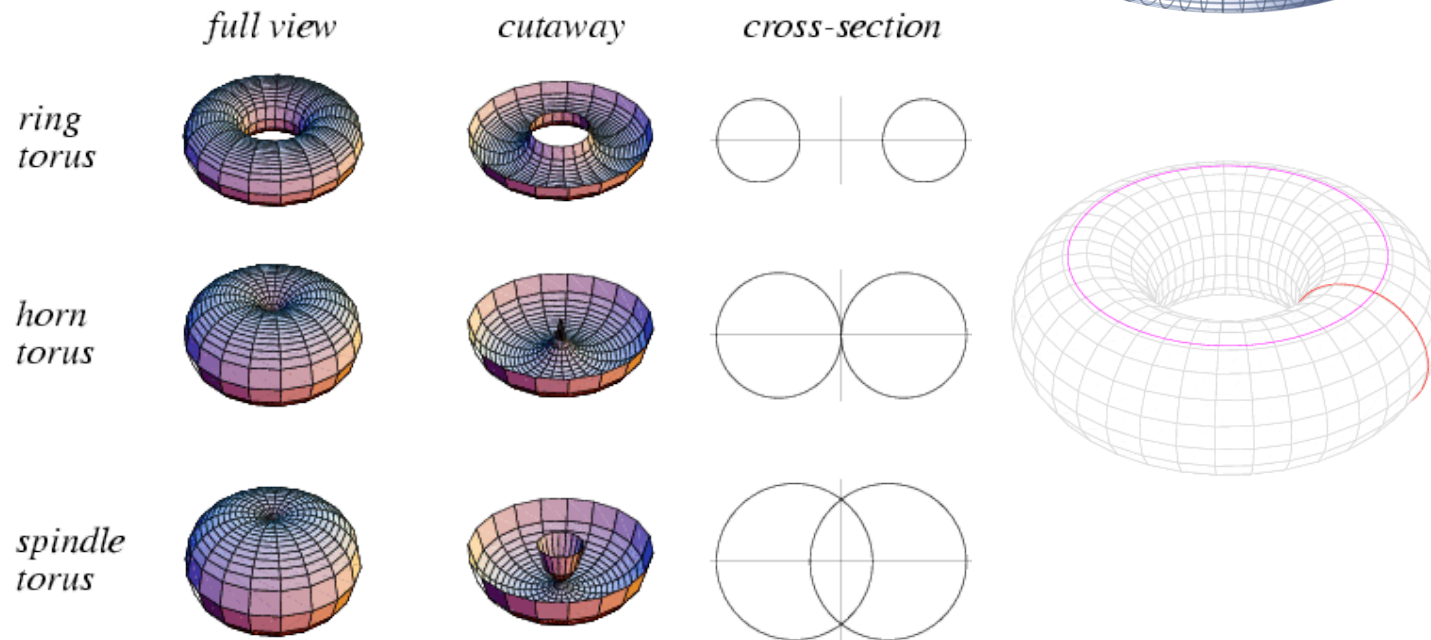
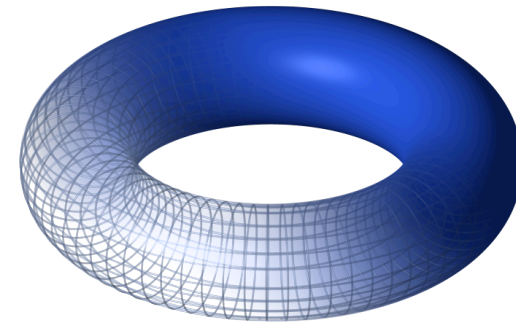
$$x^2 + y^2 - z^2 = 1$$





■ Torus:

$$\left(c - \sqrt{x^2 + y^2}\right)^2 + z^2 = a^2$$







# Superquadrics



- Verallgemeinerungen der Quadriken

- Super-Ellipsoid:

$$\left(\frac{x}{a}\right)^p + \left(\frac{y}{b}\right)^q + \left(\frac{z}{c}\right)^r = 1$$

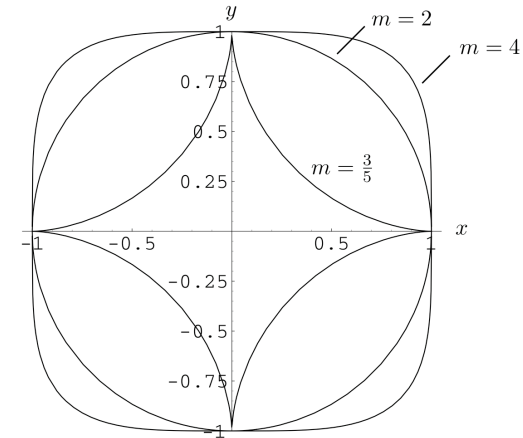
- Super-Hyperboloid:

$$\left(\frac{x}{a}\right)^p + \left(\frac{y}{b}\right)^q - \left(\frac{z}{c}\right)^r = 1$$

- Super-Toroid:

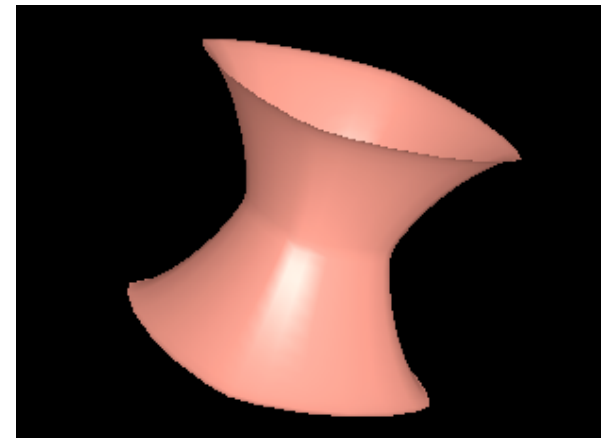
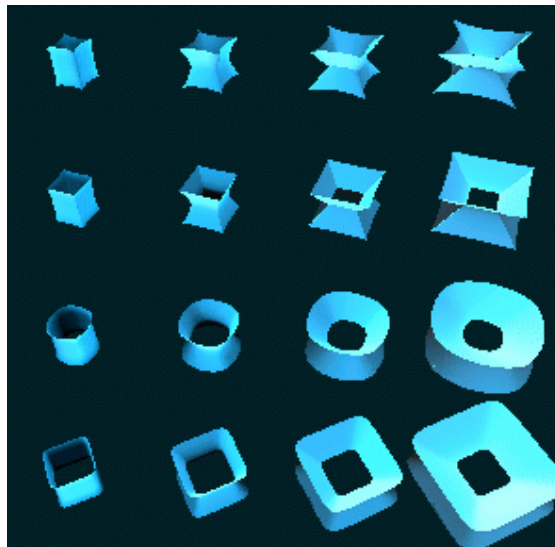
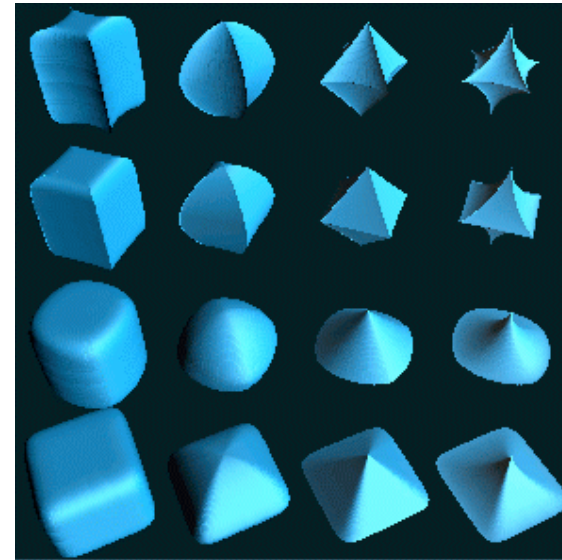
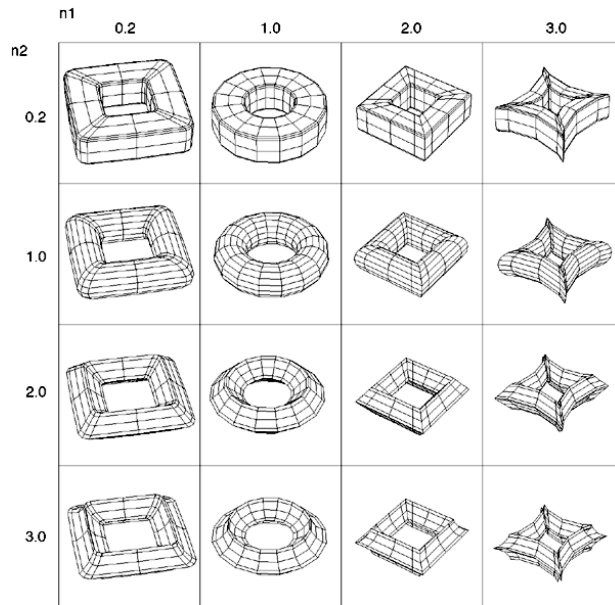
$$\left(d - \left(\left(\frac{x}{a}\right)^m + \left(\frac{y}{b}\right)^n\right)^q\right)^r + \left(\frac{z}{c}\right)^p = e^2$$

- Achtung: hier ist immer  $|x|^p$  gemeint!

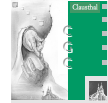




# Beispiele von Super-Quadrics



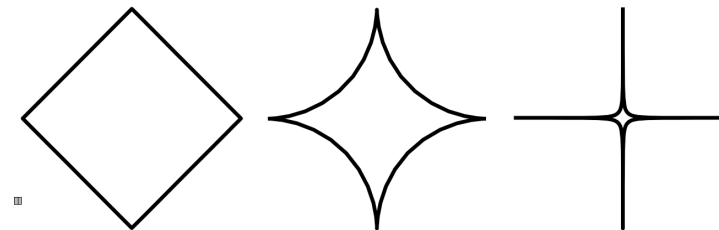
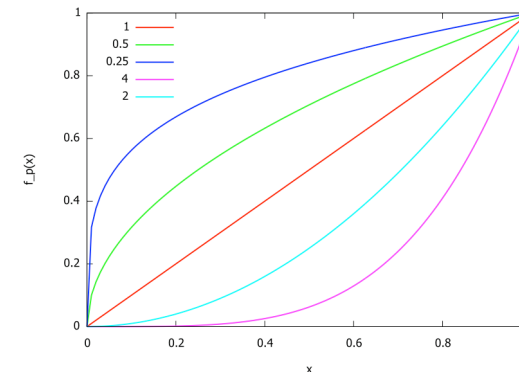
XScreenSaver demo "SuperQuadrics"  
([www.jwz.org/xscreensaver](http://www.jwz.org/xscreensaver))



- Variante der Superquadratics mit u.U. besseren Eigenschaften
- Idee der Superquadratics kann man auch so schreiben:

$$F(x, y, z) = f_p\left(\frac{x}{a}\right) + f_q\left(\frac{y}{b}\right) + f_r\left(\frac{z}{c}\right) - 1$$
$$f_p(x) = |x|^p$$

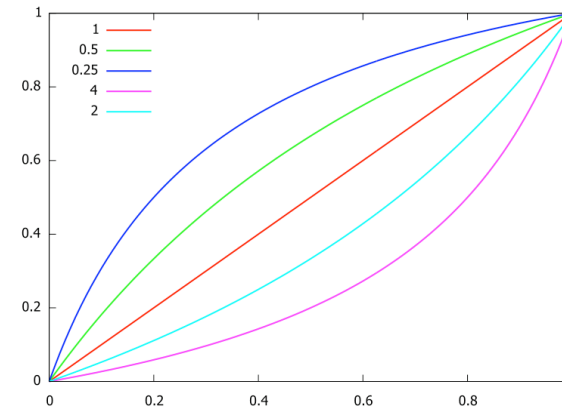
- Problem:
  - $f_p(x)$  ist an der Stelle  $x=0$  nicht differenzierbar für  $p \leq 1$
  - Dadurch entstehen für  $p < 1$  "Spitzen", die möglicherweise unerwünscht sind
  - Außerdem ist  $f_p(x)$  rel. teuer auszuwerten





- Simple Idee: verwende einfach eine andere "Potenzfunktion"
- Die Pseudo-Potenzfunktion von Blanc & Schlick:

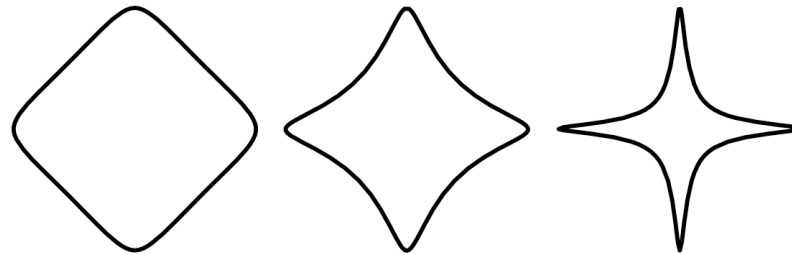
$$g_p(x) = \frac{x}{p + (1 - p)x}$$



- Die Ratioquadric für "Ratio-Ellipsoide" ist damit

$$F(x, y, z) = g_p\left(\frac{x}{a}\right) + g_q\left(\frac{y}{b}\right) + g_r\left(\frac{z}{c}\right) - 1$$

- Resultat:





# Metaballs



- Abgeschaut von den Molekülen
- Idee: betrachte Kugel als Menge aller Punkte im Raum, die dasselbe "Potential" haben, wobei das Maximum des Potentialfeldes im Kugelmittelpunkt herrscht → Isofläche
- Potentialfeld wird beschrieben durch Potentialfunktion, z.B.

$$p(r) = \frac{1}{r^2}$$

wobei

$$r = r_1(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_1\|$$

- Die Kugelfläche ist damit

$$K = \{\mathbf{x} \mid p(\mathbf{x}) = t\}$$

- $t$  heißt Schwellwert oder Isowert





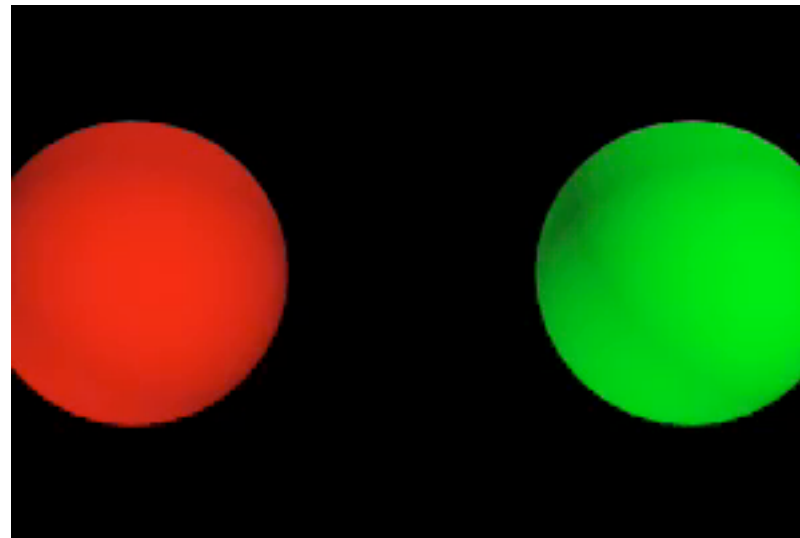
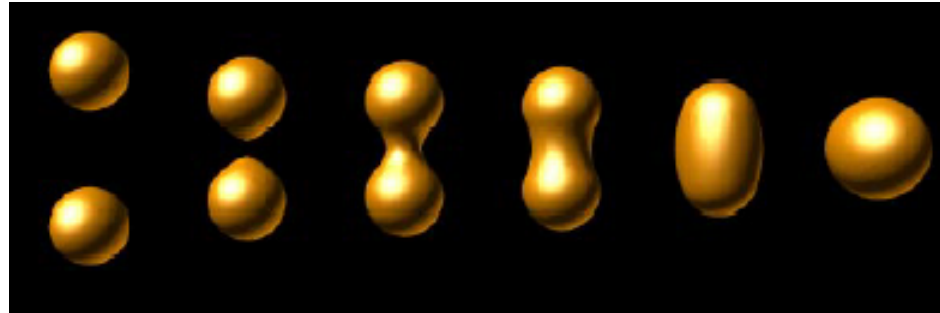
- Implizite Funktion setzt sich zusammen aus  
**Distanzfunktion + Potentialfunktion**
- Entsprechend gibt es viele Varianten und Namen: "metaballs", "soft objects", "blobs", "blobby modeling", "implicit modeling" ...
- Komplexere Objekte entstehen durch **Überlagerung (Blending)** der Potentialfelder mehrerer Punkte
  - Einfachstes Blending ist Addition der Felder:

$$P(\mathbf{x}) = \sum_{i=1}^n a_i \frac{1}{r_i^2(\mathbf{x})}$$

- Alle Punkte zusammen heißen **Skelett (skeleton)**,  
 $P$  ist das Gesamtpotential,  $a_i$  = "Feldstärke" bestimmen jew. Einfluß
- Negative Feldstärken nehmen "Material" weg (z.B. für Löcher)



- Beispiel für 2 Skelett-Punkte:



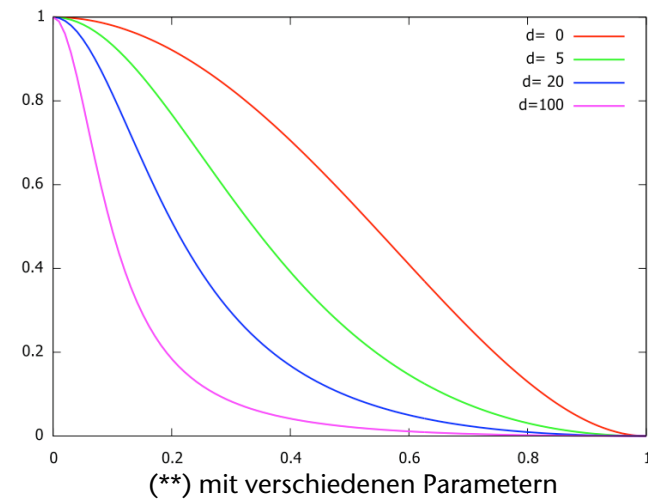
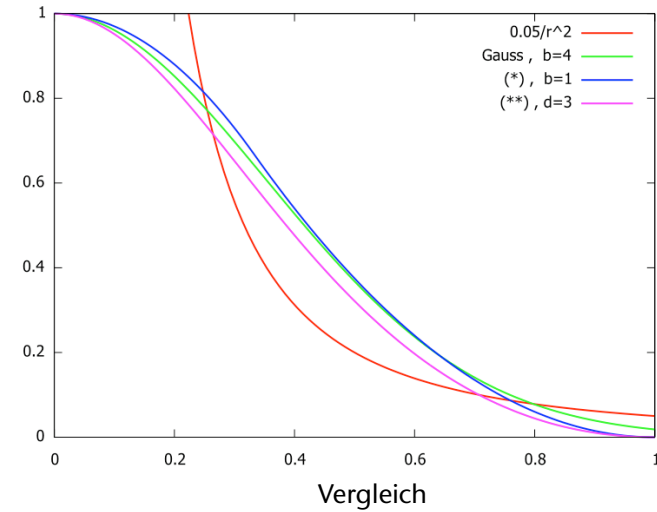


■ Andere Potentialfunktionen:

$$p_i(r) = e^{-br^2}$$

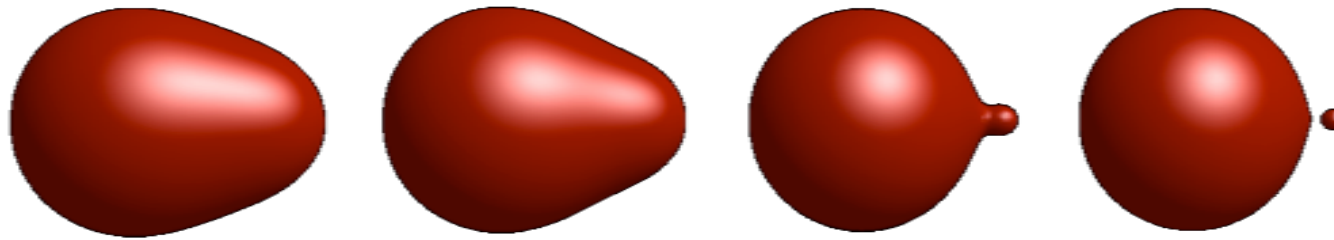
$$p(r) = \begin{cases} 1 - 3\frac{r^2}{b^2} & , r \leq \frac{1}{3}b \quad (*) \\ \frac{3}{2}\left(1 - \frac{r}{b}\right)^2 & , \frac{1}{3}b \leq r \leq b \\ 0 & , r > b \end{cases}$$

$$p(r) = \begin{cases} \frac{r^4 - 2r^2 + 1}{1 + dr^2} & , r \leq 1 \\ 0 & , r > 1 \end{cases} \quad (**)$$





- Effekt der Variation des Parameters:



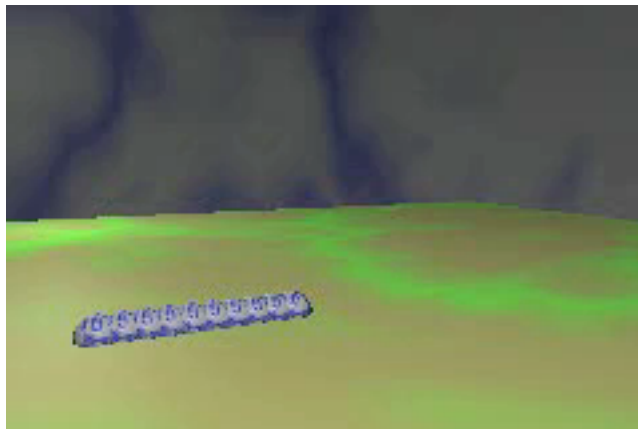
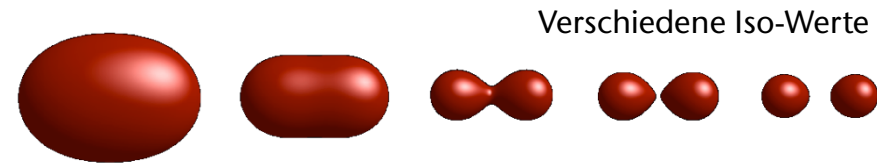
Potentialfkt (\*\*),  $d$  für den linken Skelettpunkt fest,  $d = 10 \dots 2000$  für den rechten Punkt



# Deformationen

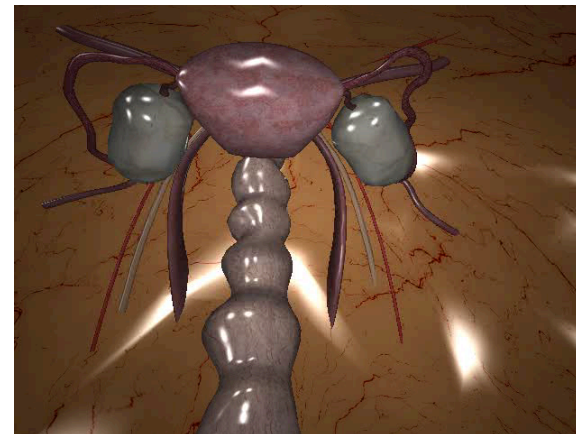


- Mit diesem Modell lassen sich Deformationen von "blob-artigen" Objekten sehr einfach modellieren:
  - Verschiebe Skelett-Punkte
  - Modifiziere Parameter  $a$ ,  $b$ , ...
  - Modifiziere den Iso-Wert  $t$



Brian Wyvill

<http://pages.cpsc.ucalgary.ca/~blob/animations.html>



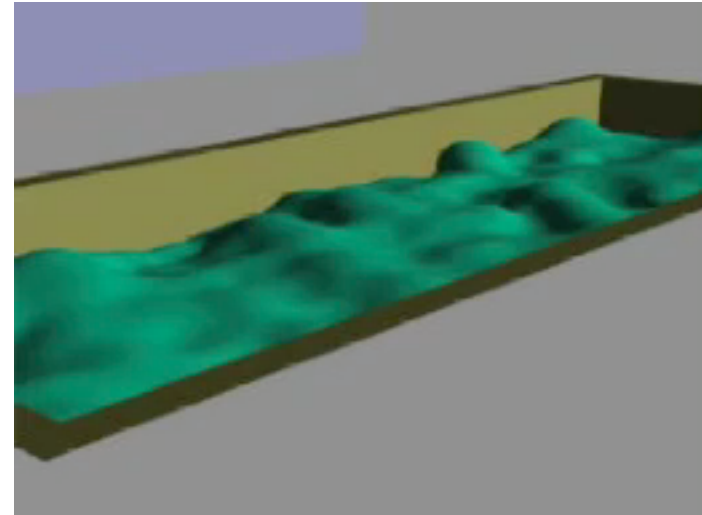
Frédéric Triquet

<http://www2.lifl.fr/~triquet/implicit/video/>





"The Great Train Rubbery" — Siggraph 1986



"Soft"

"The Wyvill Brothers"



Geoff

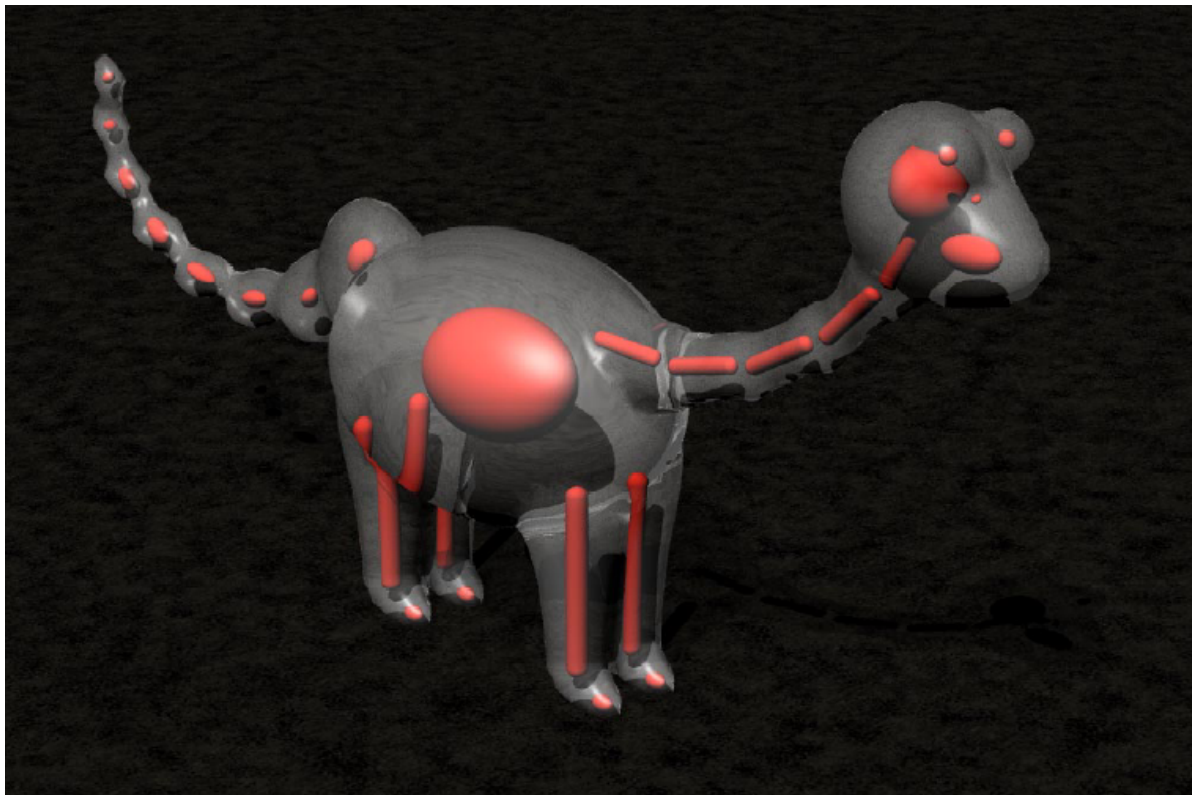


Brian

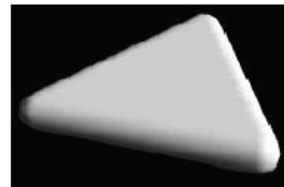
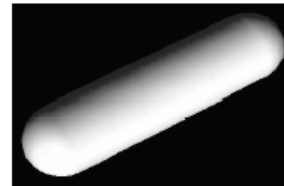


## Verallgemeinerungen / Varianten

- Punkte sind das einfachste Primitiv zur Konstruktion eines Skeletts; analog kann man Linien, Polygone, Ellipsoide, etc., verwenden:



Beispiele weiterer Skelett-Primitive:



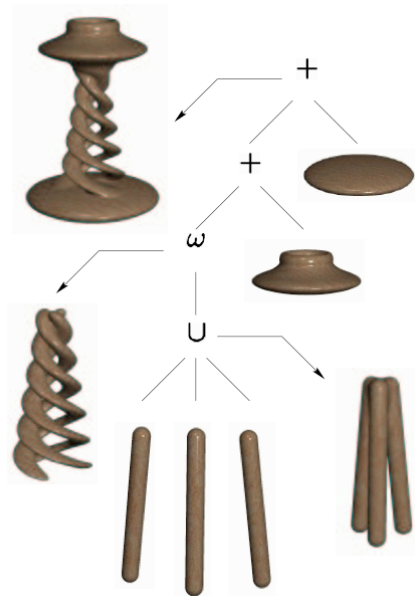


- Andere Blending-Funktionen:

$$P_{\cup}(\mathbf{x}) = \max\{p_1(\mathbf{x}), p_2(\mathbf{x})\}$$

$$P_{\cap}(\mathbf{x}) = \min\{p_1(\mathbf{x}), p_2(\mathbf{x})\}$$

- Ein Baum von Blending-Operationen — der BlobTree:





# Bemerkungen zum "*implicit modeling*"



- Man kann nette Effekte recht einfach erzielen
- Als professionelles Tool in der Animationsindustrie oder im CAD hat es sich nicht durchgesetzt, weil einfach zu viel "Magie" im Spiel ist [sagt auch Geoff Wyvill]
- Brian Wyvill arbeitet immer noch an diesen Methoden [2004]



# Normale an impliziten Flächen



- Normale in Punkt  $\mathbf{x}$  auf impliziter Fläche  $f(\mathbf{x})$

$$\mathbf{n}(\mathbf{x}) = \nabla f(\mathbf{x}) = \begin{pmatrix} \frac{\partial f}{\partial x}(\mathbf{x}) \\ \frac{\partial f}{\partial y}(\mathbf{x}) \\ \frac{\partial f}{\partial z}(\mathbf{x}) \end{pmatrix}$$

$$\approx \begin{pmatrix} f(x + \varepsilon, y, z) - f(\mathbf{x}) \\ f(x, y + \varepsilon, z) - f(\mathbf{x}) \\ f(x, y, z + \varepsilon) - f(\mathbf{x}) \end{pmatrix}$$

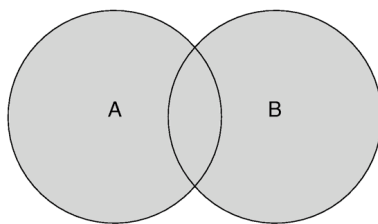
$$\approx \begin{pmatrix} f(x + \varepsilon, y, z) - f(x - \varepsilon, y, z) \\ f(x, y + \varepsilon, z) - f(x, y - \varepsilon, z) \\ f(x, y, z + \varepsilon) - f(x, y, z - \varepsilon) \end{pmatrix}$$



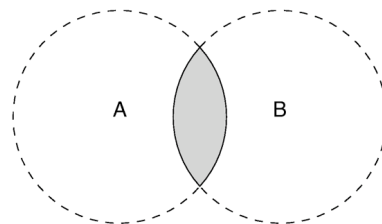


# Weitere Objekt-Repräsentation: CSG

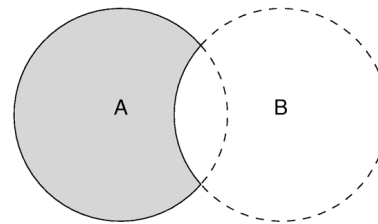
- Fügt sich genauso zwanglos ins Raytracing ein
- Zentrale Idee: konstruiere neue Objekte durch Mengen-Operationen auf einfachen Grund-Volumina (→ CSG = *constructive solid geometry*)
  - Mengen-Operationen: Schnittmenge, Vereinigung, Differenz
  - Grund-Primitive: alle Objekte, die sich leicht implizit beschreiben lassen
  - Rekursive Anwendung der Operationen → "Objekt-Arithmetik"



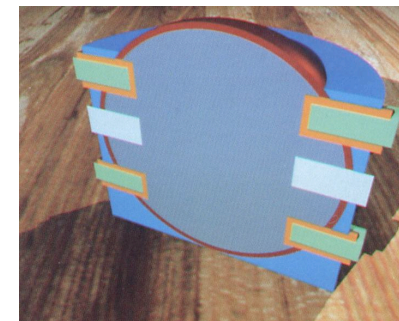
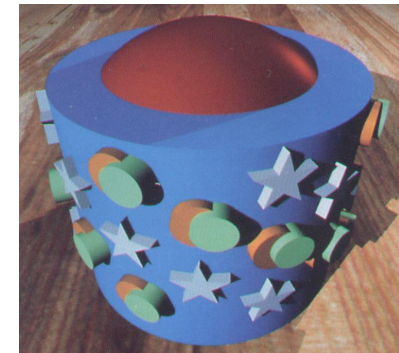
Vereinigung



Schnittmenge



Differenz

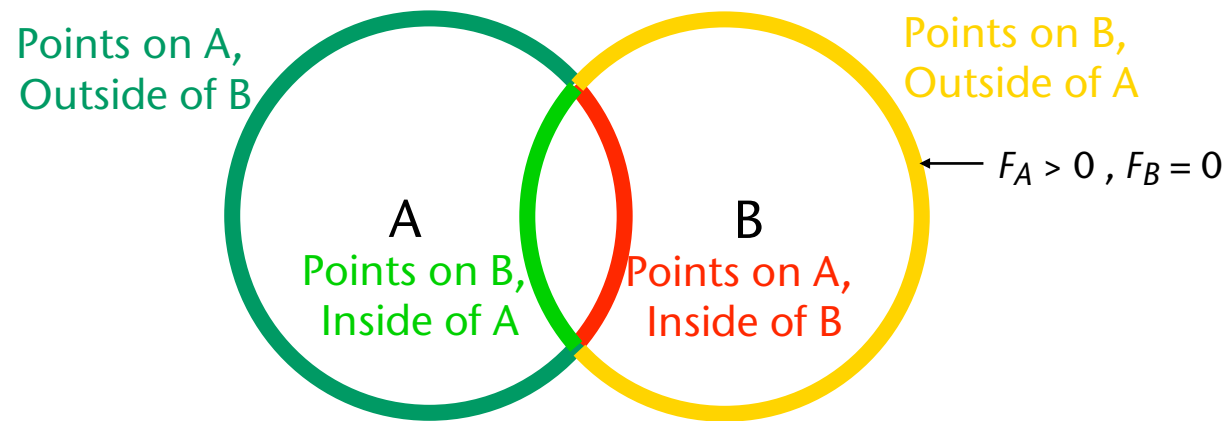




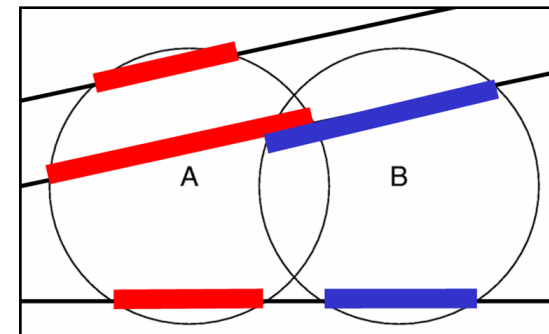
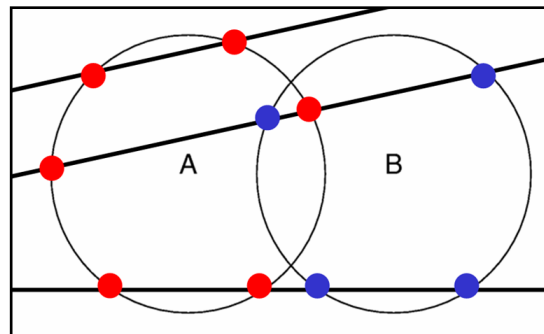
# Implementierung



- Verwende implizite Form der Grund-Objekte

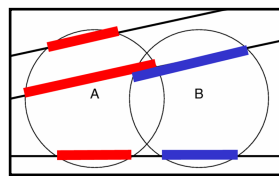
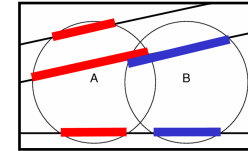


- Bestimme **alle** Schnittpunkte eines Strahls mit allen Grundobj.en
  - Falls alle Grundobj. konvex  $\rightarrow$  1 Intervall auf dem Strahl pro Grundobj.

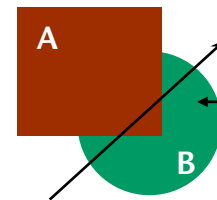




- Führe die Operation auf den Intervallen durch
- Rekursiv den CSG-Baum nach oben
- Falls an der Wurzel ein leeres Intervall entsteht → kein Schnitt
- Sonst: wähle Minimum aller Intervalle, die bis zur Wurzel übrigbleiben / entstanden sind
- Achtung:
  - Bei einer Operation können mehrere disjunkte Intervalle entstehen!



Bei Vereinigung entsteht hier ein Paar disjunkter Intervalle auf dem Strahl!



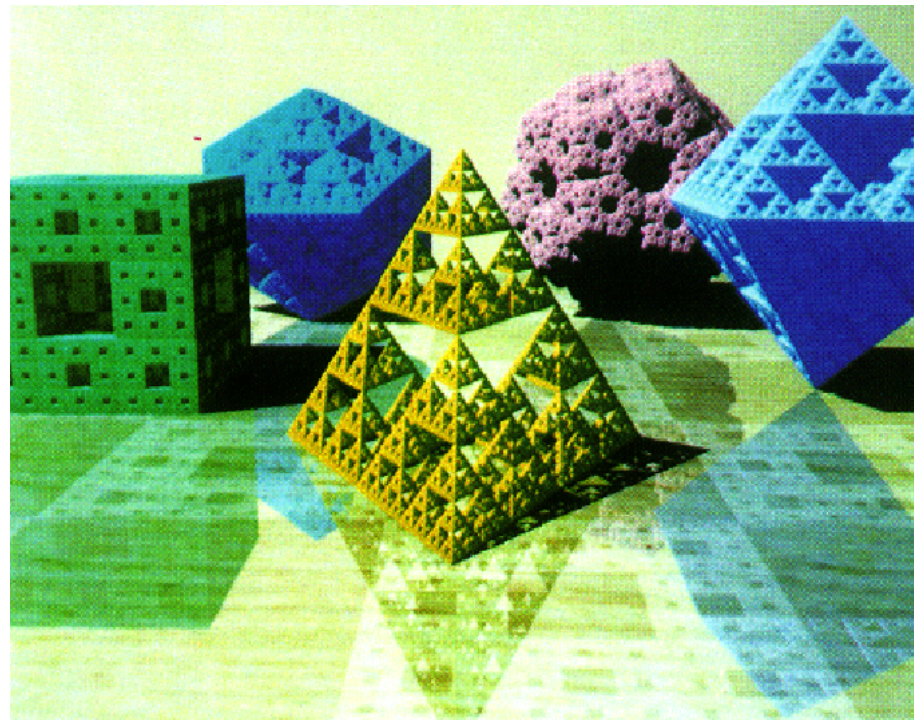
Dito hier bei der Differenz  $B - A$ !

- Achte auf numerische Stabilität (z.B.: lösche zu kleine Intervalle)



# Fraktale

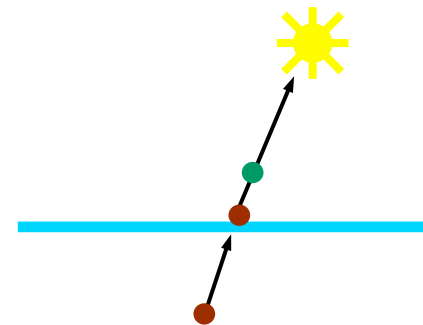
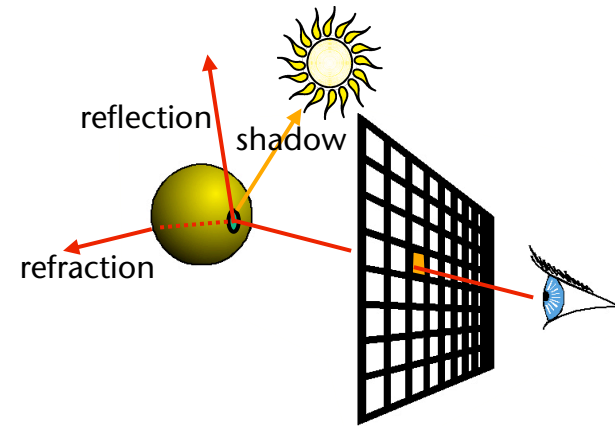
- Auch Fraktale kann man trivial ray-tracen
  - Einfach Rekursion "on demand" bis zur gewünschten Tiefe
- Prozedural beschriebene Objekte



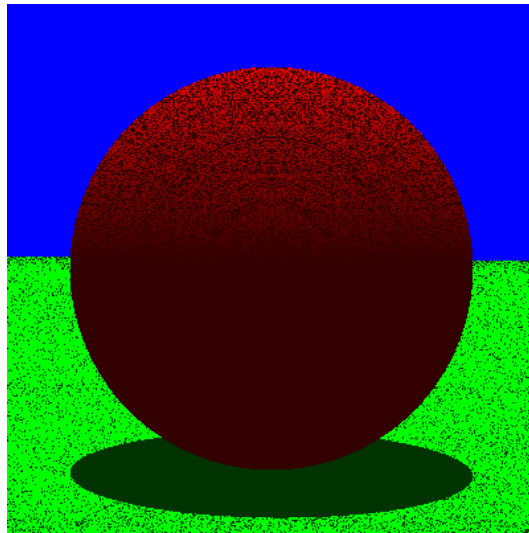


# The evil $\epsilon$

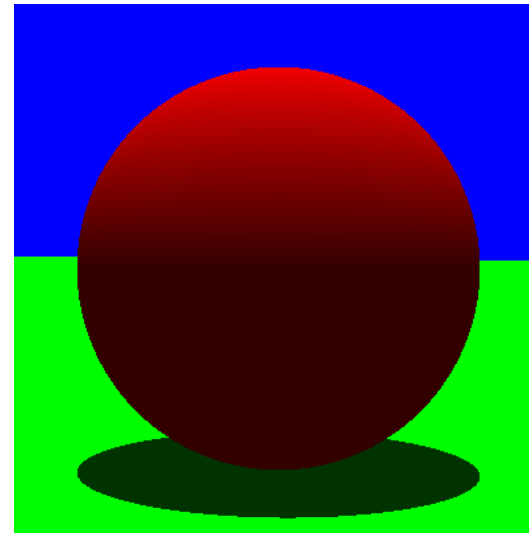
- Was passiert, wenn der Ursprung des Strahles auf der Oberfläche eines Objektes sitzt?
- Floating-Point ist immer unexakt!
  - Folge: bei den folgenden Strahltests erscheint dieser Ursprung **innerhalb** des Objektes!
  - Folge: als nächsten Schnittpunkt erhalten wir denselben Punkt wieder!
- "Lösung":  
verschiebe Aufpunkt des Strahls  
immer zuerst um ein  $\epsilon$  in Richtung  
des Strahls



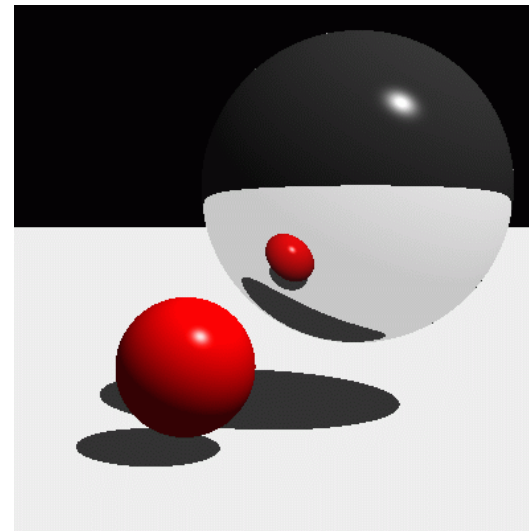
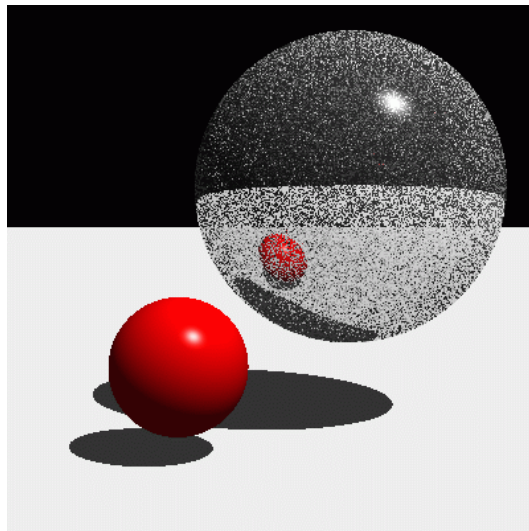




Without epsilon



With epsilon

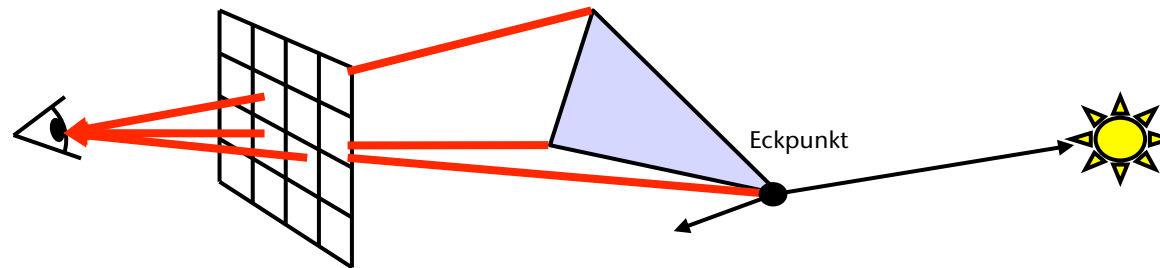




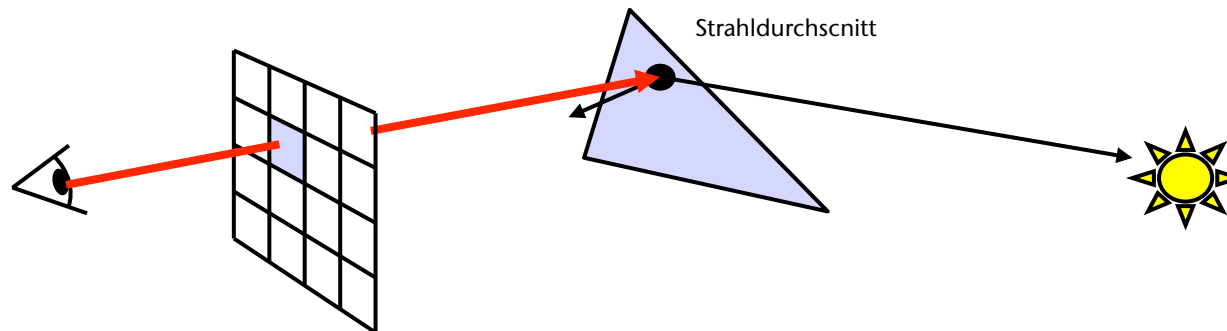
# Scankonvertierung vs. Raytracing



- Scan-Konvertierung: Auswerten eines Strahls, der durch jeden Eckpunkt eines Objektes gesendet wird

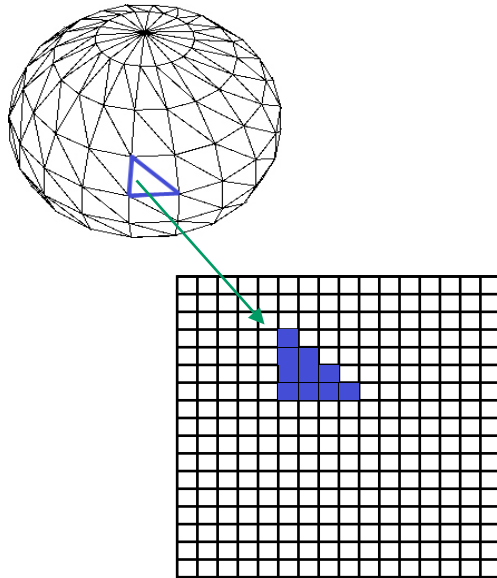


- Raytracing: Auswerten eines Strahls, der durch einen Bildschirmpixel gesendet wird



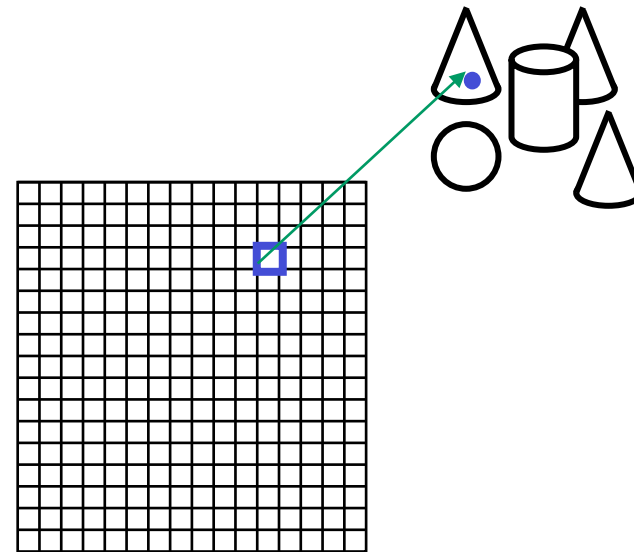


- Zum Umwandeln einer Szene mittels Scan-Konvertierung ...



... scan-konvertiere jedes Dreieck

- Zum Umwandeln einer Szene mittels Raytracing ...



... verfolge für jedes Pixel einen Strahl



# Vor- und Nachteile

- Scan-Konvertierung:
  - schnell (da nur Eckpunkte)
  - wird unterstützt von aktuellen Grafikkarten
  - geeignet für Echtzeitanwendungen
  - ad-hoc Lösung für Schatten, Transparenz
  - Keine Interreflexion
- Raytracing:
  - noch rel. langsam (Suche nach Schnittpunkten zwischen Strahlen und Objektprimitiven)
  - bisher von keiner kommerziellen Hardware unterstützt
  - Offline-Rendering-Verfahren
  - Allgemeine Lösung für Schatten, Transparenz und Interreflexion, Clipping und Culling



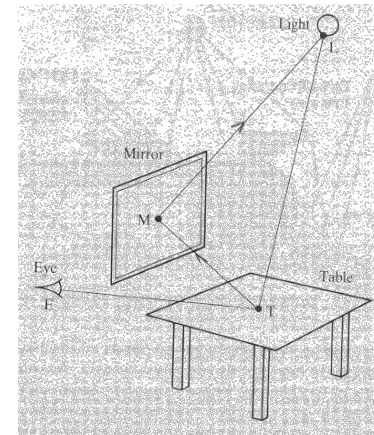


- Weitere Vorteile des Raytracings:
  - Eignet sich besonders für Szenen mit hohem spiegelndem und transparentem Flächenanteil
  - Kann beliebige Objektrepräsentationen verarbeiten (z.B. CSG, Rauch, ...)
    - Einzige Anforderung: man muß Schnitt zwischen Strahl und Objekt und die Normale in diesem Schnittpunkt berechnen können
  - Keine explizite perspektivische Transformation oder Clipping nötig



## Nachteile des (einfachen) Raytracings

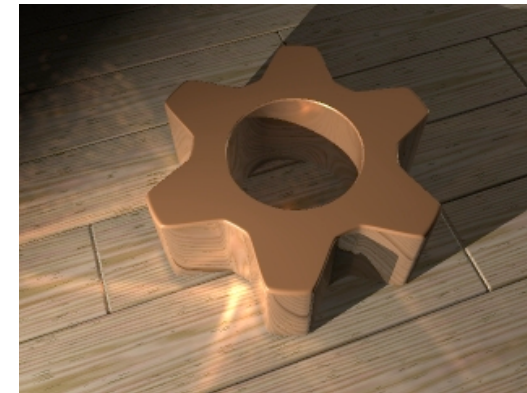
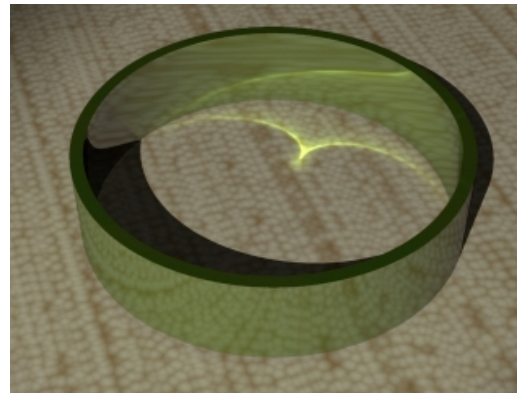
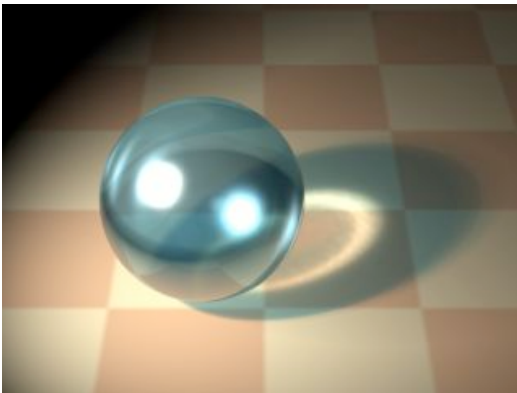
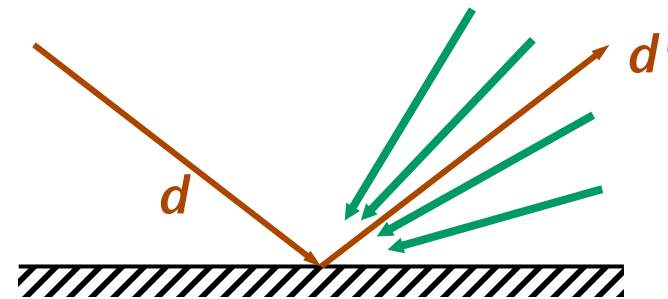
- Sehr viele Strahlen
  - Naives Ray-Casting:  $O(p \cdot n \cdot l)$ ,  $p = \# \text{ Pixel}$ ,  $n = \# \text{ Polygone}$ ,  $l = \# \text{ Lichtquellen}$
  - Anzahl Strahlen wächst exponentiell mit Rekursionstiefe!
- Keine indirekte Beleuchtung (z.B. durch Spiegel)
- Keine weichen Halbschatten
- Shading muß bei jeder Änderung der Kamera neu berechnet werden, obwohl diese nur von den Lichtquellen und den Objekten abhängen
- Für alle diese Nachteile wurden natürlich verschiedene Abhilfen vorgeschlagen





## Beispiel für das Problem der indirekten Beleuchtung: Kaustiken

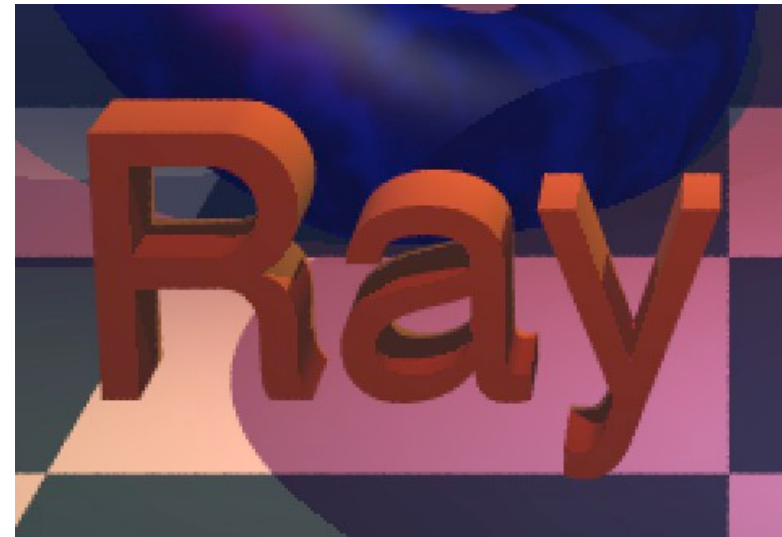
- Konzentration von Licht
- Lichtstrahlen treffen sich in einem Punkt
- Raytracing wird uneffektiv
- Nur 1 reflektierter Strahl wird betrachtet





# Aliasing

- Ein Strahl pro Pixel → typ. Aliasing-Artefakte:
  - Treppeneffekte
  - Moiré- Effekt

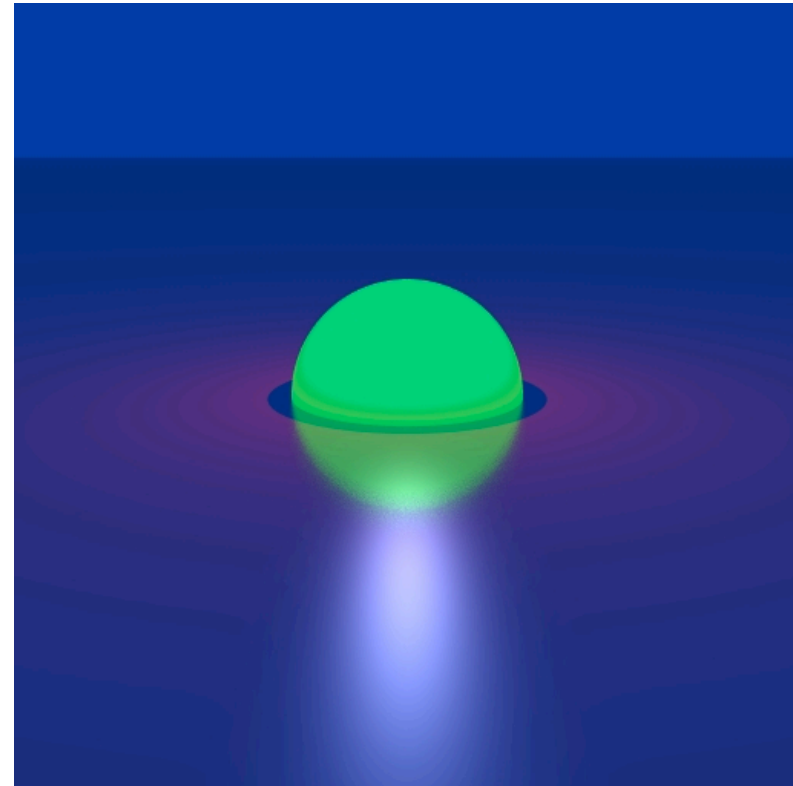






# Distribution Ray Tracing

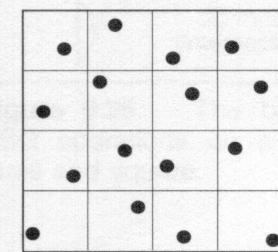
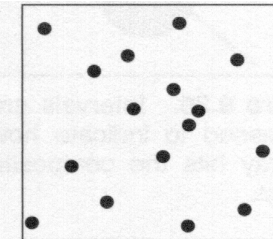
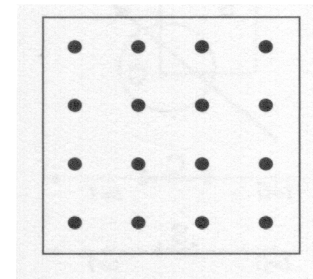
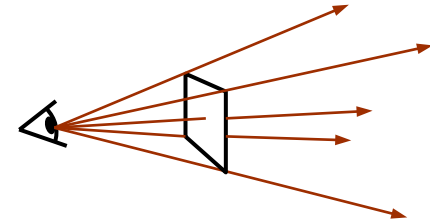
- Einfache Modifikationen des rekursiven Ray Tracings für
  - Antialiasing
  - Weiche Schatten
  - Tiefenschärfe
  - Spekulare Reflexion
- Bewegungsunschärfe
- Anderer Name früher:
  - „Distributed Ray Tracing“
  - ist aber sehr unglücklich ("distributed" = verteilt)





# Anti-Aliasing beim Ray-Tracing

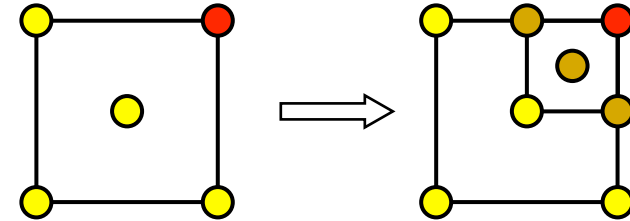
- Anstatt pro Pixel nur einen Strahl zu verfolgen werden mehrere Strahlen verfolgt und die resultierende Farbe gemittelt
- Methoden zur Auswahl der Punkte:
  - **Regelmäßige Abtastung** (Problem der Moire Muster)
  - **Zufällige Abtastung** (Problem des Rauschens)
  - **Stratifikation**, d.h. eine Kombination von regelmäßiger und zufälliger Abtastung, indem ein regelmäßiges Gitter zufällig gestört wird.



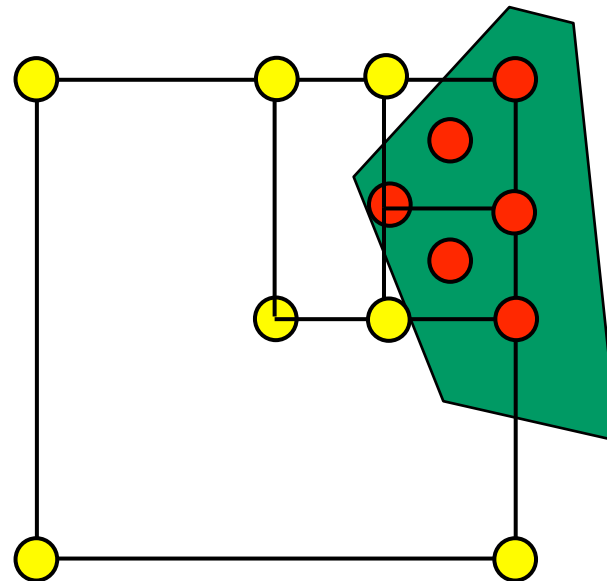


# Adaptives Supersampling

- Idee: verschieße Lichtstrahlen nur bei großen Farbunterschieden bei großen Farbunterschieden



- Beispiel:



- Resultierende Farbe = Durchschnittsfarbe aller Samples, gewichtet mit dem Flächenanteil des Pixels, den das Sample "überdeckt"



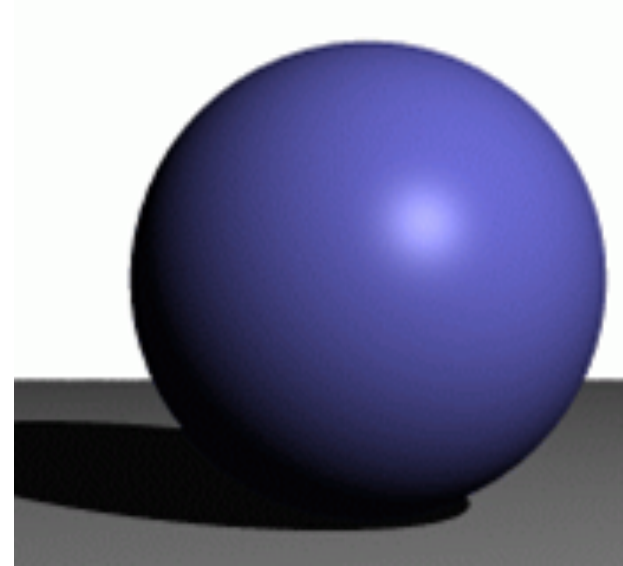
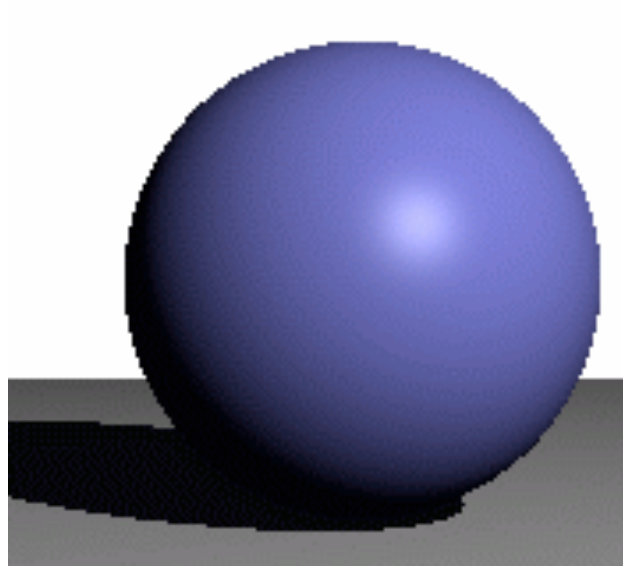
# Beispiel



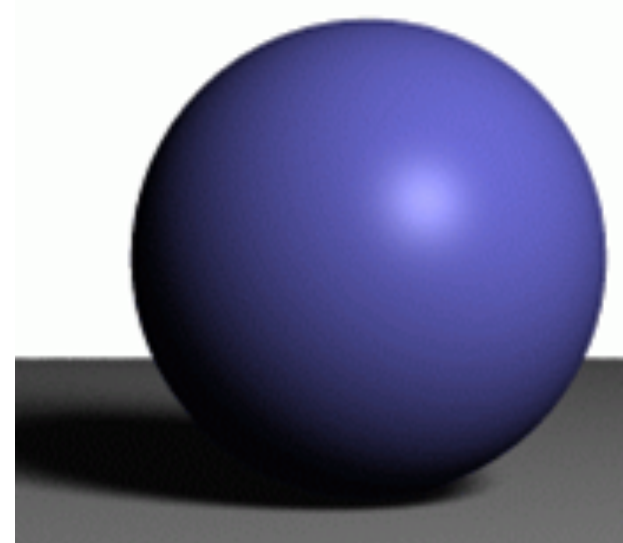
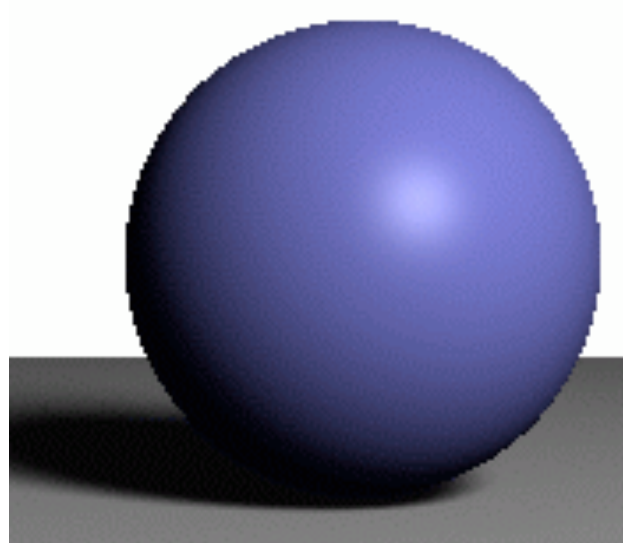
Aliasing

mit Anti-Aliasing

Punktlicht-  
quelle



Ausgedehnte  
Lichtquelle

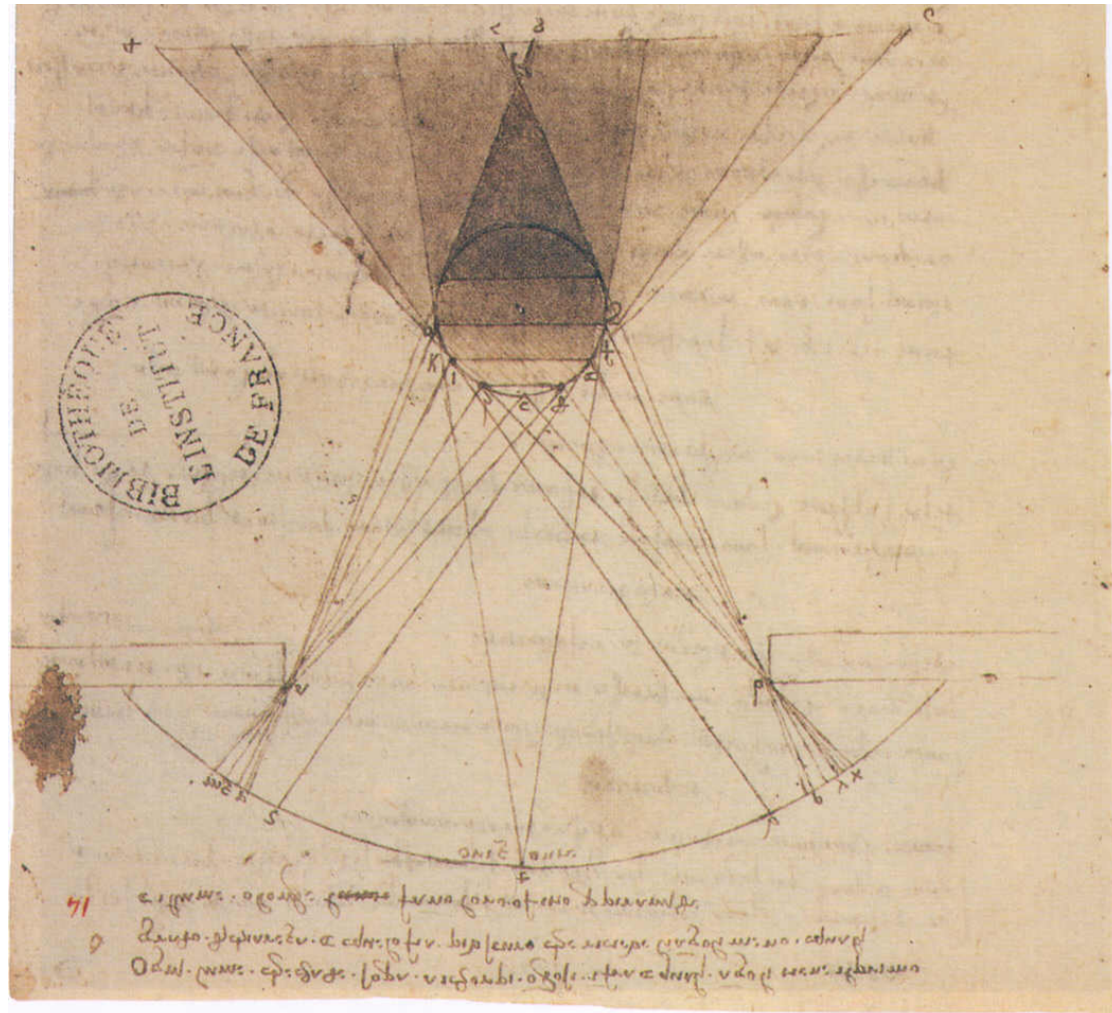




# Weiche Schatten, Halbschatten



- Regionen:
  - "Vollschatten" (*umbra*)
  - Halbschatten (*penumbra*)
  - voll beleuchtet



XVI. Léonard de Vinci (1452-1519). Lumière d'une fenêtre sur une sphère ombreuse avec (en partant du haut) ombre intermédiaire, primitive, dérivée et (sur la surface, en bas) portée. Plume et lavis sur pointe de métal sur papier, 24 x 38 cm. Paris, Bibliothèque de l'Institut de France (ms. 2185; B.N. 2038. f° 14 r°).





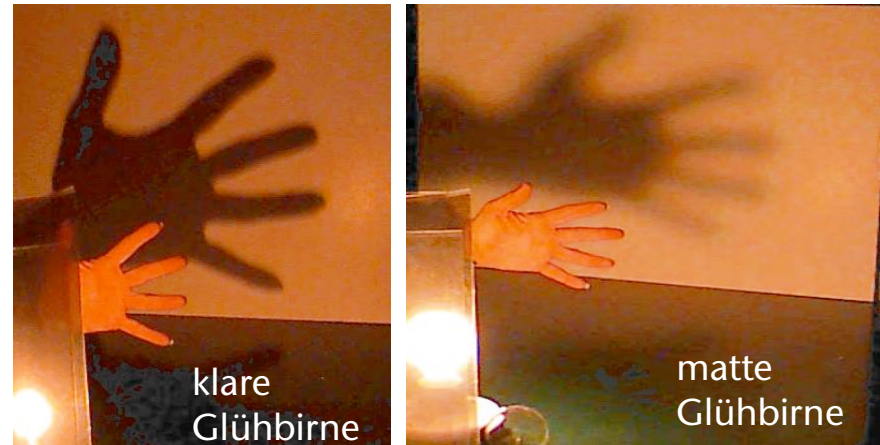
# In der Realität ...



[http://3media.initialized.org/photos/2000-10-18/index\\_gall.htm](http://3media.initialized.org/photos/2000-10-18/index_gall.htm)



<http://www.davidfay.com/index.php>



<http://www.pa.uky.edu/~sciworks/light/preview/bulb2.htm>



## ... und im Ray-Tracing

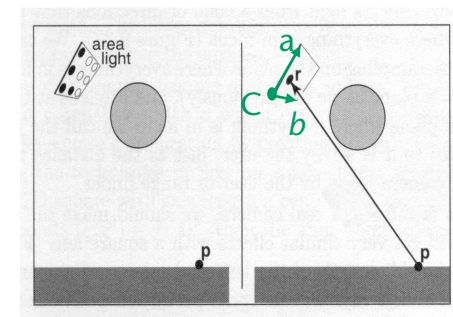
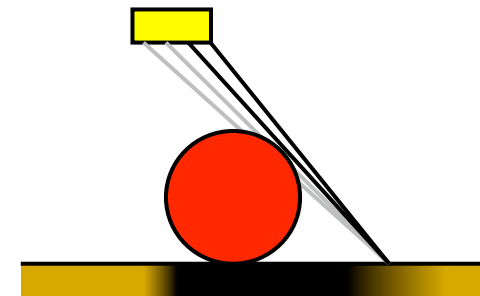
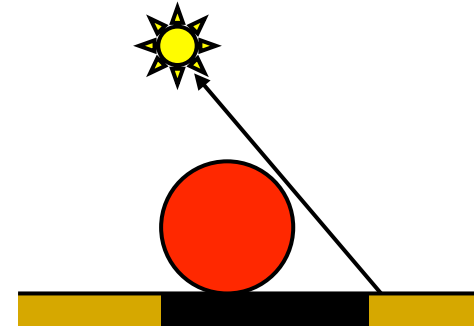
- Bisher: 1 Schattenstrahl

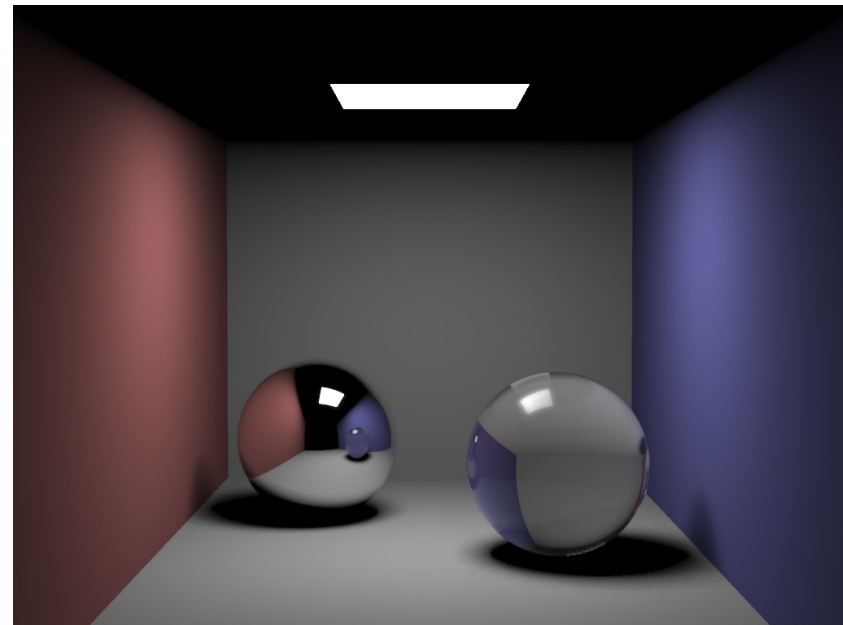
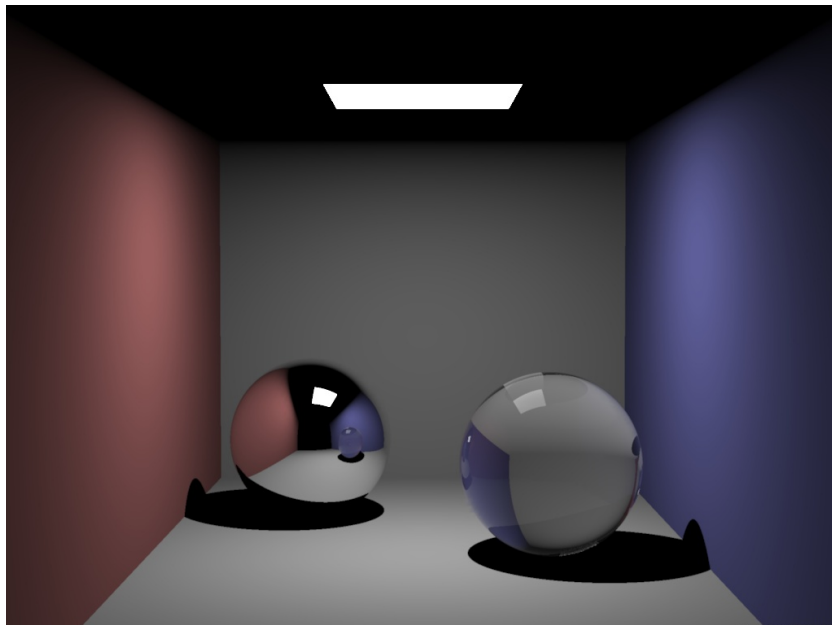
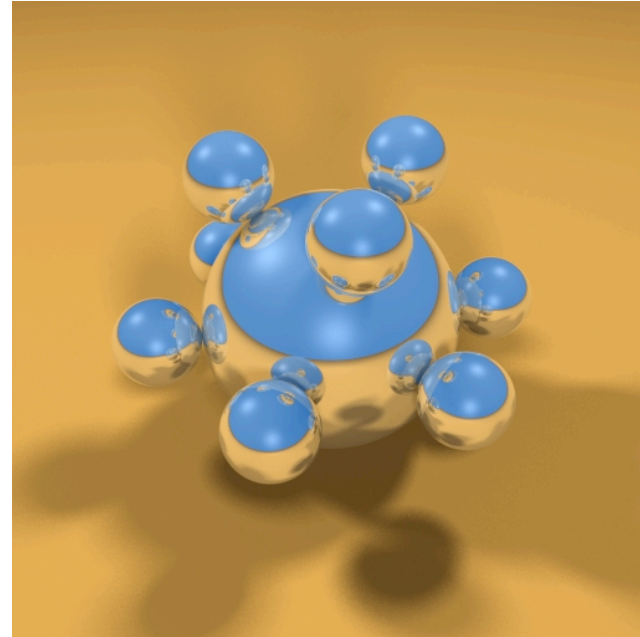
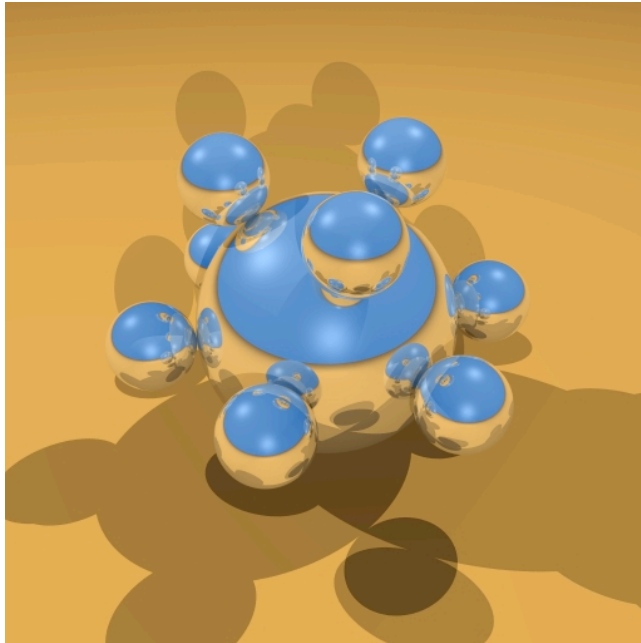
$$s_i = \begin{cases} 1, & \text{Lichtquelle sichtbar} \\ 0, & \text{nicht sichtbar} \end{cases}$$

- Jetzt: mehrere Schattenstrahlen

$$s_i = \frac{\text{Anzahl sichtbarer Samples}}{\text{Anzahl Schattenstrahlen}}$$

- Drei Arten von Sampling der Lichtquelle:
  - Regelmäßige Abtastung der Lichtquelle
  - Zufällige Abtastung der Lichtquellen
  - Stratifizierte Abtastung









- Konstruktion des Schatten- und Halbschattenbereiches:

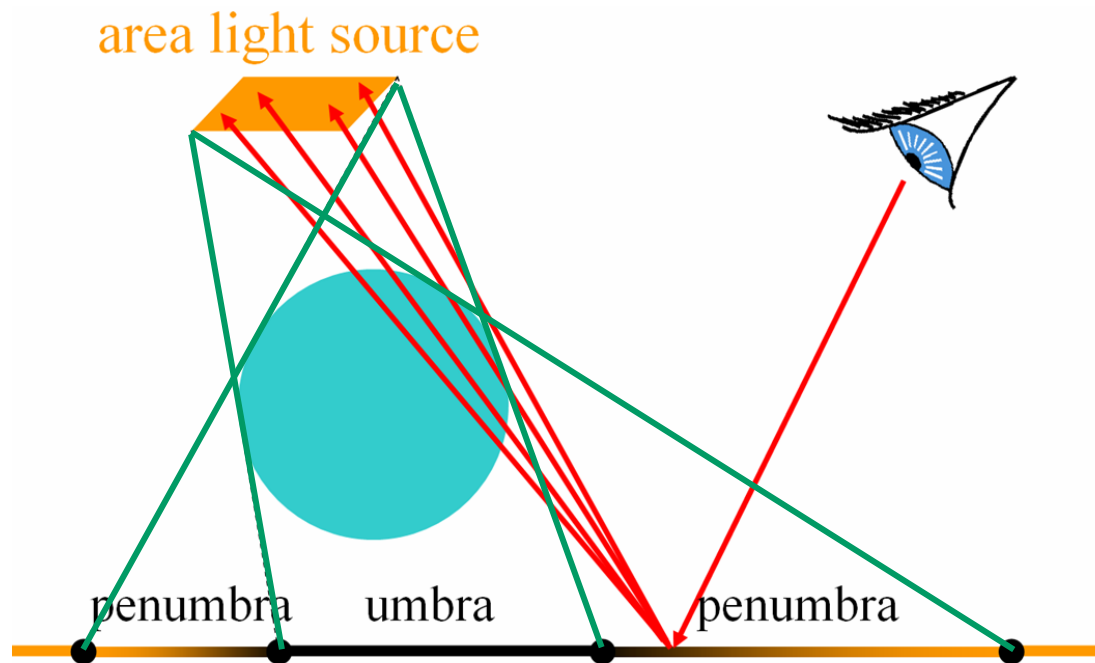
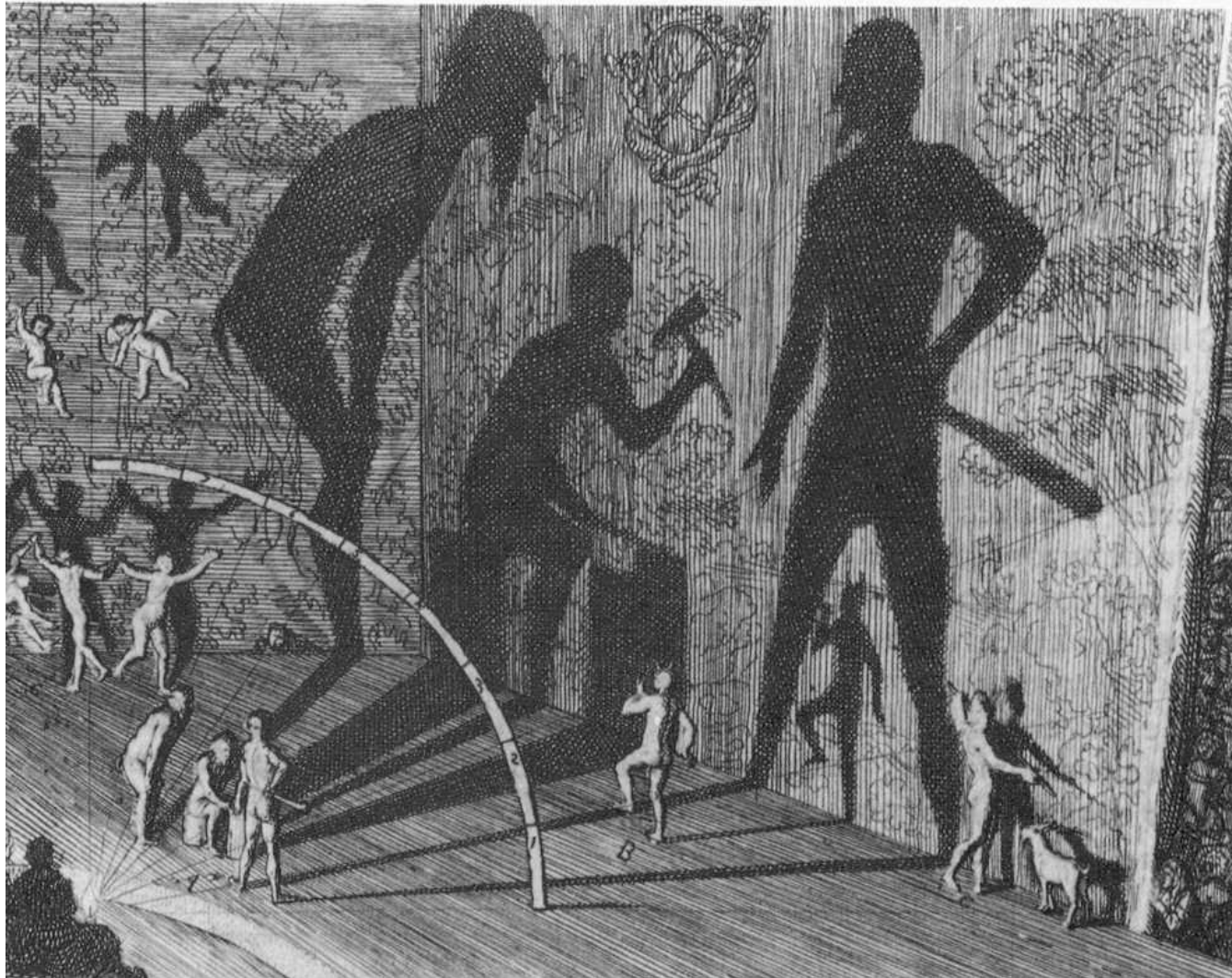




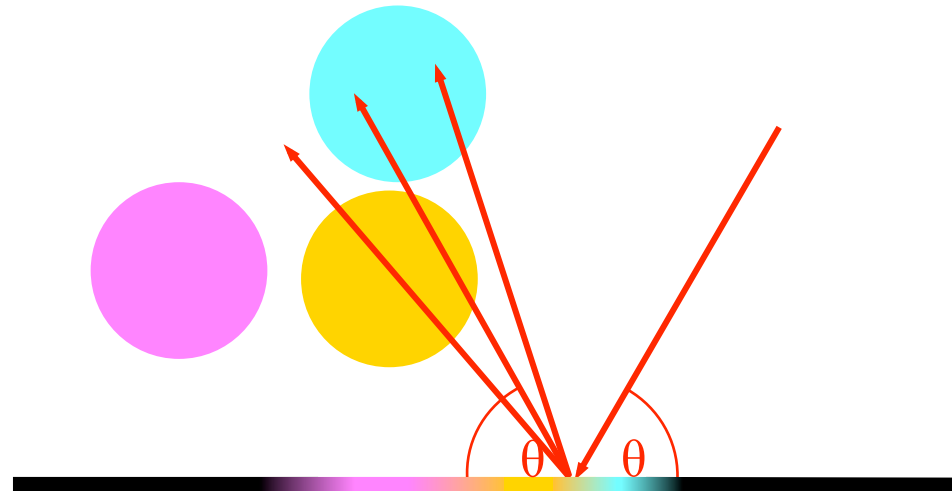
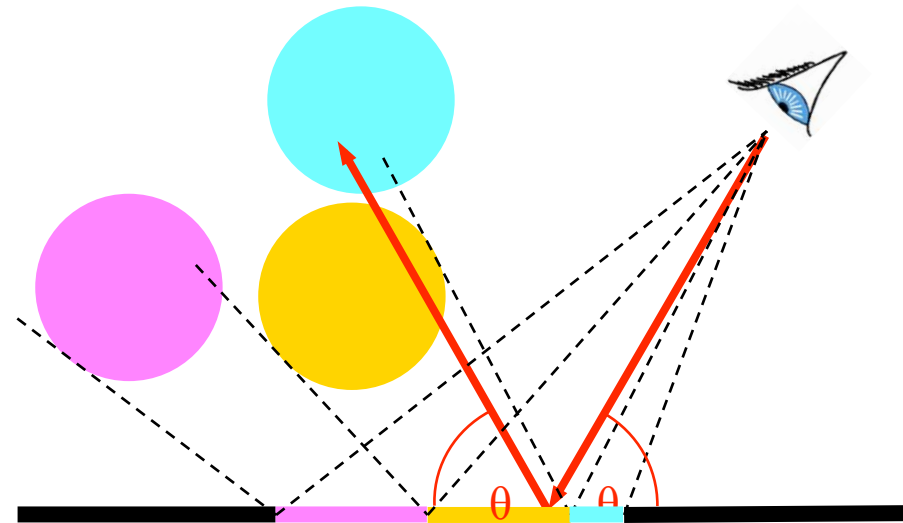
Plate 50 Samuel van Hoogstraten, *Shadow Theatre*. From *Inleyding tot de hooghe schoole der schilderkonst* 1678. (Einführung in die hohe Schule der Gemäldemalerei)





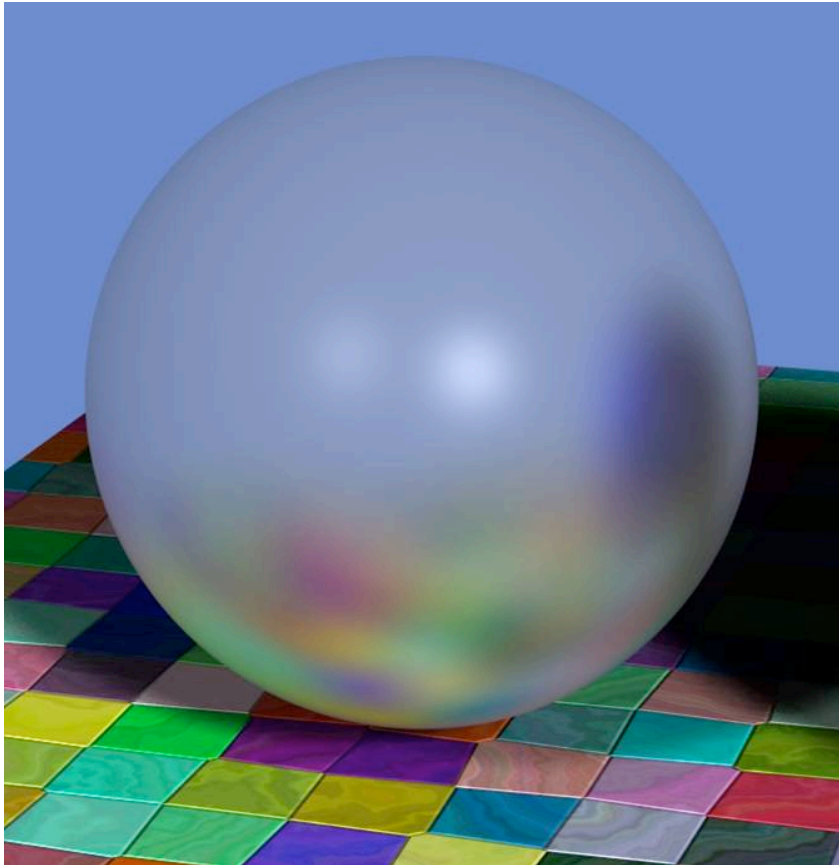
# Spekulare Reflexion

- Bisher: genau 1 reflektierter Strahl
  - Problem, falls die horizontale Fläche matt glänzend sein soll ...
- Lösung (brute-force):
  - Viele reflektierte Strahlen
  - Beiträge gemäß Kosinus-Hoch-n-Gesetz (Phong) aufaddieren

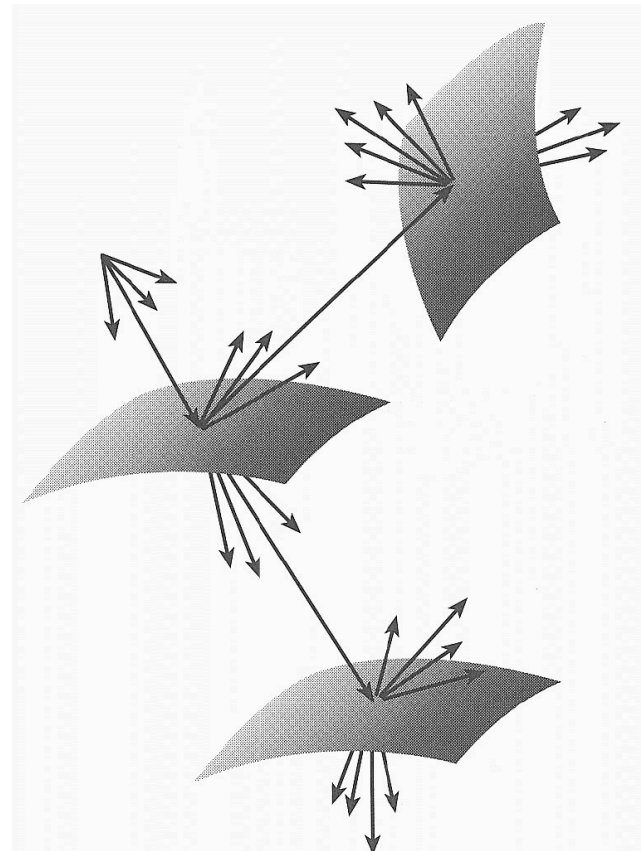




■ Beispiel:



■ Strahlbaum:

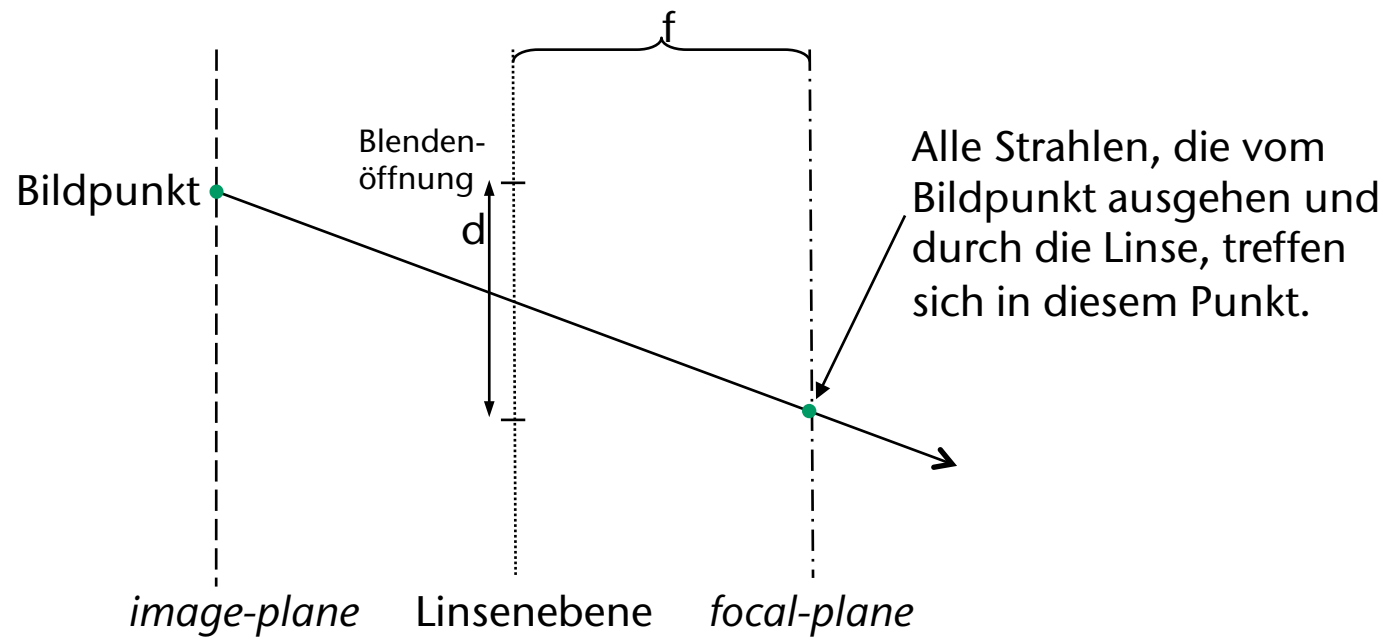
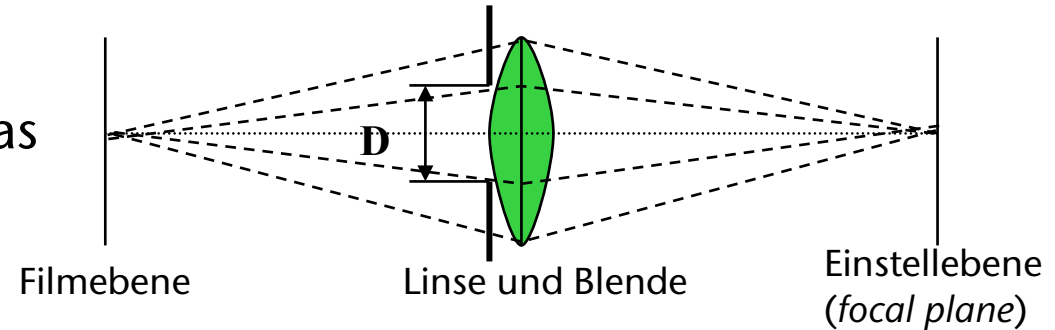




# Tiefen(un-)schärfe



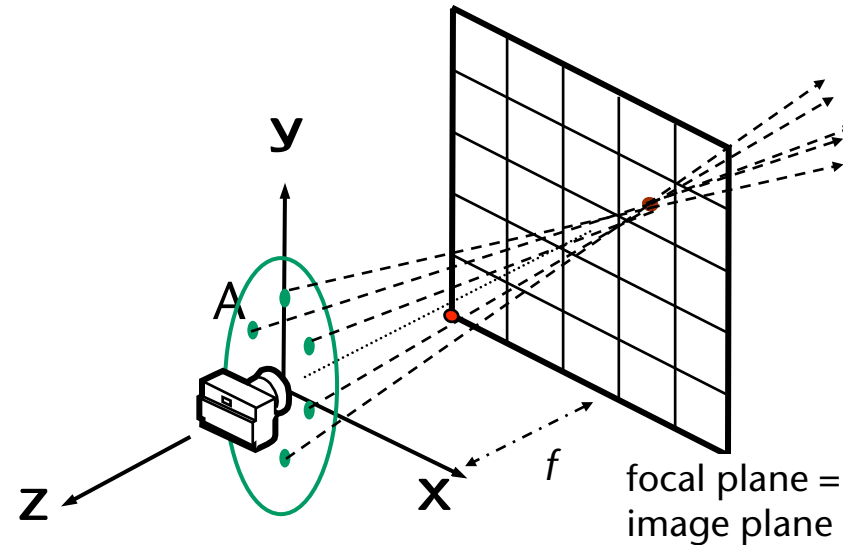
- Bisher: ideales Lochkammermodell
- Für Tiefenunschärfe muß man reale Kameras modellieren





- Eine Klasse **LensCamera** würde die Strahlen also ungefähr so erzeugen:

- Verteile Strahlen auf die gesamte Blendenöffnung und mitte



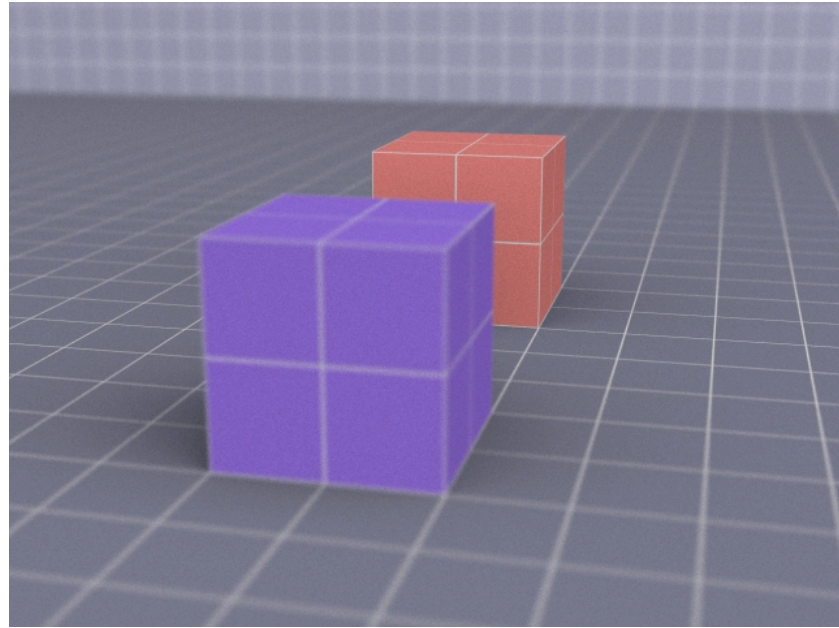
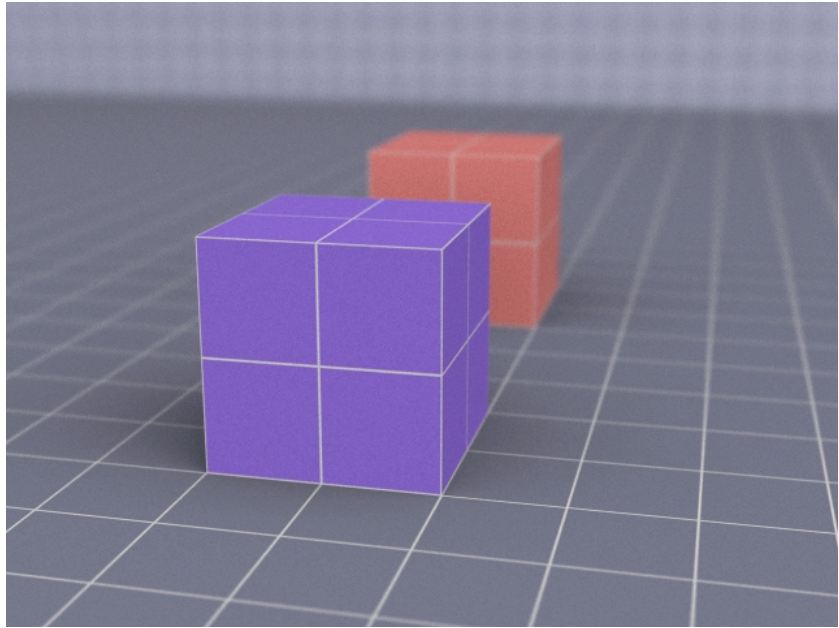
- Bemerkungen:

- Sample die Scheibe (=Linse) stratifiziert
- Achtung bei Kombination mit Anti-Aliasing





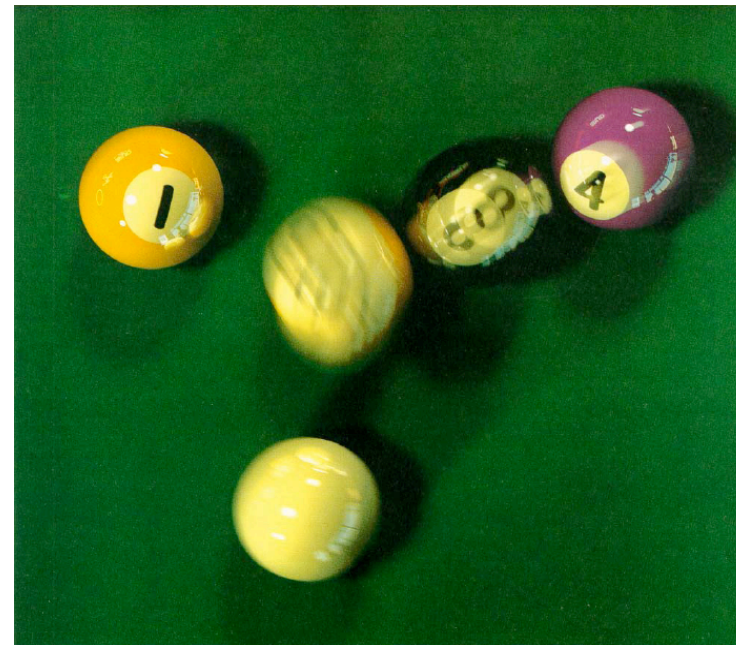
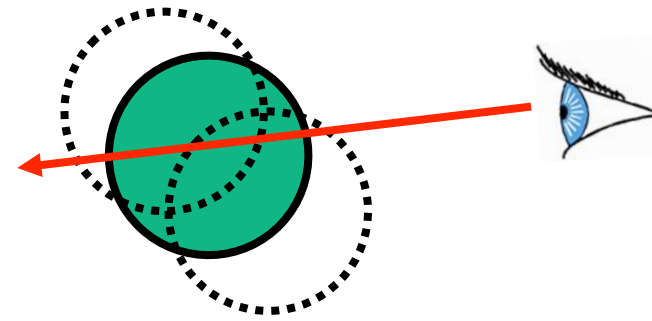
# Beispiele





# Motion Blur (Bewegungsunschärfe)

- Schieße viele Strahlen pro Pixel
- Wähle für jeden Strahl einen Zeitpunkt  $t \in [t_0, t_1]$
- Betrachte während des Schnitttests mit diesem Strahl alle Objekte an ihren Positionen  $P = P(t)$  zu diesem Zeitpunkt  $t$
- Middle Pixelwerte

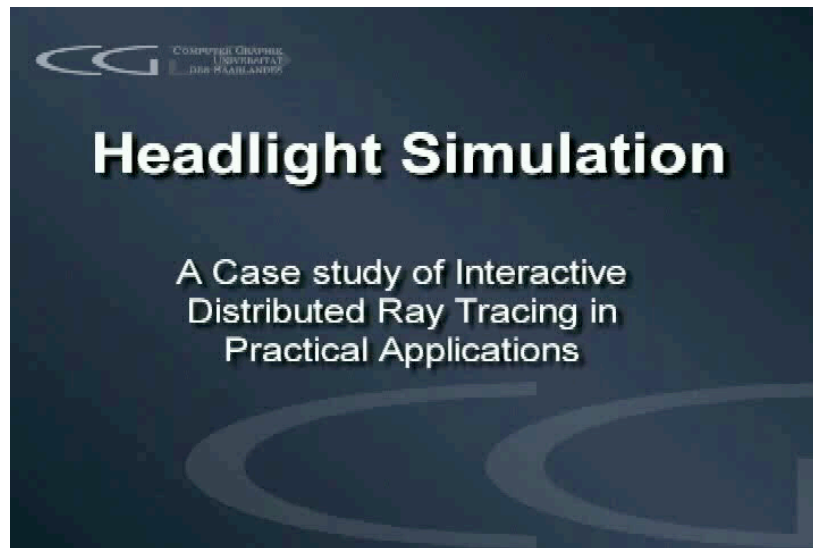




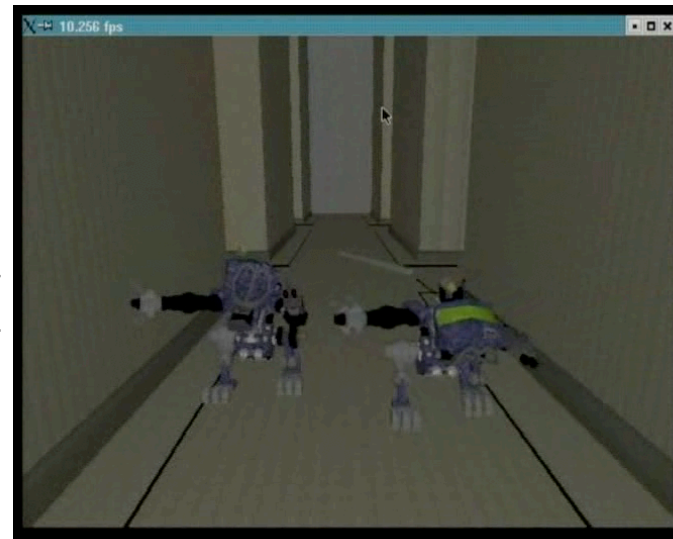


# "But is it real-time?"

- Ray Tracing in der Vergangenheit war sehr langsam
- Inzwischen Echtzeit-Fähigkeit für einige Szenen
- OpenRT-Projekt: Real-Time Ray Tracing
  - Siehe <http://www.openrt.de>
- Special-Purpose-Hardware, PC-Cluster
- Nur eine Frage der Zeit, bis Commodity-Graphics-Hardware es kann



Uni Saarbrücken





# Ray tracing in Egoshooters

Example: Quake3 Demo

<http://graphics.cs.uni-sb.de/~sidapohl/egoshooter/>



Quake 3 mit Ray-Tracing. Plattform: Cluster mit 20 AMD XP1800  
<http://graphics.cs.uni-sb.de/~sidapohl/egoshooter/>



# Eine Anmerkung zu Typos

- Typos passieren auch auf den Folien
  - Keine Angst haben zu fragen!
  - Bitte teilen Sie mir Fehler mit
- Typos passieren sogar in Lehrbüchern
  - Ich selbst habe 2 nicht-triviale Fehler im Shirley-Buch, 2-te Auflage gefunden [WS 05/06]
  - Fazit: mitdenken, nicht einfach direkt kopieren

