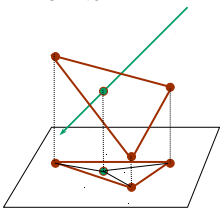


## Schnitt Strahl—Dreieck [Badouel 1990]

- Verwende Methode Strahl—Polygon; oder
- Cleverer sein: baryzentrische Koordinaten + Projektion
- Schneide Strahl mit Ebene (Normalenform)  $\rightarrow t \rightarrow$  Punkt
- Projiziere Punkt & Dreieck in Koord.ebene
- Berechne baryzentrische Koord. des 2D-Punktes
- Baryzentrische Koord. des 2D-Punktes = baryzentrische Koord. des 3D-Punktes!
- 3D-Punkt in Dreieck  $\Leftrightarrow \alpha, \beta, \gamma > 0, \alpha + \beta + \gamma < 1$
- Alternative Methode: siehe Möller & Haines "Real-time Rendering"
- Code: <http://jgt.akpeters.com/papers/MollerTrumbore97/>
- Geht noch schneller, falls Schnittpunkt nicht nötig [Segura & Feito]



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 48

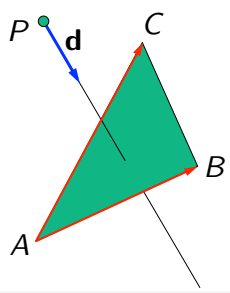
## Alternative Schnittberechnung Strahl—Dreieck [Möller]

- Geradengleichung  $X = P + t \cdot \mathbf{d}$
- Ebenengleichung  $X = A + r \cdot (B - A) + s \cdot (C - A)$
- Gleichsetzen  $-t \cdot \mathbf{d} + r \cdot (B - A) + s \cdot (C - A) = P - A$
- In Matrixschreibweise  $\begin{pmatrix} -\mathbf{d} & B - A & C - A \end{pmatrix} \cdot \begin{pmatrix} t \\ r \\ s \end{pmatrix} = P - A$

$$\mathbf{u} = B - A$$

$$\mathbf{v} = C - A$$

$$\mathbf{w} = P - A$$



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 49

$$\begin{pmatrix} t \\ r \\ s \end{pmatrix} = \frac{1}{\det(-\mathbf{d}, \mathbf{u}, \mathbf{v})} \cdot \begin{pmatrix} \det(\mathbf{w}, \mathbf{u}, \mathbf{v}) \\ \det(-\mathbf{d}, \mathbf{w}, \mathbf{v}) \\ \det(-\mathbf{d}, \mathbf{u}, \mathbf{w}) \end{pmatrix}$$

$$\det(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \mathbf{a} \cdot (\mathbf{b} \times \mathbf{c})$$

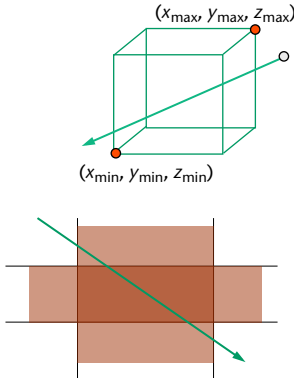
$$\begin{pmatrix} r \\ r \\ s \end{pmatrix} = \frac{1}{(\mathbf{d} \times \mathbf{v}) \cdot \mathbf{u}} \cdot \begin{pmatrix} (\mathbf{w} \times \mathbf{u}) \cdot \mathbf{v} \\ (\mathbf{d} \times \mathbf{v}) \cdot \mathbf{w} \\ (\mathbf{w} \times \mathbf{u}) \cdot \mathbf{d} \end{pmatrix}$$

- Kosten: 2 Kreuzprodukte + 4 Skalarprodukte
- Liefert: Geradenparameter + baryzentrische Koordinaten bzgl. Dreieck
- Test ob s,t im Bereich (0,1) und s+t <= 1

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 50

## Schnitt Strahl – Box

- Box (Quader) wird später noch wichtig als Bounding Box
- Hier: nur achsenparallele Boxes (AABB = *axis-aligned bounding box*)
- Definition einer AABB: durch die zwei extremen Eckpunkte  $(x_{\min}, y_{\min}, z_{\min})$  und  $(x_{\max}, y_{\max}, z_{\max})$
- Idee des Algo:
  - Eine Box ist der Schnitt von 3 *Slabs* (ein *Slab* = Schicht des Raumes, wird von 2 parallelen Ebenen begrenzt)
  - Jeder Slab schneidet vom Strahl ein Intervall heraus
  - Betrachte also sukzessive jeweils Paare von Box-Seiten



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 52

Der Algorithmus:

```

setze  $t_{\min} = -\infty$  ,  $t_{\max} = +\infty$ 
loop über alle Paare von Ebenen:
  schneide Strahl mit den
    beiden Ebenen  $\rightarrow t_1$  ,  $t_2$ 
  if  $t_2 < t_1$ :
    vertausche  $t_1$  ,  $t_2$ 
  // jetzt gilt:  $t_1 < t_2$ 
   $t_{\min} \leftarrow \max(t_{\min}, t_1)$ 
   $t_{\max} \leftarrow \min(t_{\max}, t_2)$ 
  // now:  $[t_{\min}, t_{\max}] = \text{interval inside box}$ 
  if  $t_{\min} > t_{\max} \rightarrow \text{kein Schnitt}$ 
  if  $t_{\max} < 0 \rightarrow \text{kein Schnitt}$ 

```

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 53

Bemerkungen

- Optimierung: beide Ebenen eines Slabs haben dieselbe Normale
  - Spart ein Skalarprodukt
- Bemerkung: der Algo funktioniert genauso für "schiefe" Boxes (sog. *OBBs = oriented bounding boxes*)
- Weitere Optimierung: falls AABB, nutze aus, daß die Normalen nur 1 Komponente  $\neq 0$  haben!
- Achtung: "shit happens"
  - Hier: teste auf Parallelität!
  - Im Fall der AABB:

```

if  $|d_x| < \epsilon$ :
  if  $p_x < x_{\min} \parallel p_x > x_{\max}$ :
    Strahl geht an Box vorbei
  else:
     $t_1, t_2 = y_{\min}, y_{\max}$  // evtl noch swappen!

```

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 54

## Schnitt Strahl—Kugel

- Annahme:  $\mathbf{d}$  ist normiert
- Die geometrische Methode:
 
$$|t \cdot \mathbf{d} - \mathbf{m}| = r$$

$$(t \cdot \mathbf{d} - \mathbf{m})^2 = r^2$$

$$t^2 - 2t \cdot \mathbf{m} \cdot \mathbf{d} + \mathbf{m}^2 - r^2 = 0$$
- Die algebraische Methode:
  - Punkt auf Strahl in implizite Kugelgleichung einsetzen
- Es gibt noch andere Ansätze ...

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 55

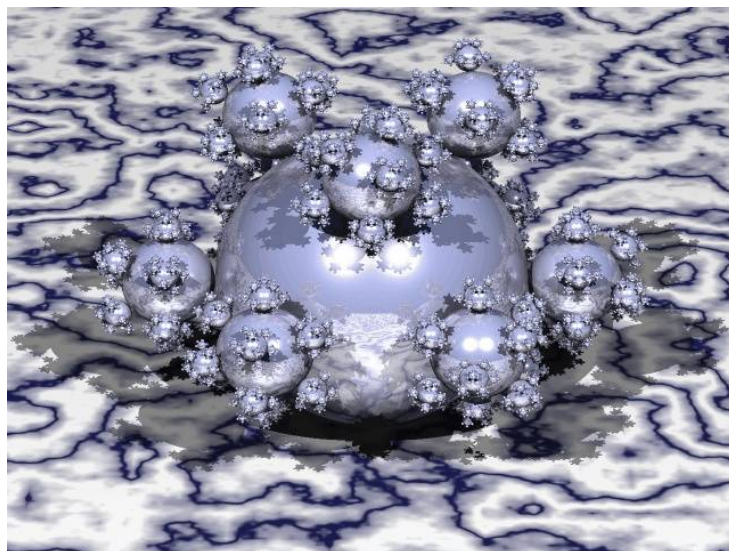
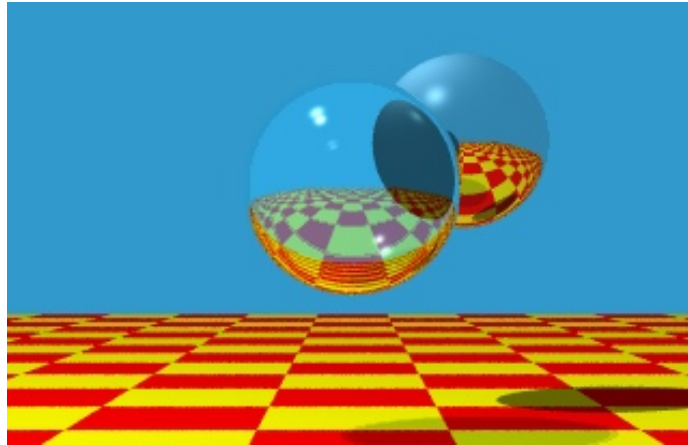
- Der Algorithmus, mit kleinen Optimierungen:

```

berechne  $\mathbf{m}^2 - r^2$ 
berechne  $b = \mathbf{m} \cdot \mathbf{d}$ 
if  $\mathbf{m}^2 - r^2 >= 0$  // Blickpunkt ausserhalb Kugel
  and  $b <= 0$  : // sieht von Kugel weg
  then
    return "kein Schnittpunkt"
  setze  $d = b^2 - \mathbf{m}^2 + r^2$ 
  if  $d < 0$ :
    return "kein Schnittpunkt"
  if  $\mathbf{m}^2 - r^2 > \epsilon$  :
    return  $t_1 = b - \sqrt{d}$  // enter;  $t_1$  is  $> 0$ 
  else:
    return  $t_2 = b + \sqrt{d}$  // leave;  $t_2$  is  $> 0$  ( $t_1 < 0$ )
  
```

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 56

- Es ist so einfach, daß alle Ray-Tracer Kugeln als Primitiv haben!



Die sog. "sphere flake"

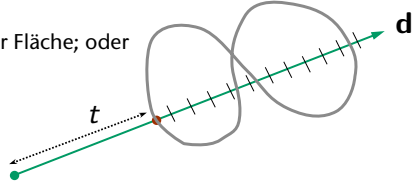
## Geometrisch vs. Algebraisch

- Die algebraische Methode ist einfach und generisch
- Die geometrische Methode ist schneller
  - Durch geometrische Einsicht
  - Frühe Tests
  - Insbesondere für die Strahlen, die weg zeigen

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 59

## Schnitt Strahl – implizite Fläche

- Implizite Fläche vom Grad  $n$ :  $F(x, y, z) = 0$
- Strahl:  $P(t) = O + t \cdot \mathbf{d}$
- Einsetzen:  $F(P(t)) = 0$   
liefert Polynom in  $t$  vom Grad  $n$
- Nullstellensuche:
  - Falls Grad  $< 5$ : nach  $t$  auflösen
  - Sonst: Intervallschachtelung, Newton-Verfahren, ...
  - Startwerte:
    - Schnitt zwischen Strahl und BBox der Fläche; oder
    - Strahl innerhalb der BBox abtasten
- Z.B.: Kugel ...



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 60

## Nullstellensuche mit Laguerre's Methode

- Eine von wenigen "*sure-fire*"-Methoden
- Algorithmus erfordert Arithmetik mit komplexen Zahlen, auch wenn alle Wurzeln reell sind (und damit auch alle Koeffizienten)
- Sehr wenig theoretische Erkenntnisse zum Konvergenzverhalten
- Sehr viel empirische Hinweise, daß Algo (fast) **immer** zu einer Wurzel konvergiert, und zwar von (fast) **jedem** Startwert aus!
- Konvergenz-Ordnung ist 3, falls die Wurzel einfach ist

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 61

## Motivation für den Algorithmus

- Gegeben (die  $x_i$  sind – evtl. komplexe – Wurzeln):
 
$$P(x) = (x - x_1)(x - x_2) \dots (x - x_n) \quad (0)$$
- Beziehungen:
 
$$\ln |P(x)| = \ln |x - x_1| + \ln |x - x_2| + \dots + \ln |x - x_n|$$

$$\frac{d}{dx} \ln |P(x)| = \frac{1}{x - x_1} + \dots + \frac{1}{x - x_n} = \frac{P'(x)}{P(x)} =: G \quad (1)$$

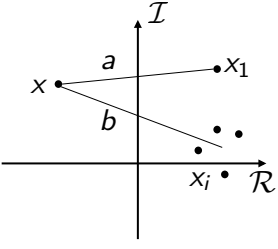
$$\frac{d^2}{dx^2} \ln |P(x)| = -\frac{1}{(x - x_1)^2} - \dots - \frac{1}{(x - x_n)^2}$$

$$= \frac{P''(x)}{P(x)} - \left( \frac{P'(x)}{P(x)} \right)^2 =: -H \quad (2)$$

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 62

- Sei  $x$  unsere aktuelle Näherung an Wurzel  $x_1$
- Eine "drastische" Annahme:
  - Abstand  $|x - x_1| = a$
  - Abstand zu allen anderen Wurzeln ist
 
$$|x - x_i| \approx b, \quad i = 2, 3, \dots, n$$
- Dann kann man (1) & (2) so darstellen
 
$$G \approx \frac{1}{a} + \frac{n-1}{b} \quad (3)$$

$$H \approx \frac{1}{a^2} + \frac{n-1}{b^2} \quad (4)$$



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 63

- Einsetzen ergibt Lösung für  $a$  :
 
$$a \approx \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}} \quad (5)$$
- VZ so wählen, daß  $|a|$  minimal wird
- Wurzel kann negativ werden
  - $a$  kann komplex werden
- Neue Näherung für  $x_1$  ist
 
$$x_1 = x - a$$

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 64



## Algorithmus

```

wähle  $x_0$ 
repeat
  berechne  $G = \frac{P'(x_i)}{P(x_i)}$ 

  
$$H = G^2 - \frac{P''(x_i)}{P(x_i)}$$


  berechne  $a = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}}$ 

  setze  $x_{k+1} = x_k - a$ 
until a "klein genug" oder  $k \geq \max$ 

```

- Achtung: möglichst Code aus *Numerical Recipes* verwenden
  - Selbst implementieren ist fehlerträchtig
  - NR-Code hat elegantes Abbruchkriterium
- Für Ray-Tracing: alle Nullstellen berechnen
  - Faktorisiere gefundene Nullstelle aus, wiederhole Laguerre  $n$  Mal

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 65

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

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 66

## Weiterer Grund für *Instancing*

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

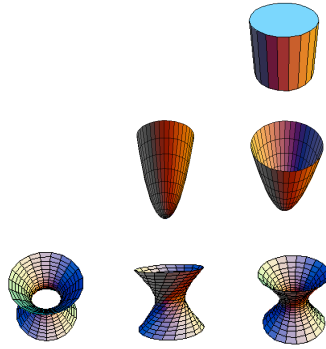


61 unique plant models, 1.1M unique triangles, 300MB —  
4000 plants in the scene, 19.5M triangles

G. Zachmann Computer-Graphik 2 — SS 10 Ray-Tracing 68

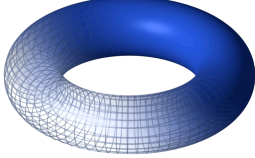
## Weitere Quadriken









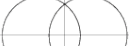
- Mit diesen Techniken kann man viele weitere Objekte testen
  - Parametrische Liniengleichung in implizite Form des Objektes einsetzen
  - Ergibt Polynom in  $t$  vom Grad 2, 4, ...
  - Nach  $t$  lösen (analytisch, Newton, ...)
- Unendlicher Zylinder:
 
$$x^2 + y^2 = 1$$
- Parabolloidd:
 
$$x^2 + y^2 - z = 0$$
- Hyperboloid (*one sheet*):
 
$$x^2 + y^2 - z^2 = 1$$

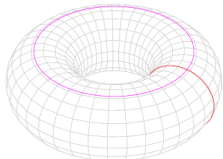


G. Zachmann Computer-Graphik 2 — SS 10 Ray-Tracing 69

■ Torus:

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


	full view	cutaway	cross-section
ring torus			
horn torus			
spindle torus			



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 70

■ Superquadrics

■ Verallgemeinerungen der Quadriken

■ Super-Ellipsoid:

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

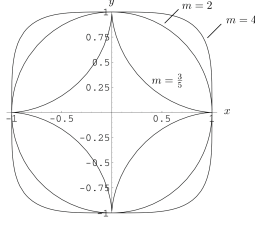
■ Super-Hyperboloid:

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

■ 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!



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 71

### Beispiele von Super-Quadrics

The image shows a 4x4 grid of wireframe models for Super-Quadrics with parameters n1 (0.2, 1.0, 2.0, 3.0) and n2 (0.2, 1.0, 2.0, 3.0). To the right, there are two sets of rendered 3D shapes: a 4x4 grid of blue shapes and a single large red shape. Below the rendered shapes is the text: "XScreenSaver demo 'SuperQuadrics' (www.jwz.org/xscreensaver)".

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 72

### Ratioquadrics [Blanc & Schlick, 1996]

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

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

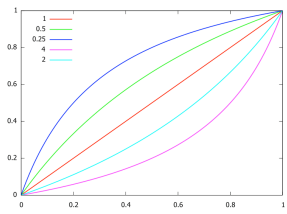
$$f_p(x) = |x|^p$$

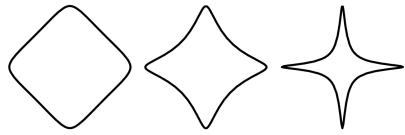
- Problem:
  - $f_p(x)$  ist an der Stelle  $x=0$  nicht differenzierbar für  $p \leq 1$
  - Dadurch entstehen für  $p < 1$  "Spitzen", die möglicherweise unerwünscht sind
  - Außerdem ist  $f_p(x)$  rel. teuer auszuwerten

The graph shows the function  $f_p(x) = |x|^p$  for  $x \in [0, 1]$  and  $p \in \{1, 0.5, 0.25, 0.1, 0.05, 0.01\}$ . The curves show that as  $p$  decreases, the function becomes more peaked at  $x=0$ . Below the graph are three 2D plots: a diamond (square rotated 45 degrees), a four-pointed star, and a cross, illustrating the shapes formed by the Ratioquadric equation for different values of  $p$ .

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 73

- Simple Idee: verwende einfach eine andere "Potenzfunktion"
- Die Pseudo-Potenzfunktion von Blanc & Schlick:
 
$$g_p(x) = \frac{x}{p + (1-p)x}$$



- Die Ratioquadrac 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:
 

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 74

## Metaballs

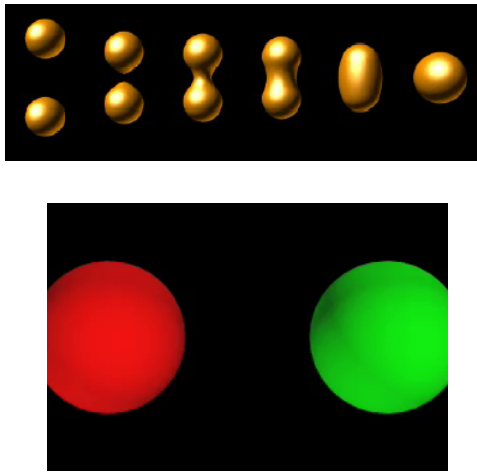
- Abgeschaut von den Molekülen
- Idee: betrachte eine Kugel als Menge aller Punkte im Raum, die dasselbe "Potential" haben, wobei das Maximum des Potentialfeldes im Kugelmittelpunkt herrscht → **Isfläche**
- Potentialfeld wird beschrieben durch Potentialfunktion, z.B.
 
$$p(r) = \frac{1}{r^2}$$
 wobei
 
$$r = r_1(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_1\|$$
- Die Kugelfläche ist damit
 
$$K = \{\mathbf{x} \mid p(\mathbf{x}) = \tau\}$$
  - $\tau$  heißt **Schwellwert** oder **Isowert**

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 75

- 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 (gewichtete) 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, die  $a_i$  (= "Feldstärke") bestimmen jew. Einfluß
  - Negative Feldstärken nehmen "Material" weg (z.B. für Löcher)

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 76

- Beispiel für 2 Skelett-Punkte:



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 77

■ Andere Potentialfunktionen:

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

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

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

G. Zachmann    Computer-Graphik 2 – SS 10 Ray-Tracing    78

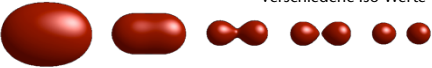
■ Effekt der Variation des Parameters:

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

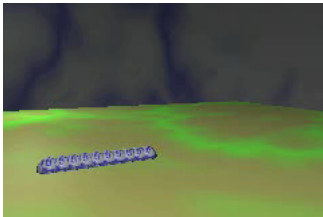
G. Zachmann    Computer-Graphik 2 – SS 10 Ray-Tracing    79

## Deformationen

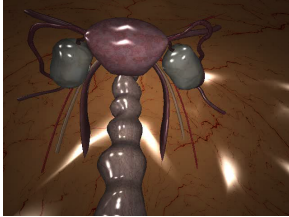
- Mit diesem Modell lassen sich Deformationen von "blob-artigen" Objekten sehr einfach modellieren:
  - Verschiebe Skelett-Punkte
  - Modifiziere Parameter  $a, b, \dots$
  - Modifiziere den Iso-Wert  $t$



Verschiedene Iso-Werte

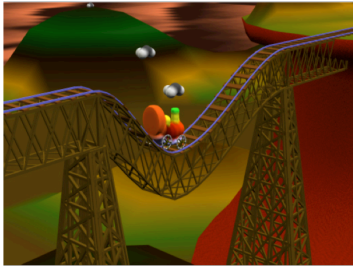


Brian Wyvill  
<http://pages.cpsc.ucalgary.ca/~blob/animations.html>

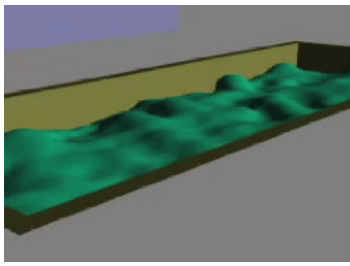


Frédéric Triquet  
<http://www2.lifl.fr/~triquet/implicit/video/>

G. Zachmann Computer-Graphik 2 – SS 10
Ray-Tracing 80





"The Great Train Rubbery" — Siggraph 1986



"Soft"

"The Wyvill Brothers"

Geoff
Brian

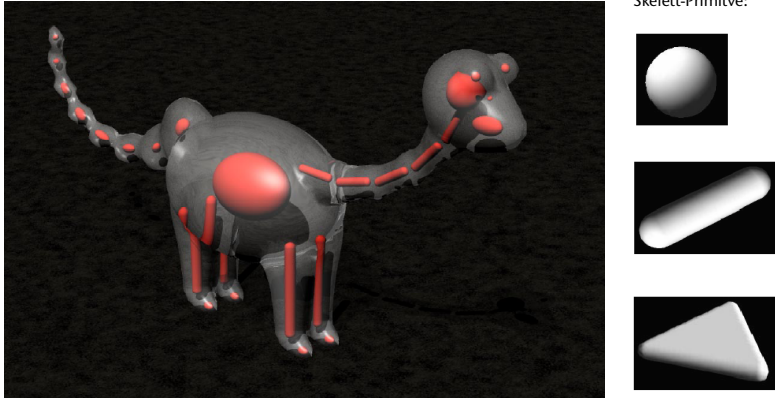
G. Zachmann Computer-Graphik 2 – SS 10
Ray-Tracing 81



Verallgemeinerungen / Varianten

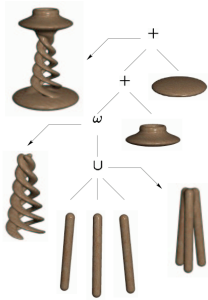
- Punkte sind das einfachste Primitiv zur Konstruktion eines Skeletts; analog kann man Linien, Polygone, Ellipsoide, etc., verwenden:

Beispiele weiterer Skelett-Primitive:



G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 82

- Andere Blending-Funktionen:
 
$$P_U(\mathbf{x}) = \max\{p_1(\mathbf{x}), p_2(\mathbf{x})\}$$

$$P_\cap(\mathbf{x}) = \min\{p_1(\mathbf{x}), p_2(\mathbf{x})\}$$
- Ein Baum von Blending-Operationen – der BlobTree:
 

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 83

## 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]

G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 84

## Normale an impliziten Flächen

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

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

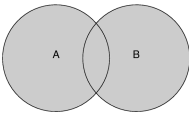
$$\approx \begin{pmatrix} f(x + \varepsilon, y, z) - f(\mathbf{x}) \\ f(x, y + \varepsilon, z) - f(\mathbf{x}) \\ f(x, y, z + \varepsilon) - f(\mathbf{x}) \end{pmatrix}$$

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

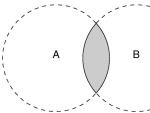
G. Zachmann Computer-Graphik 2 – SS 10 Ray-Tracing 85

## Weitere Objekt-Repräsentation: CSG

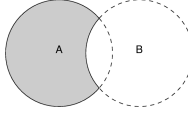
- Fügt sich genauso zwanglos ins Raytracing ein
- Zentrale Idee: konstruiere neue Objekte durch Mengen-Operationen auf einfachen Grund-Volumina (→ CSG = *constructive solid geometry*)
  - Mengen-Operationen: Schnittmenge, Vereinigung, Differenz
  - Grund-Primitive: alle Objekte, die sich leicht implizit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




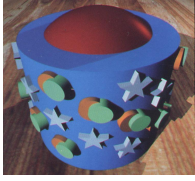
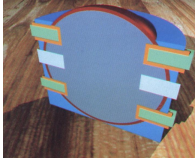
Vereinigung



Schnittmenge



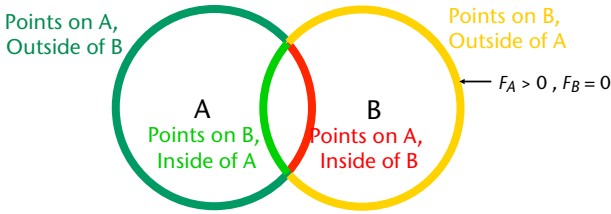
Differenz

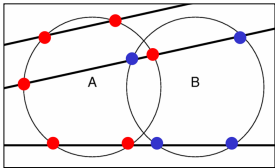
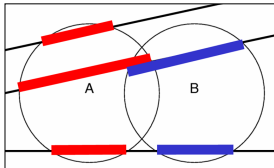
G. Zachmann Computer-Graphik 2 – SS 10
Ray-Tracing 86

## Implementierung

- Verwende implizite Form der Grund-Objekte

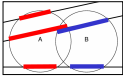


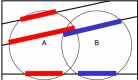
- Bestimme **alle** Schnittpunkte eines Strahls mit allen Grundobj. en
  - Falls alle Grundobj. konvex → 1 Intervall auf dem Strahl pro Grundobj.

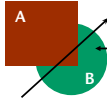
G. Zachmann Computer-Graphik 2 – SS 10
Ray-Tracing 87

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





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



Dito hier bei der Differenz  $B - A$ !

G. Zachmann    Computer-Graphik 2 – SS 10    Ray-Tracing    88