

Prüfungsunterlagen zum Modul

Interactive Visual Computing (IVC)

Vorlesungsteile

Computer Grafik (CG), 3 Kapitel und

Real Time Interactive Media (RTIM)

Werner Hansmann

Wintersemester 2010/2011

Übersicht

Computergrafik: Folien zu den Teilen

- Geometrisches Modellieren
- Grafische Algorithmen
- Photometrisches Modellieren

Real-Time Interactive Media, alle Folien

Übungsaufgabe

Interactive Visual Computing (IVC)

bzw.

Computergrafik und Bildsynthese (CGB)

(Wintersemester 2010/11)

Werner Hansmann

Geometrisches Modellieren

Geometrisches Modellieren

- Zweck des geometrischen Modellierens
- Repräsentation geometrischer Modelle
- Formen geometrischer Modelle
- Beschreibung der Geometrie
- Beschreibung der Topologie
- zur grafischen Darstellung modellierter Gegenstände

Zweck des geometrischen Modellierens

Das geometrische Modellieren ist ein Grundlagengebiet für diverse Natur- und Ingenieurwissenschaften; es dient

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

Zweck des geometrischen Modellierens (Forts.)

Typische **Analysen** sind über sog. FE-Diskretisierung (Finite Elemente) möglich:

- Massenberechnung,
- Schwerpunktberechnung (Schiffe, etc.),
- Festigkeitsberechnung, Tragfähigkeitsberechnung,
- aerodynamische und hydrodynamische Berechnungen,
- Funktionsanalysen (Kollisionstests, etc.)

Darüber hinaus ermöglichen geometrische Modelle z.B. die

- Konstruktion synthetischer Moleküle,
- Vorbereitung chirurgischer Eingriffe,
- etc.

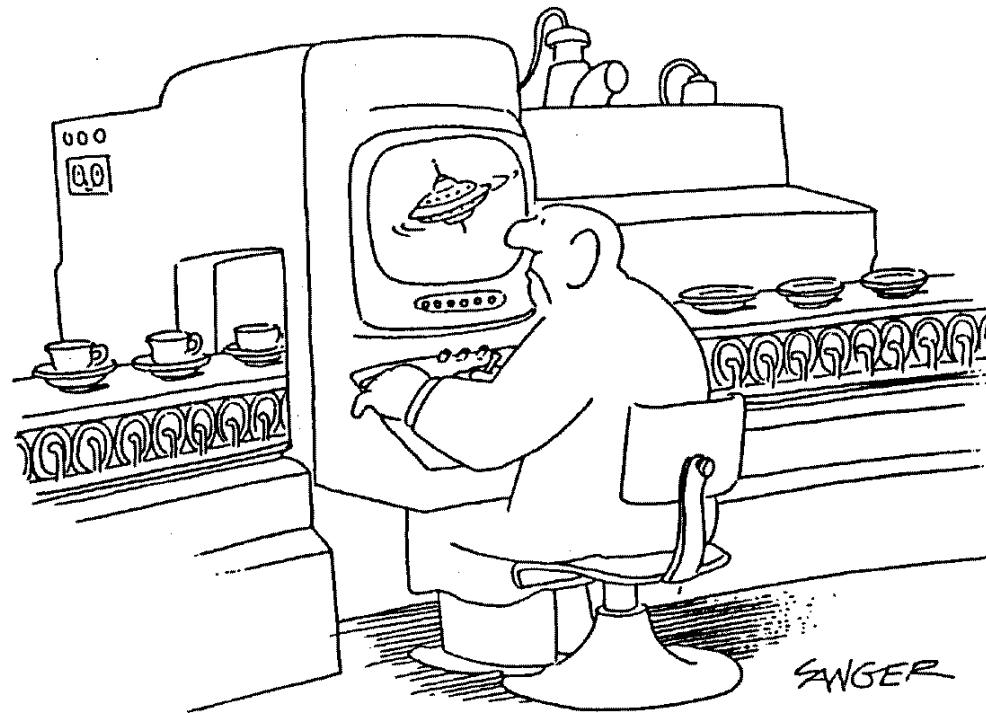
Zweck des geometrischen Modellierens (Forts.)

- Entwurf und
 - (möglichst exakte) Beschreibung der Form von Gegenständen zum Zwecke der **Fertigung**, d.h. der Erstellung von Steuerungsdaten für das
 - Blechschneiden (z.B. Schiffsbleche, Karosseriebleche),
 - Fräsen von
 - * Bauteilen,
 - * Getriebeteilen,
 - * Werkzeugen (z.B. Blechpressen, Gußformen),
 - * etc.
- in der Bildhauerei, der Bauindustrie, der Maschinenbauindustrie, der Schiffbau-, Flugzeugbau-, Automobilbauindustrie, der Chemie- und Pharmaindustrie, etc.

Zweck des geometrischen Modellierens (Forts.)

- Entwurf und
 - (möglichst exakte) Beschreibung der Form von Gegenständen zum Zwecke der
 - **Visualisation** (d.h. grafischen Wiedergabe zu Kontrollzwecken) für
 - * Bildhauer,
 - * Architekten,
 - * Bau- bzw. Maschinenbauingenieure,
 - * Kfz-Designer,
 - * Chemiker und Biologen,
 - * Mediziner,
 - * etc.
- (d.h. typisches **Anwendungsgebiet für die Computergrafik**)

Zweck des geometrischen Modellierens (Forts.)



Geometrisches Modellieren als Grundlage für computerunterstützten Entwurf (CAD: computer aided design) und computerintegrierte Fertigung (CIM: computer integrated manufacturing)

Repräsentation geometrischer Modelle

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

- Volumenbasierte Repräsentation
- Begrenzungsbasierte Repräsentation

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

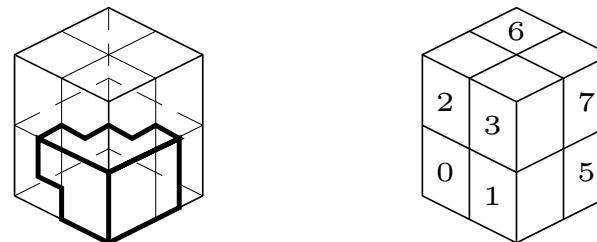
Repräsentation geometrischer Modelle (Forts.)

Volumenbasierte Repräsentationsformen

- Octree

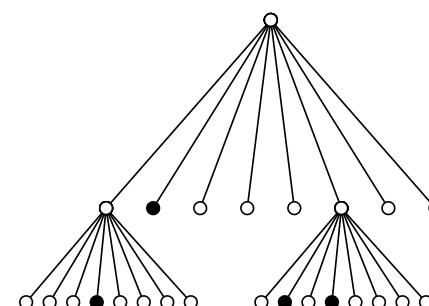
Prinzip:

sukzessive Unterteilung
der Objekthülle (Hüll-
quader) in “Achtelquader”
(bzw. Zusammenfassung
von Voxelclustern)



Anwendungsgebiet:

volumenorientierte
Objektrekonstruktion (z.B. Tomographie)



Repräsentation geometrischer Modelle (Forts.)

Volumenbasierte Repräsentationsformen

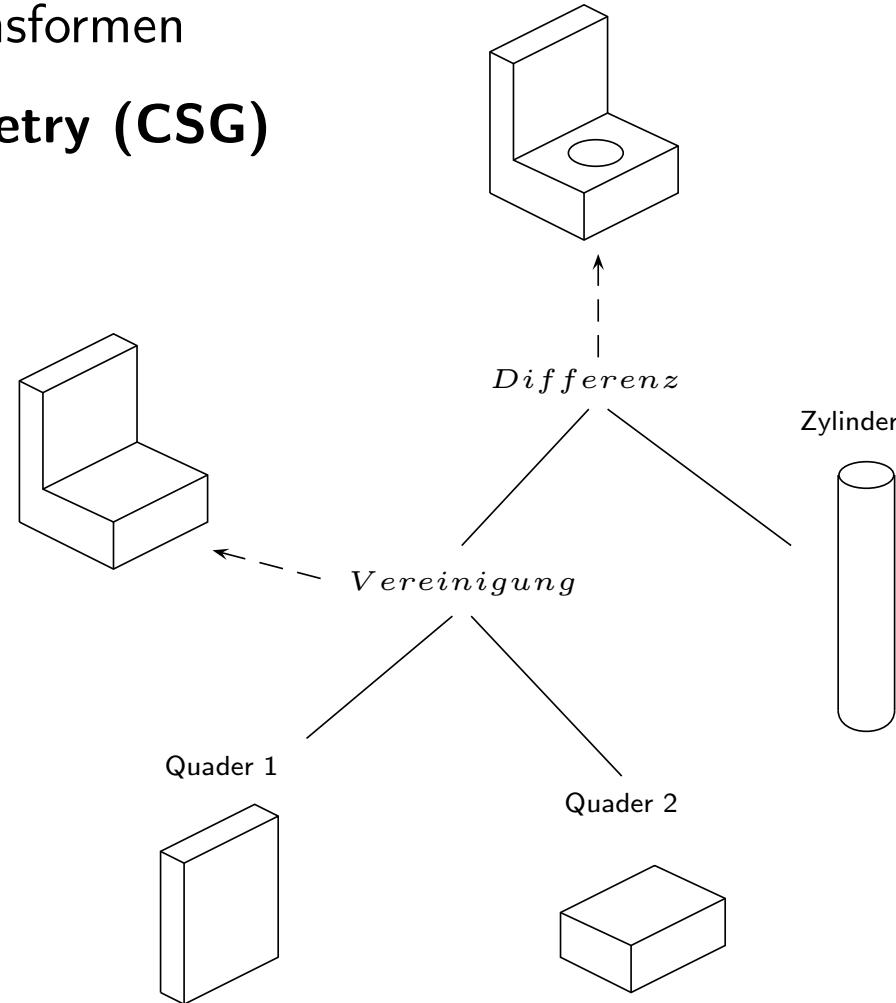
- **Constructive Solid Geometry (CSG)**

Prinzip:

Verknüpfung "primitiver" Objekte (Quader, Zylinder, Kegel, Kugel, etc.) durch Mengenoperationen

Anwendungsgebiet:

Maschinenbau
(orientiert an traditionellen Fertigungsmethoden, wie Bohren, Fräsen, Stanzen)



Beispiele zur Constructive Solid Geometry (CSG)

Nachfolgend einige Beispiele zu den Mengenoperationen

- Vereinigung (union)
- Vereinigung (merge)
- Durchschnitt (intersection)
- Differenz (difference)

sowie zu Kombinationen daraus

dazu exemplarische Realisierungen in POV-Ray

Beispiele zur Constructive Solid Geometry (CSG)

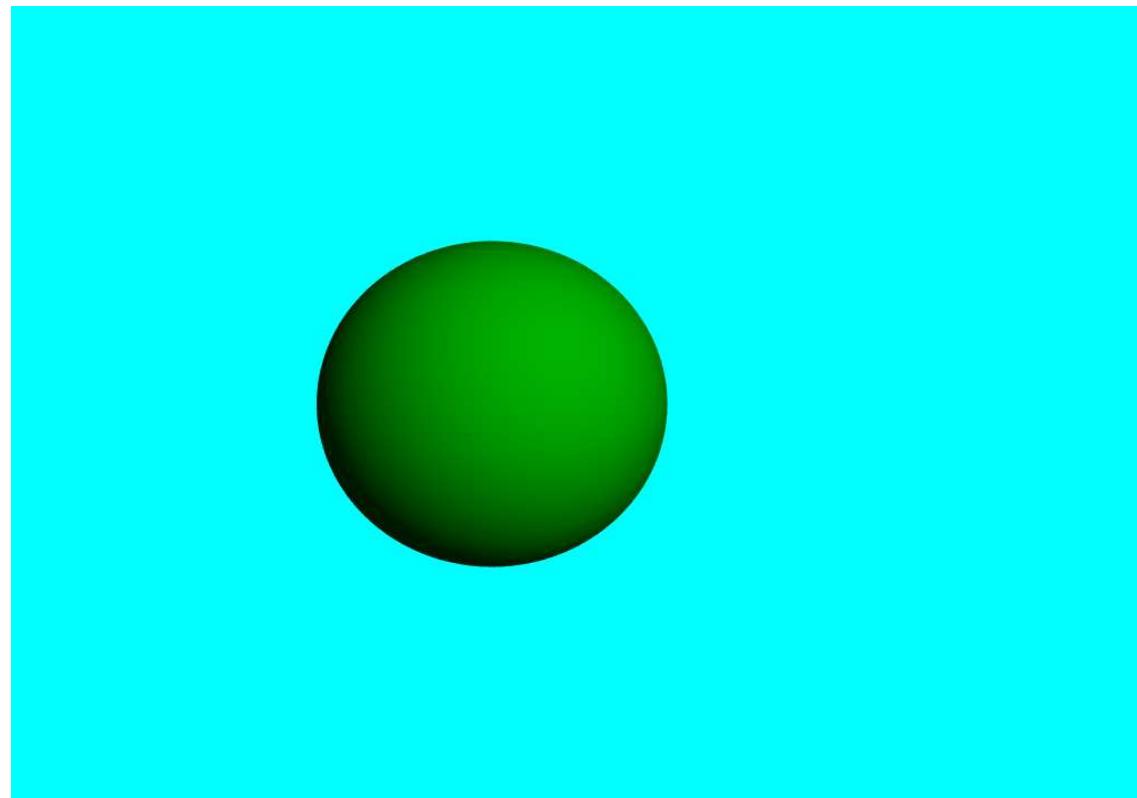
1. eine individuelle Kugel (links angeordnet)
2. eine individuelle Kugel (rechts angeordnet)
3. zwei individuelle Kugeln, ineinander „gesteckt“, individuell animiert
4. zwei individuelle durchsichtige Kugeln, ineinander „gesteckt“, individuell animiert
5. zwei durchsichtige Kugeln, Vereinigung (union), gemeinsam animiert
6. zwei durchsichtige Kugeln, Vereinigung (merge), gemeinsam animiert
7. zwei Kugeln, Durchschnitt (intersection), gemeinsam animiert
8. zwei Kugeln, Differenz (difference), gemeinsam animiert
9. L-Profil mit Bohrung (unterschiedliche Generierungsvarianten)

eine individuelle Kugel (links angeordnet)

```
#include "colors.inc"
background { Cyan }
camera {
    location <0, 1, -10>
    look_at <0, 0, 0>
    angle 36
}
light_source { <500, 500, -1000> White }
```

```
sphere { 0, 1
pigment { Green }
translate -.5*x
}
```

eine individuelle Kugel (links angeordnet)

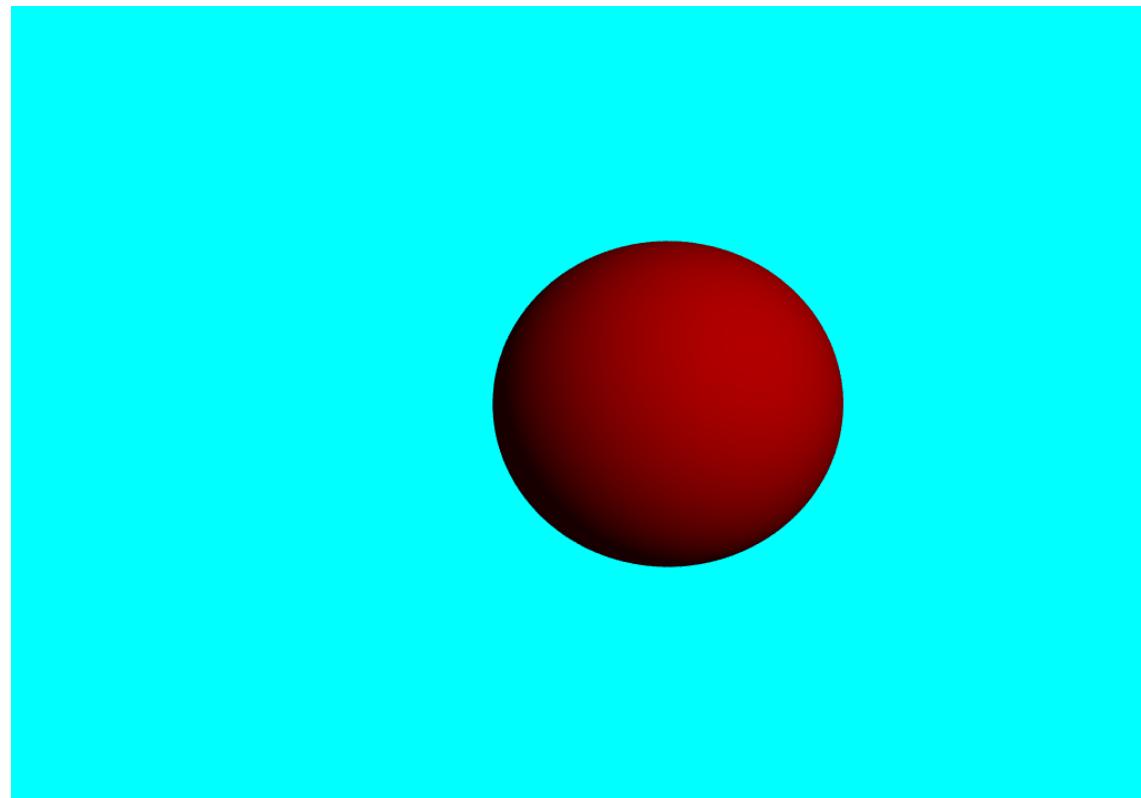


eine individuelle Kugel (rechts angeordnet)

```
#include "colors.inc"
background { Cyan }
camera {
    location <0, 1, -10>
    look_at <0, 0, 0>
    angle 36
}
light_source { <500, 500, -1000> White }

sphere { 0, 1
pigment { Red }
translate +.5*x
}
```

eine individuelle Kugel (rechts angeordnet)



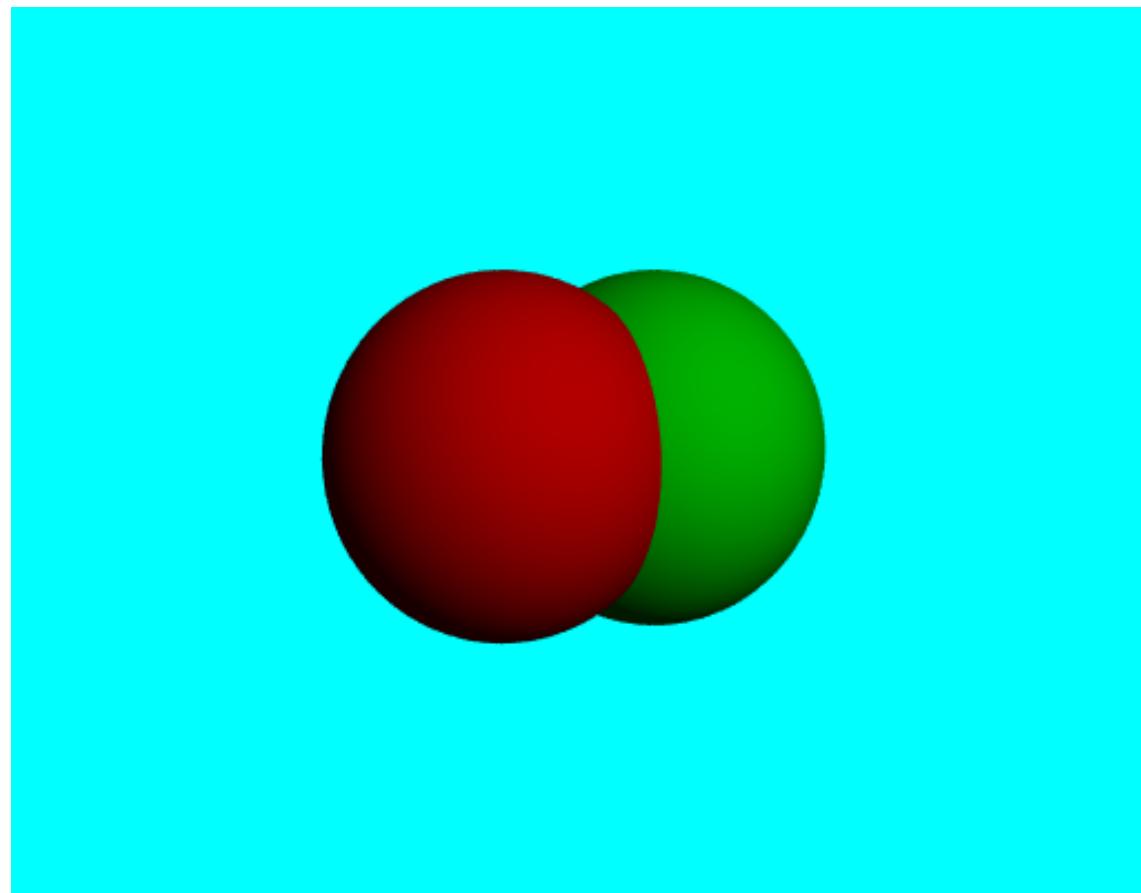
zwei individuelle Kugeln, ineinander „gesteckt“

individuell animiert

...

```
sphere { 0, 1
pigment { Green }
translate -.5*x
rotate y*360*clock
}
sphere { 0, 1
pigment { Red }
translate +.5*x
rotate y*360*clock
}
```

zwei individuelle Kugeln, ineinander „gesteckt“



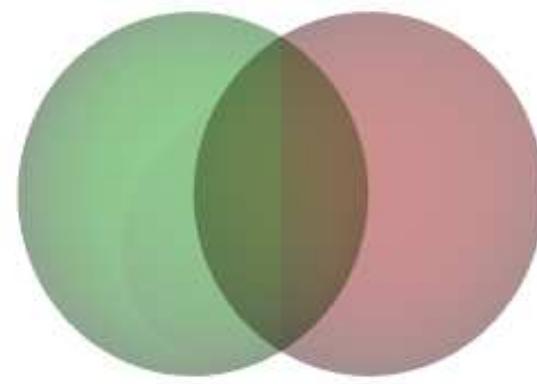
zwei individuelle durchsichtige Kugeln, ineinander „gesteckt“

individuell animiert

...

```
sphere { 0, 1
pigment { Green transmit 0.75 }
translate -.5*x
rotate y*360*clock
}
sphere { 0, 1
pigment { Red transmit 0.75 }
translate +.5*x
rotate y*360*clock
}
```

**zwei individuelle durchsichtige Kugeln, ineinander
„gesteckt“**



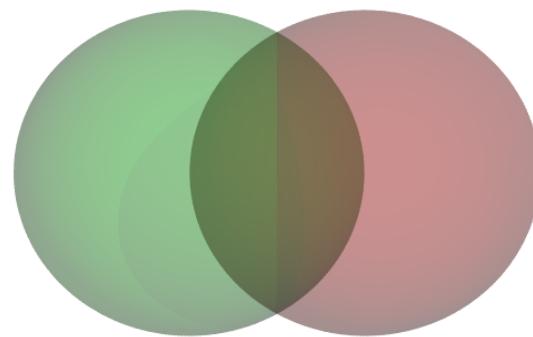
zwei durchsichtige Kugeln, Vereinigung (union)

gemeinsam animiert

...

```
union {  
    sphere { 0, 1  
        pigment { Green transmit 0.75 }  
        translate -.5*x  
    }  
    sphere { 0, 1  
        pigment { Red transmit 0.75 }  
        translate +.5*x  
    }  
    rotate y*360*clock  
}
```

zwei durchsichtige Kugeln, Vereinigung (union)



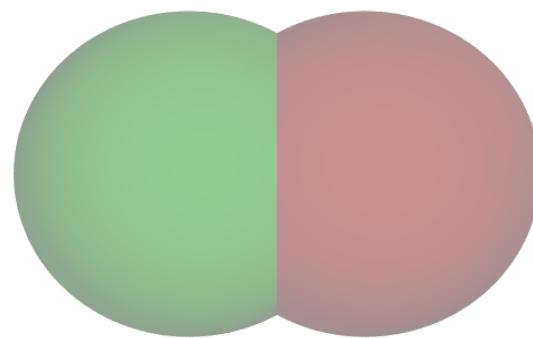
zwei durchsichtige Kugeln, Vereinigung (merge)

gemeinsam animiert

...

```
merge {  
    sphere { 0, 1  
        pigment { Green transmit 0.75 }  
        translate -.5*x  
    }  
    sphere { 0, 1  
        pigment { Red transmit 0.75 }  
        translate +.5*x  
    }  
    rotate y*360*clock  
}
```

zwei durchsichtige Kugeln, Vereinigung (merge)



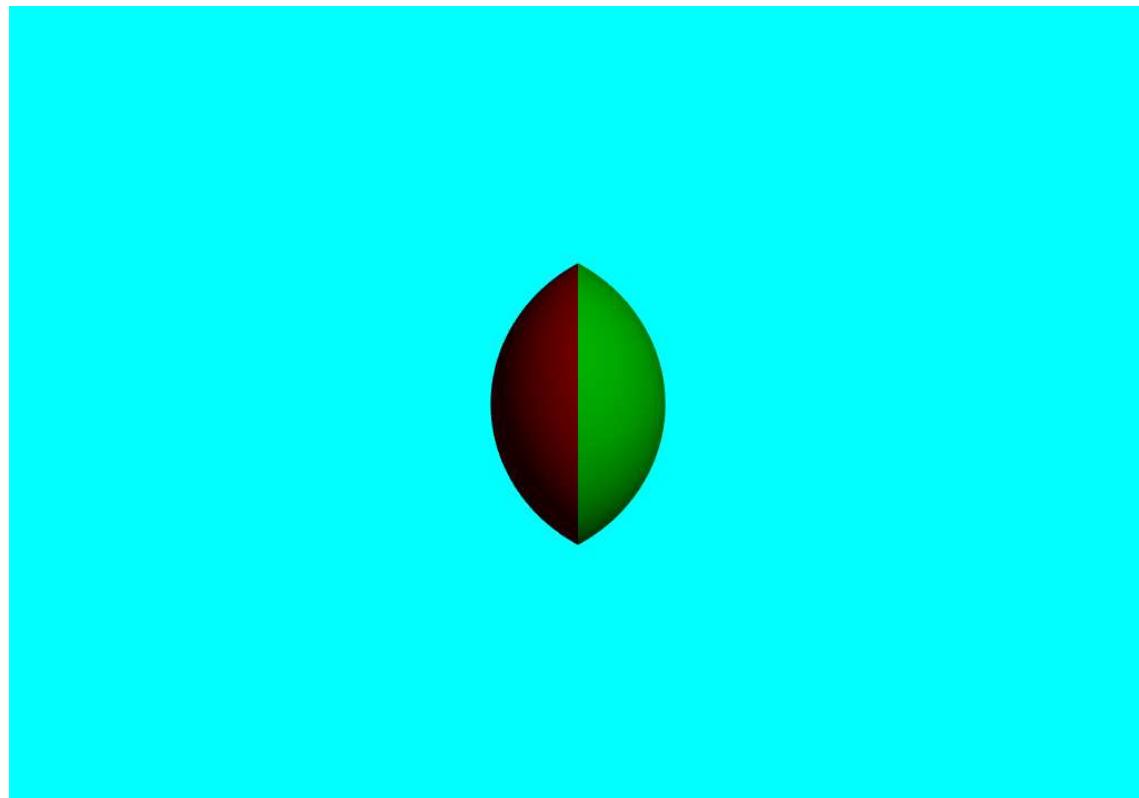
zwei Kugeln, Durchschnitt (intersection)

gemeinsam animiert — — — beachten Sie die Oberflächenfarben

...

```
intersection {  
    sphere { 0, 1  
        pigment { Green }  
        translate -.5*x  
    }  
    sphere { 0, 1  
        pigment { Red }  
        translate +.5*x  
    }  
    rotate y*360*clock  
}
```

zwei Kugeln, Durchschnitt (intersection)



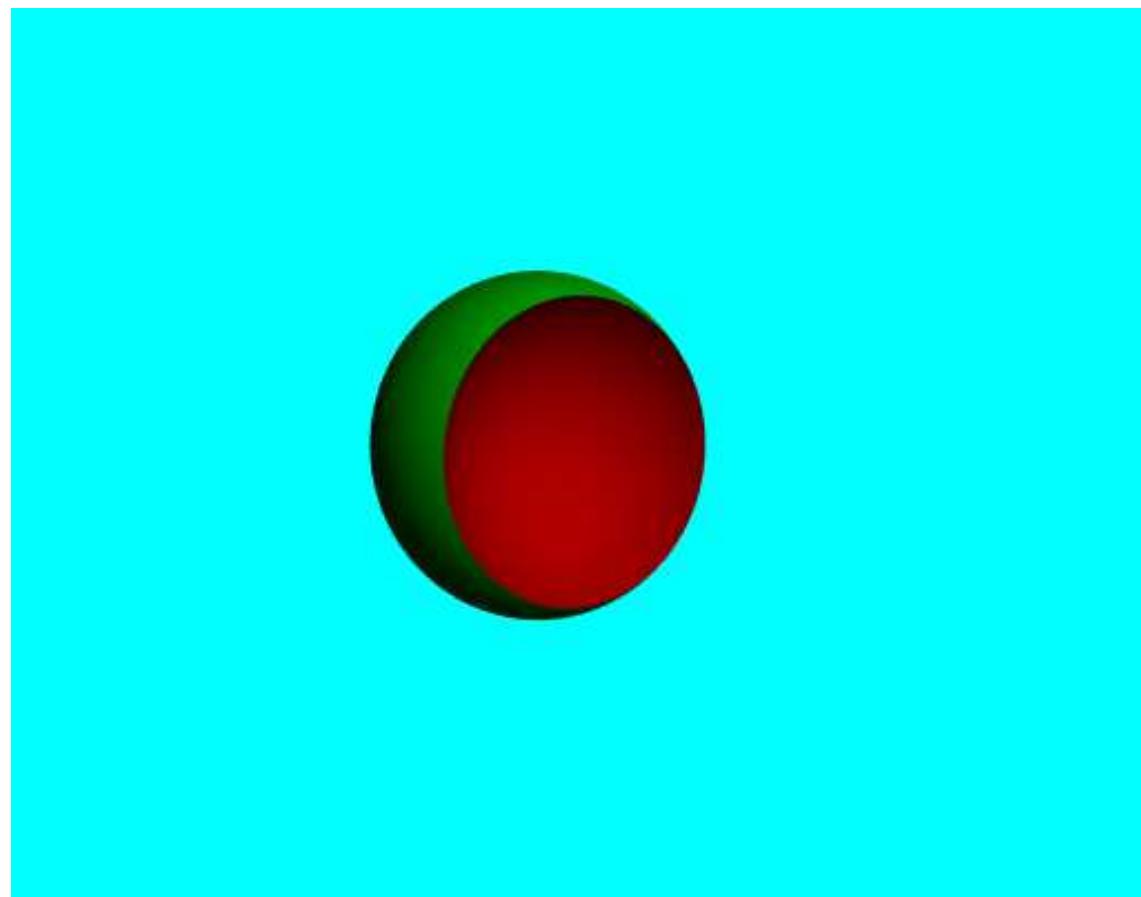
zwei Kugeln, Differenz (difference)

gemeinsam animiert — — — beachten Sie die Oberflächenfarben

...

```
difference {  
    sphere { 0, 1  
        pigment { Green }  
        translate -.5*x  
    }  
    sphere { 0, 1  
        pigment { Red }  
        translate +.5*x  
    }  
    rotate y*360*clock  
}
```

zwei Kugeln, Differenz (difference)



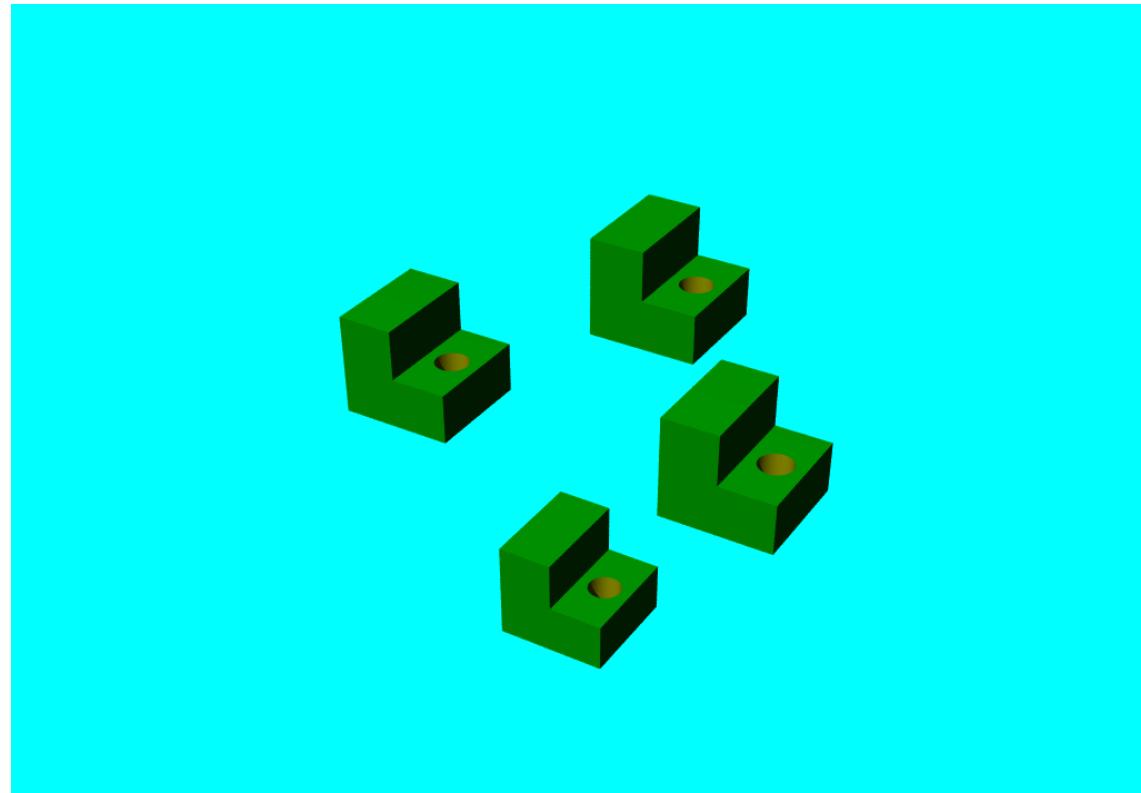
L-Profil mit Bohrung

unterschiedliche Generierungsvarianten, wie sehen die jeweiligen CSG-Bäume aus? Formulieren Sie weitere Varianten.

...

```
// Generierungsvariante 1
difference {
    union {
        box { -.5, <.5 0 .5>
              pigment { Green } }
        box { -.5*z, <-.5, .5, .5>
              pigment { Green } }
    }
    cylinder { <.25, .5, 0> <.25, -.5, 0> .15 open
               pigment { Yellow } }
    translate -1.5*x }
```

L-Profil mit Bohrung



L-Profil mit Bohrung

```
// Generierungsvariante 2  
...  
difference {  
difference {  
box { -.5, .5  
pigment { Green } }  
cylinder { <.25, .5, 0> <.25, -.5, 0> .15 open  
pigment { Yellow } }  
}  
box { -.6, .6  
translate <.6, .6, 0>  
pigment { Green } }  
translate 1.5*x }
```

L-Profil mit Bohrung

```
// Generierungsvariante 3  
...  
union {  
difference {  
box { -.5, <.5 0 .5>  
pigment { Green } }  
cylinder { <.25, .5, 0> <.25, -.5, 0> .15 open  
pigment { Yellow } }  
}  
box { -.5*z, <-.5, .5, .5>  
pigment { Green } }  
translate <0, -.5 -1.5> }
```

L-Profil mit Bohrung

```
// Generierungsvariante 4  
...  
difference {  
    box { -.5, .5  
        pigment { Green } }  
    union {  
        cylinder { <.25, .5, 0> <.25, -.5, 0> .15 open  
            pigment { Yellow } }  
        box { -.6, .6  
            translate <.6, .6, 0>  
            pigment { Green } }  
    }  
    translate <0, .5 1.5> }
```

Repräsentation geometrischer Modelle (Forts.)

Begrenzungsbasierte Repräsentationsformen (B-Rep.)

Komplexeste und flexibelste Methode der Repräsentation

Objekte werden durch die sie begrenzenden Elemente repräsentiert,

z.B. Körper durch ihre Oberfläche

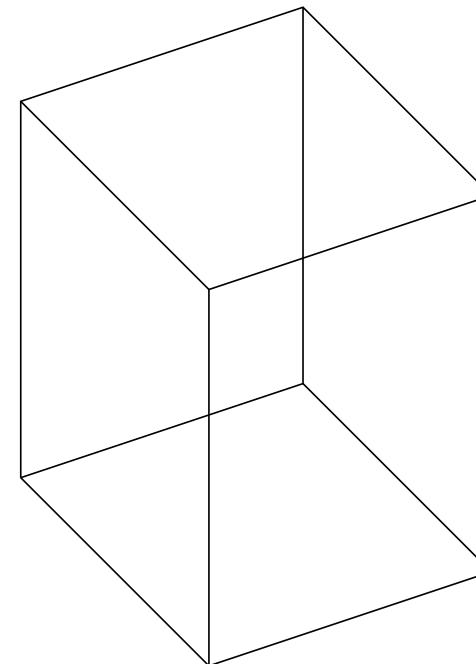
besonders geeignet für “Freiformgestaltung”, z.B.:

- Schmiedeformen,
- Gußformen,
- Skulpturen,
- Karosserien

Formen begrenzungsbasierter geometrischer Modelle

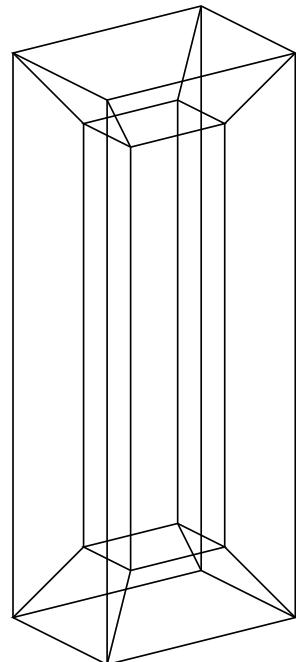
Kantenmodelle (vor 1960)

Objektrepräsentation
über Kanten (Raumkurven)
und Eckpunkte
(= Schnittpunkte der Kanten)



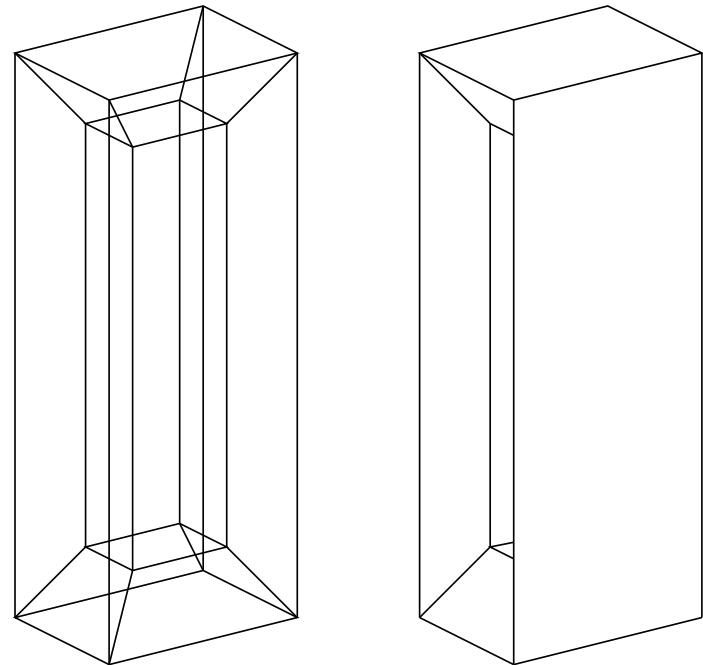
begrenzungsbasierte geometrische Modelle (Forts.)

Problem: Mehrdeutigkeiten beim Kantenmodell (Beispiel 1)



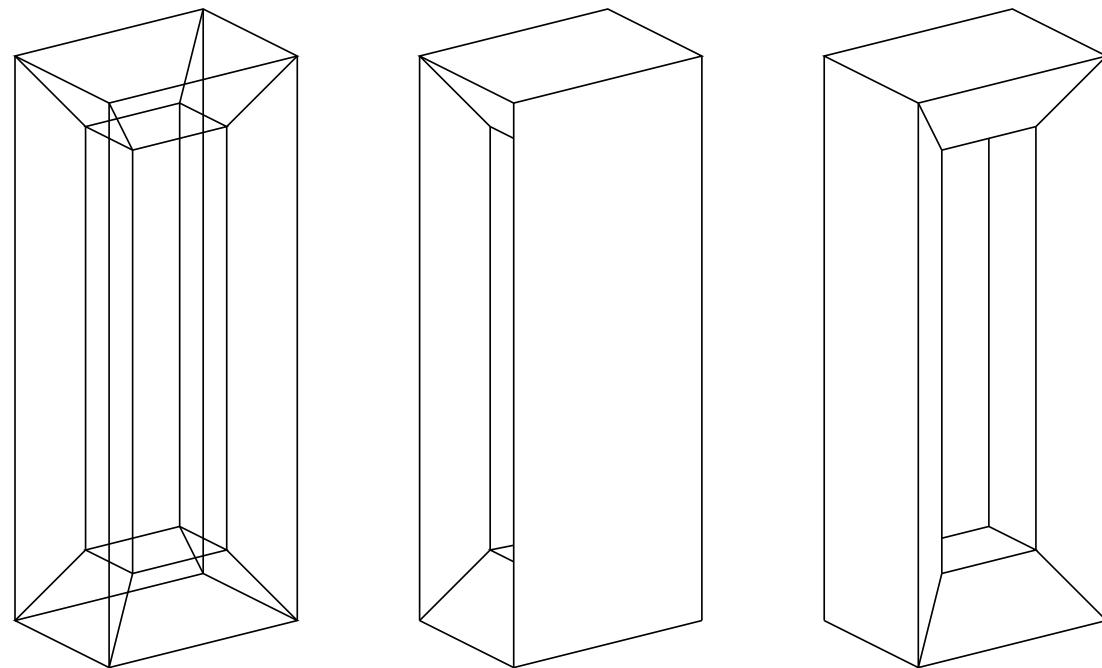
begrenzungsbasierte geometrische Modelle (Forts.)

Problem: Mehrdeutigkeiten beim Kantenmodell (Beispiel 1)



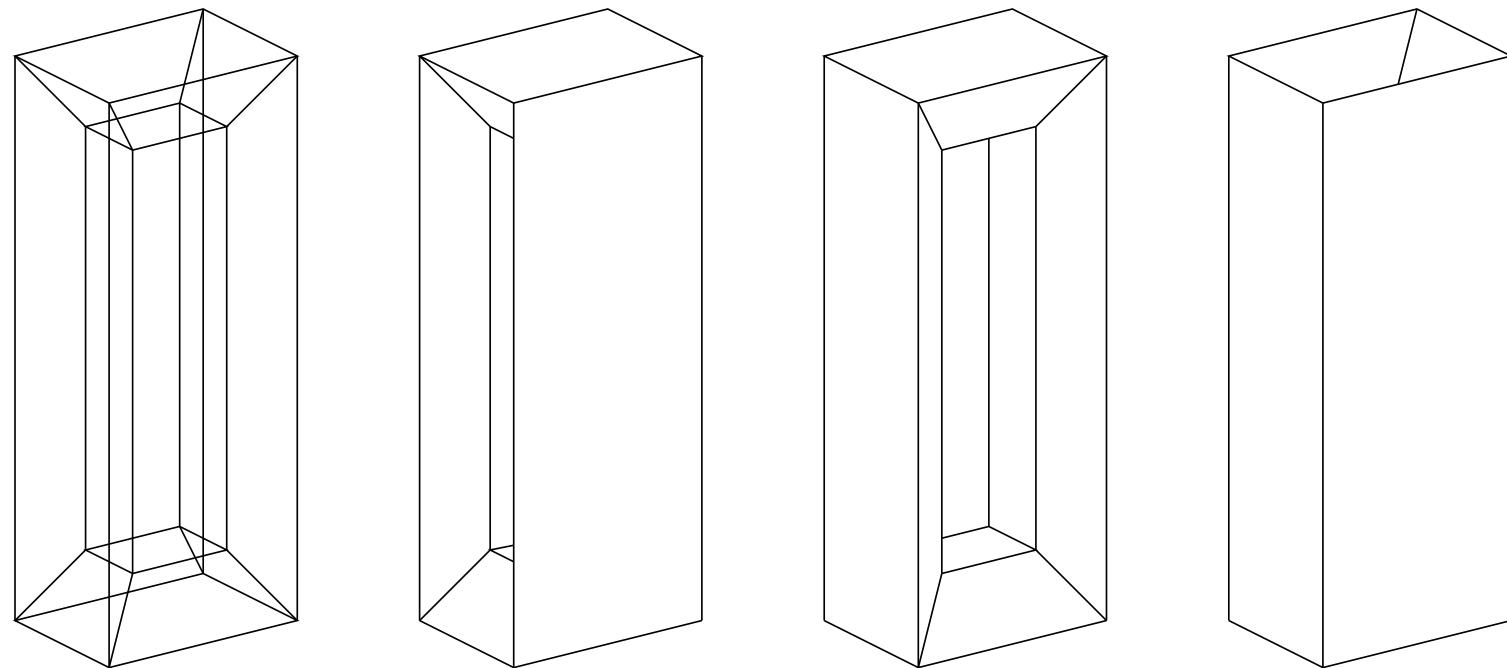
begrenzungsbasierte geometrische Modelle (Forts.)

Problem: Mehrdeutigkeiten beim Kantenmodell (Beispiel 1)



begrenzungsbasierte geometrische Modelle (Forts.)

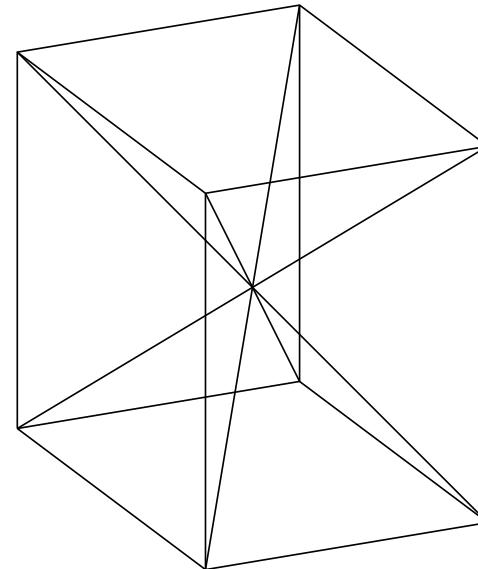
Problem: Mehrdeutigkeiten beim Kantenmodell (Beispiel 1)



begrenzungsbasierte geometrische Modelle (Forts.)

Problem: Mehrdeutigkeiten beim Kantenmodell (Beispiel 2)

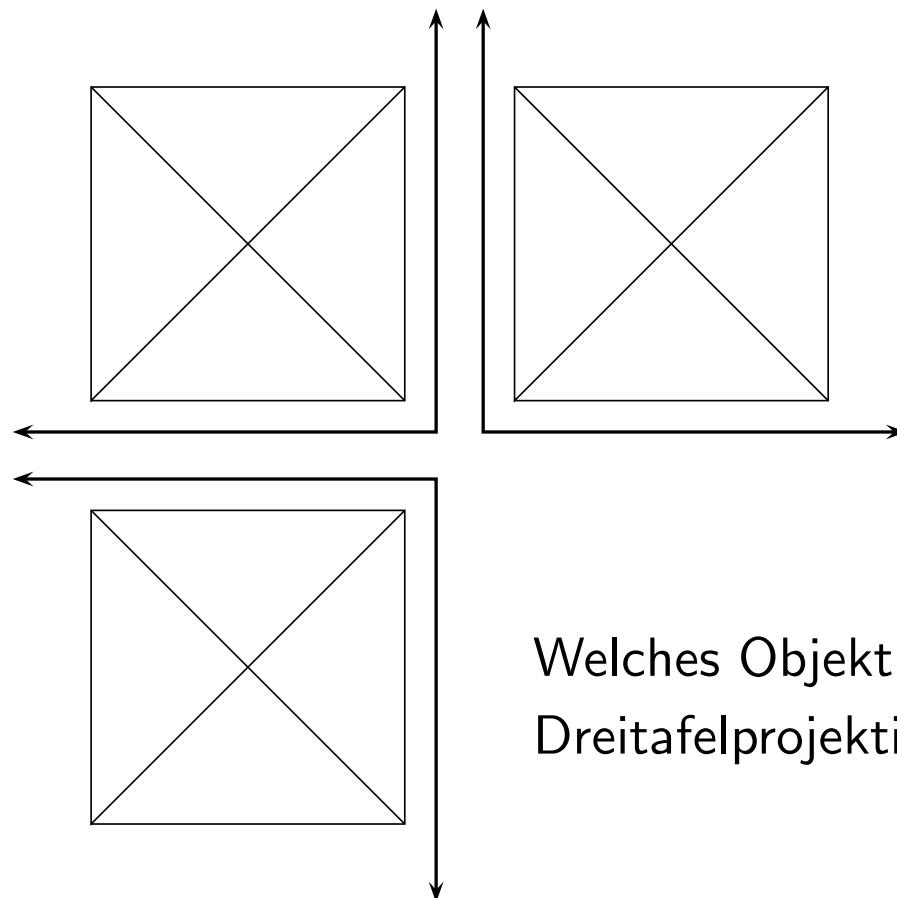
Was stellt dieses Kantenmodell dar?



Gibt die nachfolgende Dreitafelprojektion diesen Gegenstand wieder?

begrenzungsbasierte geometrische Modelle (Forts.)

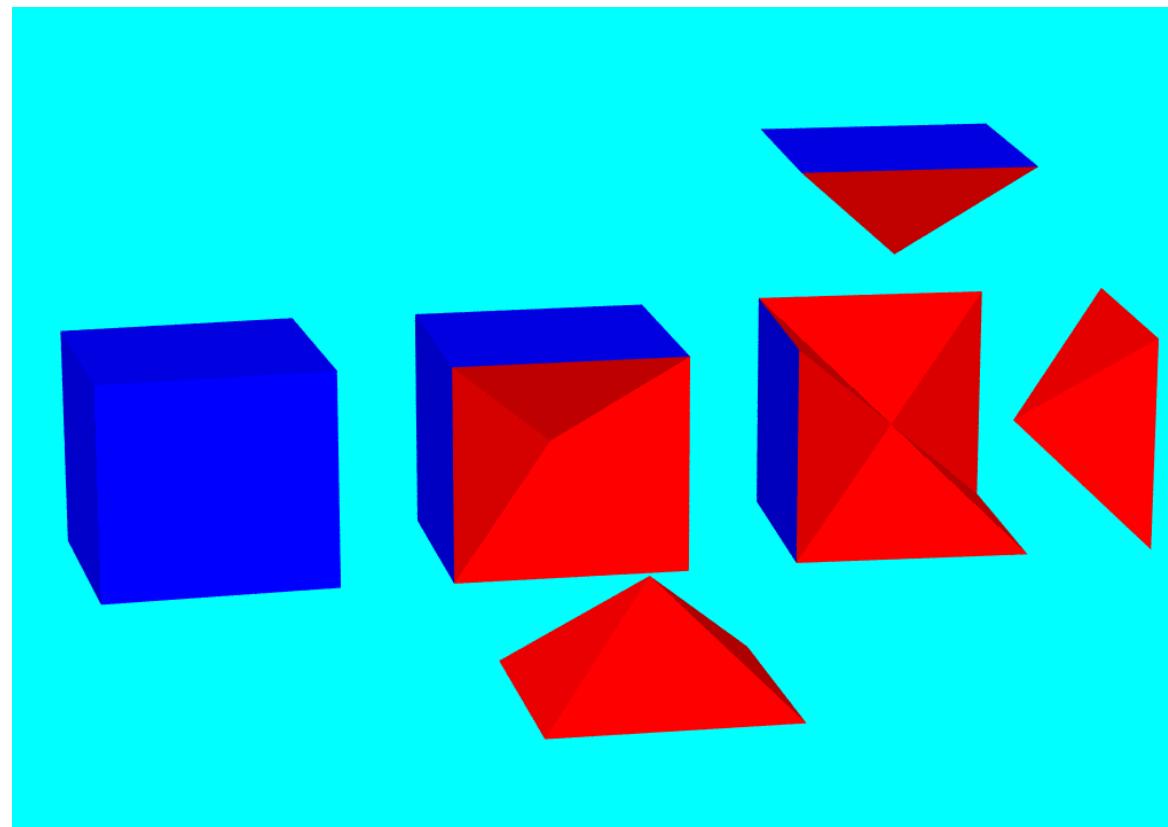
Problem: Unentscheidbarkeit bei der Dreitafelprojektion



Welches Objekt wird durch diese
Dreitafelprojektion abgebildet?

begrenzungsbasierte geometrische Modelle (Forts.)

Problem: Unentscheidbarkeit bei der Dreitafelprojektion



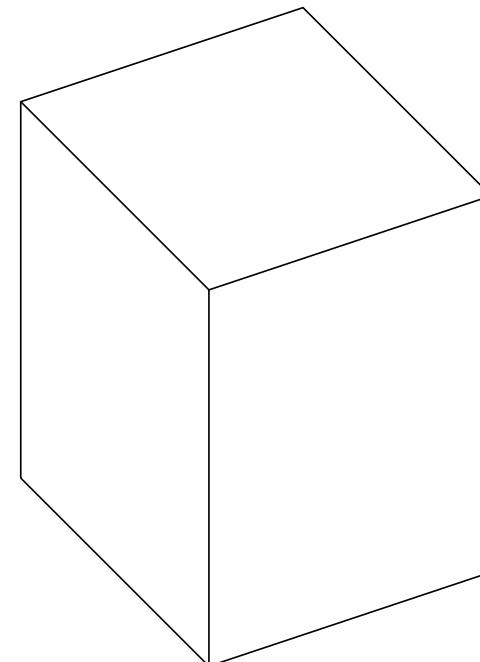
begrenzungsbasierte geometrische Modelle (Forts.)

Oberflächenmodelle (frühe 1960er)

Objektrepräsentation
zusätzlich über
geometrische Beschreibung
der Oberflächenform
zwischen Kantenpolygonen

ermöglicht:

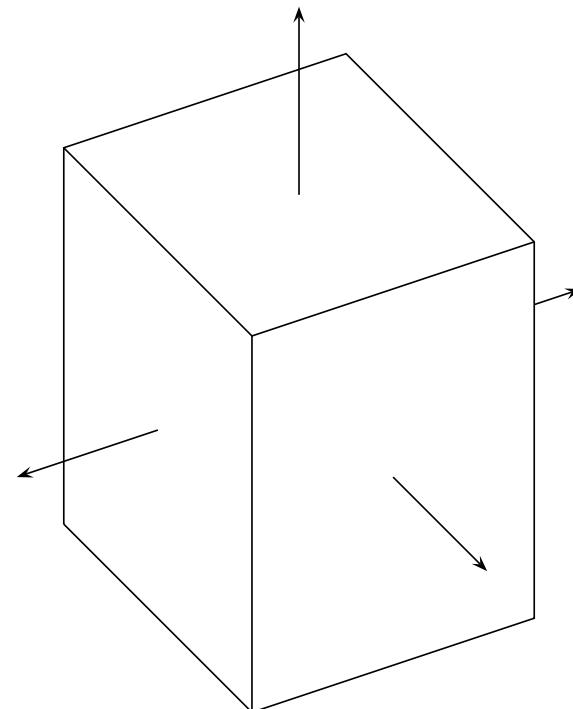
- grafische (flächige) Darstellung
- NC-Fertigung (Numerical Control)
des Gegenstandes



begrenzungsbasierte geometrische Modelle (Forts.)

(zweidimensional mannigfaltige) **Körpermodelle** (frühe 1970er)

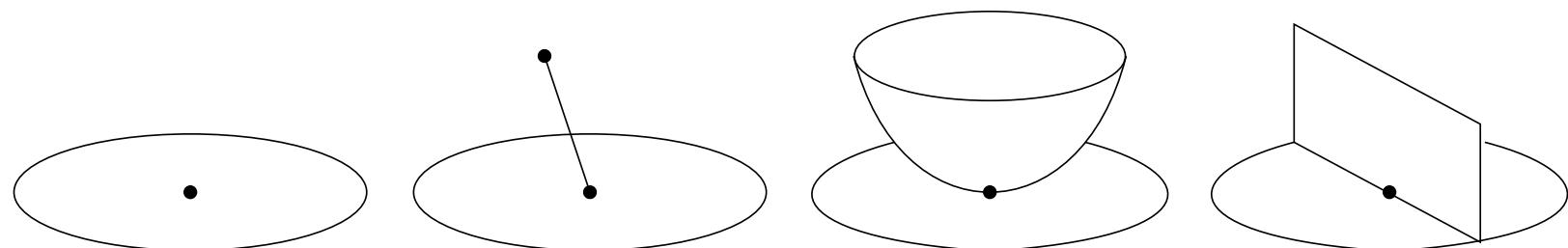
Objektrepräsentation enthält zusätzlich Information über Geschlossenheit und Zusammenhangigkeit der Volumina von Körpern d.h. "innen" und "außen" können unterschieden werden. Dadurch können z.B. Massen- und Schwerpunktberechnungen unmittelbar durchgeführt werden.



Letzte Einschränkung:
die Oberflächen müssen überall topologisch "eben" sein.

begrenzungsbasierte geometrische Modelle (Forts.)

zweidimensional mannigfaltige **Körpermodelle** (Forts.)



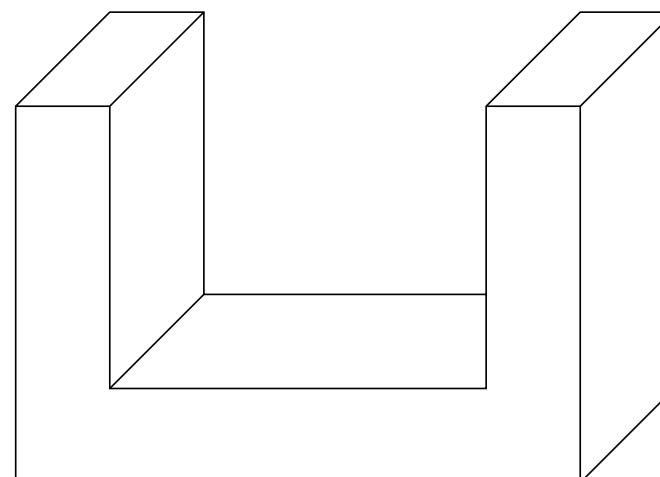
topologisch zweidimensional bedeutet, daß die unmittelbare Umgebung jedes Punktes der Oberfläche sich eindeutig in eine Ebene abbilden läßt

begrenzungsbasierte geometrische Modelle (Forts.)

nicht-zweidim. mannigfaltige **Körpermodelle** (seit Ende der 1980er)

vereinigen in geschlossener
Repräsentation

- Kantenmodelle,
- Flächenmodelle,
- Körpermodelle



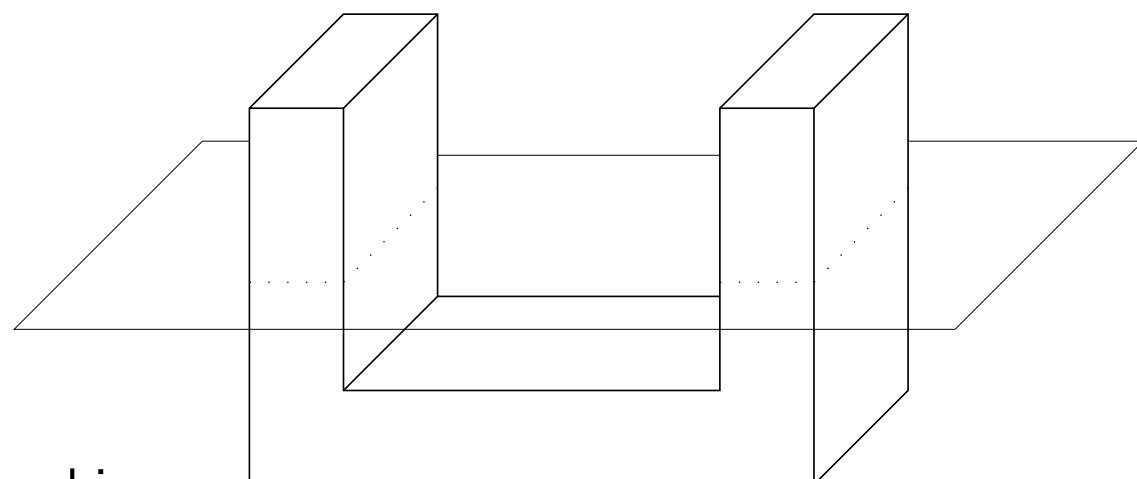
und gestatten darüber hinaus
die Behandlung topologisch nicht-zweidimensionaler Geometrien

begrenzungsbasierte geometrische Modelle (Forts.)

nicht-zweidim. mannigfaltige **Körpermodelle** (seit Ende der 1980er)

vereinigen in geschlossener
Repräsentation

- Kantenmodelle,
- Flächenmodelle,
- Körpermodelle



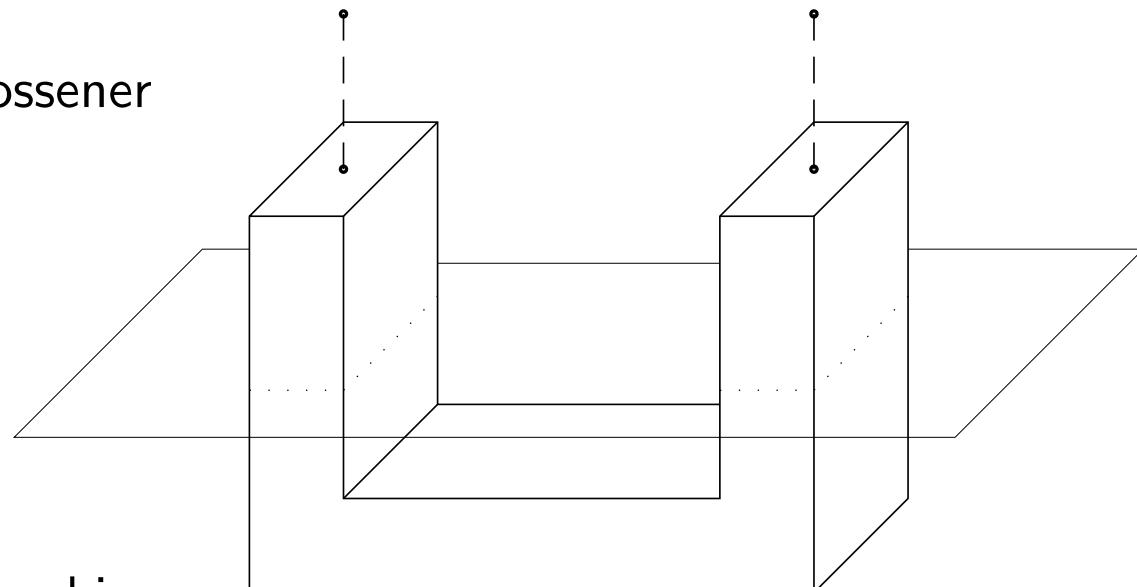
und gestatten darüber hinaus
die Behandlung topologisch nicht-zweidimensionaler Geometrien

begrenzungsbasierte geometrische Modelle (Forts.)

nicht-zweidim. mannigfaltige **Körpermodelle** (seit Ende der 1980er)

vereinigen in geschlossener
Repräsentation

- Kantenmodelle,
- Flächenmodelle,
- Körpermodelle



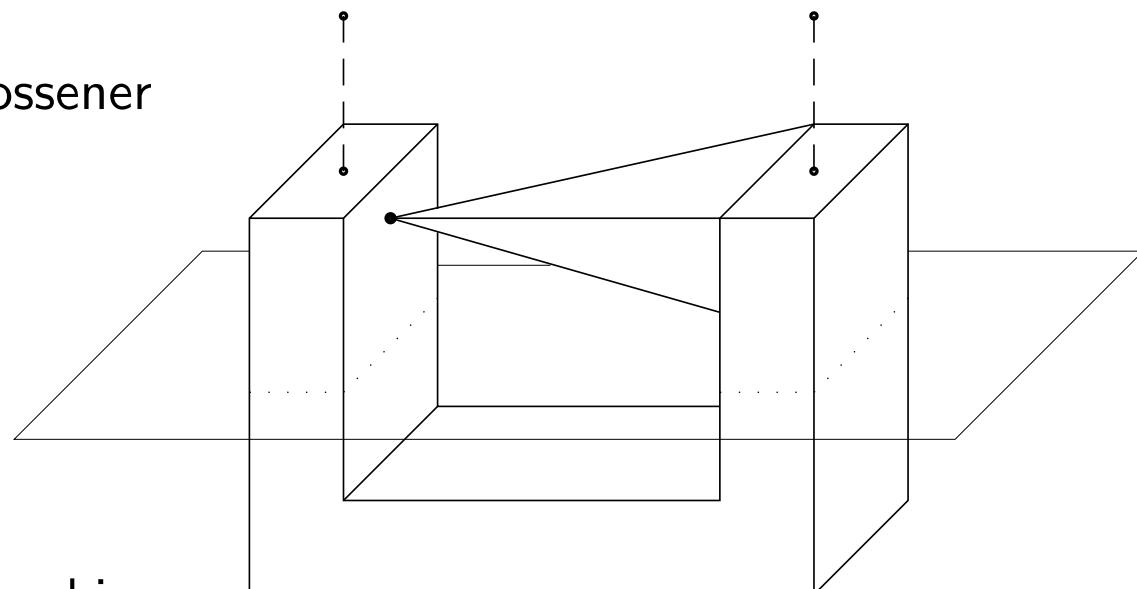
und gestatten darüber hinaus
die Behandlung topologisch nicht-zweidimensionaler Geometrien

begrenzungsbasierte geometrische Modelle (Forts.)

nicht-zweidim. mannigfaltige **Körpermodelle** (seit Ende der 1980er)

vereinigen in geschlossener
Repräsentation

- Kantenmodelle,
- Flächenmodelle,
- Körpermodelle



und gestatten darüber hinaus
die Behandlung topologisch nicht-zweidimensionaler Geometrien

Beschreibung der Geometrie

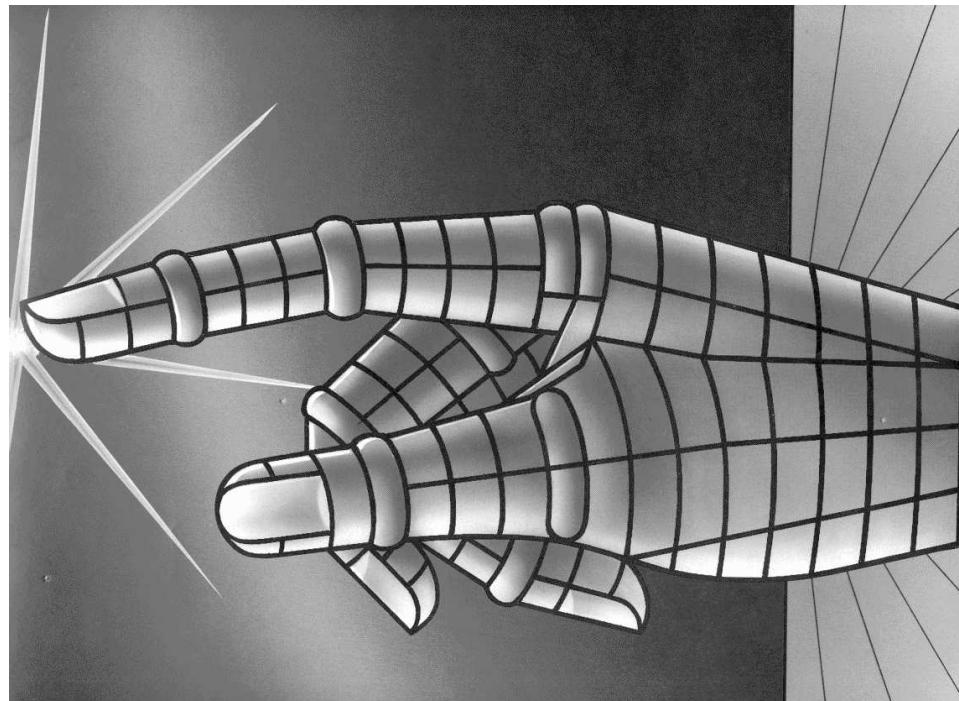
Prinzip: Oberflächen komplex gestalteter Objekte sind zusammengesetzt aus Teilflächen



geschlossene Beschreibung „am Stück“ im allgemeinen nicht möglich bzw. sinnvoll (z.B. **Karosserie**, Körperoberfläche eines Menschen)

Beschreibung der Geometrie

Prinzip: Oberflächen komplex gestalteter Objekte sind zusammengesetzt aus Teilflächen



geschlossene Beschreibung „am Stück“ im allgemeinen nicht möglich bzw. sinnvoll (z.B. Karosserie, **Körperoberfläche eines Menschen**)

Beschreibung der Geometrie (Forts.)

Die Teilflächen sind begrenzt durch ihre Ränder (sog. *Kanten*: i.d.R Raumkurven), die

- sich durch Schnitt zwischen den jeweils benachbarten Teilflächen ergeben, oder
- die Grenzen des Definitionsbereiches für eine Teilfläche repräsentieren (Freiformflächen), oder
- als Liniengitter vorgegeben sind, das durch die Oberfläche interpoliert werden soll (Coons'sche Flächen)

Beschreibung der Geometrie (Forts.)

Kurven sind repräsentierbar durch

- Folgen von Punkten
- mathematische Formeln

Im interaktiven Entwurfsprozeß können Punktfolgen intuitiv vorgegeben werden.

Aber: Kurven, die durch Punktfolgen festgelegt sind, können nur mit großem Aufwand manipuliert werden.

Beschreibung der Geometrie (Forts.)

Vorteile der Repräsentation durch Formeln:

- im allgemeinen weniger Speicherbedarf
- exakte Repräsentationen beliebiger
 - Kurvenpunkte
 - Tangenten
 - Krümmungen
 - etc.
- können gegeben werden
- ein kontinuierlicher Graph der Kurven kann im allgemeinen einfach erzeugt und dargestellt werden
- Kurven können durch Koeffizientenmanipulation verändert werden

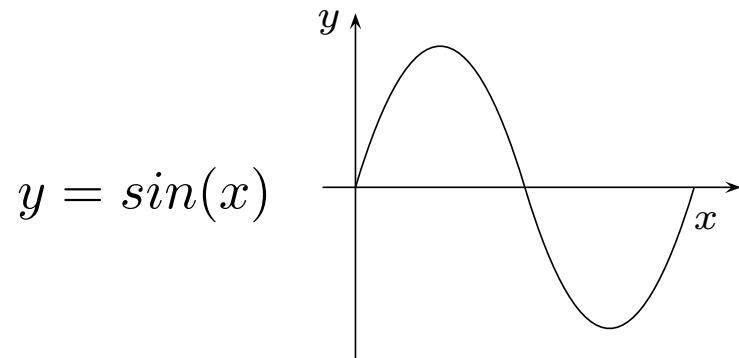
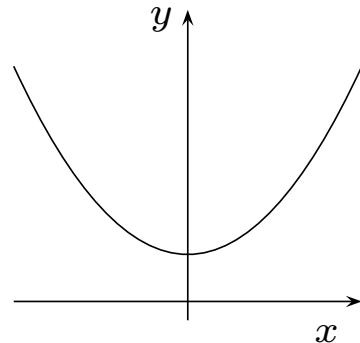
Beschreibung der Geometrie (Forts.)

Mathematische Notationsformen:

explizite Form: $y = f(x)$

Beispiele:

$$y = ax^2 + b$$



Beurteilung:

Vorteil: es existiert eine eindeutige Zuordnung zwischen abhängiger und unabhängiger Variablen

⇒ unmittelbar in grafische Darstellung umsetzbar

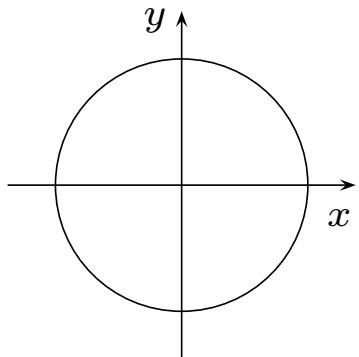
Nachteil: „rückläufige“ Kurven, Schleifen nicht direkt repräsentierbar

Beschreibung der Geometrie (Forts.)

implizite Form: $f(x, y) = \text{konst.}$

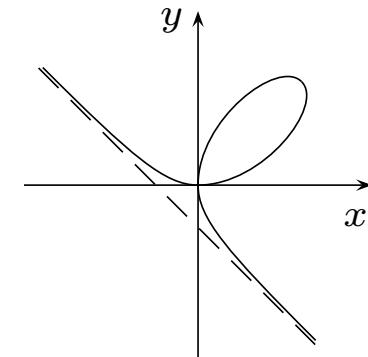
Beispiele:

$$x^2 + y^2 = r^2$$



$$x^3 + y^3 - 3xy = 0$$

Blatt des Descartes



Beurteilung:

Vorteil: vielseitige Kurvenformen mathematisch beschreibbar

Nachteil: kaum für grafische Darstellung algorithmisierbar

Beschreibung der Geometrie (Forts.)

parametrische Form: $x = f(t), y = g(t)$ (t: gemeins. Parameter)

Beispiele:

Parabel:

$$x = t$$

$$y = t^2$$

Kreis:

$$x = rt$$

$$y = r\sqrt{1 - t^2}$$

$$(-1 \leq t \leq +1)$$

Blatt des Descartes:

$$x = 3t/(1 + t^3)$$

$$y = 3t^2/(1 + t^3)$$

mögliche

Parametri-
sierungen

Beurteilung:

Vorteil: Kurven beliebiger Gestalt mathematisch beschreibbar
und leicht grafisch darstellbar

Problem: nicht immer einfach, eine parametrische Notation
für eine Kurve zu finden

Beschreibung der Geometrie (Forts.)

**allgemeine vektorwertige parametrische Notation für Kurven
(im dreidimensionalen euklidischen Raum):**

$$K(t) := K(x(t), y(t), z(t))$$

darin sind:

x, y, z : voneinander unabhängige Funktionen von

t : gemeinsamer Parameter (unabhängige Variable)

Vorstellung: eine Kurve in vektorwertiger parametrischer Notation beschreibt die Bewegung eines Punktes durch den Raum in Abhängigkeit von der Zeit

Beschreibung der Geometrie (Forts.)

- Liniengitter, die durch Oberflächen interpoliert werden sollen, können als Raumkurven in vektorwertiger parametrischer Notation beschrieben werden; z.B. als

- Bézierkurven

$$R(t) = \sum_{i=0}^n P_i \cdot B_{n,i}(t)$$

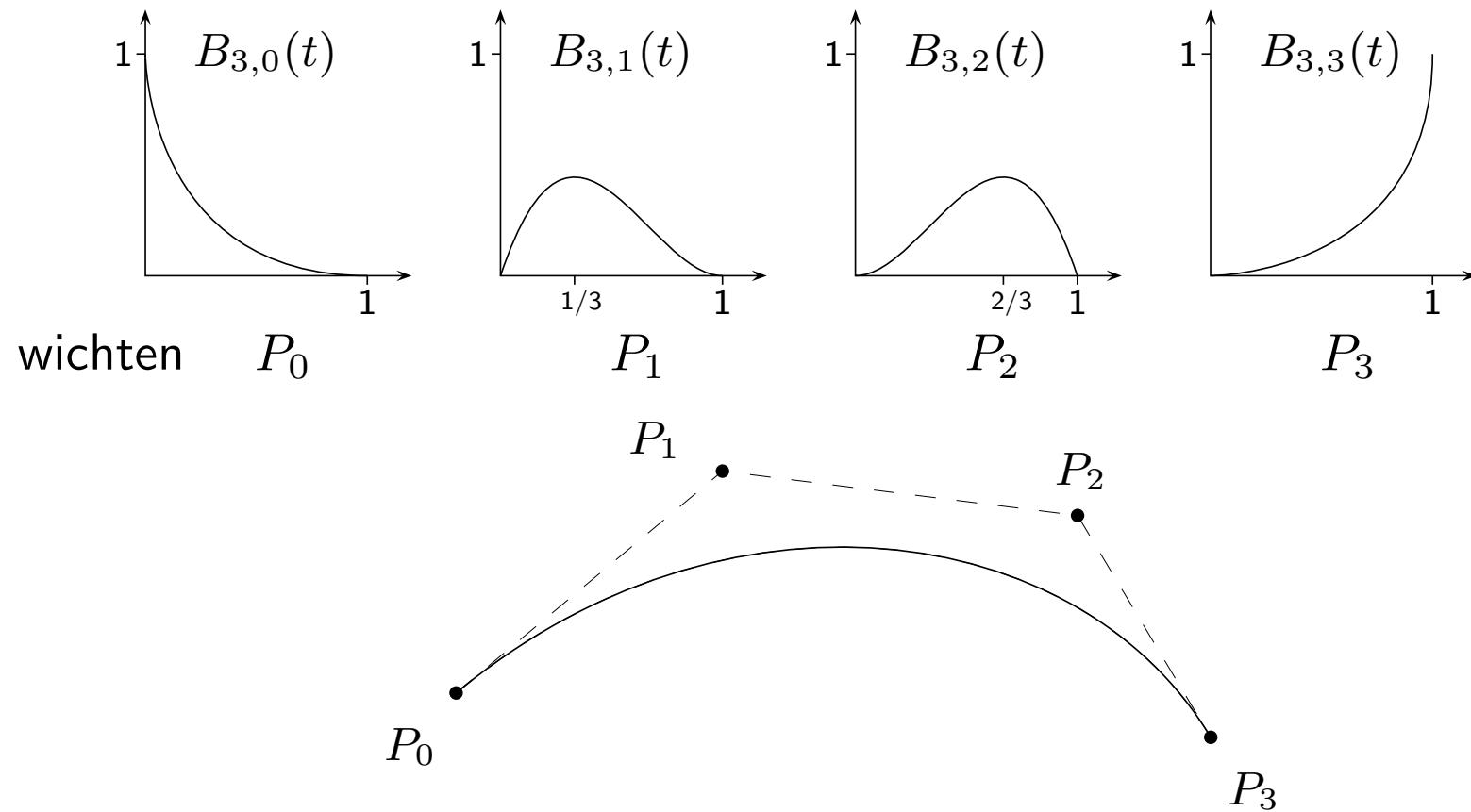
- B-Splinekurven

$$R(t) = \sum_{i=0}^n P_i \cdot N_{i,k}(t)$$

darin ist t der Kurvenparameter; die P_i sind Folgen von Punkten, die durch die jeweilige Kurve approximiert werden, und $B_{n,i}(t)$ bzw. $N_{i,k}(t)$ geeignete Wichtungsfunktionen (Bernsteinpolynome bzw. B-Splinebasisfunktionen, vgl. Freiformflächen, s.u.)

Beschreibung der Geometrie (Forts.)

Beispiel: die Bernsteinpolynome dritten Grades



zur Generierung einer vier Stützpunkte approximierenden Bézierkurve

Beschreibung der Geometrie (Forts.)

NURBS-Kurven sind neben der Freiformmodellierung geeignet auch z.B. Kegelschnitte (Kreise, Ellipsen, Hyperbeln, Parabeln) exakt zu repräsentieren (besondere Eigenschaft rationaler Polynome).

Darstellungsform:

$$R(t) = \frac{\sum_{i=0}^n w_i P_i N_{i,k}(t)}{\sum_{i=0}^n w_i N_{i,k}(t)}$$

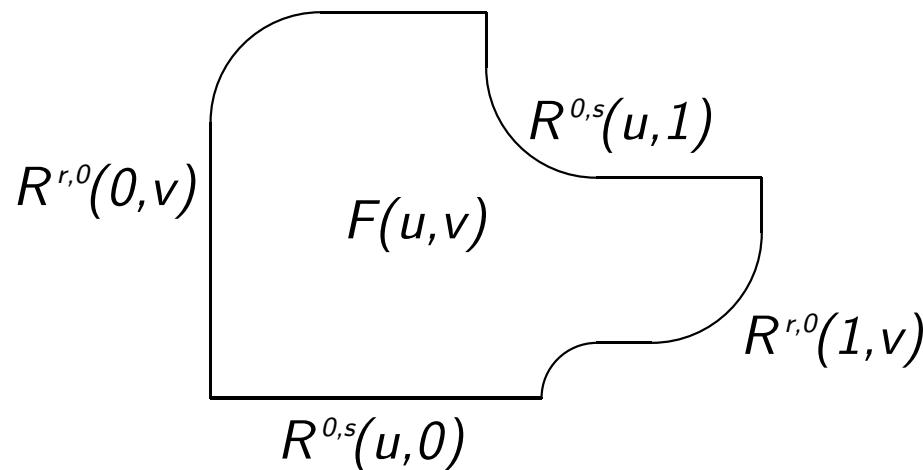
Die w_i sind zusätzliche konstante Gewichte, die noch stärkeren Einfluß auf den Kurvenverlauf ermöglichen

- ein großes w_i zieht die Kurve dichter an den Punkt P_i heran
- ist $w_i = 1$ für alle i , erhält man die nicht-rationale B-Splinekurve

Beschreibung der Geometrie (Forts.)

Vorgehensweise bei der geometrischen Definition von Teilflächen:

- Vorgabe der Ränder (Randbedingungen) und Interpolation der Fläche dazwischen (Verfahren von Coons):



$R^{r,s}$: - Funktion, die den jeweiligen Rand lokal beschreibt sowie
- Funktionen, die differentialgeometrische Eigenschaften der
Fläche quer zum jeweiligen Rand beschreiben

Beschreibung der Geometrie (Forts.)

- die allgemeine Coons'sche Fläche kann in vektorwertiger parametrischer Notation formuliert werden als:

$$\begin{aligned} F(u, v) = & \sum_{i=0}^1 \sum_{r=0}^m R^{r,0}(i, v) \cdot \beta_{r,i}(u) + \sum_{j=0}^1 \sum_{s=0}^n R^{0,s}(u, j) \cdot \beta_{s,j}(v) \\ & - \sum_{i=0}^1 \sum_{j=0}^1 \sum_{r=0}^m \sum_{s=0}^n R^{r,s}(i, j) \cdot \beta_{r,i}(u) \cdot \beta_{s,j}(v) \end{aligned}$$

für die Flächenparameter u und v gilt $0 \leq u \leq 1$ und $0 \leq v \leq 1$
 m und n geben den Grad der Ableitbarkeit am jeweiligen Rand an;
 $\beta_{r,i}(u)$ und $\beta_{s,j}(v)$ sind geeignet gewählte Verbindungsfunctionen
(blending functions)

Beschreibung der Geometrie (Forts.)

- die Verbindungsfunctionen müssen folgende Eigenschaften haben:

- für $i = 0$ bzw. $i = 1$ gilt

$$\beta_{r,i}(u) = \delta_{i,u}$$

wobei

$$\delta_{i,u} = 1 \text{ wenn } u = i$$

$$\delta_{i,u} = 0 \text{ wenn } u \neq i$$

- für alle $0 \leq u \leq 1$ gilt

$$\sum_{i=0}^1 \beta_{r,i}(u) = 1$$

- für die a -te Ableitung von $\beta_{r,i}(u)$ gilt

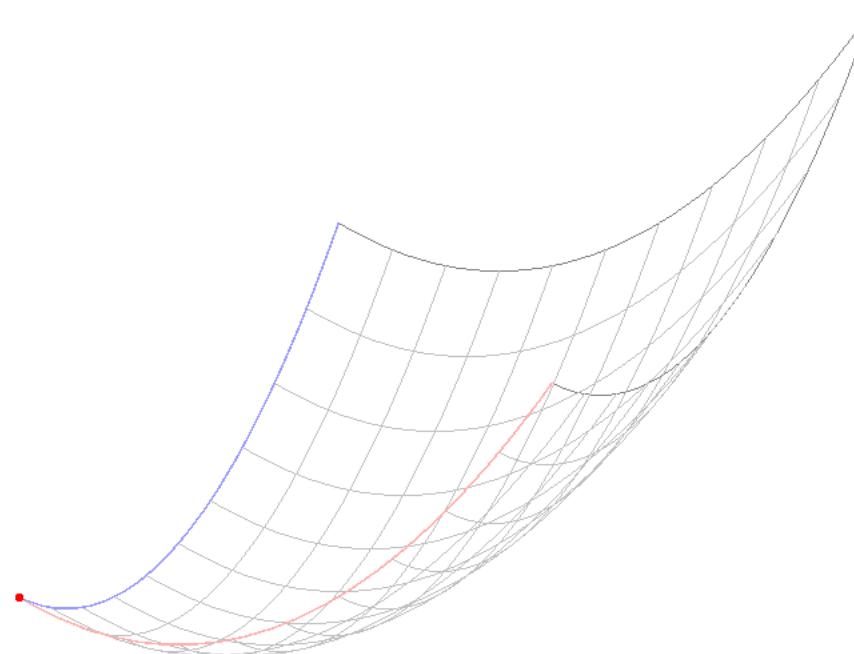
$$\beta_{r,i}^a(u_k) = \delta_{a,r} \cdot \delta_{i,k}$$

mit $u_0 = 0$ und $u_1 = 1$

dasselbe gilt analog für $\beta_{s,j}(v)$

Anmerkung zur grafischen Darstellung:

In der einfachsten Form können Oberflächen komplex gestalteter Objekte durch ihre Iso-Parameterlinien repräsentiert werden.

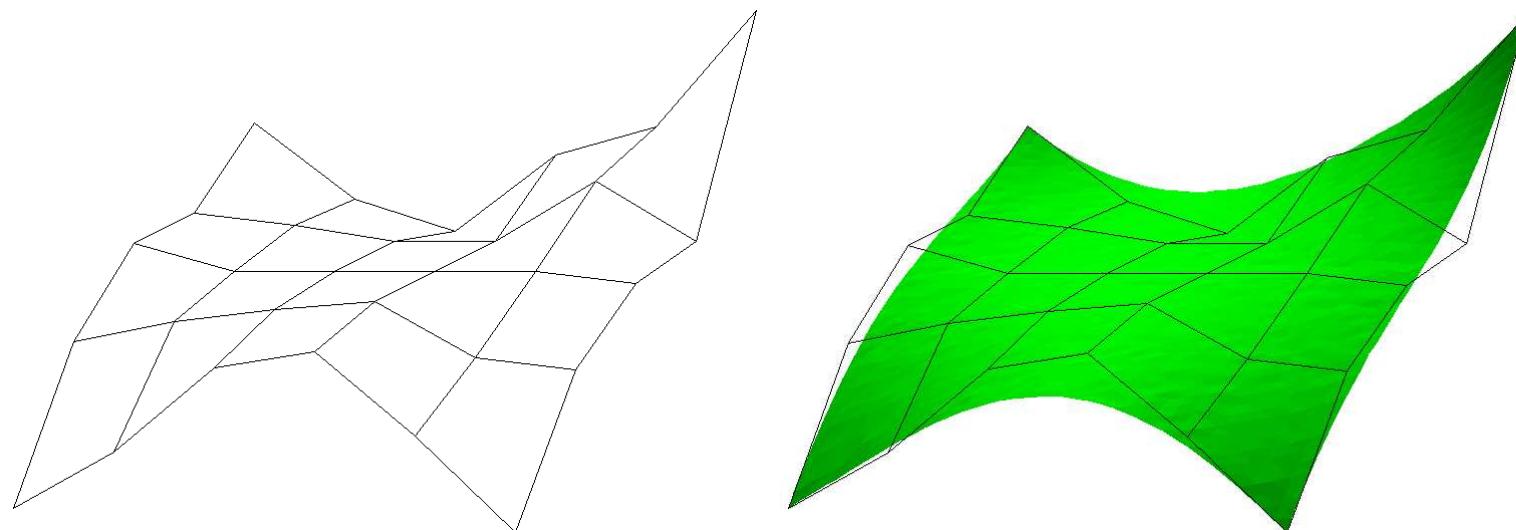


Die viereckigen Maschen des Netzes der Iso-Parameterlinien können über ihre Diagonalen in ebene Dreiecke unterteilt werden, die mit einfachen Shadern (Flat, Gouraud, Phong) flächig dargestellt werden können.

Beschreibung der Geometrie (Forts.)

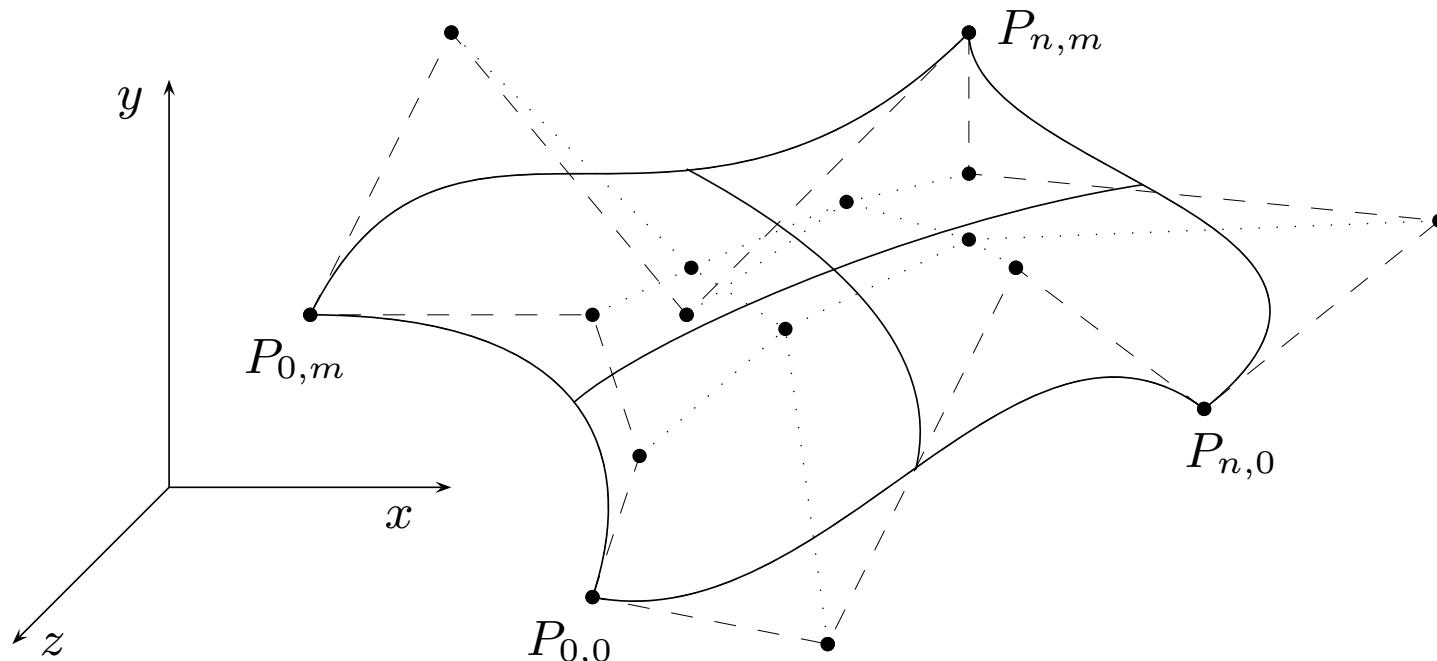
Vorgehensweise bei der geometrischen Definition von Teilflächen:

- Freiformgestaltung der Fläche durch Stütznetzvorgabe (Bézier- bzw. B-Spline-Approximation):



Einhaltung erforderlicher (lokaler, tangentialer oder Krümmungs-) Kontinuität über Flächenränder hinweg zu den Nachbarflächen wird durch die Verfahren gewährleistet

Beschreibung der Geometrie (Forts.)



- Analog zum Stützpolygon (Folge von $(n + 1)$ Punkten: $P_0 \cdots P_n$) zur Definition von approximierenden Kurven wird ein Stütznetz von $(n + 1) \times (m + 1)$ Punkten $P_{0,0} \cdots P_{n,m}$ zur Definition von approximierenden Flächen vorgegeben.

Beschreibung der Geometrie (Forts.)

- die Freiformfläche kann in vektorwertiger parametrischer Notation formuliert werden als:

$$F(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} \cdot W_i(u) \cdot W_j(v)$$

$P_{i,j}$ sind die Knotenpunkte des zu approximierenden Stütznetzes;
 W_i und W_j sind geeignete Wichtungsfunktionen, die ggf. von n bzw. m abhängen können (Verwendung finden z.B. die Bernsteinpolynome $B_{n,i}(u)$ und $B_{m,j}(v)$ für Bézierflächen bzw. die B-Splinebasisfunktionen $N_{i,k_u}(u)$ und $N_{j,k_v}(v)$ für B-Splineflächen)

Beschreibung der Geometrie (Forts.)

- Definition der Bernsteinpolynome:

$$B_{n,i}(u) = \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \quad \text{für } 0 \leq u \leq 1$$

- rekursive Definition der B-Splinebasisfunktionen:

$$N_{i,k}(u) = \frac{u-u_i}{u_{i+k-1}-u_i} N_{i,k-1}(u) + \frac{u_{i+k}-u}{u_{i+k}-u_{i+1}} N_{i+1,k-1}(u)$$

für $u_0 \leq u \leq u_{max}$ und $u_i \in (u_0, u_1, \dots, u_{max})$, $u_i \leq u_{i+1}$

mit $\frac{!}{0} = 0$ und der Endbedingung für $k = 1$:

$$N_{i,1}(u) = \begin{cases} 1 & \text{für } u_i \leq u \leq u_{i+1} \\ 1 & \text{für } u = u_{max} \text{ und } i = n \\ 0 & \text{sonst} \end{cases}$$

Beschreibung der Geometrie (Forts.)

- NURBS-Flächen:

$$F(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} \cdot R_{i,j}(u, v)$$

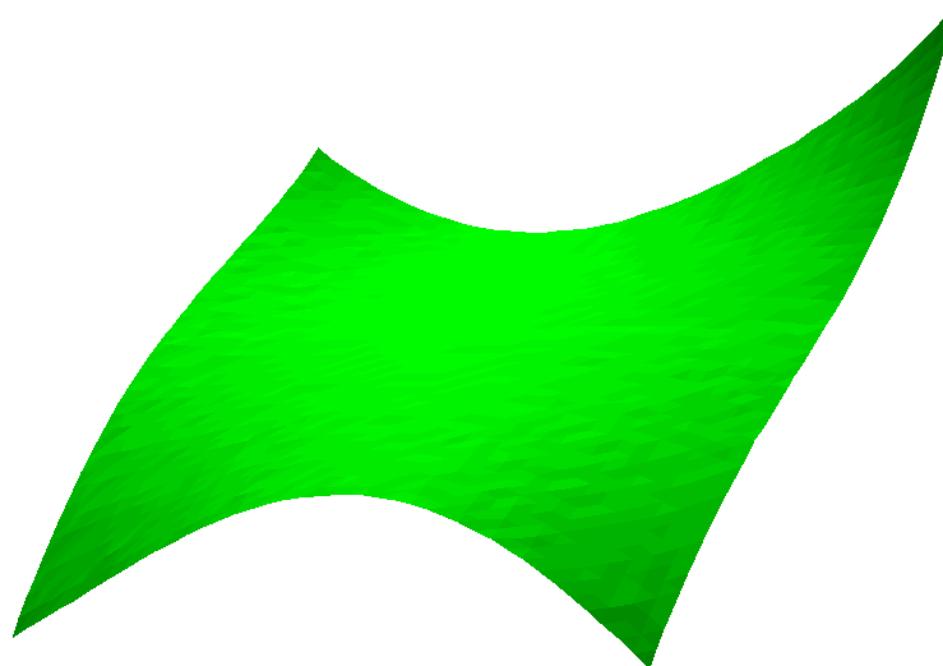
mit den Wichtungsfunktionen

$$R_{i,j}(u, v) = \frac{w_{i,j} \cdot N_{i,k_u}(u) \cdot N_{j,k_v}(v)}{\sum_{g=0}^n \sum_{h=0}^m w_{g,h} \cdot N_{g,k_u}(u) \cdot N_{h,k_v}(v)}$$

Randbedingungen, Eigenschaften, etc. dieser Flächen sind denen der entsprechenden Kurven vergleichbar.

Beschreibung der Geometrie (Forts.)

Darstellung der Approximationsfläche über ihre Iso-Parameterlinien.



Die viereckigen Maschen des Netzes der Iso-Parameterlinien wurden über ihre Diagonalen in ebene Dreiecke unterteilt, die mit einfachem Flat-Shading flächig ausgefüllt wurden.

Beschreibung der Topologie

Die vollständige geometrische Beschreibung repräsentiert im wesentlichen die gesamte Information über die geometrische Form (Gestalt) eines Gegenstandes.

jedoch: “unorganisiert”

(d.h. als ungeordnete Menge von geometrischen Elementen)

Die topologische Beschreibung “organisiert” die geometrische Information durch Darstellung von “Nachbarschaftsbeziehungen”.

Beschreibung der Topologie (Forts.)

Die in der topologischen Beschreibung repräsentierten Nachbarschaftsbeziehungen geben Auskunft über Sachverhalte wie:

- welche Flächenelemente bilden zusammen eine geschlossene Oberfläche eines Gegenstandes
- welche Flächenelemente grenzen an ein bestimmtes Flächenelement der Oberfläche an
- von welchen Kanten wird ein Flächenelement begrenzt
- welche geometrischen Elemente (Flächen, Kanten) haben welche Eckpunkte gemeinsam
- etc.

Beschreibung der Topologie (Forts.)

Die topologischen Beziehungen können in Form von Wertepaaren in einer Matrix notiert werden.

Das erste Symbol bezeichnet das jeweilige Bezugselement, das zweite die ihm benachbarte Gruppe

(P = Eckpunkt, K = Kante, F = Fläche):

$$\begin{bmatrix} PP & PK & PF \\ KP & KK & KF \\ FP & FK & FF \end{bmatrix}$$

Beschreibung der Topologie (Forts.)

Eine **topologisch ausreichende** Beschreibung ist in der Lage Nachbarschaftsbeziehungen vollständig und eindeutig wiederzugeben.

theoretisch ausreichend

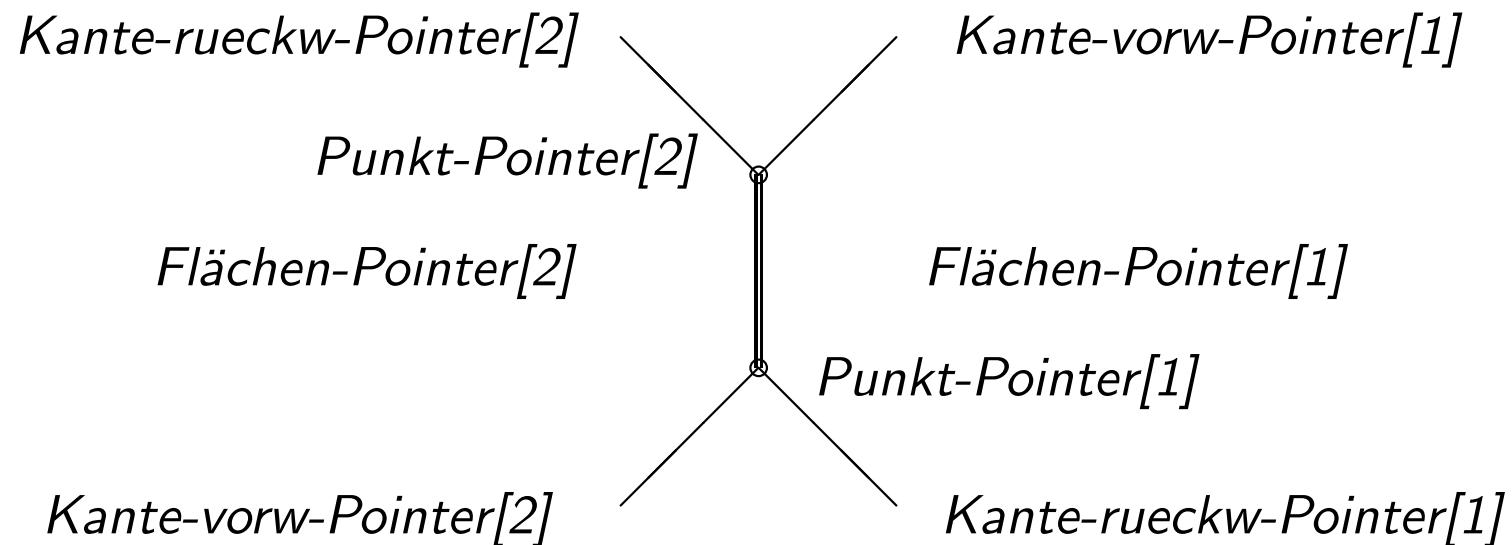
ist das absolute Minimum an Information, das benötigt wird, um die vollständige Nachbarschaftstopologie eindeutig zu repräsentieren

praktisch ausreichend

ist das Minimum an Information, das bei einer praktischen Realisierung, z.B. in einem geometrischen Modellersystem, benötigt wird
(eine gewisse Redundanz ist erforderlich, um häufig benötigte Nachbarschaftsinformationen in angemessen kurzer Zeit – z.B. im Rahmen von Interaktionen – zu erhalten)

Beschreibung der Topologie (Forts.)

Die topologische Beschreibung kann entsprechend in eine geeignete Datenstruktur umgesetzt werden, z.B. die kantenbasierte Datenstruktur “Winged Edge Structure”:



(geeignet zur Repräsentation zweidim. mannigfaltiger Körpermodelle)

Zur grafischen Darstellung modellierter Gegenstände

Vor- und Nachteile bei den unterschiedlichen Repräsentationsformen:

- **Octree** und **Voxelcluster**:
es existiert keine Information über die darzustellende geschlossene Oberfläche des Gegenstandes (muß zur grafischen Darstellung aufwendig approximiert werden)
- **CSG**:
“verwendete Teile” der Grundkörper müssen erst ermittelt werden, ehe ihre Oberfläche dargestellt werden kann
- **B-Rep**:
geometrische Beschreibung enthält genau die Flächenelemente, die dargestellt werden sollen;
topologische Beschreibung gibt Ordnung vor, die für effiziente grafische Darstellung genutzt werden kann

Ausführlichere Behandlung der begrenzungsbasierten Modellierung in der Vorlesung zum Master-Vertiefungsmodul MV-AGIS6 CF

Computergrafik

Werner Hansmann

Sommersemester 2011

Interactive Visual Computing (IVC)

bzw.

Computergrafik und Bildsynthese (CGB)

(Wintersemester 2010/11)

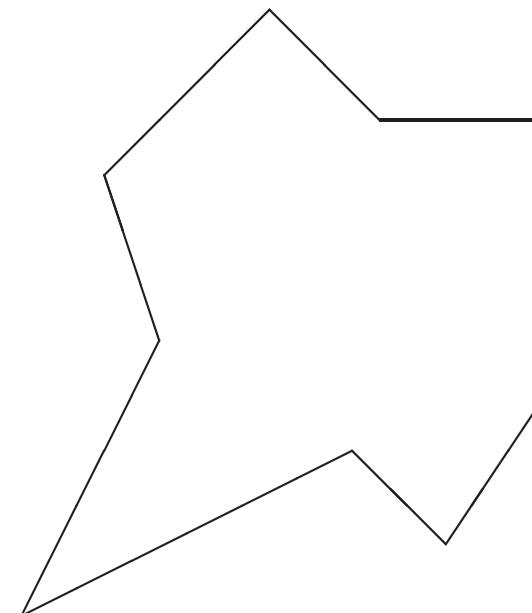
Werner Hansmann

Grafische Algorithmen

Grafische Algorithmen

Unter “Polygon” sei nachfolgend ein geschlossener Polygonzug im 2 D verstanden.

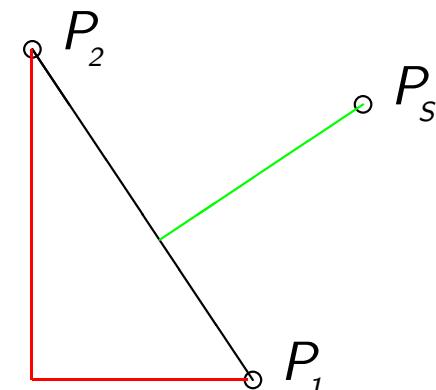
- Abstand zwischen zwei Punkten bzw. zwischen Punkt und Gerade
- Flächeninhalt eines Polygons
- Konvexe Hülle eines Polygons
- Punkt in Polygon
- Schnittpunkt von zwei Polygonkanten
- Überlappen von Polygonen
- Schraffieren von Polygonen
- Ausschnittsbildung und Kappen



Abstand zwischen zwei Punkten bzw. zwischen Punkt und Gerade

Abstand zwischen zwei Punkten P_1 und P_2 :

$$A = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{Pythagoras})$$



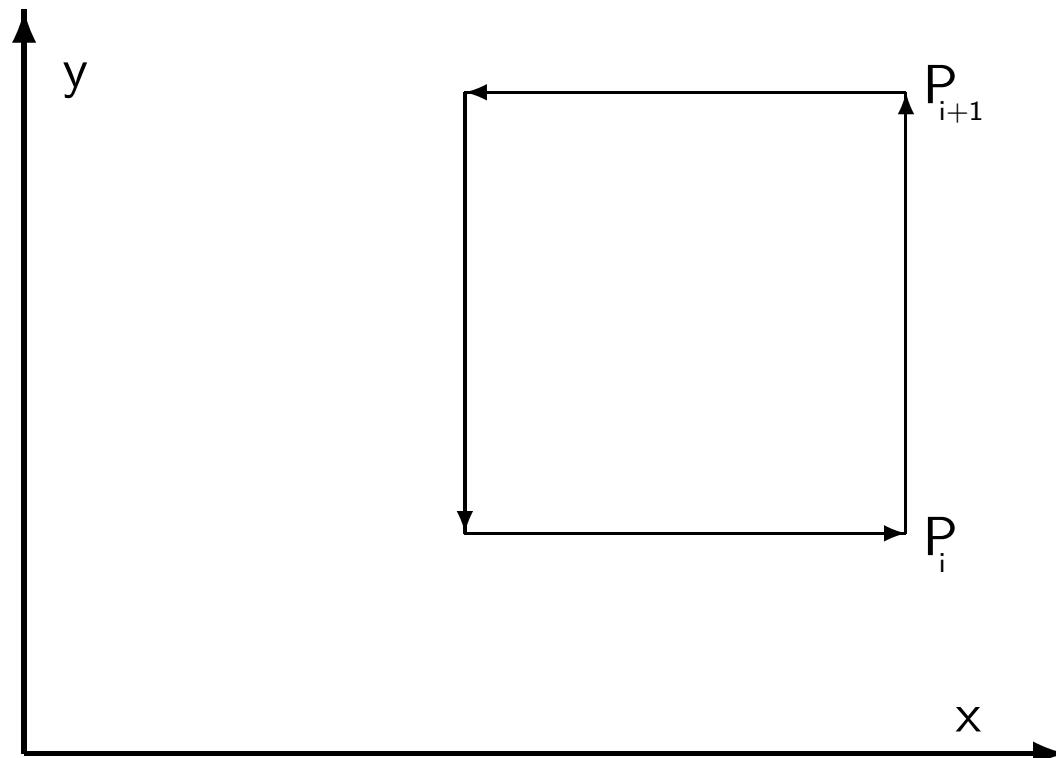
Abstand zwischen einem Punkt P_s und einer Geraden $ax + by + c = 0$:

$$A = \frac{a}{\sqrt{a^2 + b^2}} x_s + \frac{b}{\sqrt{a^2 + b^2}} y_s + \frac{c}{\sqrt{a^2 + b^2}} \quad (\text{Hessesche Normalform})$$

$A > 0$, wenn Koordinatenursprung und P_s auf verschiedenen Seiten der Geraden liegen.

Flächeninhalt eines Polygons (Polygon Area)

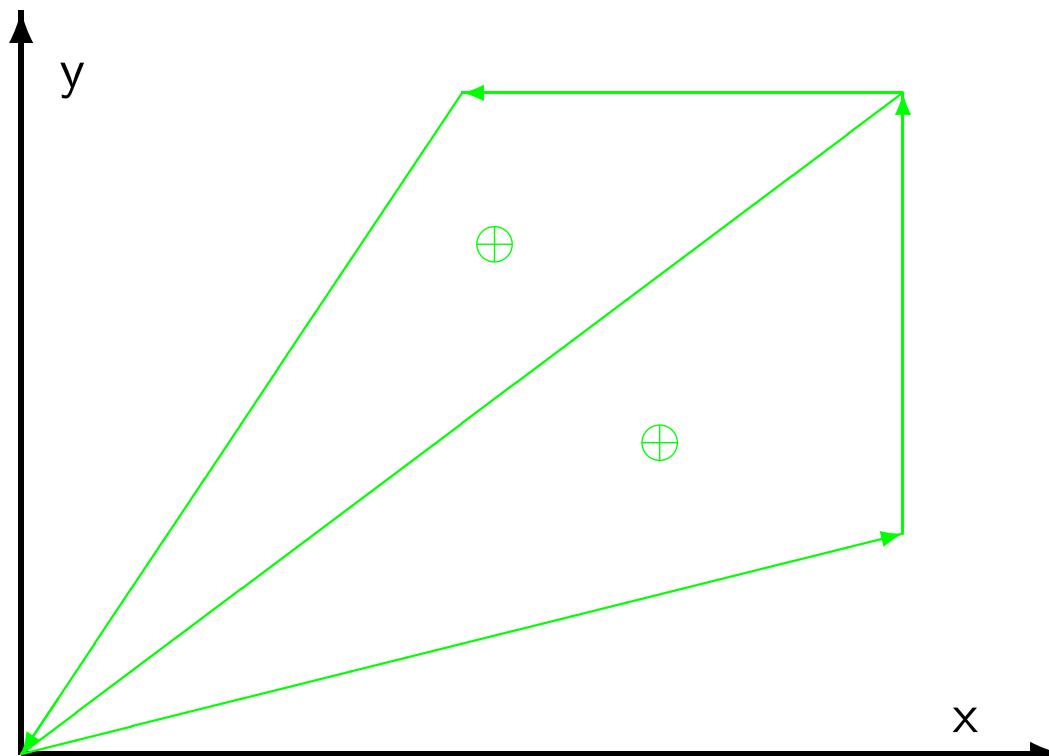
bei orientierten Polygonkanten (mathem. positiver Umlaufsinn)
und mit $P_n = P_1$ wird



$$F = \frac{1}{2} \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i)$$

Flächeninhalt eines Polygons (Polygon Area)

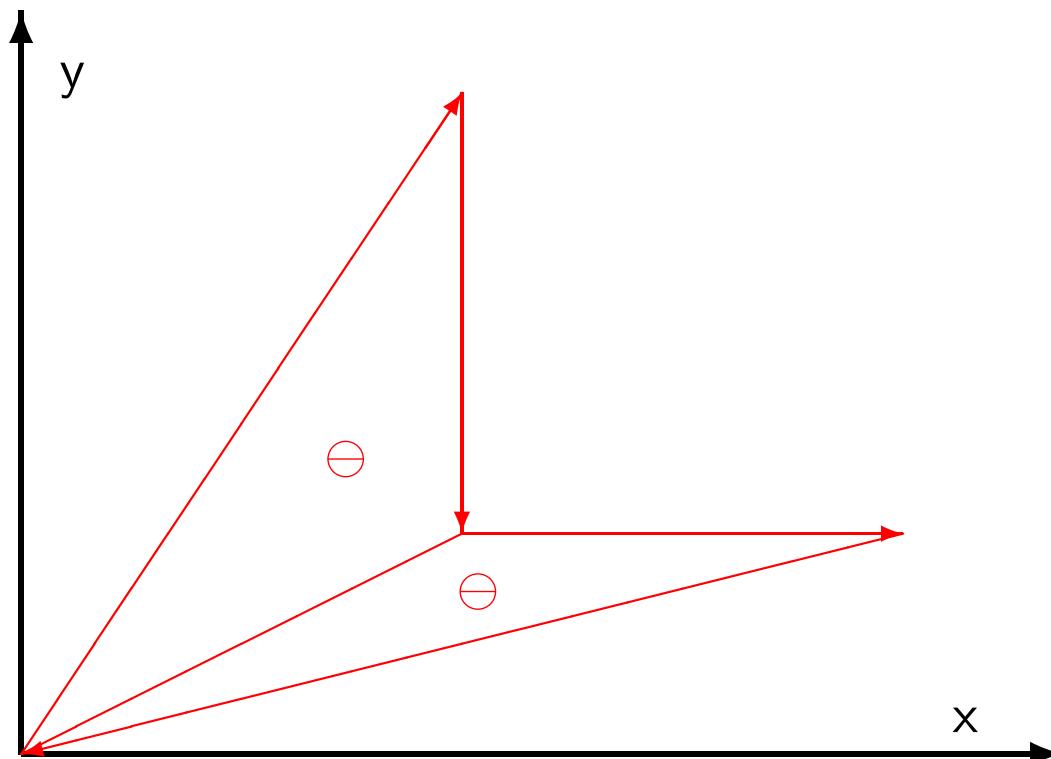
Dreiecke
mit positivem
Flächeninhalt



$$\text{je Dreieck : } F = \frac{1}{2}(x_i y_{i+1} - x_{i+1} y_i)$$

Flächeninhalt eines Polygons (Polygon Area)

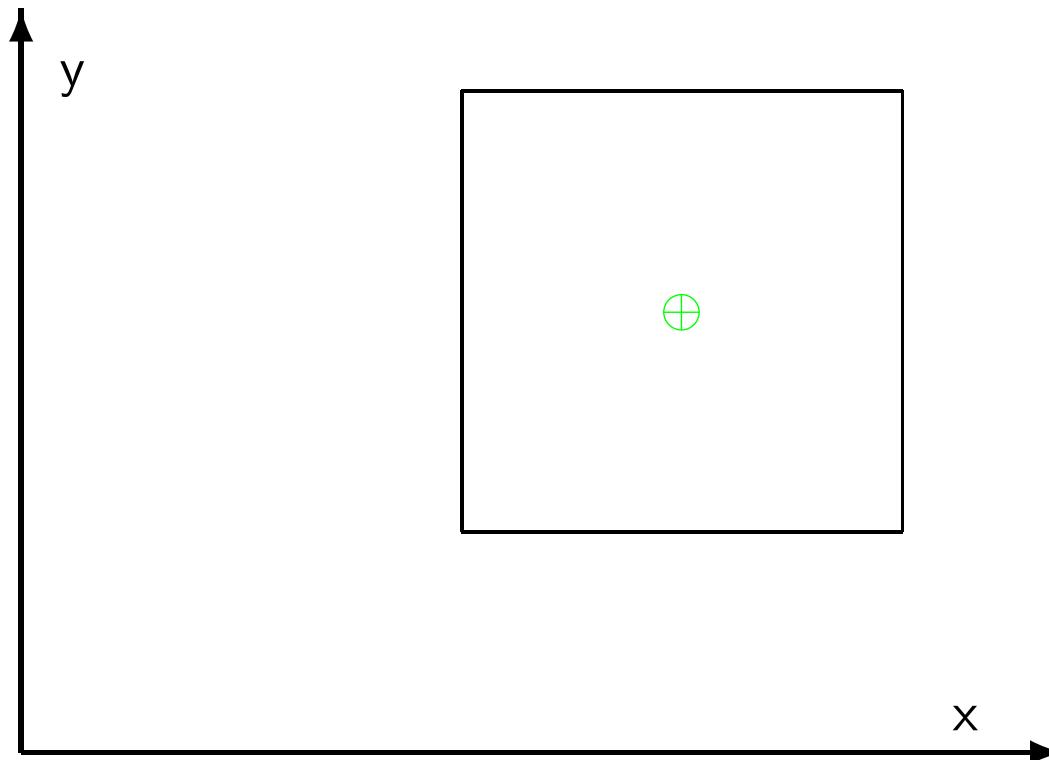
Dreiecke
mit negativem
Flächeninhalt



$$\text{je Dreieck : } F = \frac{1}{2}(x_i y_{i+1} - x_{i+1} y_i)$$

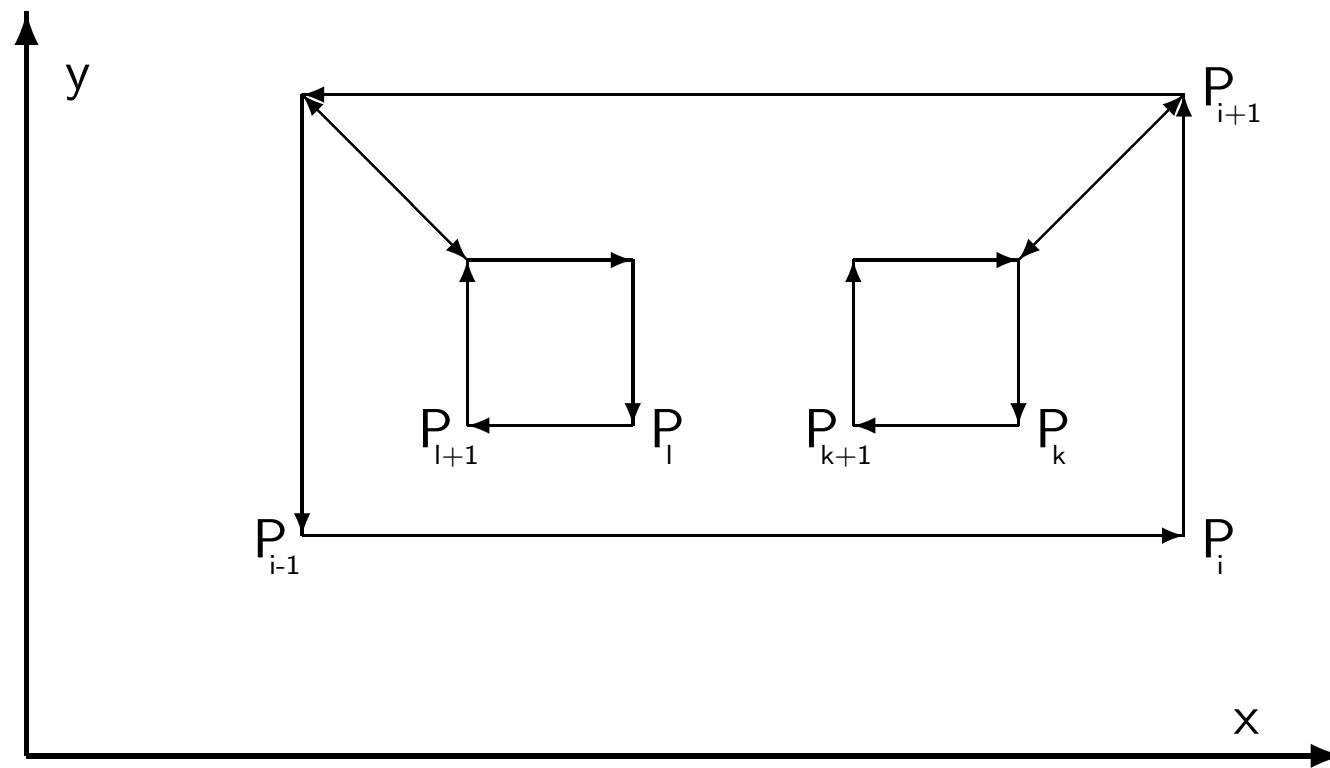
Flächeninhalt eines Polygons (Polygon Area)

verbleibender
positiver
Flächeninhalt



Flächeninhalt eines Polygons (Forts.)

mehrfach zusammenhängende Flächen



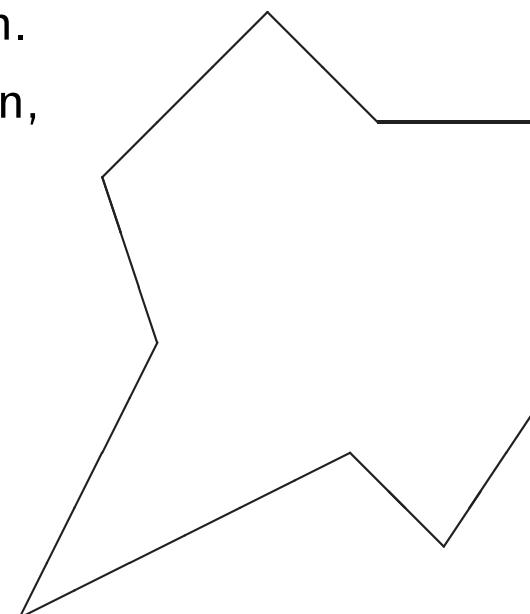
Konvexe Hülle eines Polygons (Convex Hull)

Gegeben sei ein beliebig gestaltetes Polygon.

Die „konvexe Hülle“ ist das kleinste Polygon,
das

- das gegebene Polygon völlig beinhaltet
- keine „Beulen“ nach innen hat, d.h.
jede beliebige Gerade schneidet
dieses Polygon nur zweimal

(



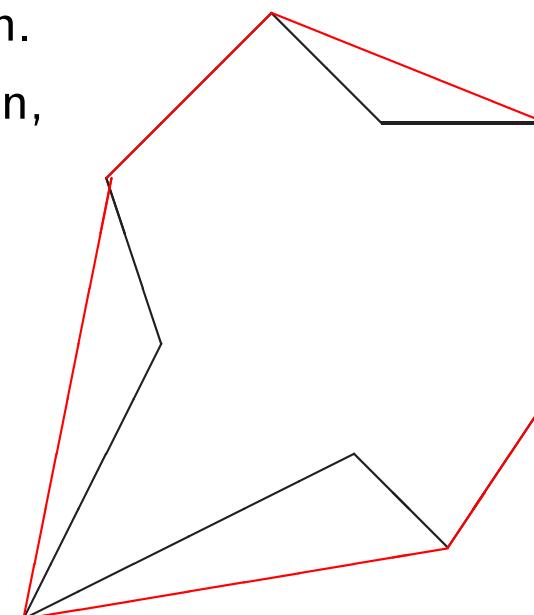
Konvexe Hülle eines Polygons (Convex Hull)

Gegeben sei ein beliebig gestaltetes Polygon.

Die „konvexe Hülle“ ist das kleinste Polygon,
das

- das gegebene Polygon völlig beinhaltet
- keine „Beulen“ nach innen hat, d.h.
jede beliebige Gerade schneidet
dieses Polygon nur zweimal

(„Gummibandprinzip“)



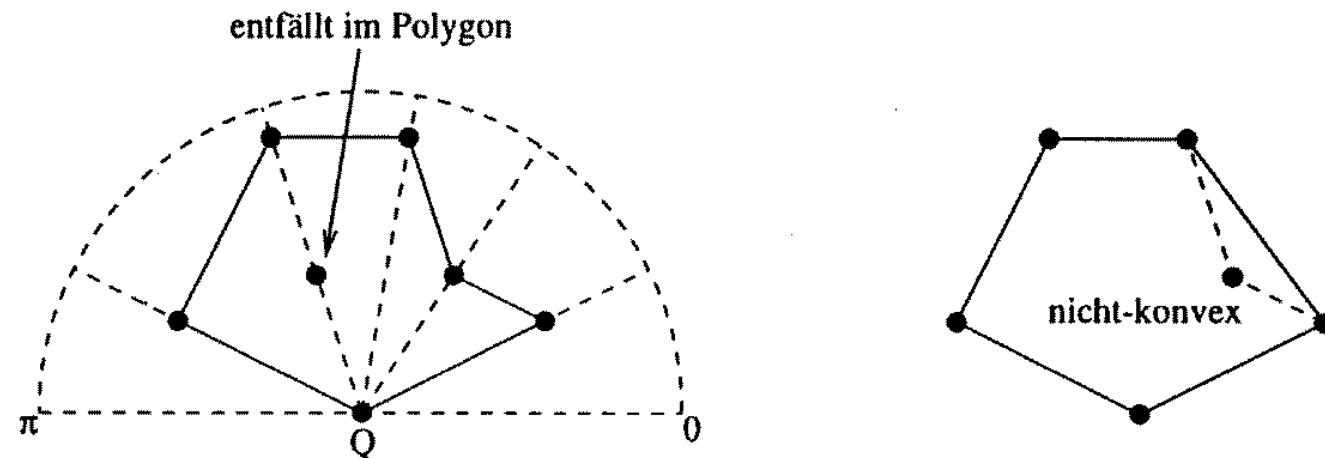
Konvexe Hülle einer Punktmenge (Convex Hull)

der Algorithmus:

1. auswählen des Referenzpunktes Q (Punkt mit kleinster Ordinate, bei Mehrdeutigkeit zusätzlich kleinster Abszisse)
2. sortieren der Punkte P_i nach Winkel, den sie mit Q und der positiven Abszissenrichtung einschließen (gleicher Winkel für mehrere Punkte: festhalten nur des von Q entferntesten)
3. verbinden der sortierten Punkte zu einem geschlossenen Polygon
4. aussortieren aller nicht-konvexen Eckpunkte

Ist bereits ein geschlossenes Polygon vorgegeben, können sofort die nicht-konvexen Eckpunkte aussortiert werden.

Konvexe Hülle einer Punktmenge (Forts.)



Aussortieren aller nicht-konvexen Eckpunkte

Teile das Polygon an den Eckpunkten P_{min} und P_{max} mit der kleinsten und der größten Abszisse (und ggf. kleinsten Ordinate) in zwei Hälften. Definiere P_{min} zum Ausgangspunkt der Untersuchung der unteren Polygonhälfte und speichere ihn als ersten Punkt des Hüllpolygons.

- a) verbinde den Ausgangspunkt mit seinem Nachfolger; ist damit P_{max} erreicht → **fertig**
- b) liegen alle auf diesen Nachfolger folgenden Eckpunkte links von der gerichteten Verbindung? – **ja:** speichere diesen Nachfolger als nächsten Punkt des Hüllpolygons und mache ihn zum neuen Ausgangspunkt; fahre fort bei a) – **nein:** verbinde den Ausgangspunkt mit dem Nachfolger des Nachfolgers; fahre fort bei b)

Verfahre analog mit der oberen Polygonhälfte.

Schnittpunkt von zwei Polygonkanten (Edge Intersection)

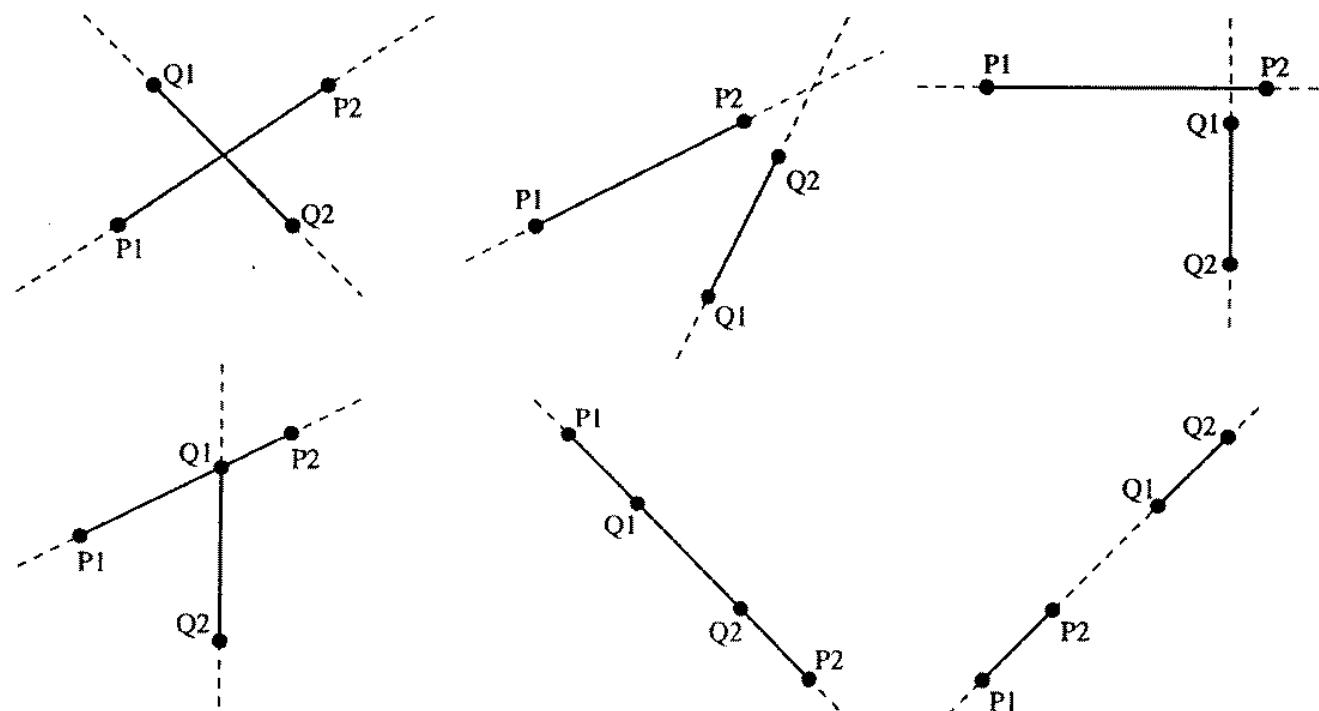
Der Schnittpunkt zweier Geraden

$a_1x + b_1y + c_1 = 0$ und $a_2x + b_2y + c_2 = 0$ ergibt sich zu

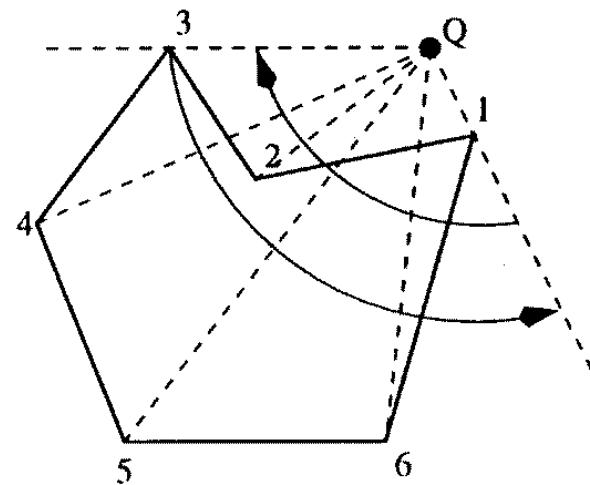
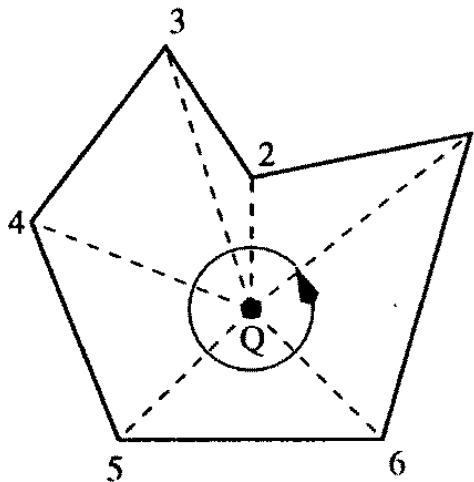
$$x_s = \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}, \quad y_s = \frac{c_1a_2 - c_2a_1}{a_1b_2 - a_2b_1}$$

Eine Punktüberschneidung liegt dann vor, wenn bez. beider Kanten die Endpunkte der jew. anderen Kante auf verschiedenen Seiten liegen.

Schnittpunkt von zwei Polygonkanten (mögliche Fälle)

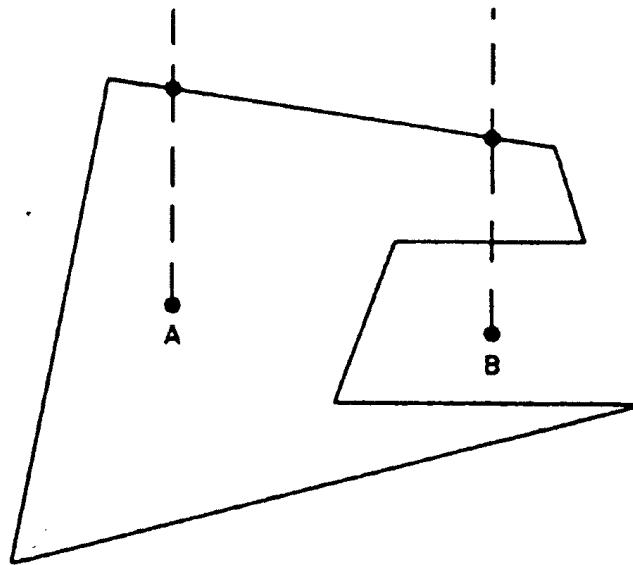


Punkt in Polygon (Point in Polygon) – Winkelsummenmethode



$Winkelsumme \simeq 2\pi \Rightarrow \text{innerhalb}, \quad Winkelsumme \simeq 0 \Rightarrow \text{außerhalb}$

Punkt in Polygon (Point in Polygon) – Schnittpunktemethode



ungeradeAnzahlSchnittpkte. \Rightarrow innerhalb, geradeAnz. \Rightarrow außerh.

Sonderbehandlung nötig, wenn Halbgerade durch Polygonecke verläuft !

Überlappen von Polygonen (Polygon Overlap)

Vorauswahl durch **Minimax-Test** (minimum box test):

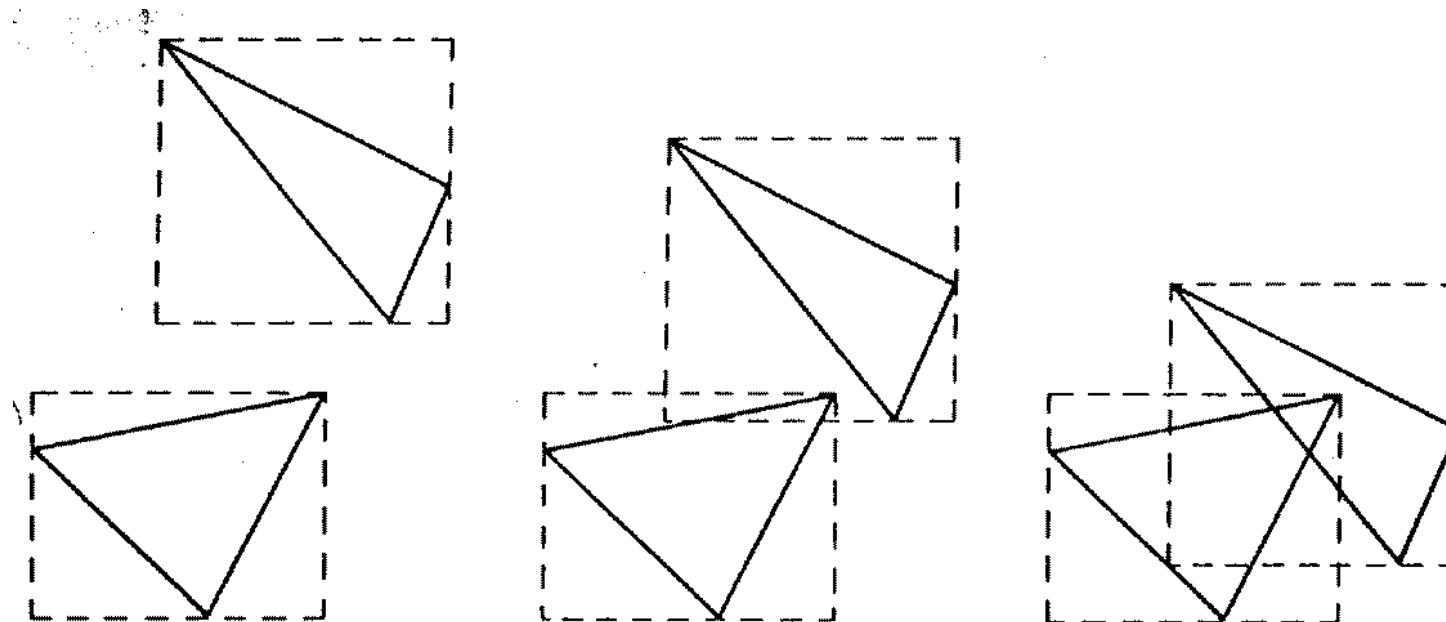
Ist

1. x_{min} des einen Polygons größer als x_{max} des anderen oder
2. y_{min} des einen Polygons größer als y_{max} des anderen,

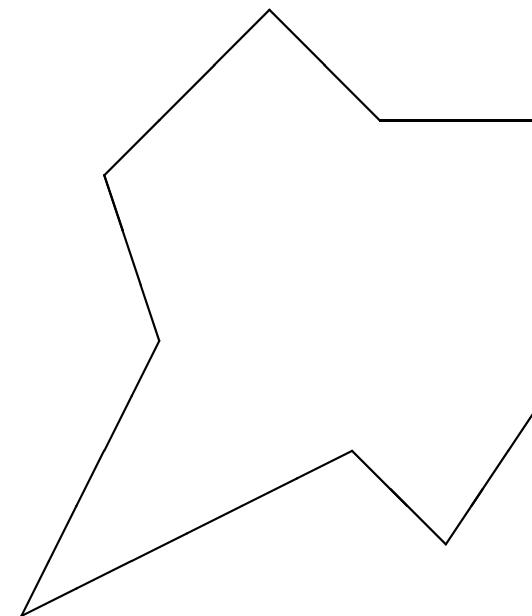
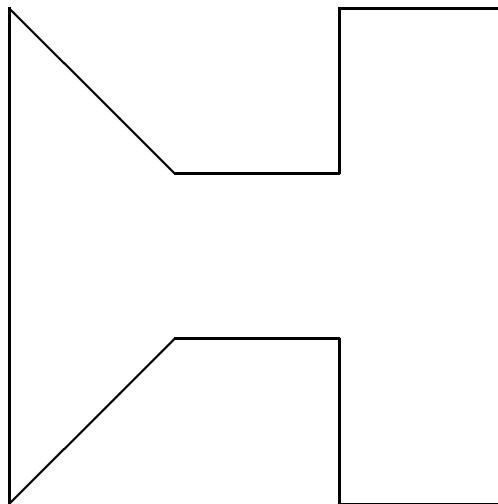
dann überlappen sich die Polygone nicht.

Andernfalls müssen die Polygonkanten jeweils direkt auf Überschneiden geprüft werden (paarweiser Minimax-Test für die Kanten);
ggf. muß der Schnittpunkt ermittelt werden.

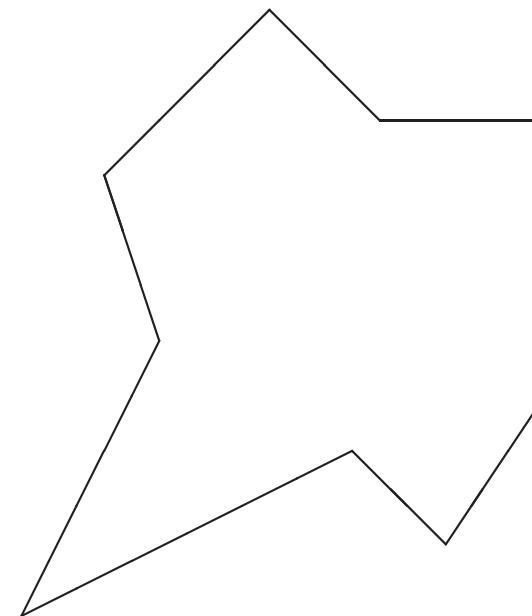
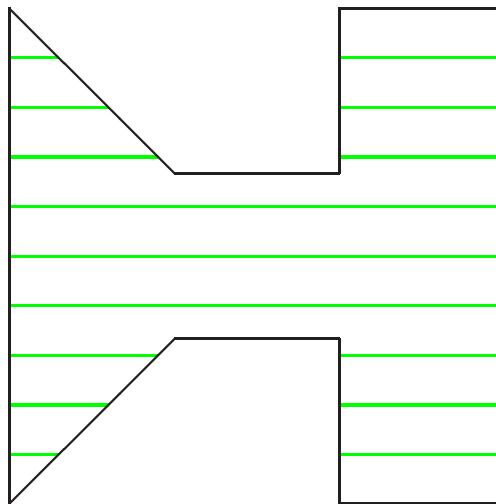
Überlappen von Polygonen (mögliche Fälle)



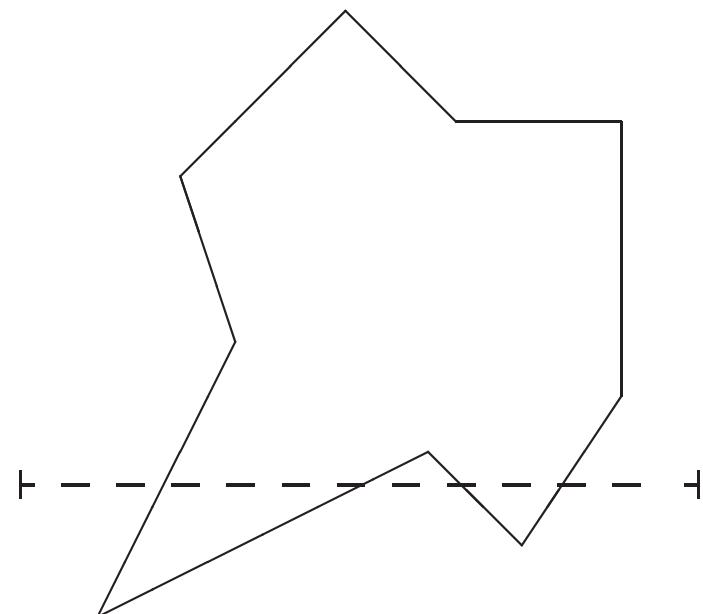
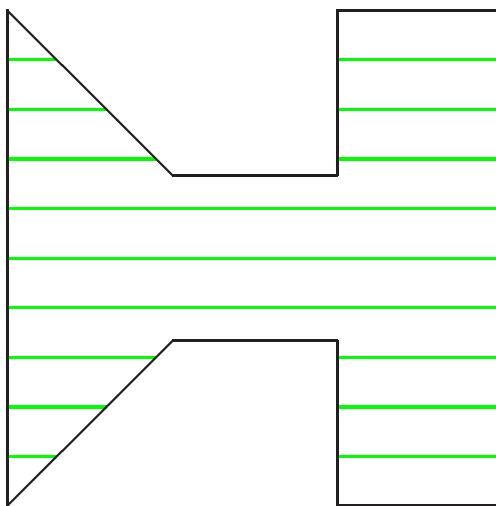
Schraffieren von Polygonen (Crosshatching)



Schraffieren von Polygonen (Crosshatching)



Schraffieren von Polygonen (Crosshatching)



Schraffieren von Polygonen (der Algorithmus)

1. ermitteln der extremen Polygon-Ordinaten y_{min} und y_{max}
2. beginnend bei y_{min} :
 - (a) inkrementieren der Schraffurlinienordinate um den Schraffurlinienabstand
 - (b) bestimmen der Schnittpunkte der Schraffurlinie mit den Polygonkanten
 - (c) sortieren der Schnittpunkte nach aufsteigender Abszisse
 - (d) zeichnen einer Verbindung der sortierten Schnittpunkte in aufsteigender Reihenfolge, und zwar jeweils
 - "zum ungeraden Punkt" unsichtbar und
 - "zum geraden Punkt" sichtbar
- fortsetzen bei (a) bis y_{max} erreicht

Schraffieren von Polygonen (der Algorithmus)

Schraffur unter einem Winkel α :

1. drehen der Polygon-Eckpunktkoordinaten um den Winkel $-\alpha$
2. ermitteln der extremen Polygon-Ordinaten y_{min} und y_{max}
3. beginnend bei y_{min} :
 - (a) inkrementieren der Schraffurlinienordinate
 - (b) bestimmen der Schnittpunkte der (jetzt horizontalen) Schraffurlinie mit den Polygonkanten
 - (c) sortieren der Schnittpunkte nach aufsteigender Abszisse
 - (d) drehen der sortierten Schnittpunkte um α
 - (e) zeichnen einer Verbindung der Schnittpunkte in aufsteigender Reihenfolge (s.o.)
- fortsetzen bei (a) bis y_{max} erreicht

Ausschnittsbildung und Kappen (Windowing and Clipping)

- **Window:** abzubildender Ausschnitt aus dem Weltkoordinatensystem
- **Viewport:** Teil des Zielkoordinatensystems, auf den das Window abgebildet werden soll (Geräteseite)
- Transformation zwischen Koordinatensystemen
(im GKS zweistufig:
 1. Normierungstransformation:
Weltkoordinaten \Rightarrow Normalized Device Coordinates (NDC)
 2. Gerätetransformation:
Normalized Device Coordinates (NDC) \Rightarrow Geräte-Viewport)

Für jede Anwendung können mehrere solche Transformationen spezifiziert werden

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Ziele des Windowing:

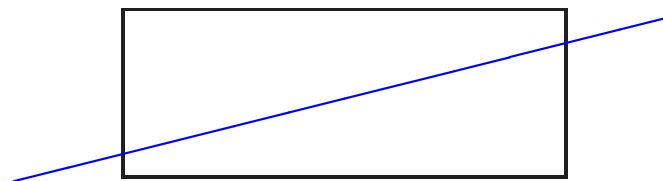
- Geräteunabhängigkeit
(Festlegung des Formats für das Ausgabegerät ist völlig entkoppelt)
- ein Window kann in unterschiedlicher Weise auf mehrere Geräte ausgegeben werden
- mehrere Windows können einen Viewport nutzen

Window und Viewport werden in der Regel als achsparallele Rechtecke realisiert.

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Clipping-Algorithmen:

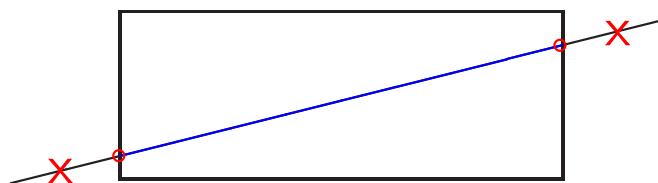
- Algorithmen, die Darstellungselemente, die ganz oder teilweise außerhalb des Viewports liegen, abschneiden
(setzen meist schon bei der Window-Viewport-Transformation an; die Gerätetransformation für so „abgeschnittene“ Objektteile kann dann entfallen)



Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Clipping-Algorithmen:

- Algorithmen, die Darstellungselemente, die ganz oder teilweise außerhalb des Viewports liegen, abschneiden
(setzen meist schon bei der Window-Viewport-Transformation an; die Gerätetransformation für so „abgeschnittene“ Objektteile kann dann entfallen)



Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Cohen-Sutherland-Algorithmus:

- Aufteilung der Bildebene in neun Regionen (Viewport im Zentrum)

1001	1000	1010
0001	0000	0010
0101	0100	0110

- Zuordnung eines 4-Bit-Schlüssels zu jeder Region; Bedeutung:
 1. Bit: Punkt liegt links vom linken Rand
 2. Bit: Punkt liegt rechts vom rechten Rand
 3. Bit: Punkt liegt unterhalb der unteren Randlinie
 4. Bit: Punkt liegt über der oberen Randlinie des Viewports

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Cohen-Sutherland-Algorithmus: Auswertung 1. Schritt

eine Gerade liegt völlig

- innerhalb des Viewports, wenn der Code für beide Endpunkte 0000 ist
- außerhalb des Viewports, wenn der Durchschnitt (logisches UND) der Codes beider Endpunkte verschieden von Null ist

andernfalls wird der Schnittpunkt zwischen Gerade und der kreuzenden Randlinie berechnet

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Cohen-Sutherland-Algorithmus: Auswertung 2. Schritt

ist m die Steigung der Geraden, ergibt sich der Schnittpunkt mit

- dem linken Rand zu: $x_s = x_l$ und $y_s = y_1 + (x_l - x_1)m$
- dem rechten Rand zu: $x_s = x_r$ und $y_s = y_1 + (x_r - x_1)m$
- dem unteren Rand zu: $x_s = x_1 + \frac{y_u - y_1}{m}$ und $y_s = y_u$
- dem oberen Rand zu: $x_s = x_1 + \frac{y_o - y_1}{m}$ und $y_s = y_o$

der Schnittpunkt wird als neuer Endpunkt der Geraden entsprechend codiert; danach wird beim 1. Schritt fortgefahren
horizontale bzw. vertikale Geraden erfahren Sonderbehandlung

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Liang-Barsky-Algorithmus (Annahmen):

- die (gerichteten) Begrenzungskanten des Viewport zerlegen die Bildebene jeweils in zwei Halbebenen, wobei der **sichtbare** Teil auf einer Seite liegt.
- der **Durchschnitt** der Halbebenen mit dem sichtbaren Teil ergibt den (sichtbaren) Viewport



Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

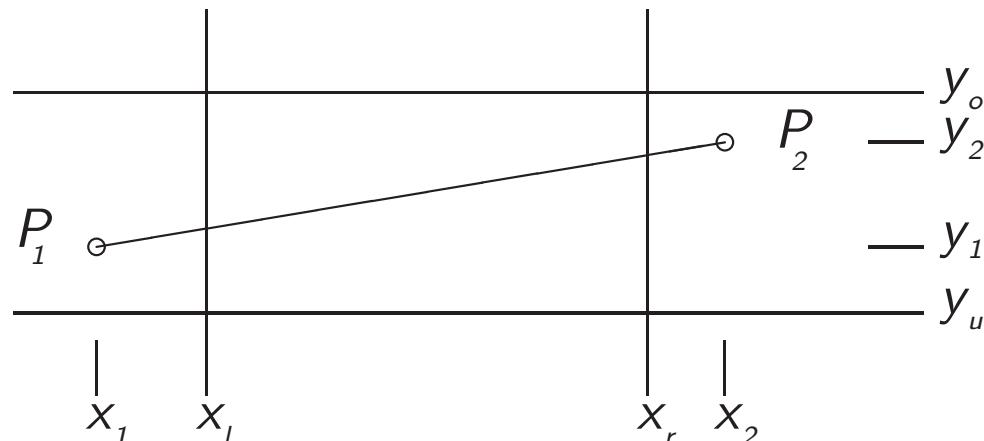
Liang-Barsky-Algorithmus (Forts.):

- eine Gerade zwischen zwei Punkten ist gegeben in Parameterform:

$$x = x_1 + t \cdot (x_2 - x_1)$$

$$y = y_1 + t \cdot (y_2 - y_1) \quad \text{und} \quad 0 \leq t \leq 1$$

- die Viewportgrenzen sind x_l , x_r , y_u und y_o



Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Liang-Barsky-Algorithmus (Forts.):

- im Viewport gilt mit

$$p_1 = x_1 - x_2, \quad q_1 = x_1 - x_l,$$

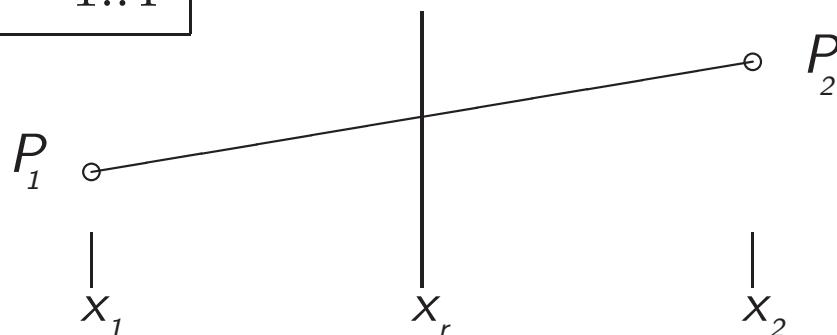
$$p_2 = x_2 - x_1, \quad q_2 = x_r - x_1,$$

$$p_3 = y_1 - y_2, \quad q_3 = y_1 - y_u,$$

$$p_4 = y_2 - y_1 \quad \text{und} \quad q_4 = y_o - y_1:$$

$$p_i \cdot t \leq q_i \quad \text{mit } i = 1..4$$

Beispiel:



Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Liang-Barsky-Algorithmus (Forts.):

Die Ungleichungen $p_i \cdot t \leq q_i$ beschreiben je eine der sichtbaren Halbebenen, die durch die Begrenzungskanten des Viewports definiert sind. Es gilt:

- wenn $q_i \geq 0 \quad \forall i$ liegt P_1 im sichtbaren Bereich
- wenn $p_i > q_i \quad \forall i$ verläßt die Gerade den sichtbaren Bereich
- wenn $p_i < q_i \quad \forall i$ geht die Gerade in den sichtbaren Bereich
- wenn $p_i = 0$ verläuft die Gerade parallel zur i-ten Begrenzungskante (mit $i = 1$: links, $i = 2$: rechts, $i = 3$: unten, $i = 4$: oben)

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Liang-Barsky-Algorithmus (Forts.):

Gesucht: die Parameterwerte t_1 und t_2 für den sichtbaren Teil der Geraden; Schnittpunkte der Geraden mit den Begrenzungskanten ergeben sich für $p_i \neq 0$ jeweils für den Parameterwert $t = q_i/p_i$

Für die beiden gesuchten Parameterwerte folgt somit:

$$t_1 = \max(\{q_i/p_i \mid p_i < q_i, i = 1, \dots, 4\} \cup \{0\})$$

$$t_2 = \min(\{q_i/p_i \mid p_i > q_i, i = 1, \dots, 4\} \cup \{1\})$$

Einsetzen dieser Werte in die Parametergleichung der Geraden ergibt die Schnittpunkte mit dem Rand des Viewport

Ausschnittsbildung und Kappen (Windowing and Clipping) (Forts.)

Vergleich der Algorithmen:

- Cohen-Sutherland kommt mit deutlich weniger Gleitkomma-Operationen aus
- Liang und Barsky haben gemessen (4 Mio zufällige Clip-Operationen), daß ihr Verfahren um 1/3 schneller ist

Interactive Visual Computing (IVC)

bzw.

Computergrafik und Bildsynthese (CGB)

(Wintersemester 2010/11)

Werner Hansmann

Photometrisches Modellieren

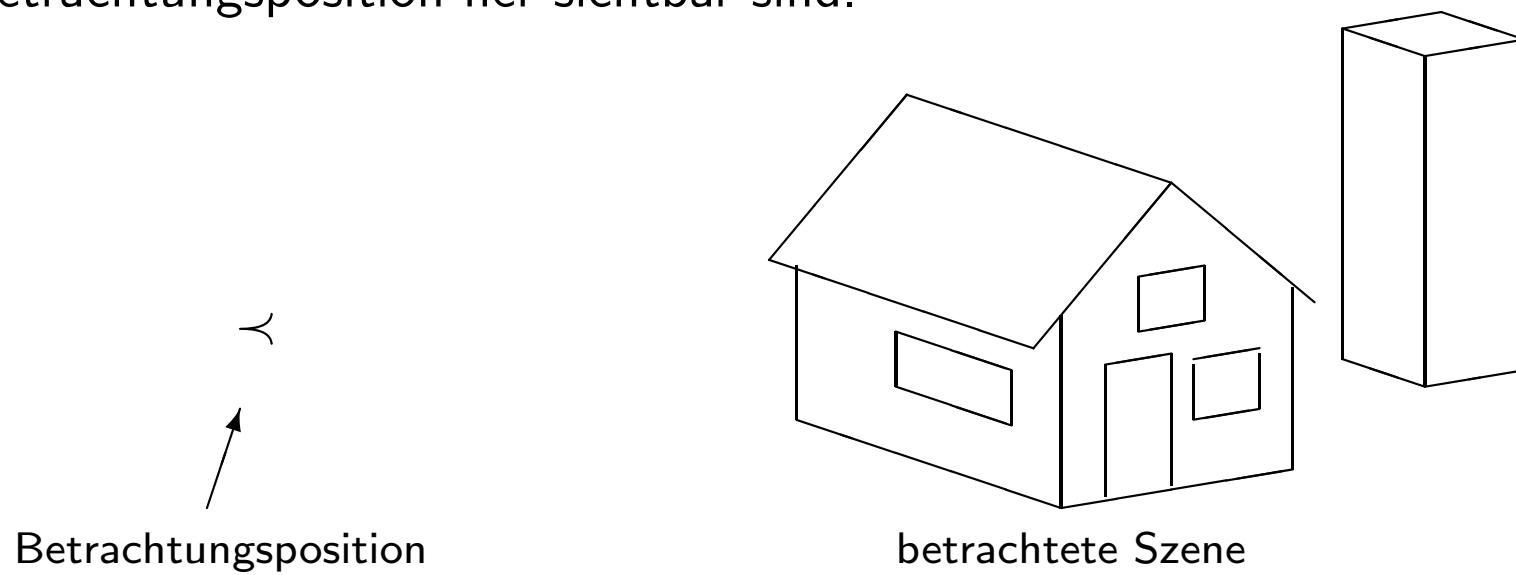
Photometrisches Modellieren

- Berücksichtigung verdeckter Kanten bzw. Flächen
- Ein Experiment
- Photometrische Modelle
- Schattierung - lokale Beleuchtungsmodelle
- Fotorealismus - globale Beleuchtungsmodelle
- Nicht-Fotorealismus – Rückkehr der Kunst

Berücksichtigung verdeckter Kanten bzw. Flächen

Hidden Line bzw. Hidden Surface Removal-Techniken (HLR / HSR)

Die entsprechenden Algorithmen untersuchen, welche Teile (Kanten bzw. Flächen) eines darzustellenden modellierten Gegenstandes von der Betrachtungsposition her sichtbar sind.



Verdeckte Kanten bzw. Flächen (Forts.)

Die existierenden Algorithmen können klassifiziert werden in

- Objektraum-Algorithmen
 - Kantenorientierte Algorithmen
 - Flächenorientierte Algorithmen
- Prioritätslisten-Algorithmen
 - A-Priori-Algorithmen
 - Dynamische Algorithmen
- Bildraum-Algorithmen
 - Flächenorientierte Algorithmen
 - Punktorientierte Algorithmen

Verdeckte Kanten bzw. Flächen (Forts.)

Die Auswahl eines Algorithmus sollte abhängig sein von:

- der Art der Anwendung
 - Animation
 - Einzelbilderstellung
 - etc.
- der Komplexität der Szene
- den Eigenschaften der Ausgabegeräte
 - Vektordisplay
 - Rasterdisplay

Verdeckte Kanten bzw. Flächen (Forts.)

Merkmale und Eigenschaften der Algorithmen

- Objektraum-Algorithmen
 - sind so genau, wie es die Maschinengenauigkeit des verwendeten Computers zuläßt
 - Ausschnittsvergrößerungen sind mit hoher Genauigkeit möglich
 - der Rechenaufwand steigt mit der Anzahl der darzustellenden Objekte

Verdeckte Kanten bzw. Flächen (Forts.)

Merkmale und Eigenschaften der Algorithmen

- Prioritätslisten-Algorithmen
 - dienen der Beschleunigung von Bildfolgenerstellungen
 - Potentielle gegenseitige Verdeckungen werden vorab (im Objektraum) ermittelt (hohe Genauigkeit, großer Rechenaufwand)
 - Abbildungsberechnungen erfolgen im Bildraum (schnell)

Verdeckte Kanten bzw. Flächen (Forts.)

Merkmale und Eigenschaften der Algorithmen

- Bildraum-Algorithmen
 - sind so genau, wie die Rasterauflösung des Ausgabegerätes
 - Vergrößerungen sind unbefriedigend
 - der Rechenaufwand ist abhängig von der Bildauflösung

Verdeckte Kanten bzw. Flächen (Forts.)

Berechnungsaufwand der Algorithmen

- Objektraum-Algorithmen $O(n^2)$
- Bildraum-Algorithmen $O(np)$
(kann durch Ausnutzung von Kohärenzeigenschaften reduziert werden)

n : Anzahl der Objekte in der Szene

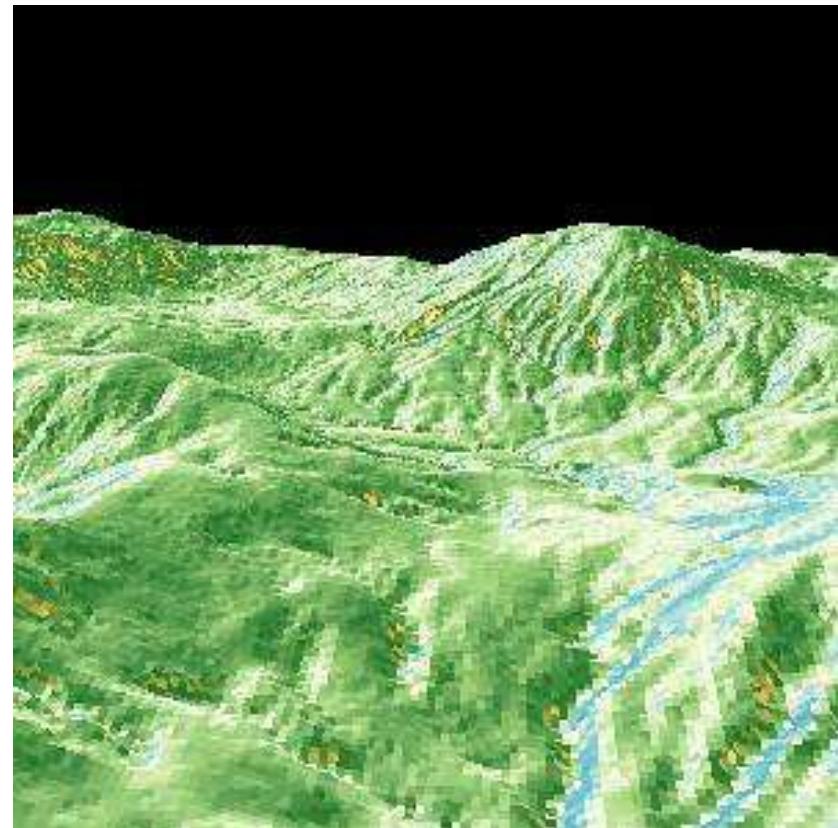
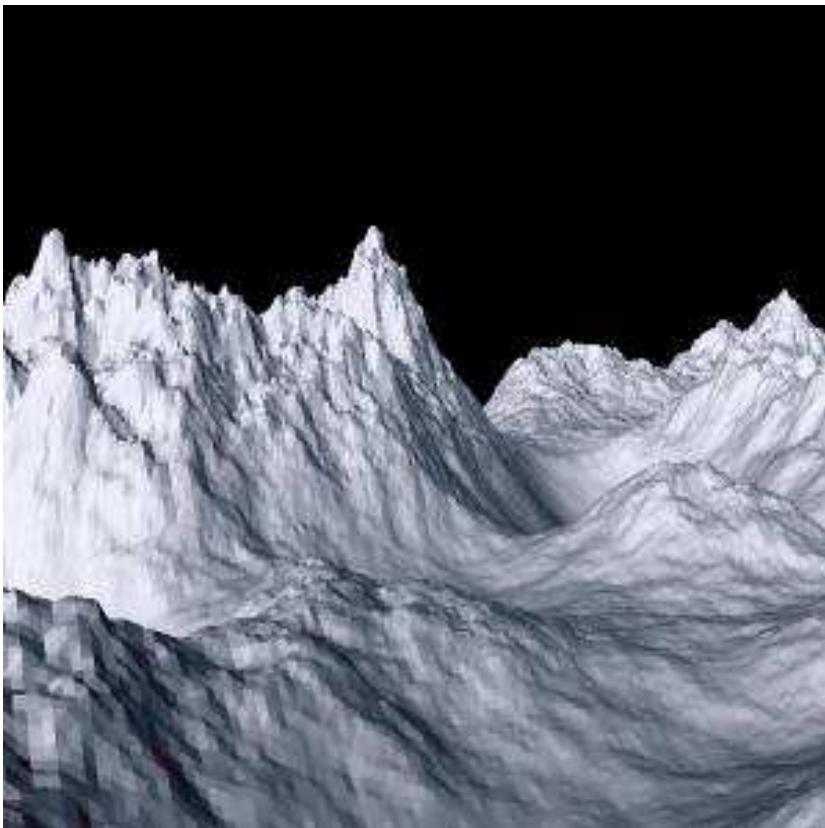
p : Anzahl der Bildpixel

Verdeckte Kanten bzw. Flächen (Forts.)

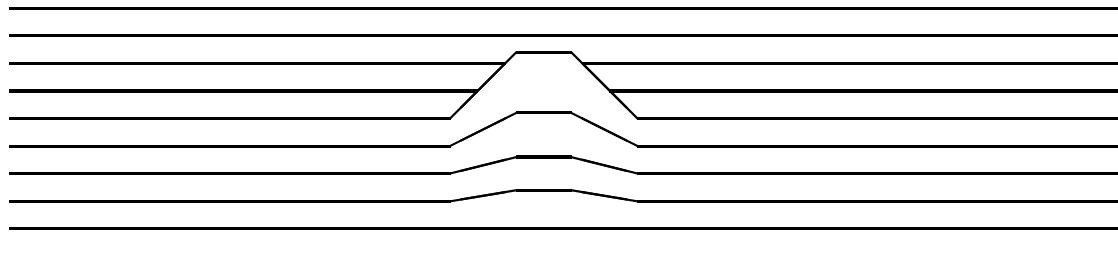
Exemplarische Algorithmen

- Methode des gleitenden Horizonts (floating horizon)
 - Entfernen abgewandter Flächen (backface culling)
 - Tiefenpuffer-Algorithmus (Z-buffer algorithm)
 - Bildschirmzeilen-Algorithmus (scan-line algorithm)
 - Maler-Algorithmus (painter's algorithm)

Methode des gleitenden Horizonts (floating horizon)



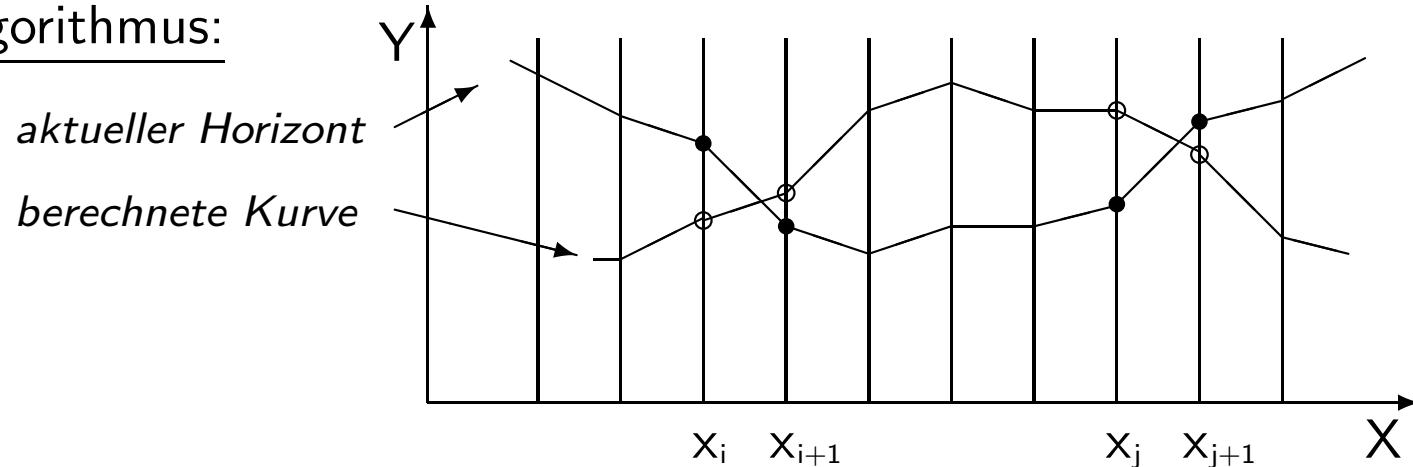
Methode des gleitenden Horizonts (floating horizon)



- das Verfahren dient der einfachen Darstellung von Flächen, die in der Form $F(x, y, z) = 0$ gegeben sind
- dargestellt werden Schnittlinien dieser Fläche mit parallelen Ebenen, z.B. mit äquidistanten Parallelten zur xy-Ebene mit den Abständen z_i
- sei also $F(x, y, z) = 0$ sowie $y = f(x, z_i)$ und $x = g(y, z_i)$ dann kann folgender Algorithmus verwendet werden:

Methode des gleitenden Horizonts (Forts.)

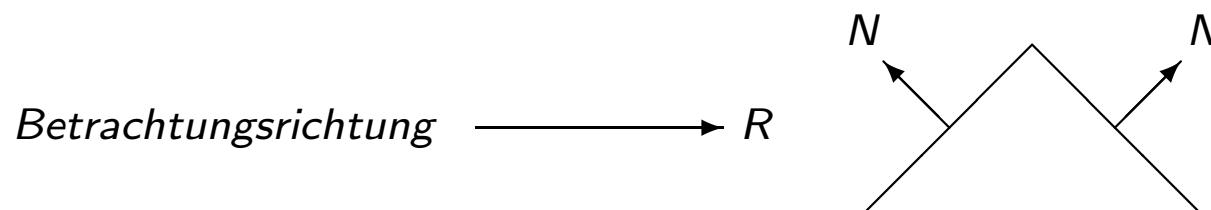
Algorithmus:



- die z_i werden nach dem Abstand vom Betrachtungspunkt aufsteigend sortiert
- beginnend mit dem ersten z_i -Wert wird für jede x-Koordinate im Bildraum der zugehörige $y(x, z_i)$ -Wert berechnet
- ist $y(x, z_i)$ größer als alle bisherigen y-Werte für diesen x-Wert, dann ist $y(x, z_i)$ sichtbar und wird dargestellt

Entfernen abgewandter Flächen (backface culling)

Grundidee: wenn die Betrachtungsrichtung R und die Flächennormale N in denselben Halbraum weisen, ist die Fläche nicht sichtbar.



Test: das Skalarprodukt aus Flächennormale N und Betrachtungsrichtung ist für „abgewandte“ (d.h. nicht sichtbare) Flächen positiv.

Hinweis: „Selbstverdeckung“ wird nicht berücksichtigt!

Entfernen abgewandter Flächen (Forts.)

Sonderfall: konvexe Polyeder

- konvexe Polyeder können sich nicht selbst verdecken und haben keine Löcher
- Hidden Surface Removal ist durch Backface Culling gelöst
- Backface Culling reduziert die Anzahl der darzustellenden Flächenelemente im Durchschnitt auf die Hälfte der vorhandenen Elemente

Resultat: hoher Effizienzgewinn für Objektraumverfahren

Tiefenpuffer-Algorithmus (Z-buffer algorithm)

Ein Bildraum-Verfahren

- **Datenstruktur:** ein 2D-Array, das für jedes Pixel des Viewports (Bildausschnitt) den Z-Wert des darzustellenden Oberflächenpunktes (d.h. seine Tiefe bzw. Entfernung von der Bildebene) enthält; es wird **Z-buffer** genannt;
das Array enthält schließlich die Z-Werte der dem Betrachtungspunkt nächsten Oberflächenpunkte;
die Größe des Arrays entspricht der des Viewports
- **Algorithmus:**
 - Löschen des Bildausschnitts (füllen mit der Hintergrundfarbe)
 - Initialisieren des Z-Buffers mit ∞

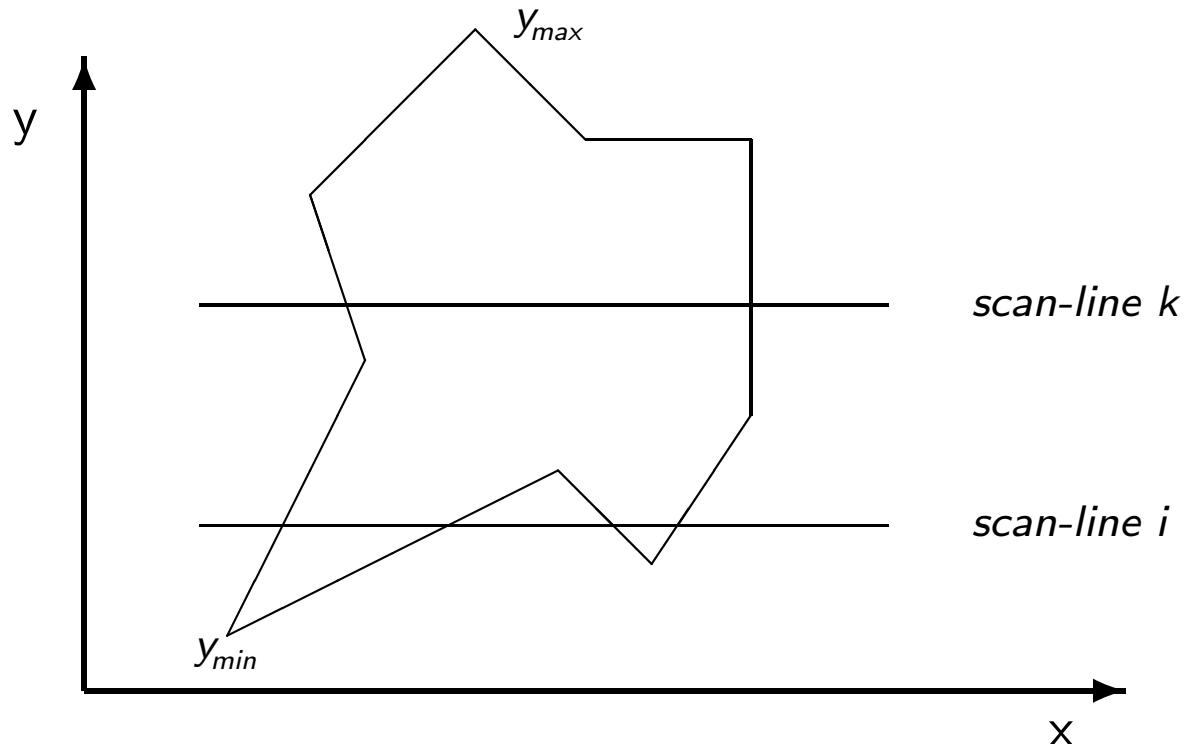
Tiefenpuffer-Algorithmus (Forts.)

- **Algorithmus:** (Forts.)
 - für jede Fläche F_i und für jedes Pixel $(x, y) \in F_i$
Berechnung der Farbe C und der Tiefe Z
ist $Z > Tiefe(x, y)$, dann
 - setze $Tiefe(x, y) = Z$ und
 - setze $Farbe(x, y) = C$andernfalls ist dieser Teil des Flächenelements verdeckt

Berechnung von Z:

die Oberfläche setzt sich aus ebenen Elementen zusammen;
d.h. Z variiert linear entlang einer Bildschirmzeile (Scanline);
 ΔZ ist konstant

Bildschirmzeilen-Algorithmus (scan-line algorithm)



Scan-Line-Algorithmen sind einfache Flächenfüll-Algorithmen, die dem gleichen Grundprinzip folgen, wie die Schraffur-Algorithmen. Dadurch daß sie aber im Bildraum arbeiten, sind sie noch erheblich einfacher.

Bildschirmzeilen-Algorithmus (Forts.)

Algorithmus:

sortieren der Liste der Polygonkanten nach größten y -Werten

1. ermitteln der extremen Polygon-Ordinaten y_{min} und y_{max}
2. beginnend bei y_{min} :
 - (a) Aktualisierung der Liste der zu betrachtenden Polygonkanten
 - (b) Ermittlung der Schnittpunkte der Scan-Line mit diesen Polygonkanten und Sortierung nach aufsteigendem x
 - (c) Darstellung der Scan-Line-Punkte
 - (d) Inkrementierung der Scan-Line
 - (e) Fortsetzung bei (a) bis y_{max} erreicht

Bildschirmzeilen-Algorithmus (Forts.)

Algorithmus (Forts.)

zu (c)

- die Elemente des Scan-Line-Puffers erhalten die Farbwerte der ursprünglichen Bildpunkte der Scan-Line (Initialisierung)
- für alle ungeraden $i > 0$:
beginnend bei dem Element, das den i -ten Schnittpunkt der Scan-Line mit einer Polygonkante repräsentiert, erhalten alle nachfolgenden Elemente bis zum $(i + 1)$ -ten den Farbwert der durchquerten Polygonfläche
- der modifizierte Scan-Line-Puffer wird ausgegeben

Maler-Algorithmus (painter's algorithm)

(Objekt- und Bildraumalgorithmus)

- alle Polygonflächen werden nach dem kleinsten z-Wert der Fläche bzgl. ihrer Entfernung vom Betrachtungspunkt sortiert
- beginnend mit der entferntesten Fläche wird für jede noch nicht bearbeitete Fläche untersucht, ob sie sich mit einer anderen in der Tiefe überlappt
- ist dies nicht der Fall, kann das Polygon angezeigt werden
- andernfalls ist zu prüfen, welches der beiden näher zum Betrachtungspunkt liegt – evtl. muß umsortiert werden

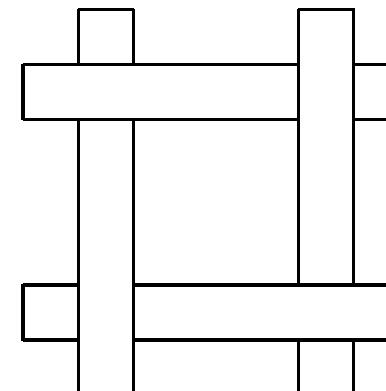
Maler-Algorithmus (Forts.)

Merkmale des Algorithmus:

- alle Polygone werden gezeichnet, aber von hinten nach vorne – sichtbare Polygone bzw. Polygonteile überschreiben automatisch verdeckte
- beim Test auf Verdeckung müssen zyklische Überlappungen erkannt werden.

Behebung:

Aufbrechen / Teilen einer Fläche



Ein Experiment

Formale Beschreibung der Geometrie:

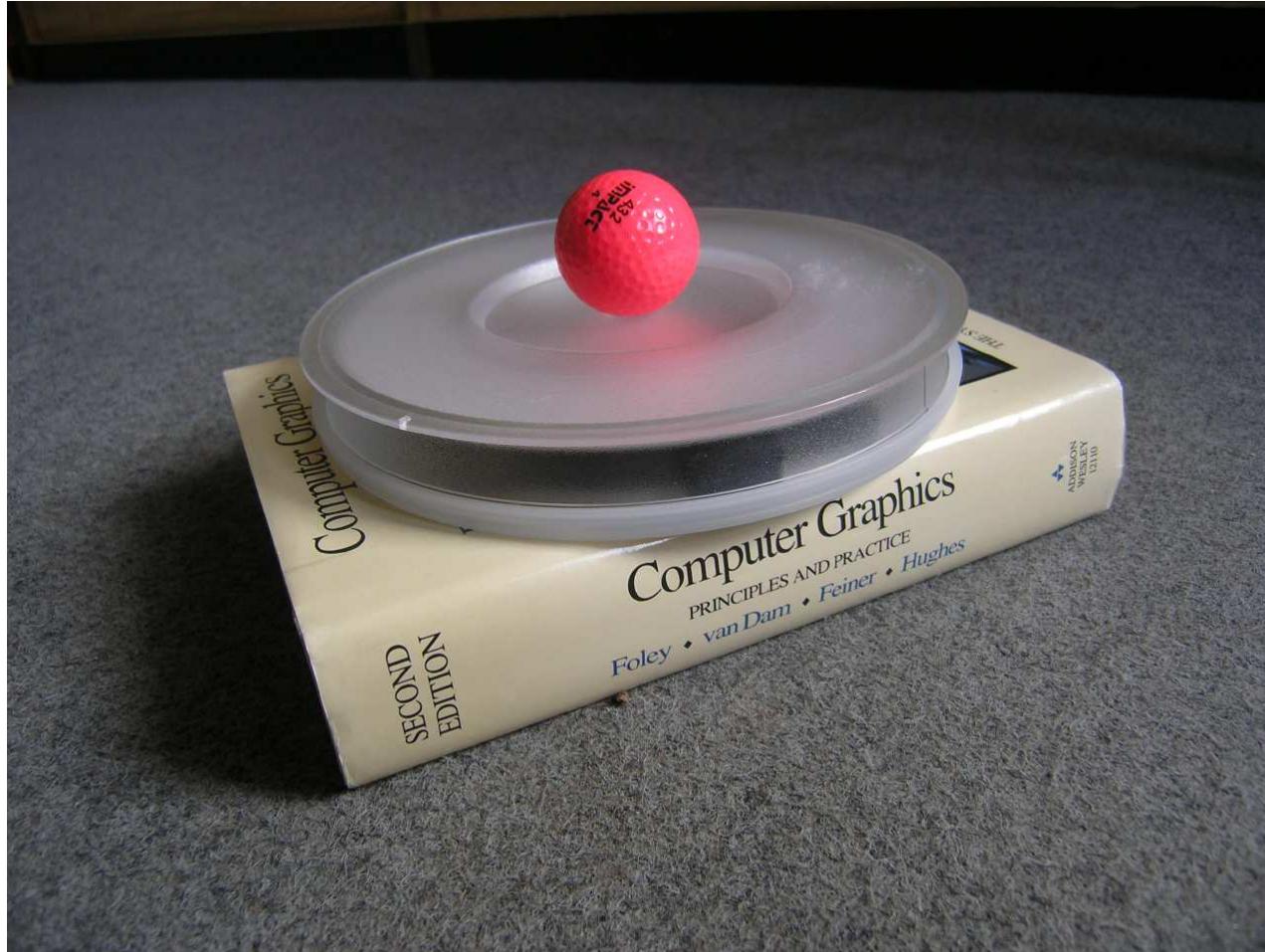
Gegenstand	Formbeschreibende Parameter
Zylinder	Radius = 9,8 cm / Höhe = 2,8 cm
Kugel	Radius = 2,0 cm
Quader	Länge = 24,3 cm / Breite = 17,0 cm / Höhe = 4,8 cm

Beschreibung der Anordnung im Raum:

- der Quader liegt zuunterst auf einer seiner beiden größten Flächen
- der Zylinder steht mittig darauf
- die Kugel liegt mittig auf dem Zylinder

Frage: Wie genau sieht diese Szene aus?

Ein Experiment (Forts.)



Photometrische Modelle

- sind Grundlage für eine möglichst fotorealistische Computergrafik
- dienen der möglichst naturgetreuen grafischen Darstellung modellierter Gegenstände bzw. Szenen
- beschreiben die Objekt- bzw. Szenenbeleuchtung
- Simulieren die einfachen Gesetze der geometrischen Optik bzw. des (Licht-) Energieaustausches
- beruhen auf Modellen der optischen Eigenschaften der darzustellenden Gegenstände (Farbe, Rauigkeit, Reflexions- bzw. Refraktionsverhalten, etc.)

Schattierung - lokale Beleuchtungsmodelle

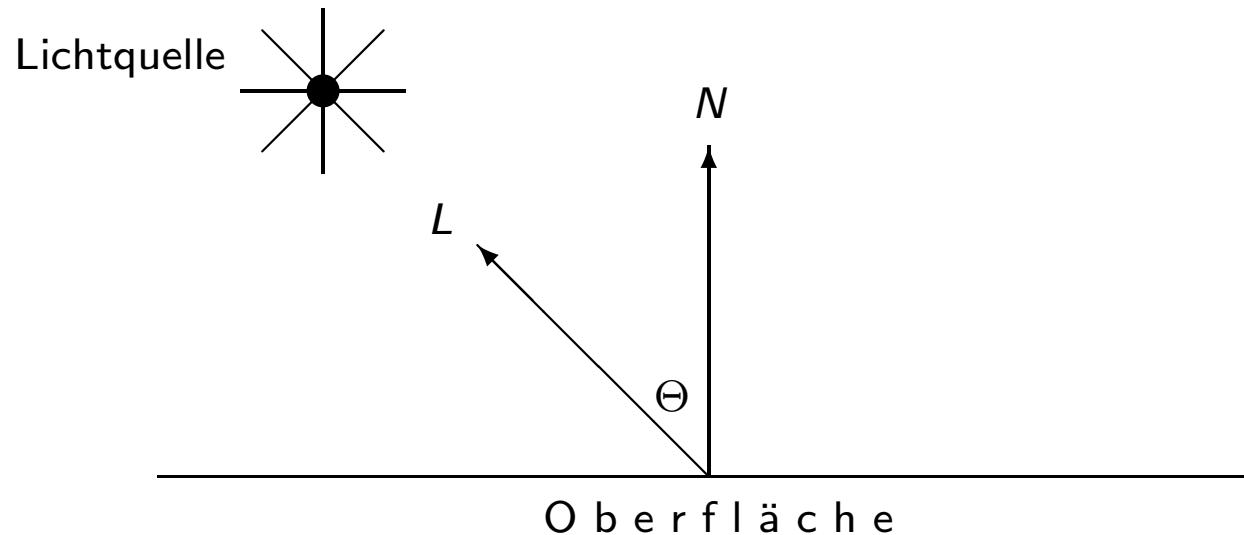
Schattierte Darstellungen berücksichtigen auf relativ einfache Weise die Wechselwirkungen zwischen der Anordnung von Lichtquellen sowie der Betrachtungsposition in Bezug auf die sichtbaren (vgl. HSR) Oberflächenteile (allgemeine Polygone, häufig Dreieckssegmente) darzustellender Gegenstände. Vereinfacht wird gleichmäßige Streuung des einfallenden Lichtes in alle Richtungen angenommen (Lambert-Reflektierer).

Man unterscheidet Verfahren für

- Konstante Schattierung (flat shading)
- Farbwertinterpolation (Gouraud shading)
- Normalenvektorinterpolation (Phong shading)

Einfache Farbwertberechnung

geometrische Zusammenhänge in einem Oberflächenpunkt:



L : Einheitsvektor in Richtung der Lichtquelle

N : Flächennormalen-Einheitsvektor im betrachteten Oberflächenpunkt

Θ : zwischen N und L eingeschlossener Winkel

Einfache Farbwertberechnung (Forts.)

$$I = I_{Lq} \cos(\theta) k_d + I_a k_a$$

I : Strahlungsintensität am betrachteten Oberflächenpunkt

I_{Lq} : Intensität der beleuchtenden Lichtquelle

θ : Winkel zwischen Lichtquelle und Flächennormale
am betrachteten Oberflächenpunkt

k_d : Koeffizient der diffusen Reflexion

I_a : Intensität des ambienten (Hintergrund-) Lichtes

k_a : Koeffizient der ambienten Reflexion

$$0 \leq k_a, k_d \leq 1$$

Konstante Schattierung (flat shading)

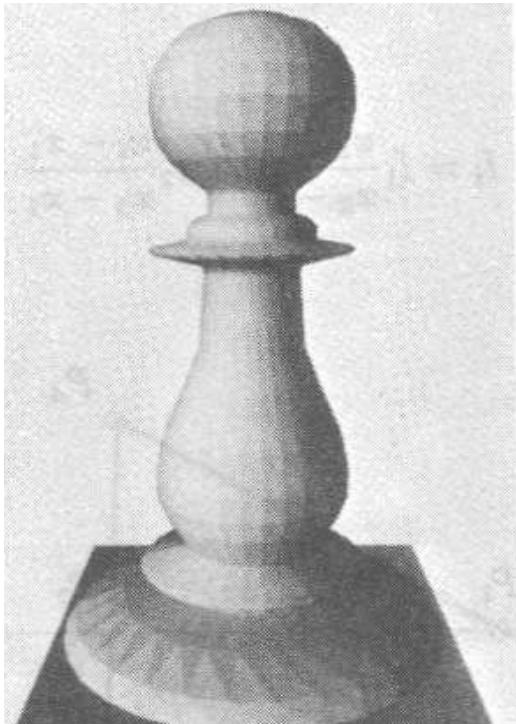
Grundprinzip:

- Berechnung des Farbwertes z.B. am Mittelpunkt des Flächensegmentes
- gleichmäßige Belegung des Flächensegmentes mit dieser Farbe

Nachteil

Gegenstände erscheinen “eckig”

Konstante Schattierung (Forts.)



Algorithmus:

WENN $Z_i > zBuffer[x, y]$, DANN :

$zBuffer[x, y] := Z_i$

$FrameBuffer[x, y] := \text{Farbe der Fläche}$

SONST tue nichts

Farbwertinterpolation (Gouraud shading)

Grundprinzip (von Henri Gouraud, 1971):

- Ermittlung (möglichst) der tatsächlichen Normalenvektoren in den Eckpunkten des Oberflächensegmentes
- Berechnung der Farbwerte in diesen Eckpunkten
- lineare Interpolation dieser Farbwerte über das gesamte Flächensegment

Vorteil:

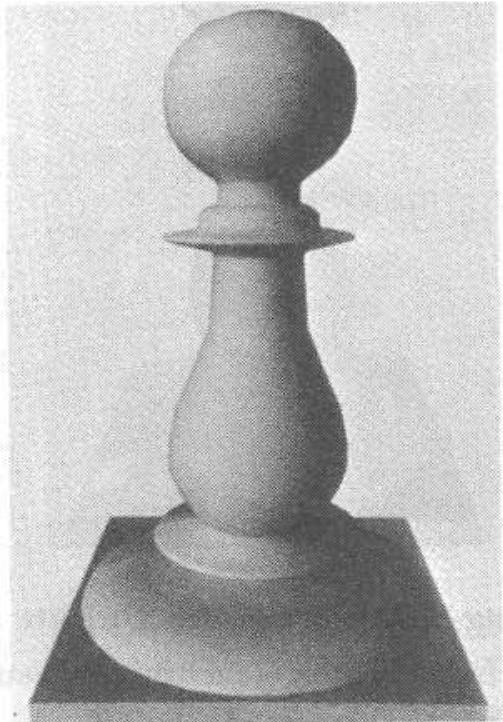
“weicher” Farbübergang zwischen den Flächensegmenten

Nachteil:

aufreten von “Mach-Band-Effekten”

(Farbverlauf zwar stetig, aber nicht stetig differenzierbar)

Farbwertinterpolation (Forts.)



Algorithmus:

WENN $Z_i > zBuffer[x, y]$, DANN :

$zBuffer[x, y] := Z_i$

$RotAnteilFrameBuffer[x, y] := R_i$

$GrünAnteilFrameBuffer[x, y] := G_i$

$BlauAnteilFrameBuffer[x, y] := B_i$

SONST tue nichts

Normalenvektorinterpolation (Phong shading)

Grundprinzip (von Bui Tong Phong, 1975):

- Ermittlung (möglichst) der tatsächlichen Normalenvektoren in den Eckpunkten des Oberflächensegmentes
- lineare Interpolation der Normalenvektoren über das gesamte Flächensegment
- Berechnung der Farbwerte in jedem Punkt des Oberflächensegmentes

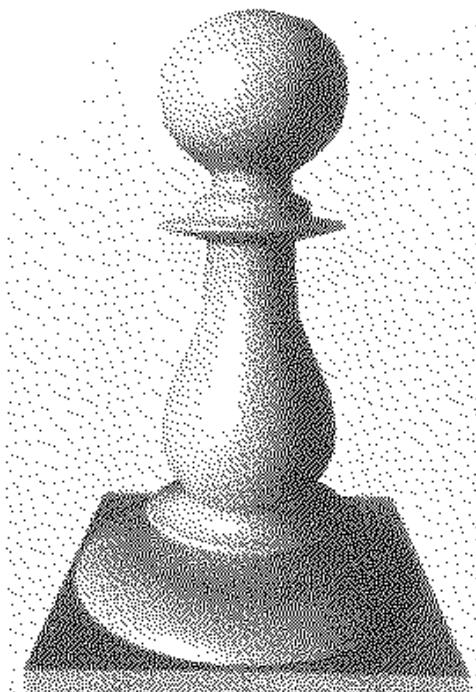
Vorteil:

deutlich bessere Bildqualität als beim Gouraud Shading

Nachteil:

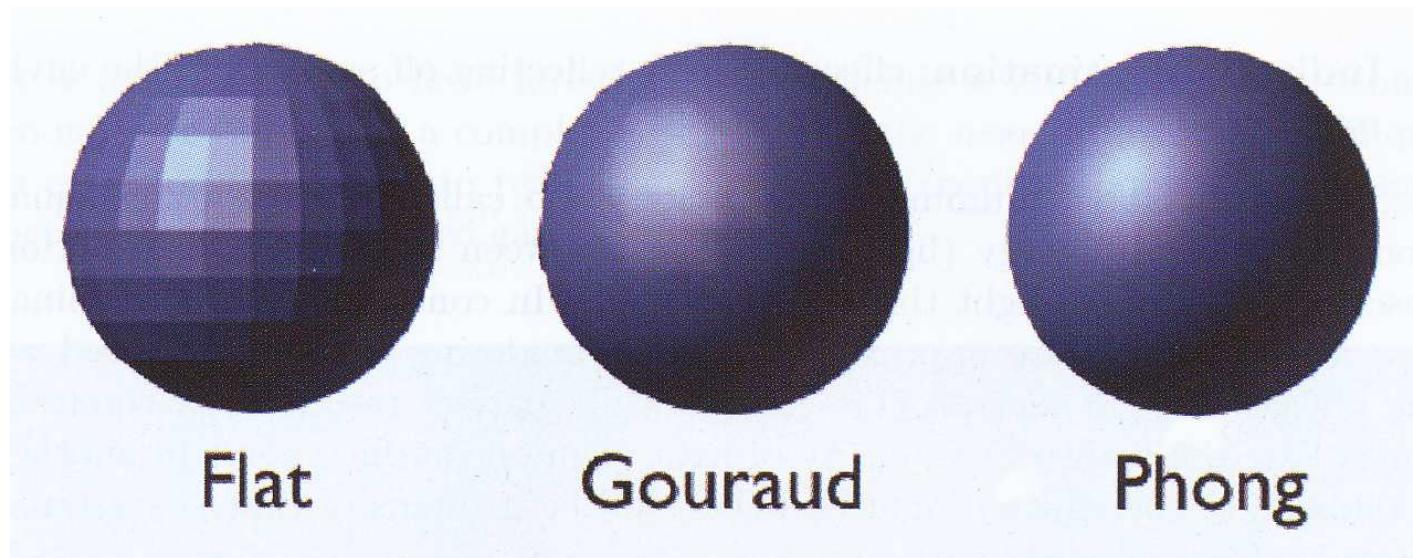
Algorithmus wesentlich aufwendiger als der Gouraud-Algorithmus

Normalenvektorinterpolation (Phong shading)



Auch Glanzlichter
werden realitätsnah
dargestellt.

Shader im Vergleich



Fotorealismus - globale Beleuchtungsmodelle

Fotorealistische Computergrafik dient der nahezu naturgetreuen grafischen Darstellung modellierter Gegenstände bzw. Szenen (wie gute Fotos von materiell existierenden Gegenständen). Sie

- benötigt globale Modelle der Objekt- bzw. Szenenbeleuchtung
- Simuliert die einfachen Gesetze der geometrischen Optik bzw. des (Licht-) Energieaustausches
- beruht auf Modellen der optischen Eigenschaften der darzustellenden Gegenstände (Farbe, Rauigkeit, Reflexions- bzw. Refraktionsverhalten, etc.)
- erfordert hohen Berechnungsaufwand

Fotorealismus (Forts.)

Um Fotorealismus zu erreichen, wurden unterschiedliche Verfahren entwickelt:

- Strahlverfolgung (ray tracing)
- Strahlungsberechnung (radiosity)
- Hybride Verfahren

Alle diese Verfahren basieren auf einer allgemeinen Darstellungsgleichung (rendering equation), die in vereinfachter Form schon von Phong benutzt wurde (s.o.).

Allgemeine Darstellungsgleichung (rendering equation)

$$I(x_1, x_2) = g(x_1, x_2)[\varepsilon(x_1, x_2) + \int_S \rho(x_1, x_2, x_3) I(x_2, x_3) dx_3]$$

darin sind

$I(x_1, x_2)$ Strahlungsintensität, die von Punkt x_2 zum Punkt x_1
gelangt

Allgemeine Darstellungsgleichung (Forts.)

- $g(x_1, x_2)$ *Geometrieterm*, gibt an, welcher Anteil der von x_2 in Richtung x_1 abgestrahlten Energie dort ankommt;
 $(g(x_1, x_2) = 0$, wenn sich die Punkte nicht sehen können,
sonst $g(x_1, x_2) = 1/r^2$
- $\varepsilon(x_1, x_2)$ *Emissionsterm*, gibt an, welche Intensität von x_2 in Richtung x_1 emittiert wird, falls x_2 Punkt einer Lichtquelle
- $\rho(x_1, x_2, x_3)$ *Streuungsterm*, gibt die von x_3 stammende Intensität an, die von x_2 in Richtung x_1 gestreut wird
- S Vereinigung aller Oberflächen in der Szene, $S = \bigcup S_i$

Strahlverfolgung (Ray Tracing)



Von spiegelnden Flächen begeisterter Künstler: Anish Kapoor (Indien),
„Tall Tree and the Eye“ (rostfreier Stahl) am Museo Guggenheim, Bilbao

Strahlverfolgung (Forts.)



Anish Kapoor (Indien), „Tall Tree and the Eye“ (Detail)
am Museo Guggenheim, Bilbao

Strahlverfolgung (Forts.)



Von spiegelnden Flächen begeisterter Künstler: Jeff Koons (USA),
„Tulips“ (rostfreier Stahl mit transparenter Farbbebeschichtung)
am Museo Guggenheim, Bilbao

Strahlverfolgung (Forts.)

- Die Grundidee wurde schon zur Zeit der Renaissance entwickelt.
- Leonardo da Vinci (1452-1519) wird die Entdeckung mathematisch berechenbarer Perspektive zugeschrieben.
- Albrecht Dürer (1471-1528) hat sich in seinem Traktat „In der Underweysung der messung mit dem zirckel un richtscheit in Linien ebenen und gantzen corporen ...“ ebenfalls mit der perspektivischen Darstellung befasst und das Ray-Tracing-Prinzip in seinen Grundzügen bereits erläutert (vgl. nachf. Holzschnitt).

Strahlverfolgung (Forts.)

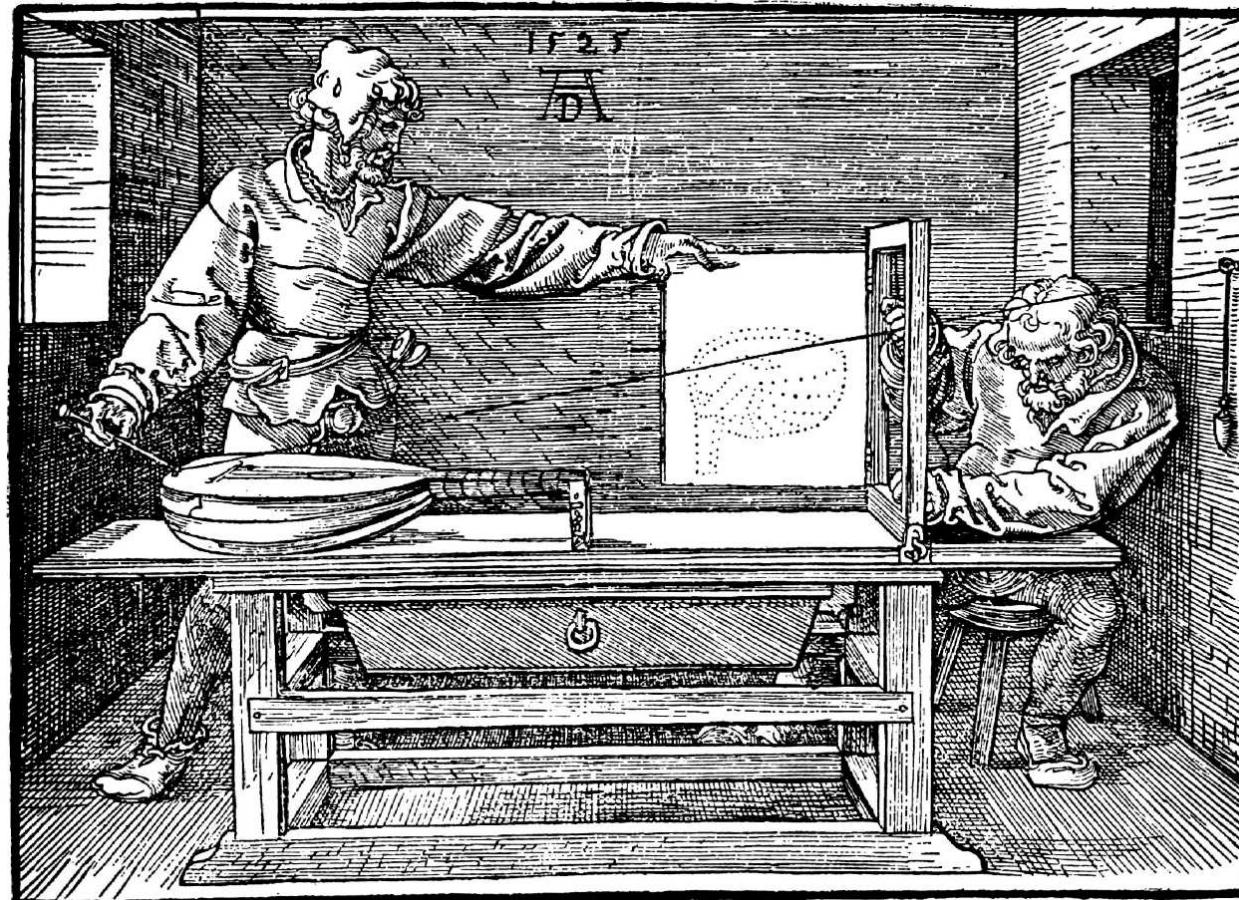


Illustration aus Dürers „... Underweysung der messung ...“

Strahlverfolgung (Forts.)



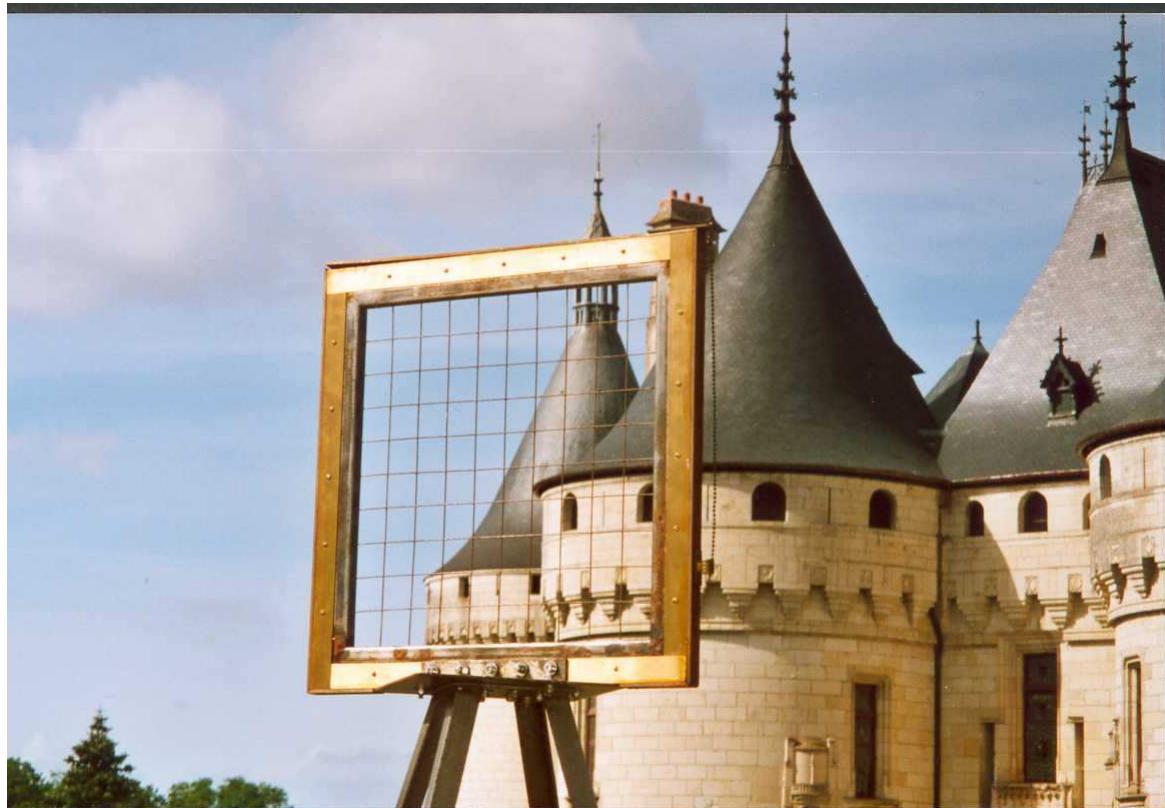
Renaissance-Instrumente für die perspektivische Darstellung
im Museo degli Uffizi, Florenz

Strahlverfolgung (Forts.)



„Gerasterte Bildebene“ am Schloss Chaumont-sur-Loire

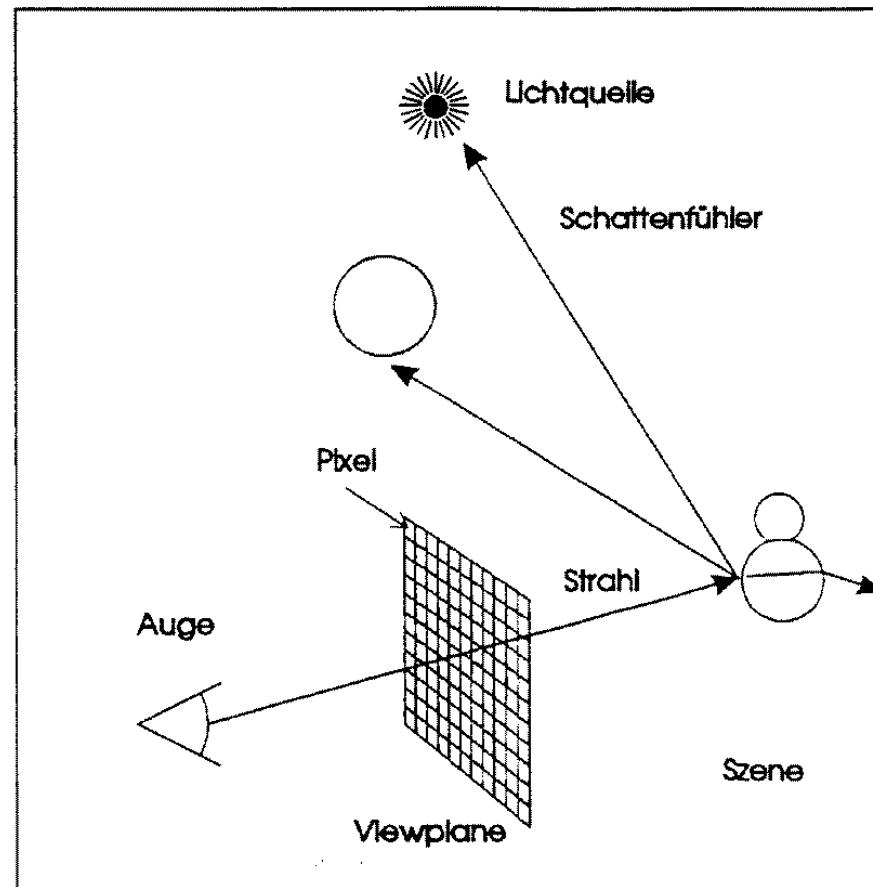
Strahlverfolgung (Forts.)



„Gerasterte Bildebene“ am Schloss Chaumont-sur-Loire

Strahlverfolgung (Forts.)

Grundprinzip
in der
Computergrafik



Strahlverfolgung (Forts.)

Grundprinzip:

Die Strahlen werden nicht von den Lichtquellen über die reflektierenden Oberflächen der Szene zum betrachtenden Auge verfolgt, sondern

- vom Augpunkt (Kamera) wird durch jeden Bildpunkt (Pixel) ein Strahl in die Szene verlängert
(d.h. ca. 1,5 Mio. Strahlen bei 1280×1024 Auflösung)
- es werden alle Schnittpunkte dieses Strahls mit Oberflächenelementen der Objekte in der Szene ermittelt
- die Schnittpunkte werden sortiert
- für den Oberflächenpunkt, der dem Augpunkt am nächsten liegt, wird zur Ermittlung der Lichtintensität der Standard - Reflexionsoperator (s.u.) ausgewertet

- ist die Oberfläche im betrachteten Punkt spiegelnd, wird nach den Gesetzen der geometrischen Optik ein reflektierter Strahl von diesem Punkt in die Szene verlängert und wie der erste behandelt (Schnittpunktermittlung, etc. → *Rekursion*)
- ist das Material im betrachteten Punkt durchsichtig, wird nach den Gesetzen der geometrischen Optik ein gebrochener Strahl von diesem Punkt durch das Objekt verlängert und wie der erste behandelt (Schnittpunktermittlung, etc. → *Rekursion*)
- die bei einer Rekursion ermittelten individuellen Lichtintensitäten werden akkumuliert

hoher Berechnungsaufwand allein durch die Schnittpunktermittlungen!!!

Strahlverfolgung (Forts.)

Standard-Reflexionsoperator (Turner Whitted, 1980):

$$I = I_a + k_d \sum_{j=1}^{l_s} I_{L_j}(\mathbf{NL}_j) + k_s S + k_t T$$

darin sind

I : Strahlungsintensität am betrachteten Oberflächenpunkt

I_a : reflektiertes ambientes Licht

k_d : Koeffizient der diffusen Reflexion

I_{L_j} : Strahlungsintensität der Lichtquelle j

\mathbf{N} : Normaleneinheitsvektor im betrachteten Oberflächenpunkt

\mathbf{L}_j : Vektor in Richtung der j-ten Lichtquelle

k_s : Koeffizient der spiegelnden Reflexion

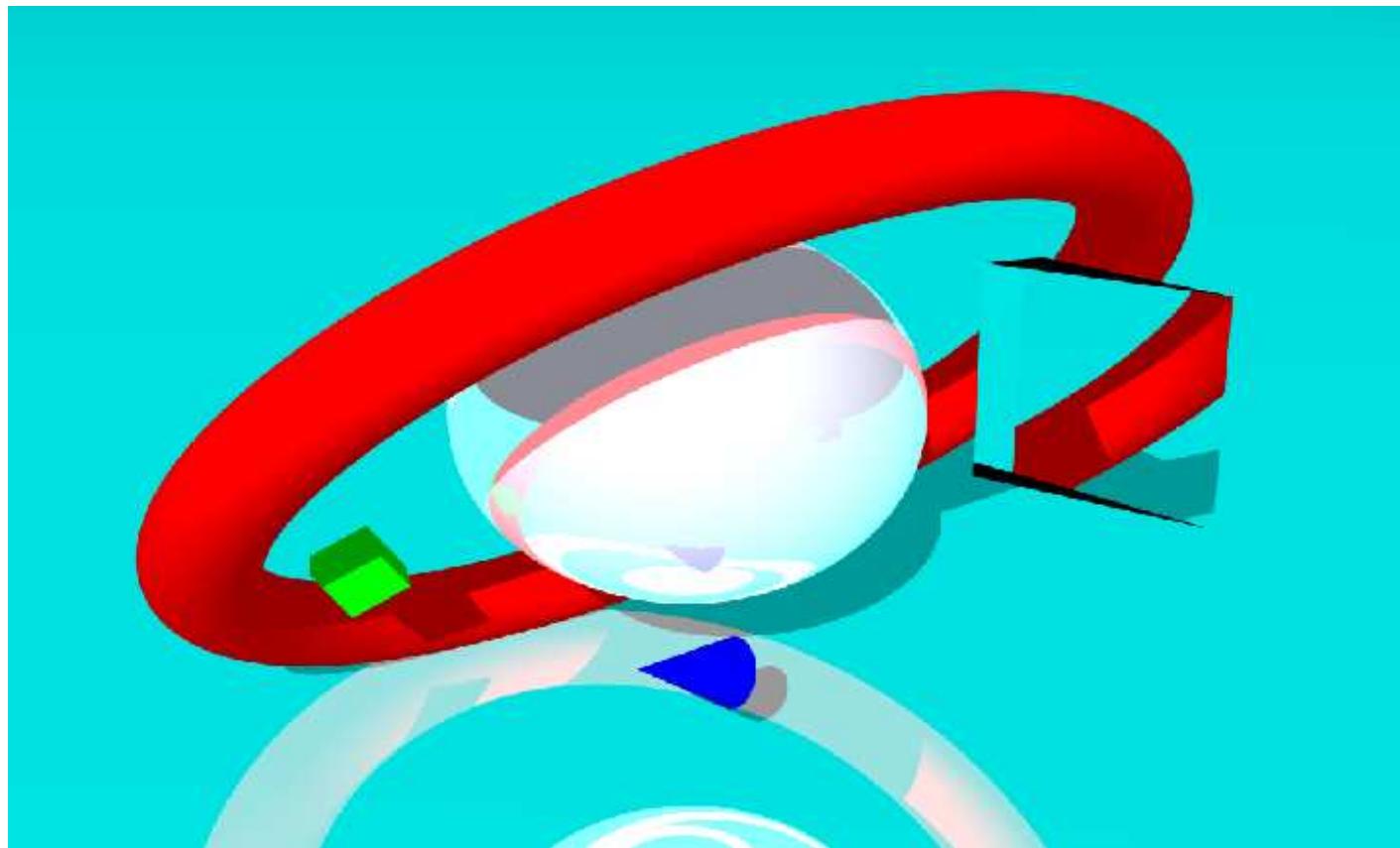
S : Intensität des spiegelnd reflektierten Lichtes

k_t : Koeffizient der spekulären Transmission

T : Intensität des spekulär transmittierten Lichtes

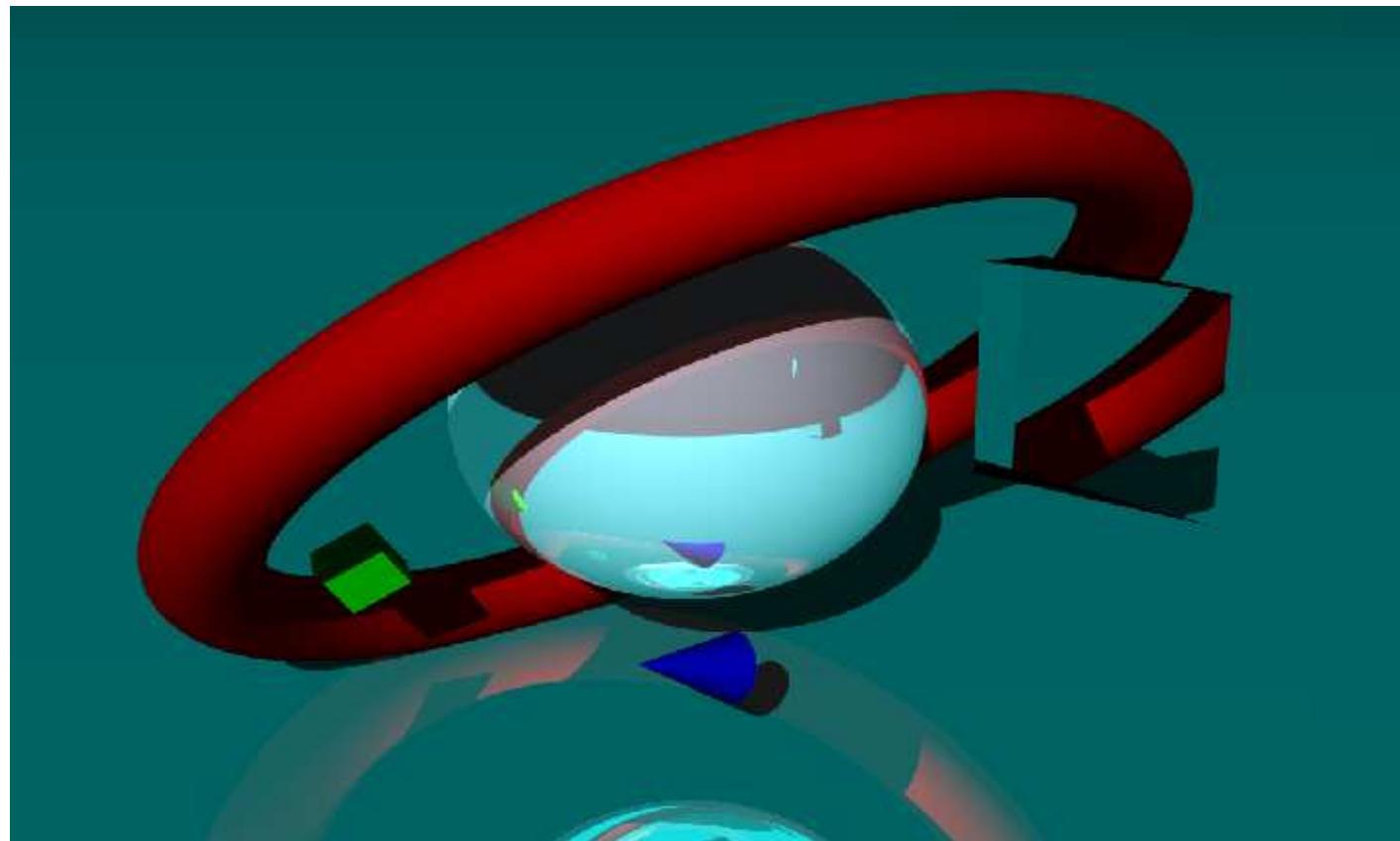
l_s : Anzahl der Lichtquellen

Strahlverfolgung (Forts.)



besondere Eigenschaften: klare Schatten, Spiegelung und Brechung des Lichts

Strahlverfolgung (Forts.)



dieselbe Szene, jetzt aber ohne ambientes Licht

Strahlungsberechnung (Radiosity)

Das Verfahren

- stammt aus der Wärmetechnik
- basiert auf dem Grundsatz der Energieerhaltung in einem abgeschlossenen Raum
- wurde 1984 in die Computergrafik übernommen und
- dient hier der Modellierung des Energieaustausches zwischen den Oberflächensegmenten der Gegenstände in einer geschlossenen Szene
- lässt jedes Flächensegment als Lichtquelle zu

Strahlungsberechnung (Forts.)

Die von dem Oberflächensegment j abgestrahlte Energie wird bestimmt m.H. der Energiegleichung:

$$B_j = E_j + \rho_j H_j$$

darin sind

B_j : die vom betrachteten Oberflächenpunkt ausgehende Energie

E_j : die vom betrachteten Oberflächenpunkt emittierte Energie

ρ_j : der diffuse Reflexionskoeffizient

H_j : die am betrachteten Oberflächenpunkt einfallende Energie

(die von allen anderen Oberflächensegmenten abgestrahlt wird)

Strahlungsberechnung (Forts.)

Die am betrachteten Oberflächenpunkt einfallende Energie lässt sich beschreiben als:

$$H_j = \sum_{i=0}^N F_{ij} B_i$$

darin sind

B_i : die vom Oberflächensegment A_i einfallende Energie

F_{ij} : Formfaktoren

Strahlungsberechnung (Forts.)

Die Formfaktoren F_{ij}

- geben an, welcher Anteil der von einem Oberflächensegment A_i abgestrahlten Energie beim Oberflächensegment A_j ankommt
- hängen nur von der Geometrie der Szene ab

Sie haben die Eigenschaften:

$$F_{ii} = 0, \quad \forall i$$

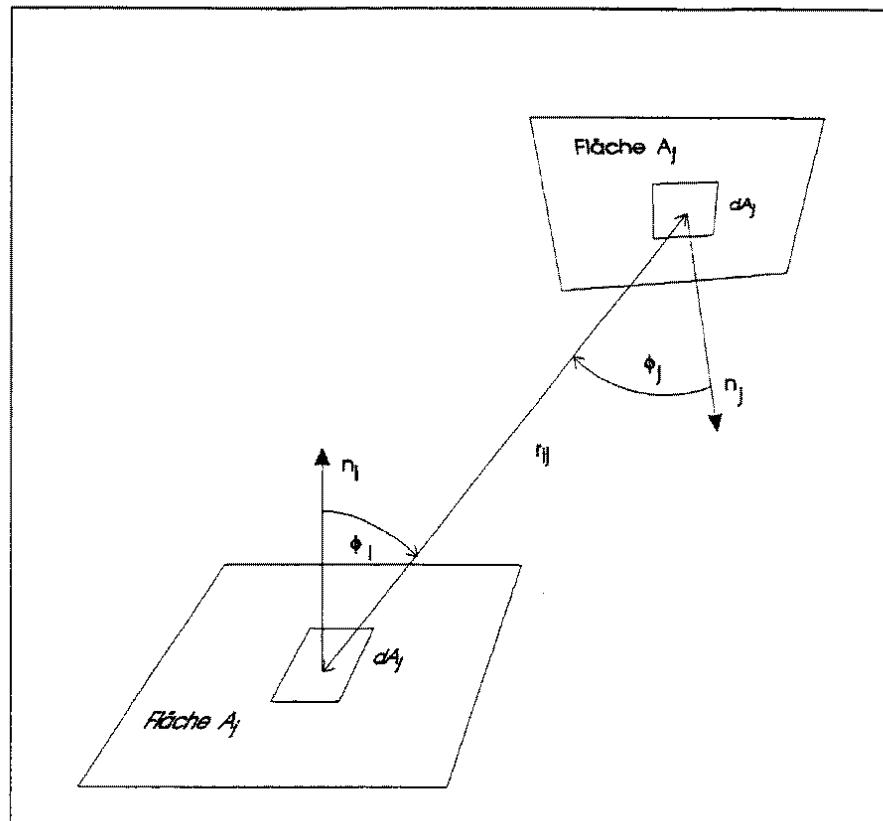
$$\sum_{i=1}^N F_{ij} = 1, \quad \forall j$$

$$A_i F_{ij} = A_j F_{ji}, \quad \forall i, j$$

wobei A_i die Fläche eines Oberflächensegmentes i ist

Strahlungsberechnung (Forts.)

Formfaktorgeometrie

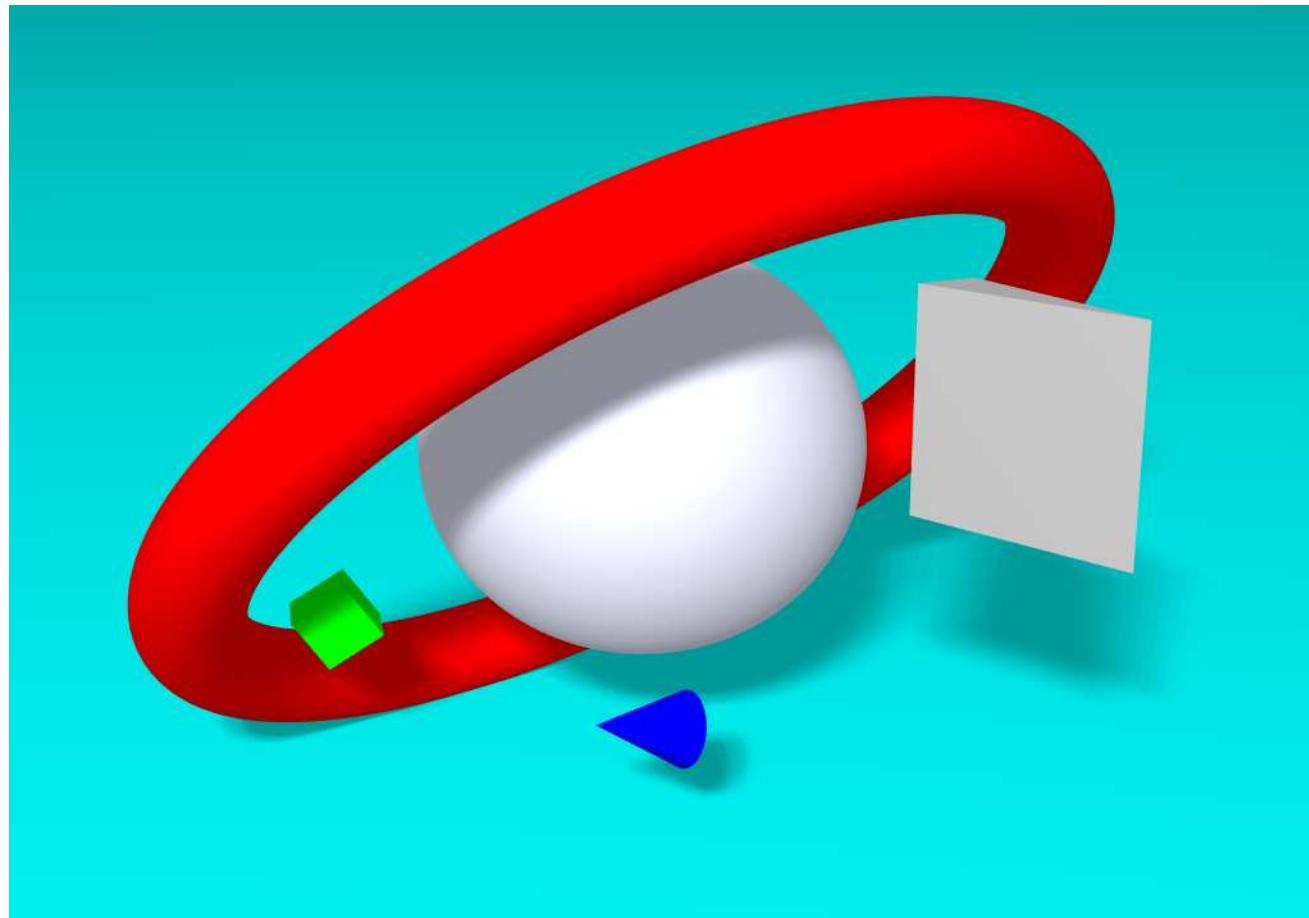


Strahlungsberechnung (Forts.)

Nach der Ermittlung der Formfaktoren können die abgestrahlten Energien aller Oberflächensegmente aus folgendem Gleichungssystem ermittelt werden:

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \dots & -\rho_1 F_{1N} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & \dots & -\rho_2 F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_N F_{N1} & -\rho_N F_{N2} & \dots & 1 - \rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

Strahlungsberechnung (Forts.)



Merkmale: realistische diffuse Lichtausbreitung und flächige Lichtquellen

Vergleich der Verfahren

Strahlverfolgung (ray tracing) und Strahlungsberechnung (radiosity) haben jeweils ihre individuellen Stärken und Schwächen:

Verfahren	Ray Tracing	Radiosity
Vorteil	sehr gutes Modell für Spiegelung und Brechung	sehr gutes Modell für diffuse Lichtausbreitung sowie flächige Lichtquellen
Nachteil	diffuse Lichtausbreitung nur grob angenähert, Lichtquellen nur punktförmig	Spiegelung und Brechung nicht berücksichtigt

Hybride Verfahren

Eine Kombination aus beiden Verfahren, bei dem die diffuse Reflexion aus der Strahlungsberechnung in den Reflexions-Operator für die Strahlverfolgung eingesetzt wird; ein ambienter Term und ein Schattenfühler erübrigen sich:

$$I = B + k_s S + k_t T$$

I : abgestrahlte Intensität

B : Radiosität am Oberflächenpunkt (aus der Strahlungsberechnung)

k_s : Koeffizient der spiegelnden Reflexion

S : Intensität des spiegelnd reflektierten Lichtes

k_t : Koeffizient der spekulären Transmission

T : Intensität des spekulär transmittierten Lichtes

Nicht-Fotorealistische Computergrafik

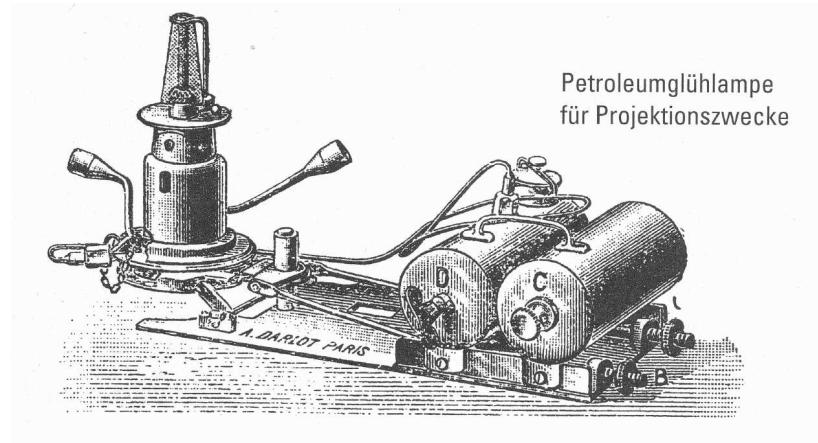


Nicht-Fotorealistische Computergrafik (Forts.)

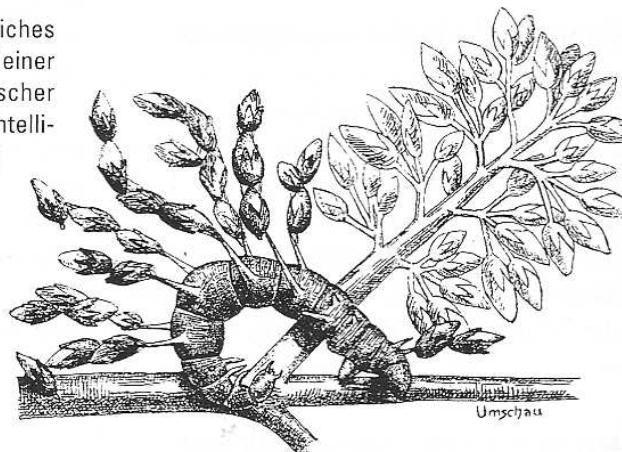
Non-photorealistic rendering

- fotorealistische Detailtreue ist nicht immer erforderlich bzw. wünschenswert (z.B. Zeichentrickfilm, Cartoon, techn. Illustration)
- traditionelle Abstraktionsformen bei der grafischen Darstellung sind für manche Anwendungen weitaus geeigneter, die Aufmerksamkeit betrachtender Personen auf das Wesentliche zu lenken
- nicht-fotorealistische Computergrafik versucht (u.a. auf der Basis geometrischer Eigenschaften bzw. der Darstellungsgleichung), Techniken zu simulieren, die in der Kunst seit dem 15. Jahrhundert angewandt und bis ins 19. Jahrhundert stetig verfeinert wurden, und die bis weit in das Zeitalter der Fotografie hinein auch der Illustration in technischen und naturwissenschaftlichen Gebieten dienten

Nicht-Fotorealistische Computergrafik (Forts.)



► Ungewöhnliches Versteckspiel einer Raupe. Forscher vermuteten Intelligenz dahinter.



Nicht-Fotorealistische Computergrafik (Forts.)



Dampfwaschmaschine im Gebrauch

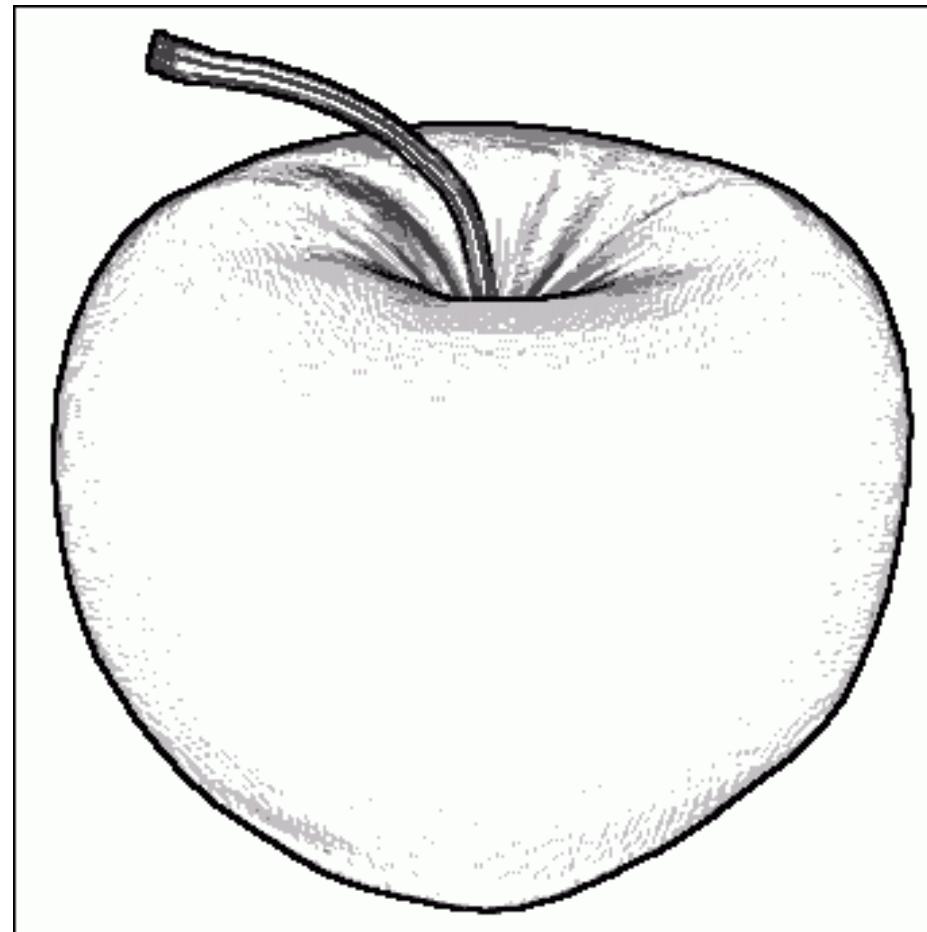


Ansicht von Hamburg um 1818 – historischer Stich von Anton Radl, Detail



Nicht-Fotorealistische Computergrafik (Forts.)

Betonung
von Konturen,
Schattierung
durch Punkt muster



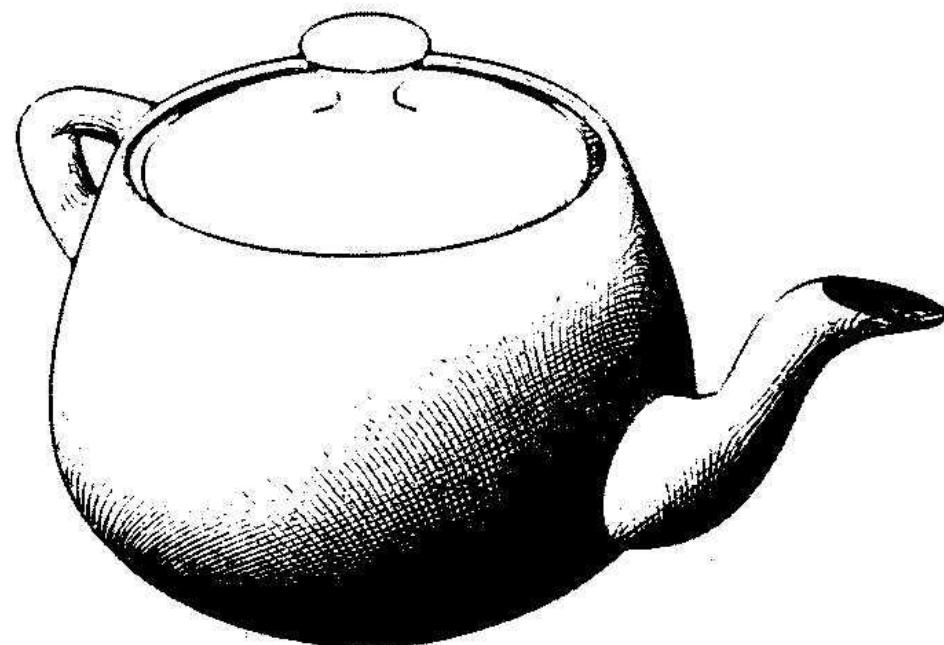
Nicht-Fotorealistische Computergrafik (Forts.)

Betonung
von Konturen,
und Schatten

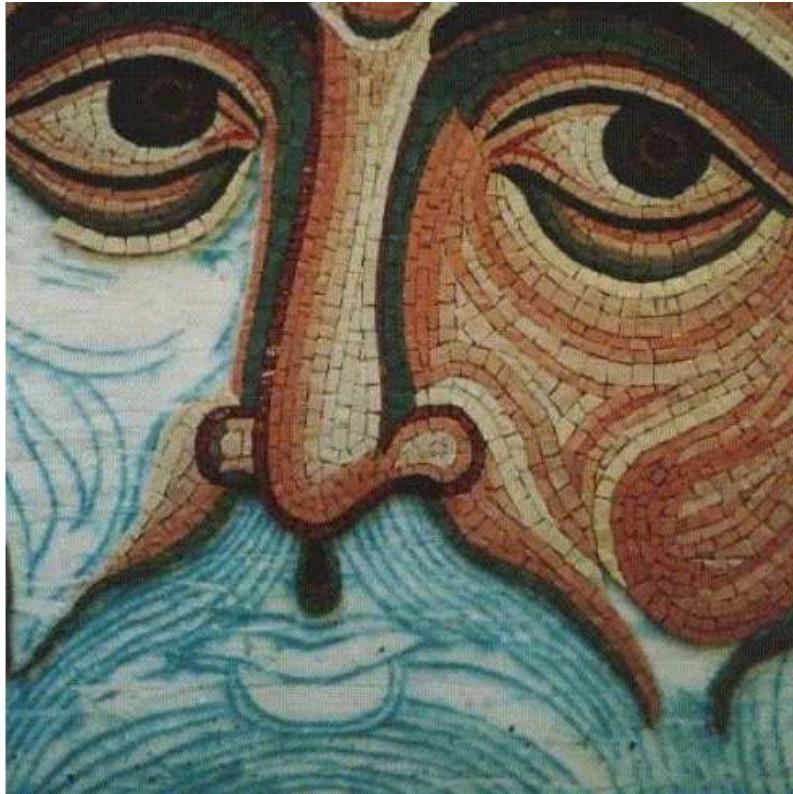


Nicht-Fotorealistische Computergrafik (Forts.)

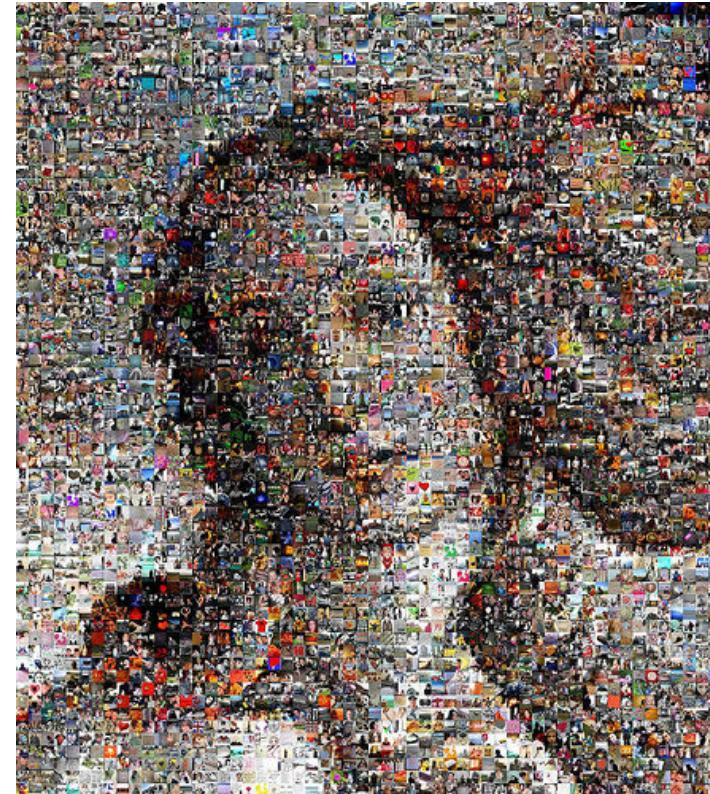
Betonung
von Konturen,
Schattierung
durch kreuzweise
Schraffur



Nicht-Fotorealistische Computergrafik (Forts.)



traditionelles Mosaik: betonte
Konturlinien, Flächenfüllung



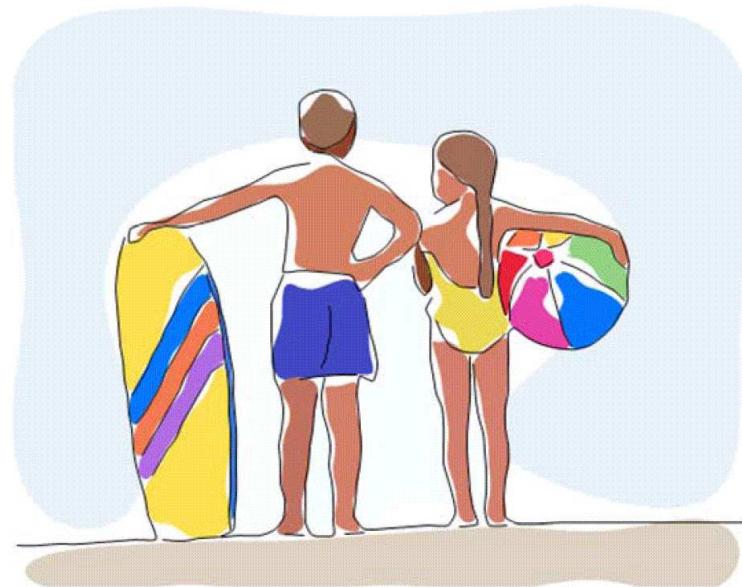
Fotomosaik: zusammengesetzt aus
„Kacheln“ geeignetfarbiger Fotos (*)

(*) nach Sandro Botticelli: *La nascita di Venere* (1485) (aus: Flickr Group Mosaics, www.flickr.com)

Nicht-Fotorealistische Computergrafik (Forts.)



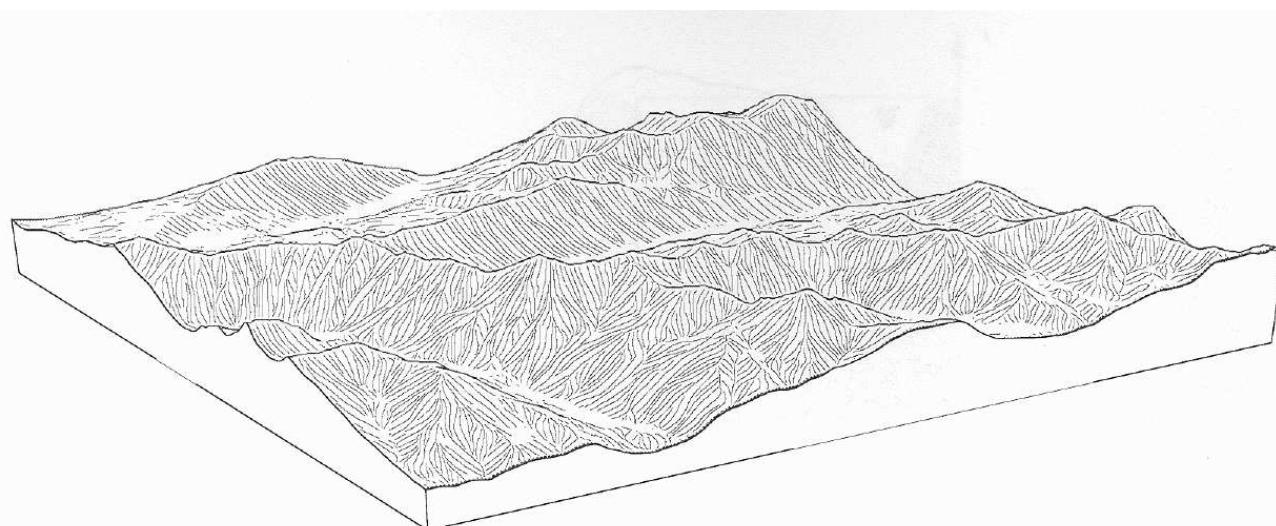
Vorlage: Fotografie



daraus generierte Farbskizze
in Aquarelltechnik (*)

(*) Fang Wen et al: Color Sketch Generation, vorgestellt bei der NPAR 2006, Annecy, Frankreich

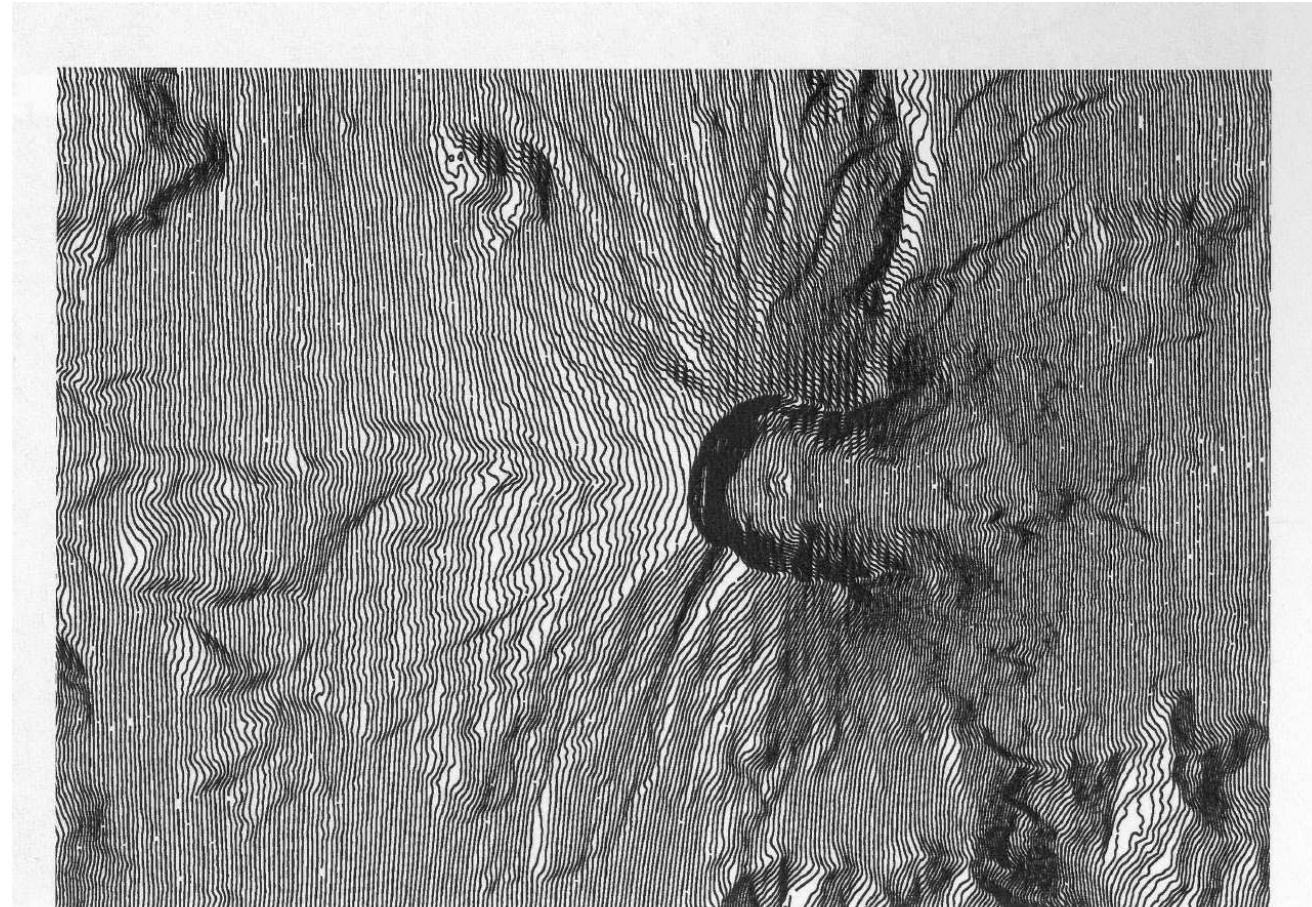
Nicht-Fotorealistische Computergrafik (Forts.)



Blockmodell eines Geländes; Strukturierung durch Falllinien

Nicht-Fotorealistische Computergrafik (Forts.)

Gelände in
Draufsicht,
Strukturie-
rung durch
Konturlinien



Prüfungsunterlagen zum Modul

Interactive Visual Computing (IVC)

Vorlesungsteil

Real Time Interactive Media (RTIM)

Werner Hansmann

Wintersemester 2010/2011

Übersicht

1. Einführung
(Anwendungsgebiete, Aktivitäten, Literatur)
2. Computergestützte Animation
3. Virtuelle Darsteller
 - äußere Erscheinung (Haut)
 - Interaktion mit virtuellen Objekten
 - Gesichtsanimation
 - autonome virtuelle Darsteller
4. Hardware für die Interaktion mit virtuellen Welten

1. Einführung

- Virtuelle Realität (VR) = Wissenschaft von der Illusion
- Geschichtliches
- Grundlagen aus Computergrafik und Geometriemodellierung
- VR-Grundkonzepte: Immersion – Präsenz
- Realität-Virtualität-Kontinuum
- Anwendungen von VR-Systemen
- Konferenzen, Symposia, Workshops
- Literatur

Virtuelle Realität (VR) = Wissenschaft von der Illusion

RTIM: Voraussetzung für VR-Systeme ist die Möglichkeit der Mensch-Computer-Interaktion in Echtzeit (über möglichst viele Sinneskanäle).

VR soll

- die Illusion einer Umgebung wecken, die als glaubwürdig wahrgenommen werden kann,
- effiziente und bequeme Interaktions- und Navigationsmöglichkeiten bieten

aber: aktuelle Technologie gestattet die akzeptable Reproduktion realer Objekte oder Umgebungen für Ausbildung und Unterhaltung

Geschichtliches

- 1962 Norton Heilings „Sensorama“: Multisensorisches Fahrzeug mit vorgefertigten Aufzeichnungen – sehr immersiv aber kaum interaktiv
- 1965 Ivan Sutherlands Veröffentlichung „The ultimate display“ (Vision von einem Fenster in eine virtuelle Welt)
- 1968 Ivan Sutherlands erstes Head-Mounted Display (einfache Liniengrafik)
- 1978 „Aspen Movie Map“ von Lippmann und Fisher (MIT) – mit vier Kameras aufgenommene Fotografien aller Straßen der Stadt dienen als optische Grundlage für einen „Fahrsimulator“
 - seit Mitte der 1980er Entwicklung von VR-Systemen (z.B. NASA pilot training system) mit typischer Hardware
 - Anfang der 1990er erste CAVE (Cave Automatic Virtual Environment)

Grundlagen aus Computergrafik und Geometriemodellierung

- Computergrafik
 - Koordinatensysteme, Transformationen, Projektionen
 - Physiologie des Sehens, Licht, Farbe, Raumwahrnehmung
 - Kameramodelle
 - lokale und globale Beleuchtungsmodelle – Fotorealismus
 - Texturen
 - etc.
- geometrisches Modellieren
 - Formbeschreibungsverfahren (CSG, B-Rep, etc.)
 - kontinuierlich gekrümmte und triangulierte Oberflächen
 - Formbeschreibung in Detailstufen (Level of Detail)
- Rendering Pipeline

Level of Detail: Formbeschreibung in Detailstufen

dient der Erhöhung der Geschwindigkeit in VR-Anwendungen:

- hohe Detailgenauigkeit im Nahbereich
- niedrige Detailstufe im Fernbereich

durch

- Anpassung der Komplexität (Anzahl der Polygone) der 3D-Objekte
- Auflösung der Texturen

in Abhängigkeit von der Entfernung und Position der Objekte zum Beobachter

Problem bei wenigen Detailstufen: plötzliches Erscheinen und Verschwinden von Details (Popping Effect);

Vermeidung: kontinuierliche LOD, Blending (Überblenden) und Morphing

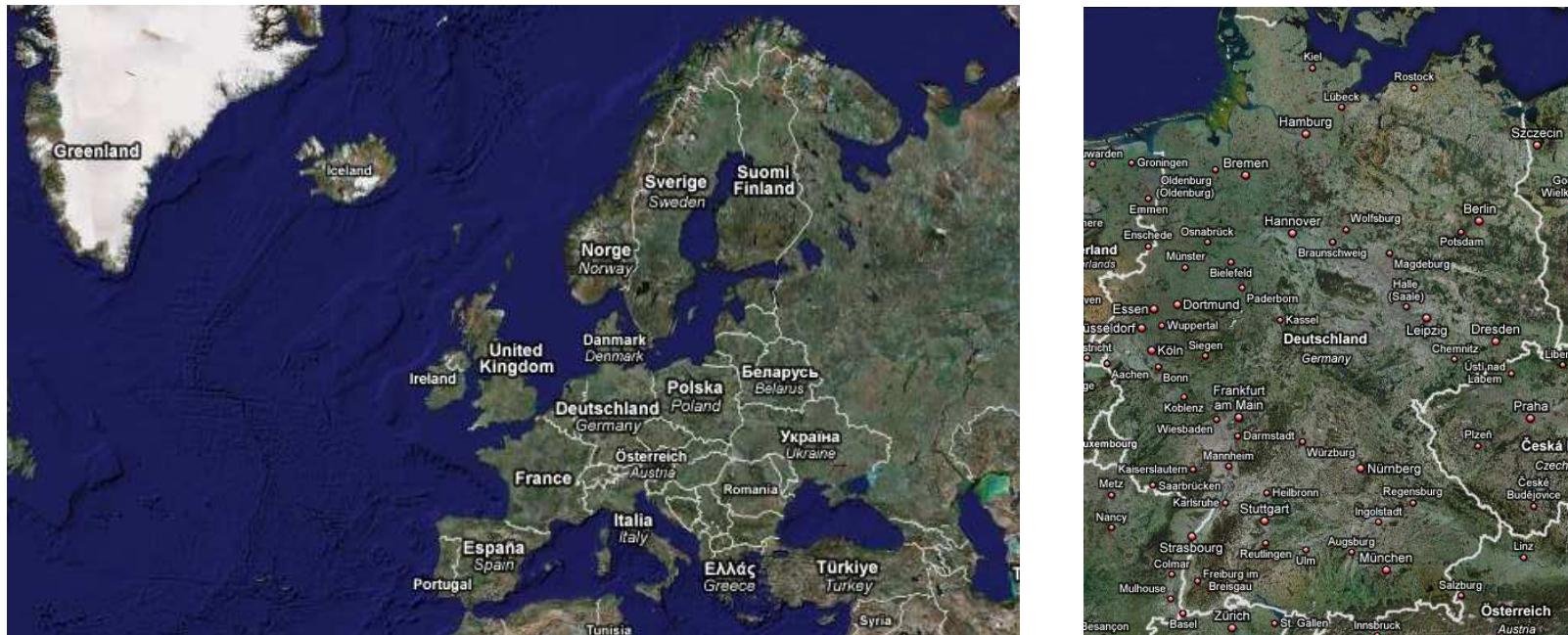
Level of Detail: Formbeschreibung in Detailstufen

in dreidimensionalen Landschafts- und Stadtmodellen:

- LOD 0: Regionalmodell – 2,5D Geländemodell mit Luftbildtextur
- LOD 1: Klötzchenmodell – Gebäudeblock (Grundfläche hochgezogen)
- LOD 2: 3D-Modell der Außenhülle und Dachstrukturen, einfache Texturen
- LOD 3: Architekturmodell – 3D-Modell der Außenhülle mit Textur
- LOD 4: Innenraummodell – 3D-Modell des Gebäudes mit Etagen, Innenräumen, etc. und Texturen

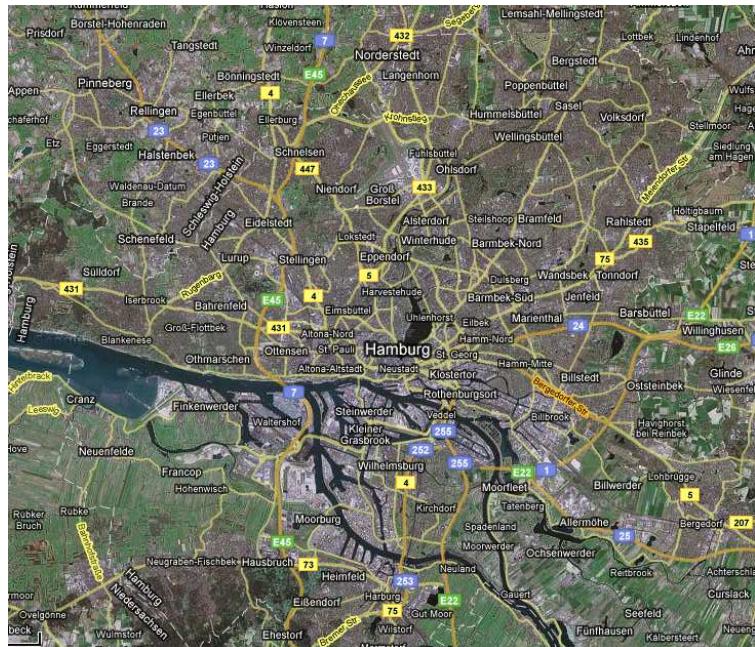
(Detailstufen für die CityGML-Spezifikation (City Geography Markup Language), die derzeit vielen deutschen Stadtmodellen zu Grunde liegen)

Level of Detail: Formbeschreibung in Detailstufen



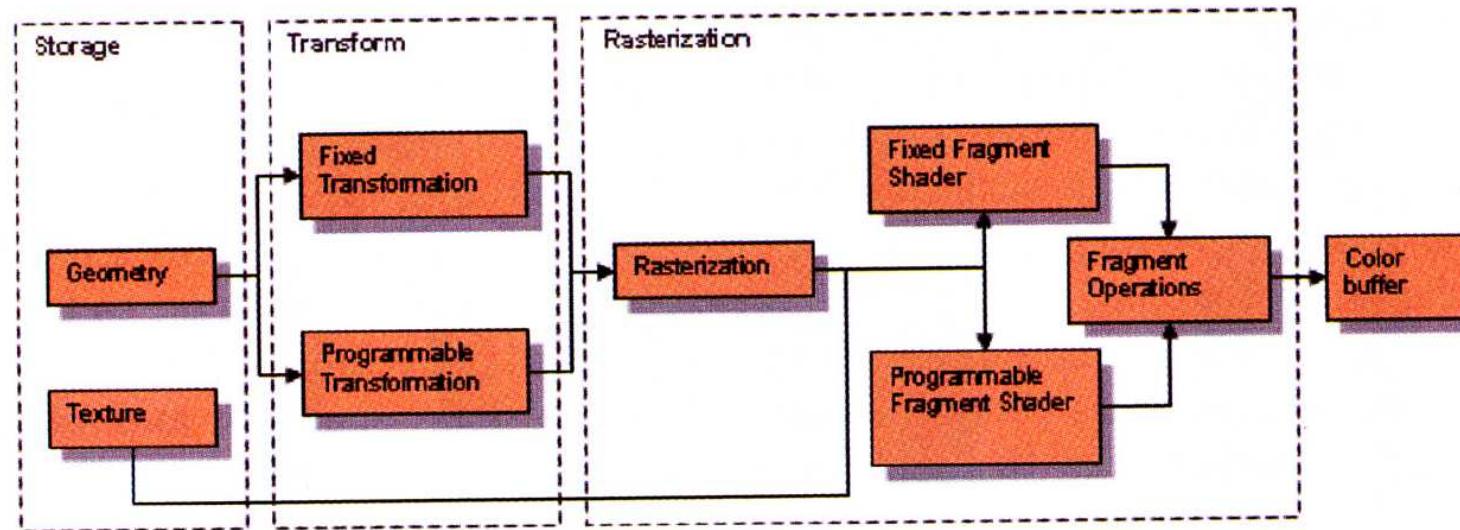
das Informatikum, dargestellt in unterschiedlichen Detailstufen
(aus Google Maps)

Level of Detail: Formbeschreibung in Detailstufen



das Informatikum, dargestellt in unterschiedlichen Detailstufen
(aus Google Maps)

Rendering Pipeline



Sie organisiert (z.B. in OpenGL) die schrittweise Verarbeitung vom geometrischen Modell (im einfachsten Fall repräsentiert durch Punkte, Kanten und Polygone) bis hin zur Farbdarstellung im Bildpuffer.
Die Funktionalität (z.B. Gouraud-Shading) ist i.d.R. auf Grafikkarten „fest verdrahtet“.

VR-Grundkonzepte

- **Immersion**

- voll-immersive Systeme verwenden Head-Mounted-Displays
- halb-immersive Systeme verwenden Großbildprojektionen
(z.B. CAVE: Cave Automatic Virtual Environment)
- nicht-immersive Systeme verwenden normale Bildschirme
(„Aquarium-VR“, z.B. für Videospiele)

- **Präsenz**

wird erzeugt durch multimodale Anregung (Bilder, Klänge, haptisches Feedback, etc.),

kann emotionales Engagement und entsprechende Reaktionen bewirken;

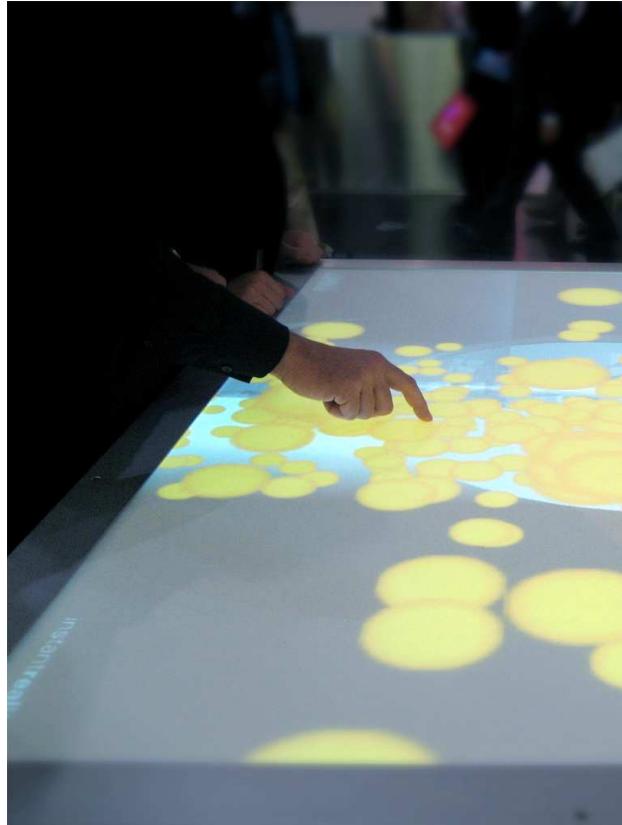
Benutzer sind sich bewusst, in einer virtuellen Umgebung zu sein,
Benutzer verhalten sich, als seien sie in einer realen Umgebung,

Voll-immersive VR



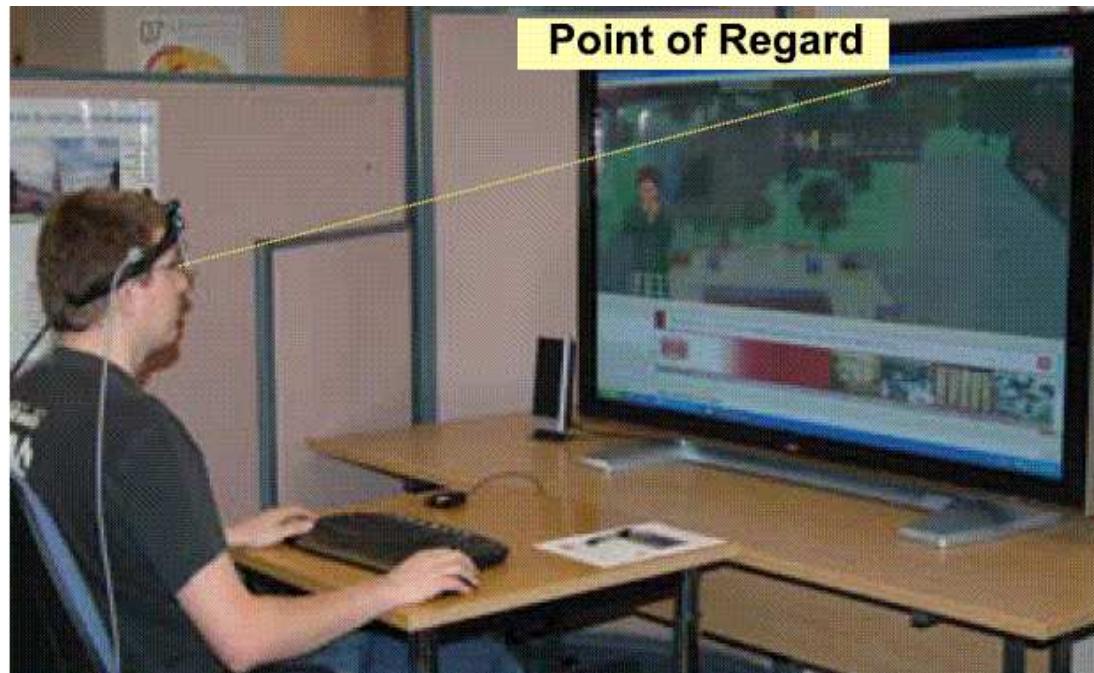
Stereo-Display (head mounted display)
und Datenhandschuh (data glove)

CAVE: halb-immersive VR



Touchtable: Auswahl von Elementen durch einfaches Berühren
(Quelle: Fraunhofer Institut für Graphische Datenverarbeitung)

Nicht-immersive VR (Aquarium-VR)



Verfolgung der Kopfbewegung (head tracking) zur Identifikation der
Blickrichtung

(aus: Rodriguez-Echavarri, et al: Developing Effective Interfaces for Cultural Heritage 3D Immersive Environments, in: VAST'08)

Realität-Virtualität-Kontinuum

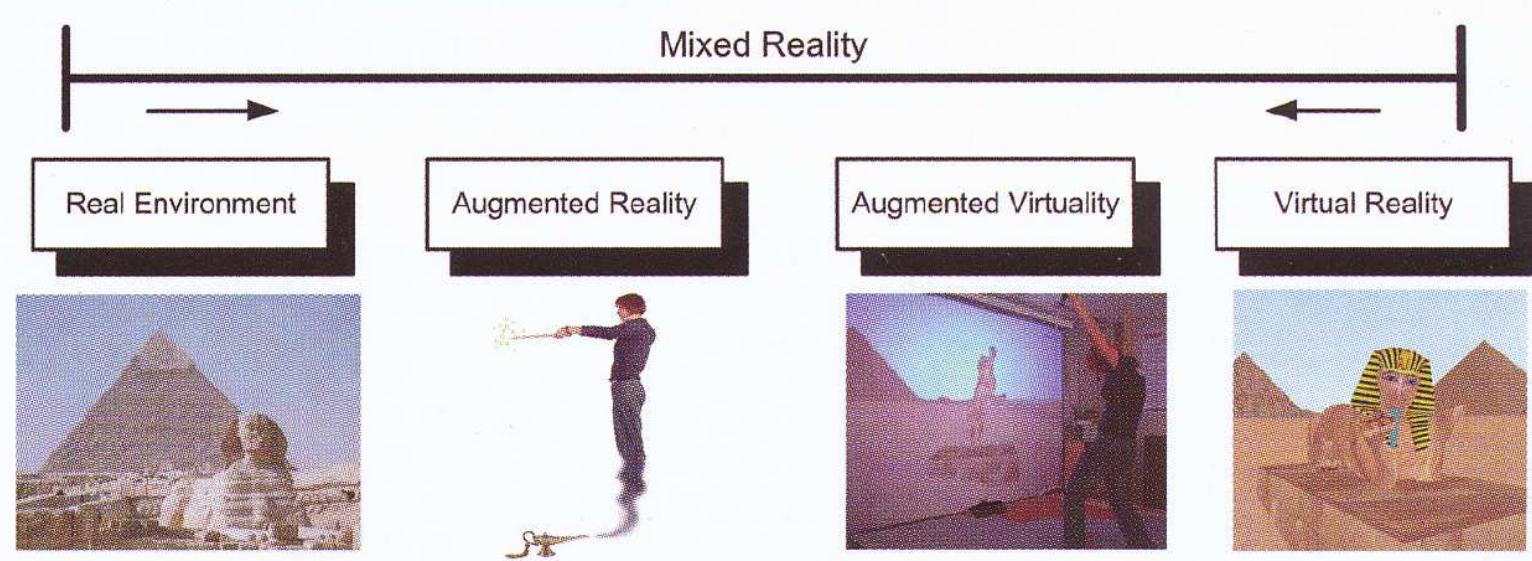
Kontinuum zwischen zwei Polen:

- **wirkliche Realität (real environment):** keine computererzeugten Stimuli
- **virtuelle Realität (virtual reality):** alles Wahrnehmbare ist künstlich – Benetzende von Wirklichkeit isoliert

dazwischen liegt als Übergangszone die „gemischte Realität“ (mixed reality), bestehend aus einer Kombination aus Realität und Virtualität:

- **erweiterte Realität (augmented reality):** virtuelle Objekte werden in reale Umgebung projiziert
- **erweiterte Virtualität (augmented virtuality):** reale Darsteller bewegen sich in virtuellen Umgebungen (z.B. Wetterkarte)

Realität-Virtualität-Kontinuum



Anwendungen von VR-Systemen

- Archäologie/Geschichte
- Fahrsimulation
- Fertigung
- Medizin
- Spiele/Unterhaltung
- Theater

Archäologie/Geschichte

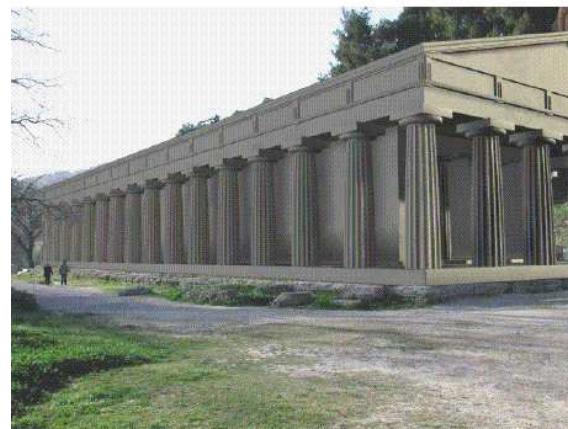
Beispiele für praktische VR-Anwendung

- VR-Präsentation der Heiligtümer von Dun Huang an der Seidenstraße in Westchina
- Virtuelle Wiederbelebung der Terrakotta-Armee in Lintong, China
- Archeoguide (Führer durch archäologische Fundstätten)
- Lifeplus (lebensnahe Rekonstruktion historischer Szenen)
- Campeche und Calakmul (Rekonstruktion historischer Stätten in Mexiko)
- etc.

Archäologie/Geschichte



(a)



(b)

ArcheoGuide

- a) wirkliche Realität
- b) erweiterte Realität
- .

Archäologie/Geschichte



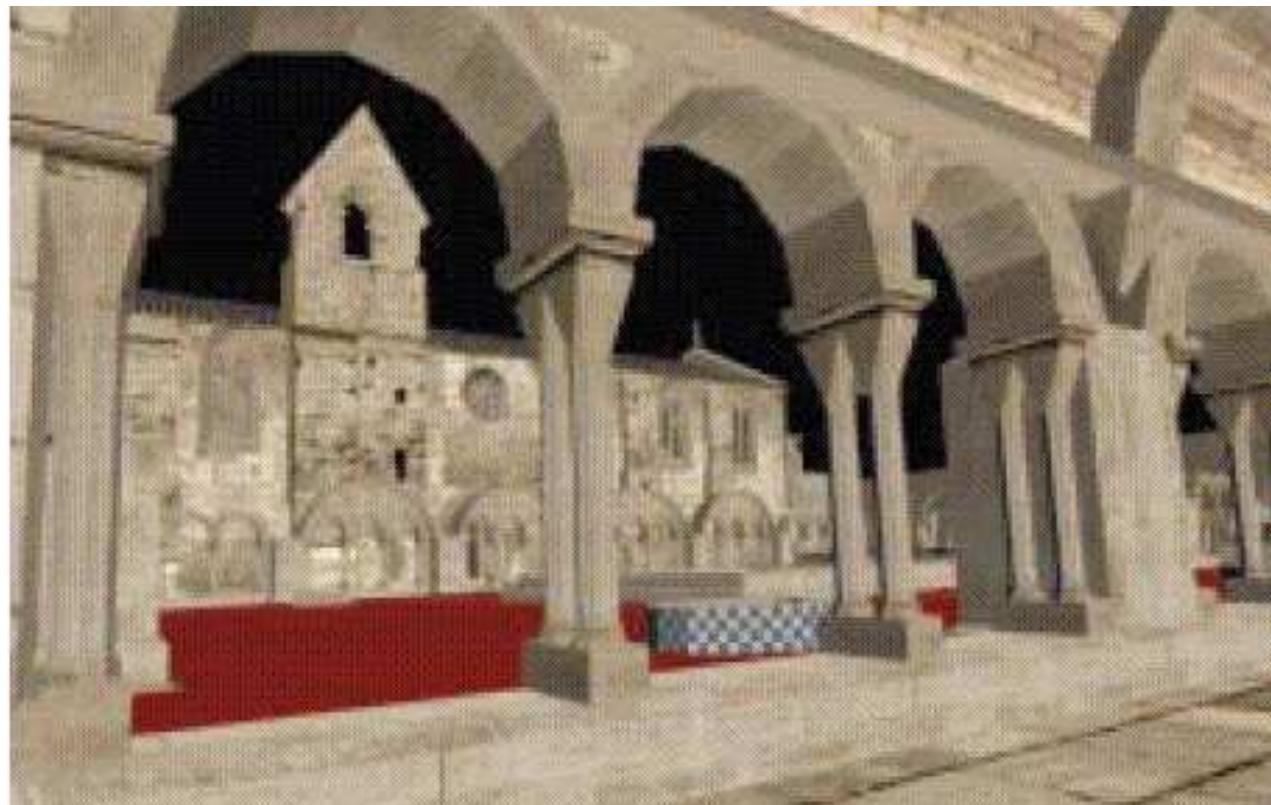
Erweiterte Realität (augmented reality, aus: ArcheoGuide)

Archäologie/Geschichte



Wiedergabe der Wandmalereien von Dun Huang (Westchina)

Archäologie/Geschichte



Virtuelle Rekonstruktion des Klosters Santa Clara (Coimbra, Portugal)

Archäologie/Geschichte



MIRALab - University of Geneva

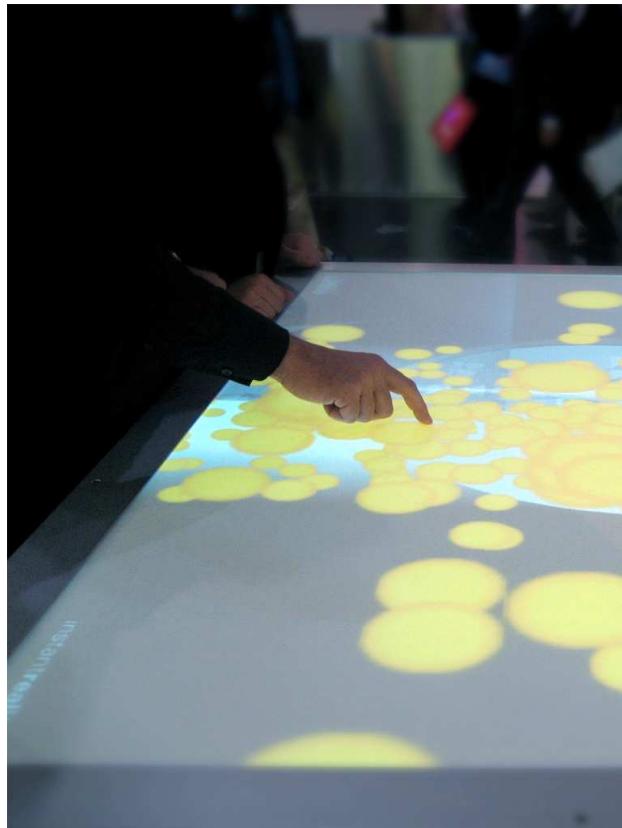
Rekonstruktion einer Lebensszene in Pompeji

Anwendungen: Simulatoren



Schiffsführungssimulator (vollständige Immersion)

Anwendungen: Virtuelle Fabrik

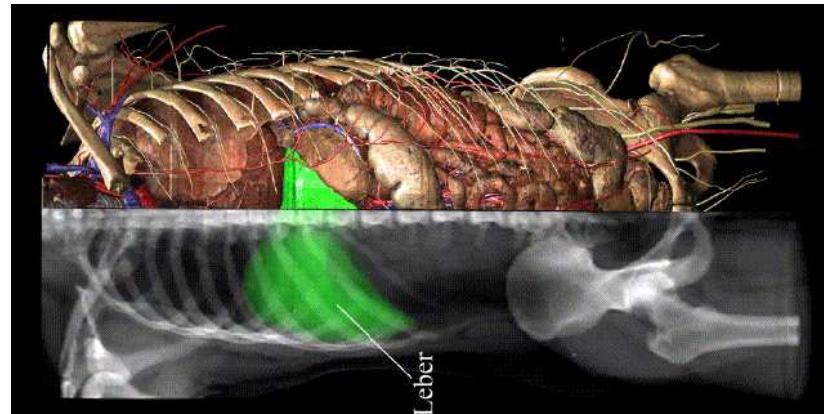
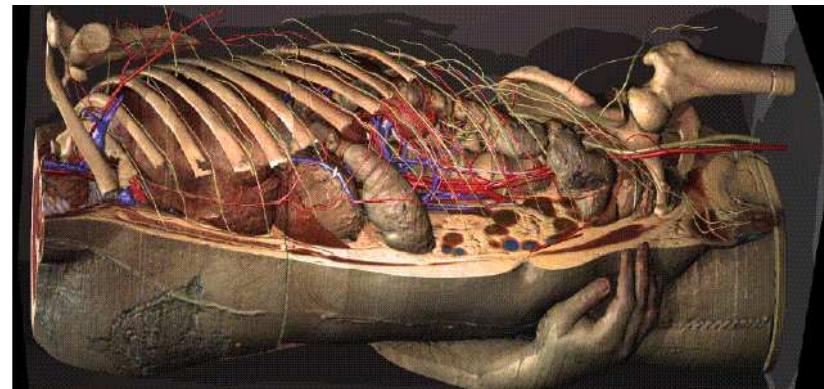


Touchtable als Werkzeug zur Repräsentation und Manipulation von Prozessen –
Auswahl von Elementen durch einfaches Berühren (Fraunhofer IGD)

VR in der Medizin

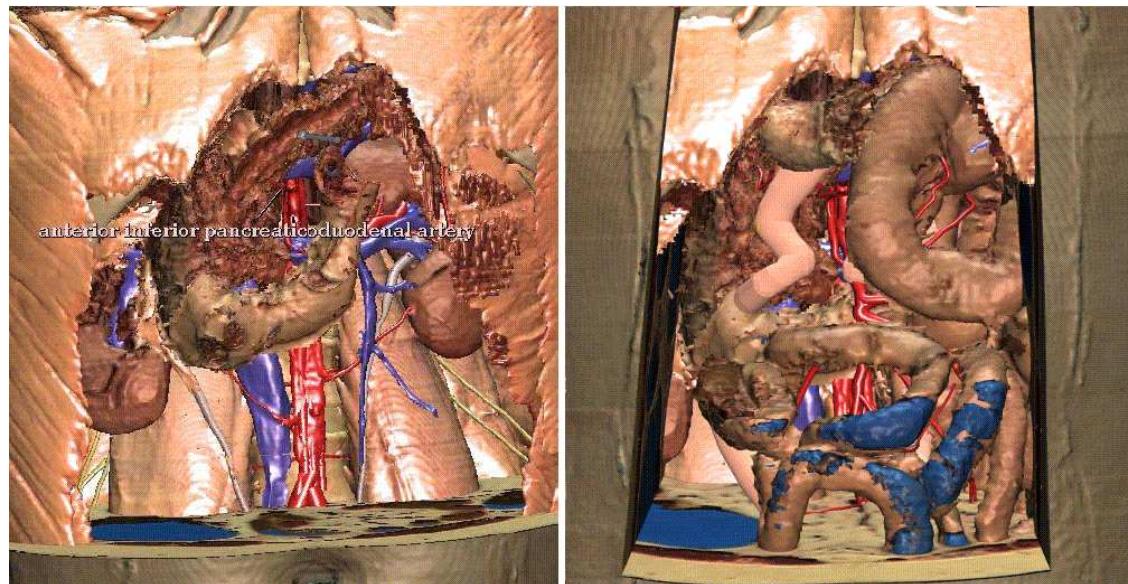
- virtuelle Anatomie
- virtuelle Chirurgie
- VR in der Rehabilitation
- VR in der Physiotherapie
- VR in der Psychotherapie

Virtuelle Anatomie (Visible Human Projekt im IMDM des UKE, Hamburg)



Interaktiver Anatomieatlas

Virtuelle Chirurgie



Virtuelle v-shape-Resektion des Pankreas

(aus: S. Gehrmann, 3D-Atlas des kardiovaskulären Systems des Menschen auf der Basis des
Visible Human, Dissertation 2003)

Virtuelle Chirurgie

- Simulation chirurgischer Eingriffe erfordert Realzeitsysteme mit visuellen und haptischen Interfaces
- Formveränderungen der Organe beim simulierten Eingriff müssen realistisch modelliert werden
- Üben von Eingriffen an virtuellen Patienten verbessert die Leistung im Operationssaal

Bereits weit entwickelt (im Bereich virtueller Anatomie): virtuelle Endoscopie und virtuelle Laparoskopie

Virtuelle Physiotherapie

benötigt Realzeitsysteme mit visuellen und komplexen haptischen Interfaces

- Identifikation von Bewegungsmustern (s.u. „motion capture“)
- motorische Rehabilitation von Schlaganfallpatienten
- Behandlung von Gleichgewichtsstörungen
- durch Web-Einsatz (Telerehabilitationssysteme) auch Fernbehandlung möglich

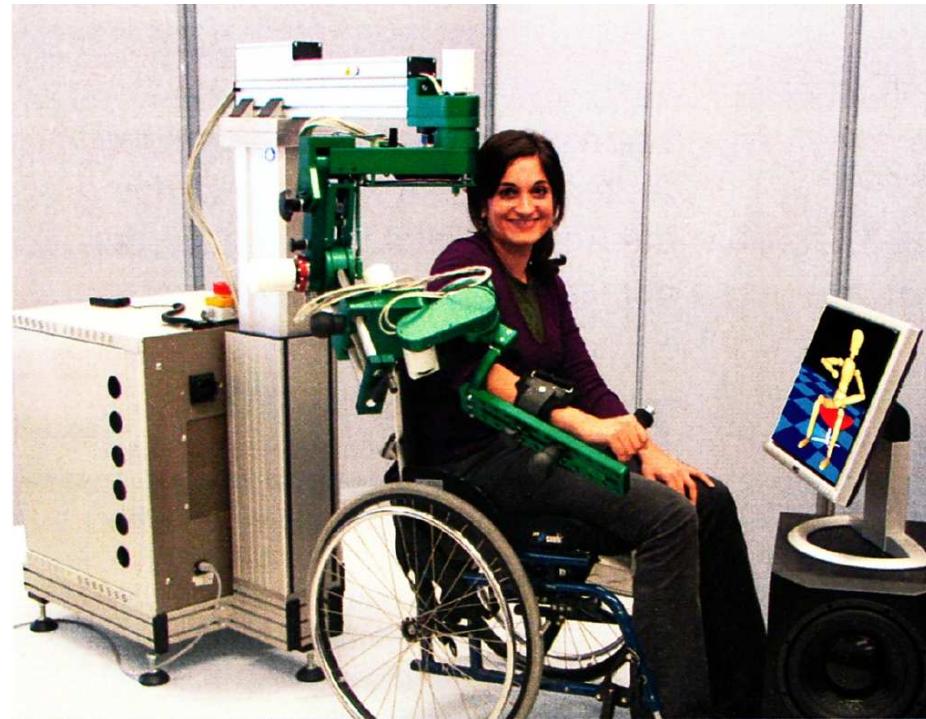
Anwendungen in der Rehabilitation



Interaktives System zur Wiedererlernung des Gehens

(aus: Apotheken Umschau 01/09)

Anwendungen in der Rehabilitation



Interaktives System zur Wiedererlernung des Greifens

(aus: Apotheken Umschau 01/09)

Virtuelle Psychotherapie

Behandlung von Phobien wie Höhenangst, Flugangst, Klaustrophobie, Angst vor Spinnen, Angst vor Publikum oder allgemein vor Menschen

- traditionelle Behandlungsweise: Patienten werden betreffenden angsterregenden Situationen mit zunehmender Intensität ausgesetzt, um die Angst zu mildern
 - in vivo: die entsprechende Situation wird real hergestellt
 - in vitro: Patienten stellen sich die Situation lebhaft vor
- Nachteil: aufwendig/teuer, ggf. peinlich, gefährlich

Virtuelle Psychotherapie

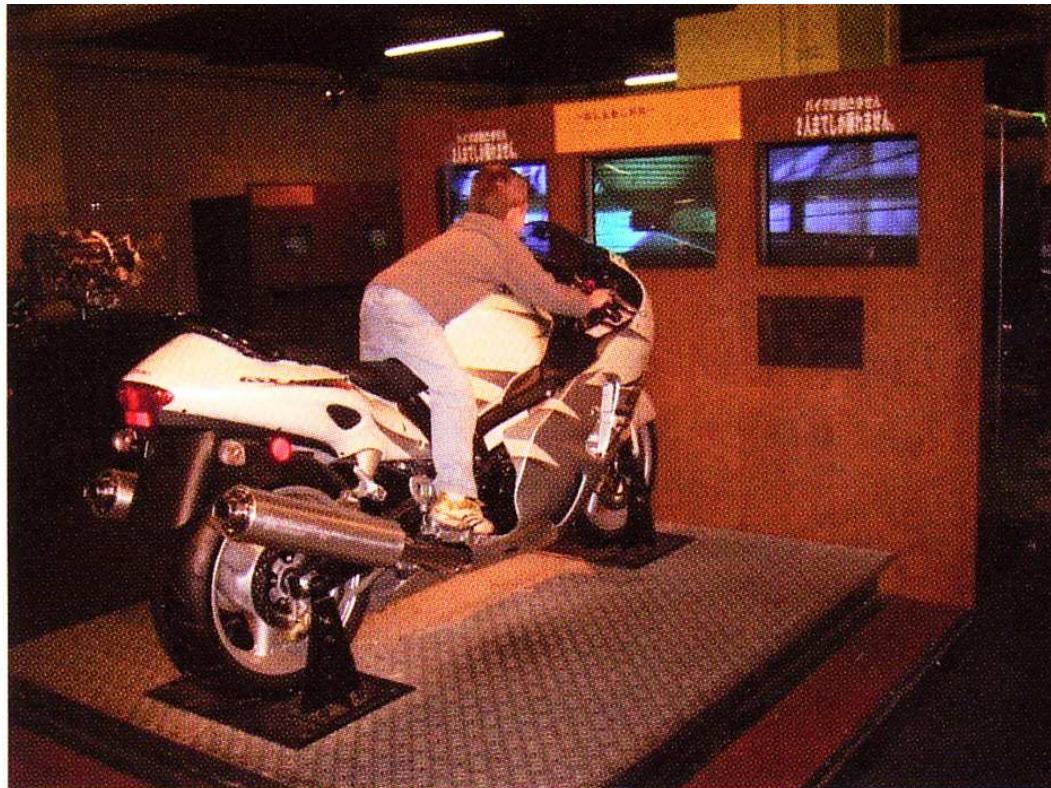
- VR-Simulation der angstauslösenden Situation (z.B. vor einem Auditorium zu sprechen, in großer Höhe zu stehen, etc.) hat i.d.R. diese Nachteile nicht
- mit VR-Simulation können Situationen geschaffen werden, die im realen Leben schwer herzustellen sind
- VR-Systeme werden bereits erfolgreich für die Behandlung von Phobien eingesetzt

Anwendungen: Simulatoren



Immersive Videospiele bzw. Fahrsimulatoren

Anwendungen: Simulatoren



Immersive Videospiele bzw. Fahrsimulatoren

Virtuelles Theater

- Aufführung auf einer virtuellen Bühne
- menschliche Darsteller steuern Avatare
- autonome virtuelle Darsteller könnten mitspielen
- Bühnenarbeiter steuern Beleuchtung und Kulissenumbau
- Regisseure bestimmen die Einsätze
- Darsteller reagieren auf Zuschauerreaktionen
- virtuelles Theater kann „verteilt“ gespielt werden; d.h. die einzelnen Akteure sowie die Zuschauer können sich an unterschiedlichen Orten aufhalten

Konferenzen (exemplarisch)

- International Conference on Animation, Effects, Games and Digital Media (2009 in Stuttgart, Deutschland)
- International Conference on Articulated Motion and Deformable Objects (2008 auf Mallorca, Spanien)
- International Conference on Computer Graphics and Artificial Intelligence (2011 in Athen, Griechenland)
- International Conference on Cyberworlds (2008 in Hangzhou, China)
- International Conference on Technologies for Interactive Digital Storytelling and Entertainment (2006 in Darmstadt, Deutschland)
- International Conference on Virtual Systems and Multimedia (2007 in Brisbane, Australien)

Symposia und Workshops (exemplarisch)

- SCA: Eurographics/SIGGRAPH Symposium on Computer Animation (2011 in Vancouver, Kanada)
- Workshop on Human Motion – Understanding, Modeling, Capture and Animation (2007 in Rio de Janeiro, Brasilien)
- LAMDa: Workshop on Location Awareness for Mixed and Dual Reality (2011 in Palo Alto, USA)
- ISMAR: International Symposium on Mixed and Augmented Reality (2010 in Seoul, Korea)
- EGMM: Eurographics Multimedia Workshop (2004 in Nanjing, China)

Symposia und Workshops (Fortsetzung)

- EGVE: Eurographics Virtual Environments Symposium (2010 in Stuttgart, Deutschland)
- VAST: International Symposium on Virtual Reality, Archaeology and Cultural Heritage (2010 in Braga, Portugal)
- VRIPHYS: Workshop on Virtual Reality Interaction and Physical Simulation (2010 in Kopenhagen, Dänemark)
- VCBM: Workshop on Visual Computing for Biomedicine (2010 in Leipzig, Deutschland)

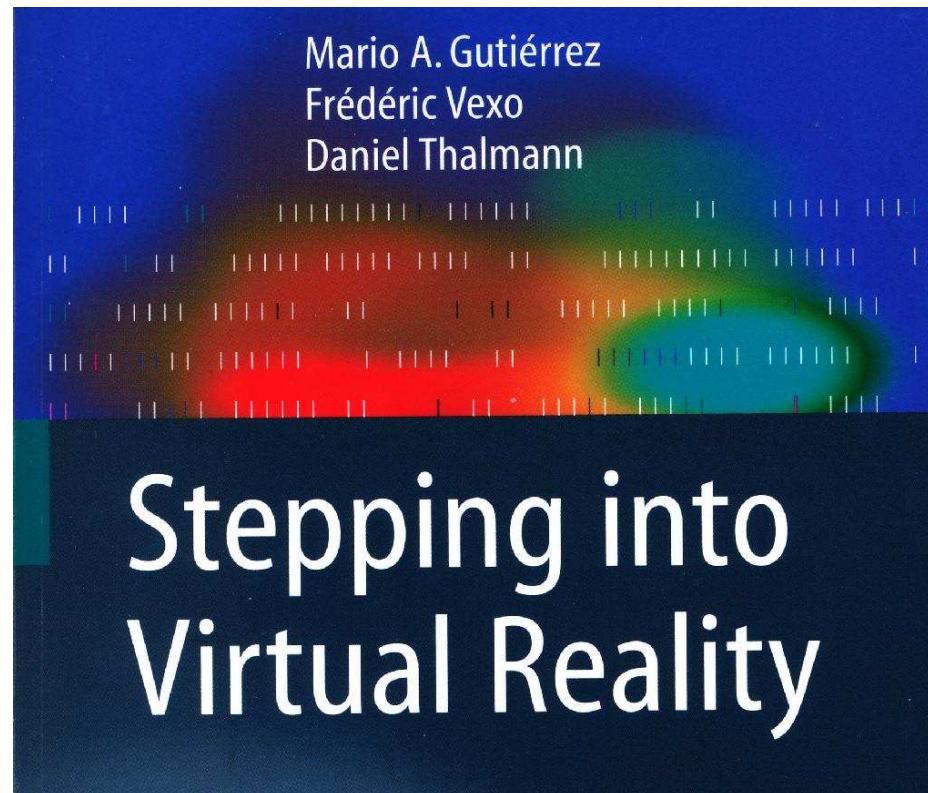
Literatur

- Norman I. Badler, Cary B. Phillips und Bonnie Lynn Webber: Simulating Humans, in: Computer Graphics. Animation and Control. Oxford University Press, New York, 1993
- Chi Chung Ko und Chang Dong Cheng: Interactive Web-Based Virtual Reality with Java 3D, Information Science Reference, Hershey/New York, 2009
- Roy Featherstone: Robot Dynamics Algorithms, Kluwer International Series in Engineering & Computer Science, 1986
- Mario A. Gutiérrez, Frédéric Vexo und Daniel Thalmann: Stepping into Virtual Reality, Springer 2008
- Ronald Huston: Multibody Dynamics, Butterworth-Heinemann, Stoneham, 1990

- Dietmar Jackèl: Methoden der Computeranimation, Springer, Berlin, 2006
- Jean-Claude Latombe, Robot motion planning, Kluwer Academic Publishers, Norwell, MA, 1996
- Ian Millington: Artificial intelligence for games, Morgan Kaufmann/Elsevier, 2006
- Rick Parent: Computer animation – algorithms and techniques, Elsevier/Morgan Kaufmann, 2008
- Daniel Thalmann und Soraia Raupp Musse (Hrsg.): Crowd Simulation, Springer, Berlin, etc., 2007
- The Open Virtual Reality Journal (Open Access Zeitschrift, herausgegeben von Bentham Open)

Grundlage:

die Vorlesung dieses Teils des Moduls IVC basiert im Wesentlichen auf



erschienen 2008 beim Springer-Verlag (i.d. Informatik-Bibliothek vorh.)

2. Computergestützte Animation

- Einführung
- Animationsformen
- Bewegungserfassung (Motion Capture)
- Key-Frame-Animation (Schlüsselbild-Animation)
- Direkte und inverse Kinematik
- Bewegungsanpassung (Motion Retargeting)
- Prozedurale Animation

Einführung

- Virtuelle Welten sind dynamisch,
- Objekte werden bewegt, transformiert
- virtuelle Personen „leben“

Computergestützte Animation erzeugt die Illusion von Bewegung durch Darstellung einer Folge individueller Zustände einer dynamischen Szene z.B. Gelenkwinkel = Zustandsvariable von Menschen u.ä.

Animationsformen

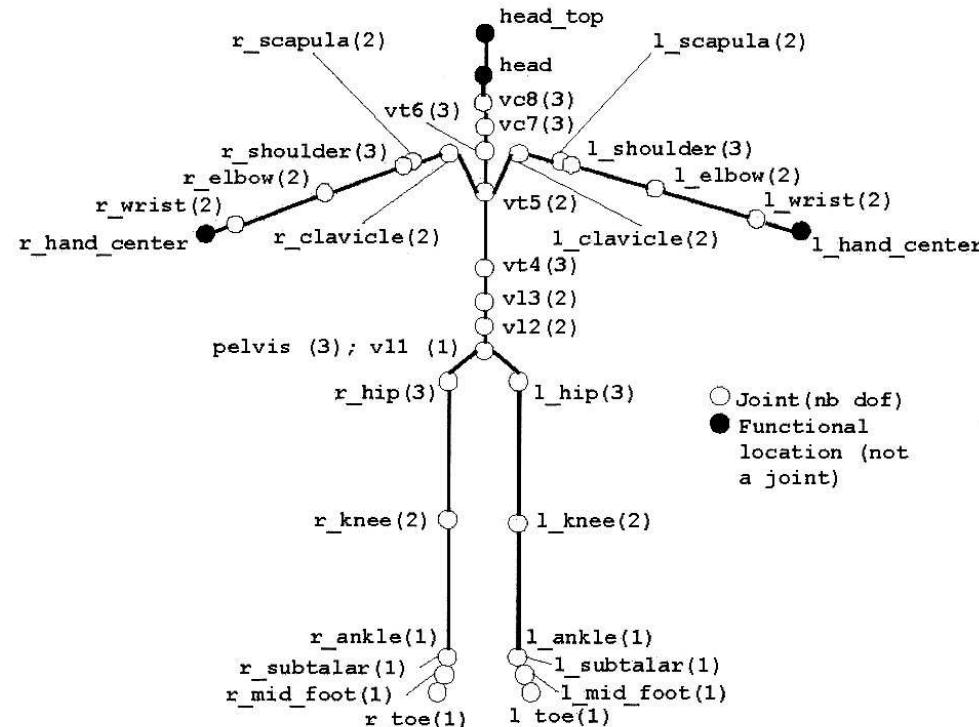
Man unterscheidet derzeit

- Real-Time-Animation
 - schnelle Animationsform für VR-Systeme (z.B. Video-Spiele, Fahr- und Flugsimulatoren) – viele Bildbereiche kommen mit relativ niedrigen Detailstufen aus, da sie nicht im (gewünschten) Zentrum der Aufmerksamkeit liegen
- bildweise Animation
 - mit oft hoher Detailgenauigkeit in allen Bildbereichen – darum sehr langsam (Anwendung: computergenerierte Filme)

Virtuelle Darsteller mit beweglichen Körpern

- Virtuelle Darsteller werden modelliert durch
 - die „Haut“, die die äußere Erscheinung wiedergibt und
 - das Skelett, auf dem die Modellierung von Bewegungsabläufen basiert.
- Das Skelett besteht aus einer hierarchisch geordneten Menge von Gliedmaßen und Gelenken („kinematische Kette“).

Virtuelle Darsteller mit beweglichen Körpern



Formales Skelett entsprechend der Humanoid Animation Specification (H-ANIM) für VRML und MPEG4 (Standard des Web3D Consortium)

Virtuelle Darsteller mit beweglichen Körpern

- Die Detaillierung orientiert sich an den Animationsanforderungen (für realitätsnahe Anwendungen in der Physiotherapie müssen alle Knochen und Gelenke – z.B. auch Wirbel – exakt modelliert werden, für Computerspiele reichen i.d.R. vereinfachte Skelette aus).
- Je detaillierter ein Skelett ist, desto größer ist der Berechnungsaufwand für die Animation.

Virtuelle Darsteller mit beweglichen Körpern

- Gelenke haben einen oder mehrere Bewegungs-Freiheitsgrade (degrees of freedom – DOF).
- Jedem Gelenk ist eine 4D Transformationsmatrix zugeordnet, in der seine Verdrehung(en) sowie die Länge und Orientierung des zugehörigen Gliedes abgelegt sind.
- Zusätzlich werden zu jedem Gelenk die Bewegungsgrenzen gespeichert, um unnatürliche Bewegungen zu vermeiden.
- Die endgültige Körperhaltung des Skeletts ergibt sich aus der Auswertung der einzelnen Gelenkmatrizen entsprechend der hierarchischen Struktur.
- Für die Animation werden die Transformationsmatrizen der Gelenke entsprechend dem gewünschten Bewegungsablauf manipuliert.

Bewegungssteuerung (Motion Control)

Man unterscheidet drei Ansätze:

- **geometrisch:** animierte Objekte werden lokal m.H. geometrischer Daten gesteuert; verwendete Methoden:
 - Bewegungserfassung (Motion Capture)
 - Form-Transformation
 - parametrische Key-Frame-Animation
- **physikalisch:** Simulation der physikalischen Gesetze, die Bewegungen in der realen Welt zugrunde liegen, durch funktionale Methoden (z.B. basierend auf biomechanischen Gesetzen)
- **verhaltensorientiert:** animierte Objekte werden als autonome „Wesen“ betrachtet, deren Verhalten von ihrer Beziehung zueinander bestimmt ist.

Bewegungserfassung (Motion Capture)

- derzeit wesentliche Grundlage der Bewegungssteuerung
- beruht auf Messung und Aufzeichnung der Bewegungen realer Lebewesen (Menschen, Tiere)
- die gemessenen Bewegungsdaten werden auf virtuelle Darsteller abgebildet:
 - direkt: Bewegung eines realen Menschen wird in Echtzeit auf einen virtuellen Darsteller übertragen
 - indirekt: Mausbewegung wird auf virtuellen Darsteller übertragen
- in der Praxis werden derzeit zwei unterschiedliche Methoden zur Bewegungserfassung eingesetzt:
 - optische Bewegungserfassungssysteme
 - magnetische Systeme zur Positions-, Orientierungs- bzw. Bewegungserfassung

Optische Bewegungserfassung

- **passive optische Systeme** nutzen i.d.R. mehrere (4-6) unterschiedlich positionierte Kameras; an ausgewählten Stellen (z.B. Gelenke) der zu vermessenden Person werden reflektierende Markierungen angebracht, die von den Kameras her beleuchtet werden; die 3D-Positionen der Markierungen können aus den synchronen Kamerabildern ermittelt werden
Nachteil: verdeckte oder zu dicht beieinander liegende Markierungen erschweren die Positionsbestimmung
- **aktive optische Systeme** verwenden anstelle reflektierender Markierungen LEDs, die einzeln angesteuert und über mehrere Kameras trianguliert werden können – aufwendiges Verfahren

Bewegungserfassung (Motion Capture)



Bewegungserfassung (motion capture) mit magnetischen Sensoren

Magnetische Systeme zur Positions-, Orientierungs- bzw. Bewegungserfassung

- ermitteln Position und Orientierung über die relative Stromstärke der magnetischen Felder von drei zueinander orthogonalen Spulen jeweils im Sender und Empfänger
- optische Verdeckungen sind irrelevant **aber**
- Stromquellen, Lampen, Computer und schlichte Metallteile können die Magnetfelder stören
- „Verkabelung“ der Person stört
- Erfassungsraum ist deutlich kleiner als bei optischen Systemen
- neuere Systeme haben diese Probleme weitgehend im Griff; drahtlose Systeme werden für die die Positions- und Orientierungserfassung von Head-Mounted Displays und Data Gloves eingesetzt

Vorteile der Bewegungserfassung

- Systeme arbeiten in Echtzeit
- Arbeitsumfang hängt nicht von Länge bzw. Komplexität des Einsatzes ab
- komplexe Bewegungen und realistische physikalische Interaktionen können einfach physikalisch korrekt wiedergegeben werden
- durch Bewegungserfassung kann ein Darsteller mehrere Rollen in einem Film übernehmen

Nachteile der Bewegungserfassung

- zur Erfassung und Bearbeitung der Daten wird spezielle Hard- und Software benötigt, deren Kosten abschreckend sein können
- Systeme zur Bewegungserfassung benötigen meist besonders ausgestattete Räume
- aufgrund rasanter Entwicklung veraltet die einmal erworbene Technik schnell
- ohne Datennachbearbeitung sind die Ergebnisse auf das beschränkt, was im Erfassungsraum gemacht werden kann
- es können i.d.R. keine Bewegungen durchgeführt werden, die nicht physikalischen Gesetzen gehorchen
- virtuelle Darsteller müssen in ihren Proportionen im wesentlichen den Personen entsprechen, von denen die Bewegungsdaten erfasst wurden (z.B. Kollision übertrieben großer Hände von Cartoon-Figuren)

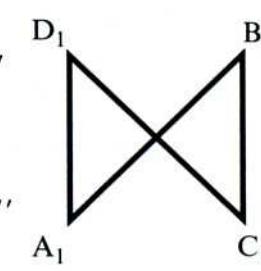
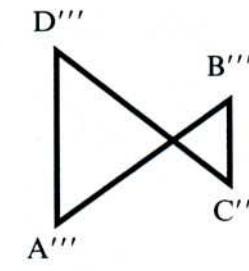
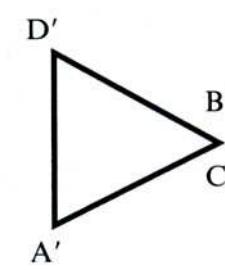
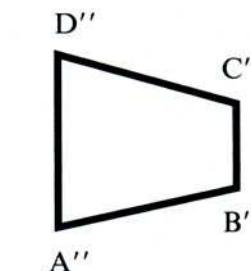
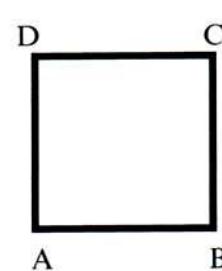
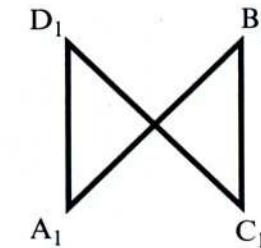
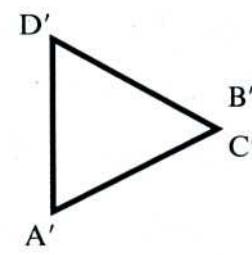
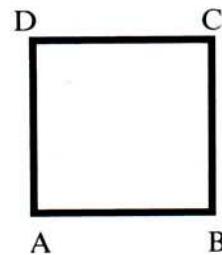
Key-Frame-Animation (Schlüsselbild-Animation)

Wegen des hohen Rechenzeitbedarfes werden i.d.R. nicht alle Bilder einer Sequenz komplett neu generiert.

Basierend auf charakteristischen „Schlüsselbildern“ werden dazwischenliegende Bilder (in-betweens) einer Sequenz automatisch generiert; verwendete Techniken sind

- direkte Interpolation einer geometrischen Form in eine andere m.H. mathematischer Methoden (z.B. mit interpolierenden Splines);
Nachteil: die Anzahl der Eckpunkte von Ausgangs- und Zielgeometrie muss übereinstimmen
- Interpolation formbeschreibender Parameter (z.B. physikalische Parameter), die das Verhalten der Figur bestimmen, anstelle der Form selbst – führt i.d.R. zu besseren Resultaten

Form-Interpolation (In-betweening)

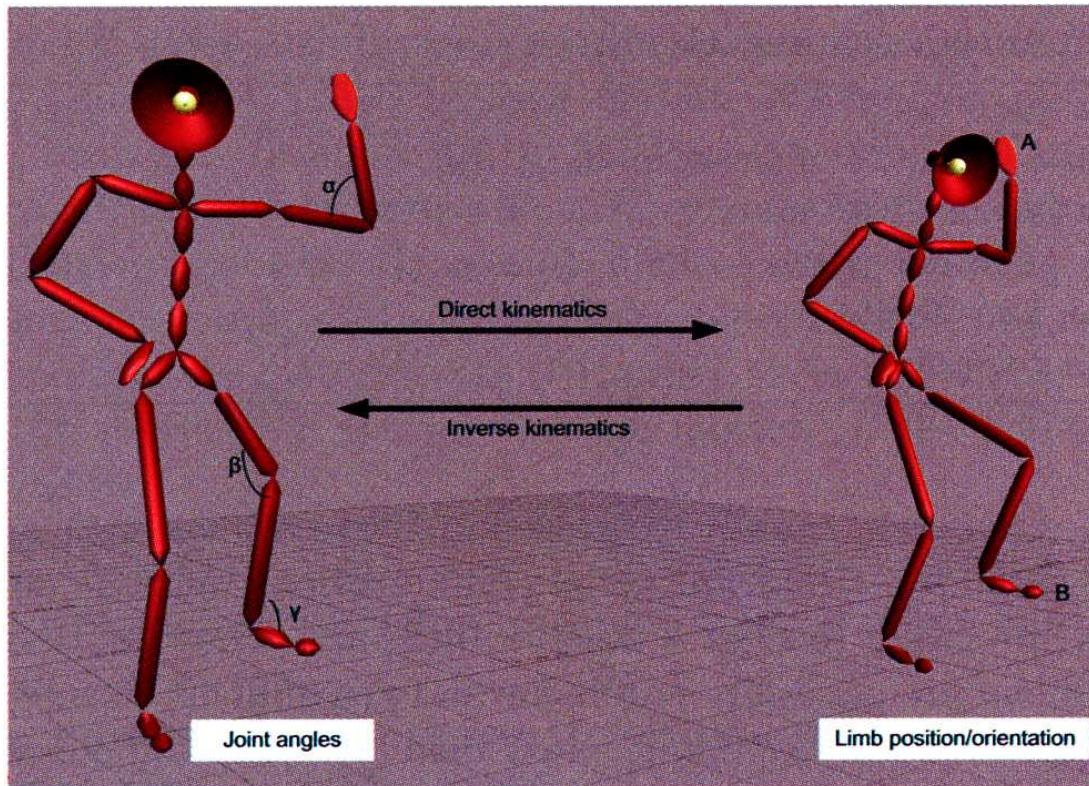


Interpolationen zwischen zwei Bildern
(oben bei 50 %, unten bei 25 %, 50 % und 75 %)

Direkte und inverse Kinematik

- **Direkte Kinematik** dient in der Robotik der Ermittlung der Position und Orientierung eines Endeffektors in Abhängigkeit von den eingestellten Gelenkwinkeln; analog dient sie bei der Computeranimation z.B. menschenähnlicher Darsteller der Ermittlung der Position und Orientierung des Kopfes, einer Hand oder eines Fußes.
- **Inverser Kinematik** dient umgekehrt der Ermittlung der für das Erzielen einer gewünschten Position und Orientierung eines Endeffektors durchzuführenden (sinnvollen) Gelenkwinkelveränderungen.

Direkte und inverse Kinematik



Probleme der inversen Kinematik

- Mehrdeutigkeit: eine gewünschte Position und Orientierung eines Endeffektors kann auf unterschiedlichen Positionen und Orientierungen der einzelnen Elemente der „kinematischen Kette“ beruhen
- es können unzulässige Konstellationen auftreten (z.B. unmögliche Gelenkwinkel)

wichtige Aufgabe: eine sinnvolle Konstellation muss gefunden werden

Lösungsansätze für die inverse Kinematik

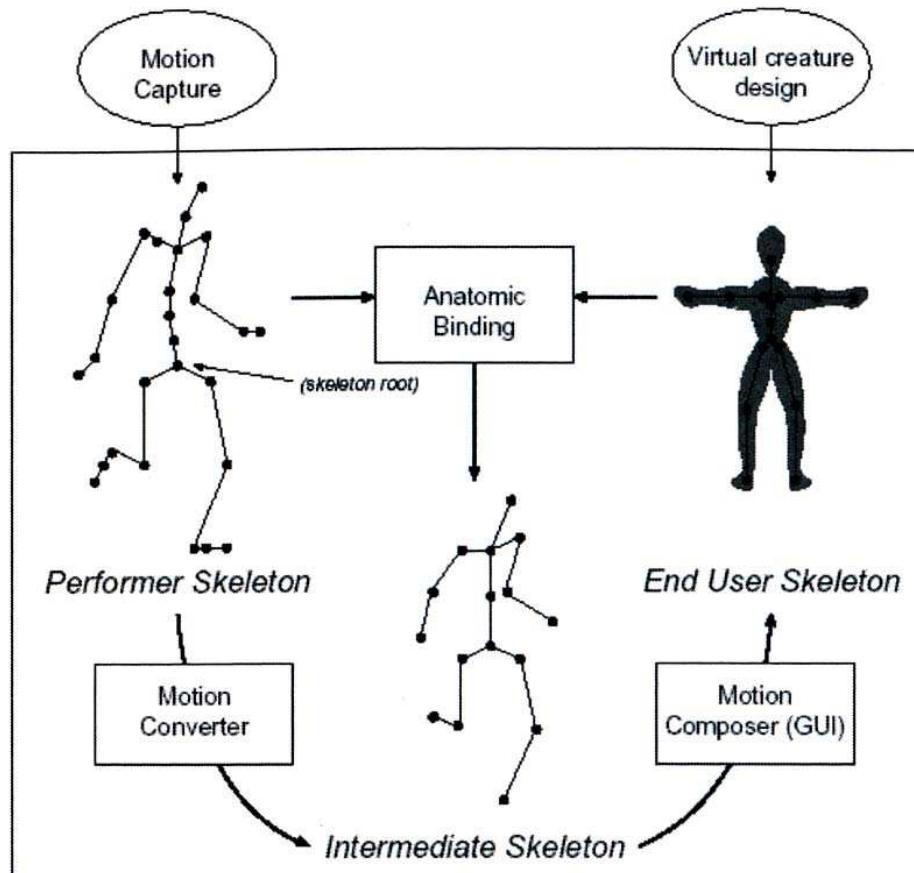
Als grundlegende Methoden werden eingesetzt:

- algebraische Methoden,
die durch sukzessive Invertierung der Transformationsmatrizen
schrittweise die einzelnen Gelenkwinkelvektorkomponenten
berechnen, oder
- geometrische Methoden,
die ausgehend von der Geometrie des Skeletts die Komponenten des
Gelenkwinkelvektors ermitteln, oder
- numerische Methoden,
mit deren Hilfe iterativ die Komponenten des Gelenkwinkelvektors
ermittelt werden (Problem hierbei: Identifikation eines geeigneten
Startwertes für die Iteration)

Bewegungsanpassung (Motion Retargeting)

- **Aufgabe:** Übertragen der Bewegungsabläufe, die beim Motion Capture von einer realen Person erfasst wurden, auf einen virtuellen Darsteller
- **Problem:** wenn Körperproportionen nicht übereinstimmen, entstehen in der virtuellen Welt Kollisionen, oder der virtuelle Darsteller erreicht gewünschte Ziele nicht
- möglicher **Lösungsansatz:** vereinfachtes „Zwischenskelett“, auf das zunächst die erfassten Positionsdaten abgebildet werden – diesem werden dann die Gelenkmatrizen für den virtuellen Darsteller entnommen

Bewegungsanpassung (Motion Retargeting)



Verwendung eines „Zwischenskeletts“ für das Motion Retargeting

Fort-Bewegungen (Locomotion)

Charakterisierung des Gehens:

- Gehen ist eine periodische Bewegung, die normalisiert werden muss, um an unterschiedliche Anatomien angepasst werden zu können,
- beim Gehen bewegt sich der Körperschwerpunkt zwischen der rechten und der linken Körperseite hin und her,
- ein Fuß steht immer auf dem Boden,
- für einen kurzen Augenblick berühren beide Füße den Boden

Fort-Bewegungen (Locomotion)

Bewegungsbedingte Winkeländerungen der Gelenke können beschrieben werden durch

- Sinusfunktionen mit unterschiedlichen Amplituden und Frequenzen für globale Translationsbewegungen (vertikal, quer, frontal) sowie speziell für Bewegungen des Beckens (vor- bzw. rückwärts, links bzw. rechts und Drehung)
- periodische Spline-Funktionen zur Beschreibung von Hüft-, Knie-, Fußgelenk-, Schulter- und Ellbogenbeugung sowie der Drehung des Oberkörpers

Resultierende Gehbewegungen wirken unrealistisch!

Da die verwendeten kinematischen bzw. dynamischen Modelle recht komplex und die Berechnungen i.d.R. sehr aufwendig sind, wird hier meist Key-Frame-Animation eingesetzt.

Fort-Bewegungen (Locomotion)

Analyse von Bewegungsgrundkomponenten
(PCA: Principal Component Analysis)

- basiert auf Daten aus Bewegungserfassungen (motion capture)
- da die Daten von unterschiedlichen Personen und bei unterschiedlichen Geh- bzw. Laufgeschwindigkeiten erfasst werden, können individuelle Eigenheiten (z.B. Alter, Geschlecht, Größe, Laufen, Gehen) wiedergegeben werden
- die erfassten Daten werden in 2-Schritt-Zyklen segmentiert und normalisiert, die dann wiederholt zusammengesetzt werden können
- zusätzlich wird zu jedem Typ eine neutrale Position (aufrecht stehen) erfasst und normalisiert
- Bewegungsabläufe wirken natürlich

Fort-Bewegungen (Locomotion)



Bewegungsabläufe beim Gehen unterschiedlicher Darsteller

Prozedurale Animation

Die Bewegungsabläufe können durch Algorithmen oder Formeln bzw. durch Prozeduren beschrieben werden, die z.B. physikalische Gesetze wiedergeben (Physikbasierte Animation), wie

- die Bewegung eines Fahrzeugs entlang einer Bahnkurve,
- den Flug eines Geschosses entlang einer Wurfparabel,
- die Pendelbewegung eines Perpendikels (einer Schaukel)

oder regelbasiertes Verhalten (Verhaltensanimation), wie z.B.

- Vogelschwärme, Vieherden,
- Menschenmengen,
- autonome Darsteller

3. Virtuelle Darsteller

- Virtuelle Menschen in virtuellen Welten
- Modellierung der Haut
- Gesichtsanimation
- Interaktion zwischen virtuellen Menschen und Objekten
- Autonome Darsteller
- Simulation von Menschenmengen

Virtuelle Menschen in virtuellen Welten

- Identifikationsgrad steigt mit zunehmender Immersion
- je natürlicher sie ausgestattet sind, desto realistischer werden virtuelle Welten wahrgenommen
- eine natürliche Welt ist bevölkert von autonom agierenden und miteinander interagierenden Lebewesen
- virtuelle Welten müssen darum glaubwürdig erscheinende virtuelle Lebewesen, vor allem virtuelle Menschen, enthalten, die autonom agieren, kommunizieren, kooperieren, möglichst gar „Gefühle zeigen“, etc.

Modellierung der Haut

Es wurden unterschiedliche Ansätze entwickelt, eine verformbare Haut auf das Skelett aufzutragen:

- Skelett basierte Verformungen
- Daten gesteuerte Methoden
- physikalisch basierte Ansätze

Für Verformungen in Realzeit werden vornehmlich die beiden ersten Methoden eingesetzt.

Skelett basierte Verformung der Haut

- klassische, weit verbreitete Methode
- erste Ansätze wurden bereits 1987 vorgestellt
 - z.B. gelenkabhängige lokale Verformung:
jeder Hautoberflächenstützpunkt wird durch eine gewichtete Kombination der beeinflussenden Gelenke – ausgehend von einer „neutralen Position“ – verschoben
Problem: unerwünschte (unnatürliche) Verformungen bei starken Gelenkwinkeländerungen
- Verbesserung der Resultate z.B. durch Einführung einer „Muskelschicht“ zwischen flexibler Haut und starrem Knochen;
 - einfaches Modell: an den „Knochen“ orientierte Ellipsoide,
 - realistischer: biomechanisches Modell der Muskelverformung (aufwendig)

Daten gesteuerte Methoden der Hautmodellierung

- Grundlage: realistische Hautformen, die
 - durch Oberflächenvermessung (3D-Scanner) realer Menschen oder
 - aus Simulationen mit realistischen Modellen gewonnen oder
 - von erfahrenen Designern modelliert wurden
- Ziel: Echtzeit-Animation
- Verfahren:
 - Formen in Schlüsselpositionen werden vom Modell übernommen,
 - Formen in Zwischenpositionen werden durch geeignete Blending-Verfahren ermittelt

Neuere Verfahren basieren nicht auf einem vordefinierten Skelett; ausgehend von einer Menge verformter Oberflächennetze, die unterschiedliche Positionen repräsentieren, werden Knochen geschätzt und Zwischenpositionen ermittelt – funktioniert nur bei relativ starren Strukturen

Gesichtsanimation

- Ziel: hoher Grad an Realismus in Bezug auf
 - Bewegung
 - Mimik / Ausdruckskraftund Animationsgenerierung in Echtzeit mit geringstmöglichen Datenaufwand
- spezielles Anwendungsgebiet (neben „normalen“ virtuellen Darstellern): realitätsnahe virtuelle NachrichtensprecherInnen oder virtuelle Ansprechpartner in synthetischen Auskunftssystemen
- Mittel:
 - Verwendung sehr feiner Gittermodelle zur Formbeschreibung
 - Verwendung effektiver Verformungsverfahren
 - muskulaturbasierte Bewegungsmodelle
 - Definition von „Visemen“ (visuelles Gegenstück zu Phonemen)

Gesichtsanimation



Überzeugend realistische Ausdruckskraft / Mimik
(aus dem Film *Avatar* von James Cameron, 2009)

Gesichtsanimation

Das Facial Action Coding System (FACS)

- beruht auf Messung von Gesichtsbewegungen,
- wurde durch Analyse der anatomischen Grundlagen von Gesichtsbewegungen gewonnen,
- stellt High-Level-Parameter für die Gesichtsanimation zur Verfügung,
- definiert dazu 46 grundlegende „Action Units“ (AU).
- Jede AU beschreibt die Kontraktion eines Gesichtsmuskels oder einer Gruppe von Gesichtsmuskeln (nur willkürlich bewegliche).
- Die Sammlung von AUs repräsentiert einen Baukasten zur Erstellung von Gesichtsausdrücken.

Gesichtsanimation

Gesichtsanimationssysteme arbeiten i.d.R. in folgenden Schritten:

1. geeignete Parametrisierung der Animation des Gesichtsmodells
2. Definition von Grundbausteinen für die Animation in Abhängigkeit von den gewählten Parametern (z.B. unbewegtes Gesicht, Viseme) als Key-Frames
3. Definition unterschiedlicher Interpolations- bzw. Blendingfunktionen um die Viseme zu „Wörtern und Sätzen“ zu verknüpfen (die Qualität dieser Funktionen trägt wesentlich zur realistischen Wirkung bei)
4. Generierung eines Gitternetzes der Oberfläche für jedes berechnete Einzelbild

Gesichtsanimation



Gesichtsausdruck eines Singenden

Gesichtsanimation

Ein weiterer Ansatz zur Gesichtsanimation ist die Merkmalspunkt (feature point) basierte Netzverformung.

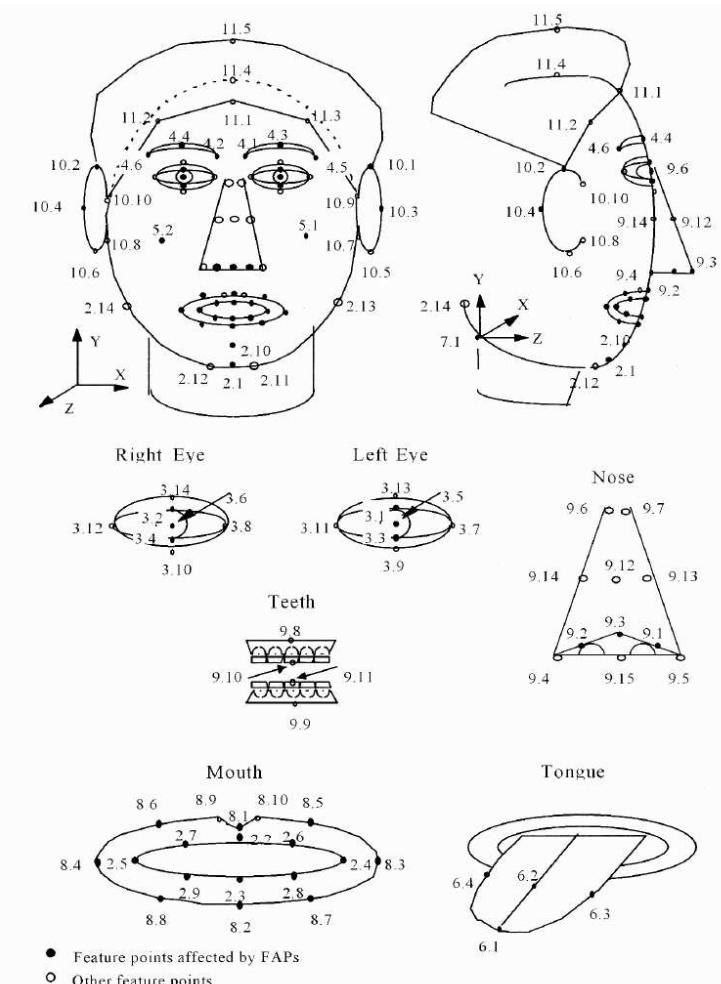
- die Gesichtsform wird durch
 - eine Menge von Gesichtsdefinitionsparametern (FDP: Facial Definition Parameters) und
 - eine Menge von Gesichtsanimationsparametern (FAP: Facial Animation Parameters) festgelegt
- die Animation reproduziert damit Gesichtsausdruck, Emotion und Aussprache
- die FDPs werden durch die Position von Merkmalspunkten definiert (Mundwinkel, Augenwinkel, Nasenspitze, etc.)
- m.H. der FDPs kann ein Gesichtsmodell auf ein bestimmtes Gesicht abgebildet werden

Gesichtsanimation



Merkmalspunkte als Basis für die Gesichtsanimation
(aus dem Film *Avatar* von James Cameron, 2009)

Gesichtsanimation



Merkmalspunkte zur Steuerung des Gesichtsausdrucks

Gesichtsanimation

- die FAPs (vergleichbar den AUs, s.o.) bestimmen die animationsabhängige Verschiebung von Merkmalspunkten und damit die Verformung des Gesichts
- die MPEG-4 Facial Animation Engine kann m.H. von 66 Low-Level-FAPs in Echtzeit eine Vielzahl von Gesichtsausdrücken (inkl. Sprachartikulation) darstellen
- aktuelle Systeme extrahieren aus natürlich gesprochener und synthetischer Sprache Phoneme und generieren daraus Sprechanimationen mit Lippensynchronisation („Talking Head Systeme“)
- neueren Entwicklungen gelingt eine natürlich wirkende Integration von „Emotionen“ in den Gesichtsausdruck von „Talking Heads“

Interaktion zwischen virtuellen Menschen und Objekten

Problem: wie können virtuelle Darsteller (überzeugend) virtuelle Objekte manipulieren?

Aufgabe gliedert sich in zwei unterschiedliche Teile:

1. formale (abstrakte) Spezifikation des Objektverhaltens
2. sinnvolle und glaubwürdige (primär grafische) Darstellung des Manipulationsvorgangs in einer Animation

Dies sei am Vorgang des Greifens erläutert: hierbei müssen die Bewegungen der Hand, einer komplizierten beweglichen Struktur bestehend aus 27 Knochen, sowie die damit verbundene Arm- und Körperbewegung genau modelliert werden.

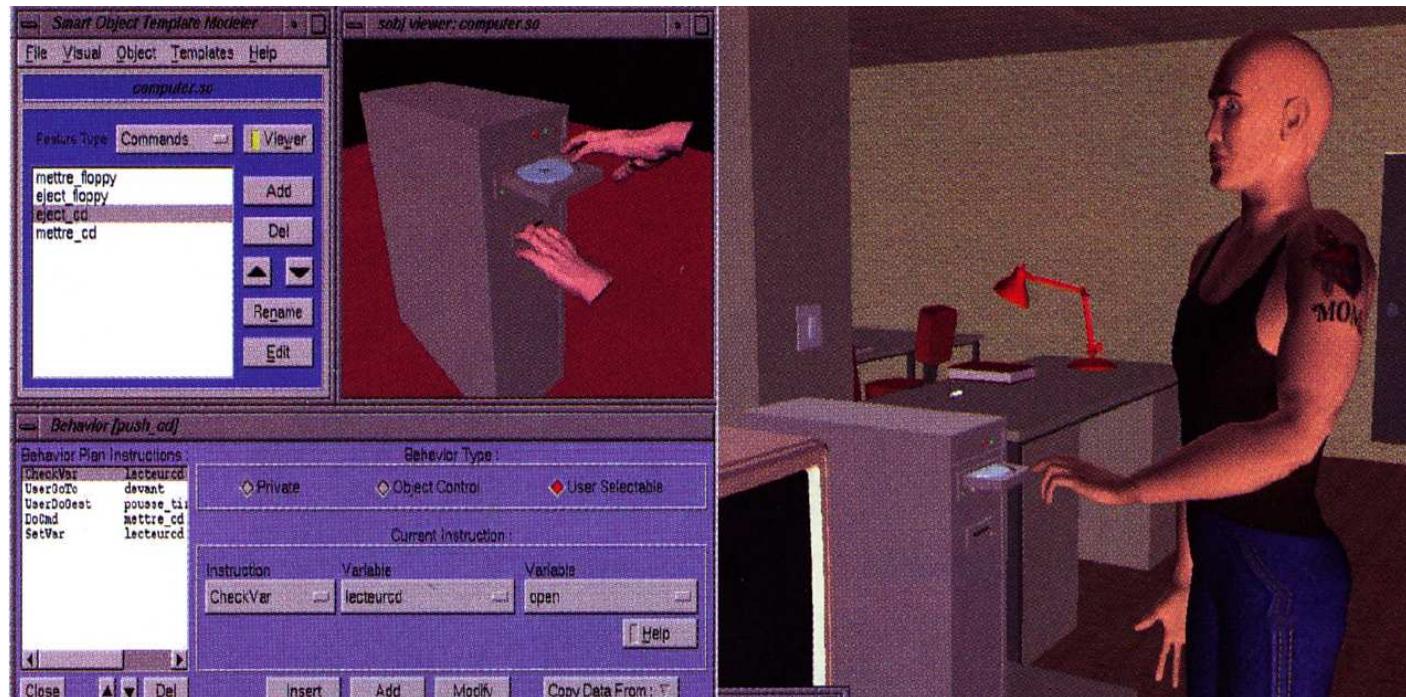
Darüber hinaus müssen mögliche Kollisionen mit Objekten identifiziert werden.

Interaktion zwischen virtuellen Menschen und Objekten

Semantische Objektbeschreibung („Smart Objects“):

- Objekte müssen nach der Art ihrer Manipulation klassifiziert werden
- handhabungsbezogene Objekteigenschaften sind zu modellieren; d.h. neben der Formbeschreibung enthält die Objektbeschreibung semantische Information darüber
 - wo und wie das Objekt für eine sinnvolle Manipulation angefasst wird,
 - wie sich das Objekt bei der Manipulation verhält (z.B. wie sich eine Tür öffnet oder wie sich ein weicher Gegenstand bei Berührung verformt)
 - ggf. welche Teile des Objekts im Zuge der Handhabung angesehen werden müssen, etc.

Interaktion zwischen virtuellen Menschen und Objekten



(a) Definition of smart object

(b) Visualization

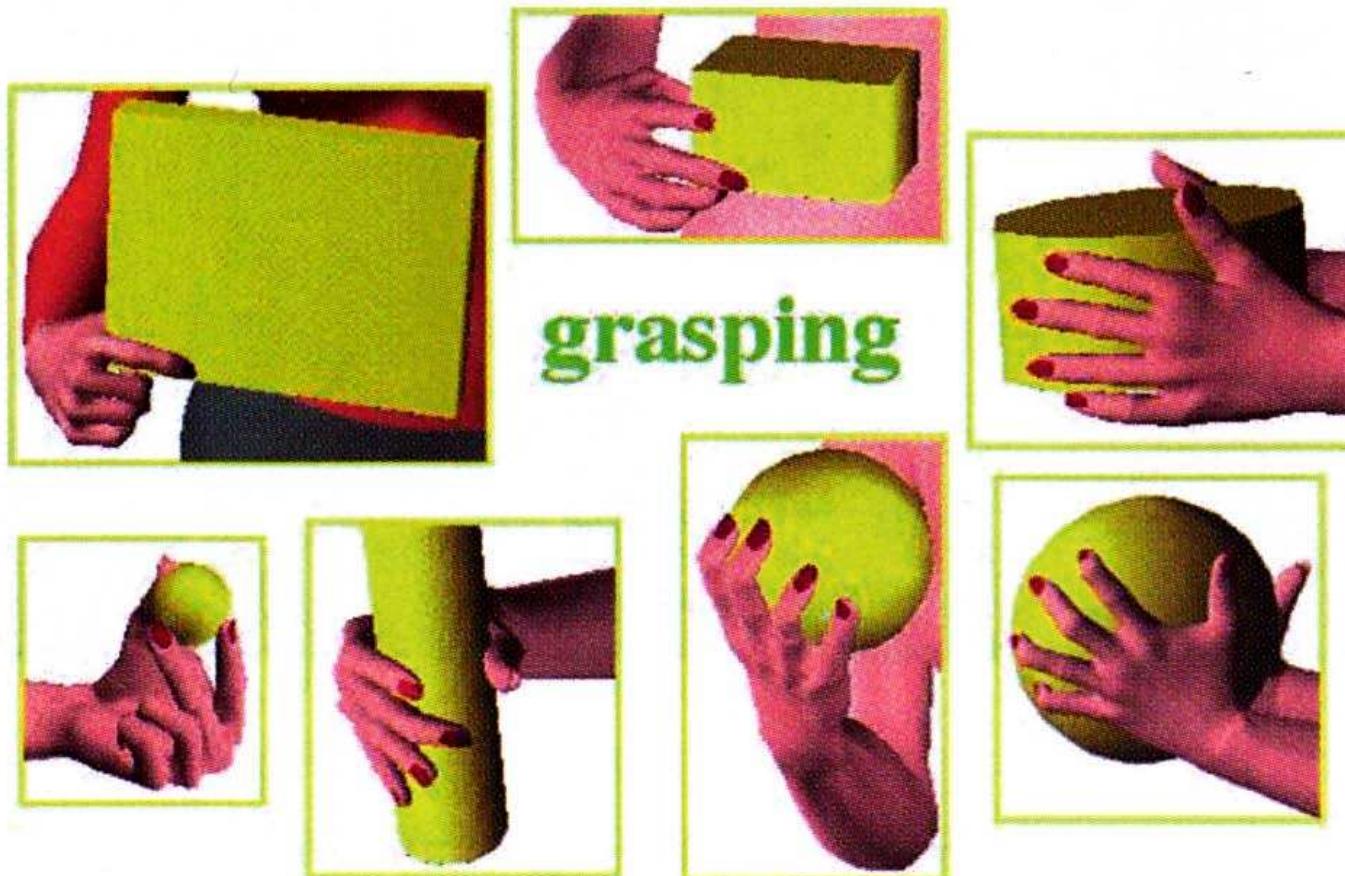
Definition und Verwendung von „Smart Objects“

Interaktion zwischen virtuellen Menschen und Objekten

Modellierung des Greifens beruht auf

- Systematik des Greifens: welcher Gegenstand kann wie gegiffen werden (benötigt man evtl. ein Werkzeug, z.B. eine Pinzette)
- Inverse Kinematik, um die endgültige Handposition zu erreichen (bei einer sinnvollen Körperhaltung!)
- „Handsensoren“, die „fühlen“, wenn die virtuelle Hand das Objekt berührt (damit sie nicht hineingreift oder es bewegt, ohne es zu berühren) – realisiert z.B. durch kugelförmige Kollisionsdetektoren an der Berührungs punkten der Hand

Interaktion zwischen virtuellen Menschen und Objekten



grasping

Systematik des Greifens

Interaktion zwischen virtuellen Menschen und Objekten

Greifen ist i.d.R. Bestandteil von komplexen Bewegungsabläufen

- Darsteller muss zum Objekt gehen, das er ergreifen soll
- Darsteller muss das Objekt ansehen
- Darsteller muss das Objekt ergreifen
- Darsteller muss das Objekt ggf. fortbewegen
- Darsteller darf in dem Prozess nicht mit Objekten in der Szene zusammenstoßen (Kollisionsdetektion erforderlich)

Lösungsansatz: Bewegungsplanung m.H. probabilistischer Landkarten
(probabilistic roadmaps)

Interaktion zwischen virtuellen Menschen und Objekten

Probabilistische Landkarten (probabilistic roadmaps)

- bilden den Zusammenhang (die Konnektivität) des Suchraumes in einem Graphen ab
- Knoten im Graphen repräsentieren beliebige Darsteller-Positionen
- Kanten zwischen Knoten bedeuten, dass sich der Darsteller von der einen Position in die andere bewegen kann
- Roadmap-Techniken werden oftmals zusammen mit Motion-Capture-Daten eingesetzt
- die generierten Bewegungsabläufe sind kollisionsfrei, ausgewogen und sie halten Bewegungsgrenzen ein
- Bewegungsplanung wird sehr komplex, wenn Szene neben statischen auch bewegte Objekte (z.B. andere virtuelle Darsteller) enthält

Interaktion zwischen virtuellen Menschen und Objekten



Beispiele für komplexe Greifaktionen

Autonome virtuelle Darsteller

- wozu autonome virtuelle Darsteller?
- Eigenschaften autonomer virtueller Darsteller
- Verhalten von autonomen virtuellen Darstellern
- Simulation von Menschenmengen

wozu autonome virtuelle Darsteller

- Außer in Filmen und Spielen werden virtuelle Darsteller in Simulationen eingesetzt, in denen sie autonom (d.h. selbstbestimmt, nicht nach vorgegebenem Skript) menschliche Rollen spielen, z.B. in der Ausbildung/Schulung oder der Psychotherapie.
- In solchen Situationen müssen autonome virtuelle Darsteller (AVC: Autonomous Virtual Character) Veränderungen in der virtuellen Umgebung wahrnehmen, eine entsprechende Aktionsentscheidung treffen und angemessen reagieren.
- Hauptproblem: die jeweilige Aktionsentscheidung sollte zum einen den AVC seinem aktuellen Ziel näher bringen, sie muss aber auch langfristige Zielvorgaben berücksichtigen.
- AVCs verändern ggf. auch selbst die Umgebung (s. Interaktion)

Eigenschaften autonomer virtueller Darsteller

- **Wahrnehmungsvermögen**

- ermöglicht AVCs, Veränderungen in der Umgebung zu registrieren
- muss möglichst realistisch modelliert werden, um glaubwürdig zu erscheinen
- sollte daher die Funktionalität menschlicher Sinnesorgane simulieren (z.B. auch den Informationsfluss aus der Umgebung filtern)
- muss dem biologischen Vorbild aber nicht exakt entsprechen, z.B.
 - Sehen: Z-Puffer-Farbbild im Umfang des Gesichtsfeldes des AVC
 - Hören: Funktion, die aktuelle Klangereignisse zurückliefert
 - Tasten/Riechen: Funktion, die das globale Kraftfeld in der Nähe des entsprechenden „Sensors“ auswertet

Eigenschaften autonomer virtueller Darsteller

- **Auffassungs- und Anpassungsvermögen**

- Auffassungs- und Anpassungsvermögen bestimmen, wie ein AVC darüber „nachdenken“ kann, was er wahrnimmt (insbesondere bei unvorhersehbaren Ereignissen)
- um in seiner Umgebung überstehen und seine Ziele erreichen zu können, sollte ein AVC immer die bestmögliche Aktion durchführen
- auf Veränderungen der Umgebung sollte ein AVC dynamisch reagieren können, damit seine Annahmen und Ziele sich mit der Zeit entwickeln können
- für die Simulation von großen AVC-Gruppen können bottom-up Artificial-Life-Techniken verwendet werden (im Gegensatz zu top-down Plan basierten Ansätzen der KI) –
damit können sich neue, ungeplante Verhaltensweisen entwickeln

Eigenschaften autonomer virtueller Darsteller

- **Erinnerungsvermögen**

- beim Auftreten ähnlicher Situationen sollte sich ein AVC ähnlich verhalten wie vorher
- um glaubwürdig zu sein, sollte auch das Gedächtnis eines AVC nicht „perfekt“ sein („nobody is perfect“)

- **Emotionalität**

- Emotion ist eine gefühlsmäßige Reaktion auf eine Wahrnehmung
- der Ausdruck von Emotion durch Gesichtsausdruck und/oder Gestik macht einen AVC glaubwürdiger
- emotionales Verhalten von AVCs beeinflusst auch den emotionalen Zustand der Nutzer von VR-Systemen
- wichtige Aufgabe: Entwicklung eines guten Computermodells emotionalen Verhaltens

Verhalten von autonomen virtuellen Darstellern

- **Reflexe**

- Reflexe sind spontanes Verhalten als Reaktion auf externe Stimuli (z.B. als Schutzmechanismen)
- Reflexbewegungen beruhen auf unterschiedlichen Faktoren wie Geschlecht, Alter, körperliche und psychische Verfassung, etc.
- Reflex-Algorithmen verarbeiten neben Parametern, die diese Faktoren repräsentieren, die Eigenschaften der Stimuli und setzen sie in angemessene, der Umgebung angepasste Bewegungsabläufe um

Verhalten von autonomen virtuellen Darstellern

- **Motivation**

- Motivation ist ein Auslöser für Aktivitäten bzw. Interaktionen mit der Umgebung
- Motivation basiert auf natürlichen Bedürfnissen (Kommunikationsbedürfnis, Hunger, Durst, etc.)
- sinnvolle Handlungsabläufe müssen modelliert werden, die zur Befriedigung des jeweiligen Bedürfnisses führen
- auszuführende Handlungen ergeben sich aus der jeweiligen Motivation und aus Informationen, die in der Umgebung wahrgenommen werden

Verhalten von autonomen virtuellen Darstellern

- **soziales Verhalten**

- soziales Verhalten ist geprägt durch die Beziehungen, die die Akteure zueinander haben
- es äußert sich in verbaler und nicht-verbaler Kommunikation
- Grundlagen von Gruppenverhalten sind:
 - Macht/Einfluss (Dominanz und Unterwürfigkeit),
 - Attraktion (Freundlichkeit und Unfreundlichkeit),
 - Rangstellung,
 - emotionales Ausdrucksvermögen
- Bildung, Geschlecht, Alter kennzeichnen die Rolle innerhalb der Gruppe

Simulation von Menschenmengen

- im Gegensatz zur Simulation von Einzelpersonen erfordert die Simulation von Menschenmengen in Echtzeit auf jeder Ebene eine effiziente Vielfalt:
- Menschen in einer Menge sehen anders aus, bewegen sich anders, reagieren anders, klingen anders, etc. als Einzelpersonen
- auch in Menschenmengen müssen Einzelpersonen individuelle Eigenschaften (Aussehen, Verhalten, etc.) aufweisen
- jede Person verfolgt ein individuelles Ziel, versucht es auf kürzestem Wege zu erreichen, muss Hindernissen ausweichen
- Verhalten hängt von der Dichte der Menschenmenge ab
- es muss u.a. zwischen planmäßigem und chaotischem Verhalten (Panik) unterschieden werden

Simulation von Menschenmengen

- Bewegungsabläufe können in zwei Teilen geplant werden:
 - **Wegplanung:** welches ist der beste Weg zum Ziel?
Entscheidungskriterien: Vermeidung verstopfter Bereiche,
Verringerung des Abstandes zum Ziel sowie der Zeit es zu erreichen,
Anzahl möglicher Wege muss hinreichend groß sein, damit sich viele
Individuen gleichzeitig bewegen können
 - **Hindernisvermeidung:** Gegenständen oder anderen Akteuren
ausweichen
bei kleinen Mengen werden Agenten basierte Methoden für realistische
Fußgängerplanung eingesetzt (aufwendig bei großen Mengen),
Potentialfeld-Ansätze dienen dazu Kollisionen auf lange wie auf kurze
Sicht zu vermeiden (weniger realistisch)
 - bei der grafischen Darstellung werden LOD-Techniken eingesetzt

Simulation von Menschenmengen



Simulation von Menschenmengen (crowd simulation)

4. Hardware für die Interaktion mit virtuellen Welten

Wahrnehmung von und Interaktion mit virtuellen Welten

- Sehen (visuelle Wahrnehmung)
- Hören (akustische Wahrnehmung)
- Fühlen (haptische Wahrnehmung)
- Geruch (olfaktorische Wahrnehmung)
und Geschmack

speziell für Eingaben

- Bewegungserfassung
- Brain Computer Interface

Visuelle Wahrnehmung in virtuellen Welten

Übliche Displaytechniken sind

- Kathodenstrahlröhren (CRT: Cathode Ray Tubes)
immer noch gut geeignet für VR-Anwendungen wegen hoher Auflösung und kurzer Antwortzeiten – Nachteil: klobige Geräte
- Flüssigkristall-Bildschirme (LCD: Liquid Crystal Displays)
Passivmatrixsysteme wegen langsamer Antwortzeiten nicht gut geeignet
Aktivmatrixsysteme (TFT-Bildschirme) sind gut geeignet für VR-Anwendungen wegen hoher Auflösung, kurzer Antwortzeiten und ziemlich flimmerfreier Darstellung
Nachteil: (noch) eingeschränkte Betrachtungswinkel

Visuelle Wahrnehmung in virtuellen Welten

Übliche Displaytechniken (Forts.)

- Plasmabildschirme (Plasma Displays)
gut geeignet für VR-Anwendungen wegen hoher Auflösung, kurzer Antwortzeiten, großer Helligkeit, nahezu beliebiger Betrachtungswinkel und großer Bildschirmfläche (Diagonalen über 100 Zoll),
ihre Bildqualität ist der von guten CRT-Bildschirmen vergleichbar
- Projektoren („Beamer“)

Visuelle Wahrnehmung in virtuellen Welten

In VR-Systeme werden eingesetzt

- Stereo-Displays (Head-Mounted Displays)
meist mit LCD-Technik
- Aquarium-VR (Fish Tank VR)
mit CRT-, LCD- oder Plasmabildschirmen
- Taschenbildschirme (Hand Holds)
meist mit LCD-Technik
- große Projektionsschirme – häufig Plasmabildschirme
- Shutter Brillen
(in Verbindung mit Bildschirmen oder Projektionsschirmen)
- CAVE-Systeme
meist mit Beamern und großen Projektionsflächen

Visuelle Wahrnehmung in virtuellen Welten



Stereo-Display (head mounted display)
und Datenhandschuh (data glove)

Visuelle Wahrnehmung in virtuellen Welten



GeForce Shutter Brille

Akustische Wahrnehmung in virtuellen Welten

Bedeutung von Geräuschen in virtuellen Welten

- akustische Wahrnehmung ist wesentlicher Orientierungskanal
- akustische Wahrnehmung ergänzt visuelle Wahrnehmung
(insbes. bei Sehbehinderung)
- gesprochene Sprache ist wesentliches Kommunikationsmittel
- Musik ist bedeutendes Unterhaltungsmedium
- etc.

Akustische Wahrnehmung in virtuellen Welten

Aufnahme und Wiedergabe von Raumklang

- monaural („einohrig“)
Aufnahme mit einem Mikrofon – Position der Klang- bzw. Geräuschquelle kann nicht identifiziert werden
- binaural („zweiohrig“)
Aufnahme mit zwei Mikrofonen, die in einen Modellkopf integriert sind – „Kunstkopfstereophonie“ wirkt fast wie natürlicher Raumklang
- Stereo
Aufnahme mit zwei Mikrofonen, die einen größeren Abstand voneinander haben – räumlicher Klangeindruck, jedoch nicht wie bei Kunstkopfstereophonie, Klangquelle scheint im Kopf der/des Hörenden zu liegen

Akustische Wahrnehmung in virtuellen Welten

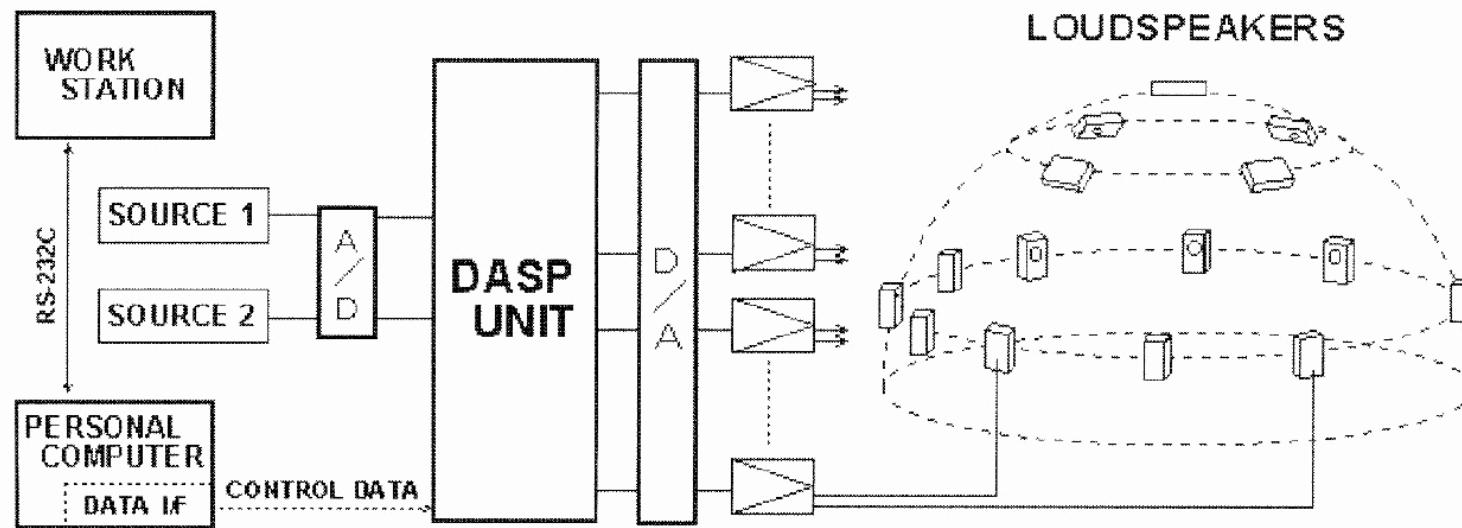
Synthese von Raumklang

- Geräuschquellen können synthetisch oder „natürlich“ sein
- Geräuschquellen senden Schallwellen i.d.R. in alle Richtungen aus
- Schallwellen werden an Objekten reflektiert bzw. gebrochen
- Sound Rendering ist ähnlich wie Ray Tracing

Geräusch-/Klangsysteme für VR

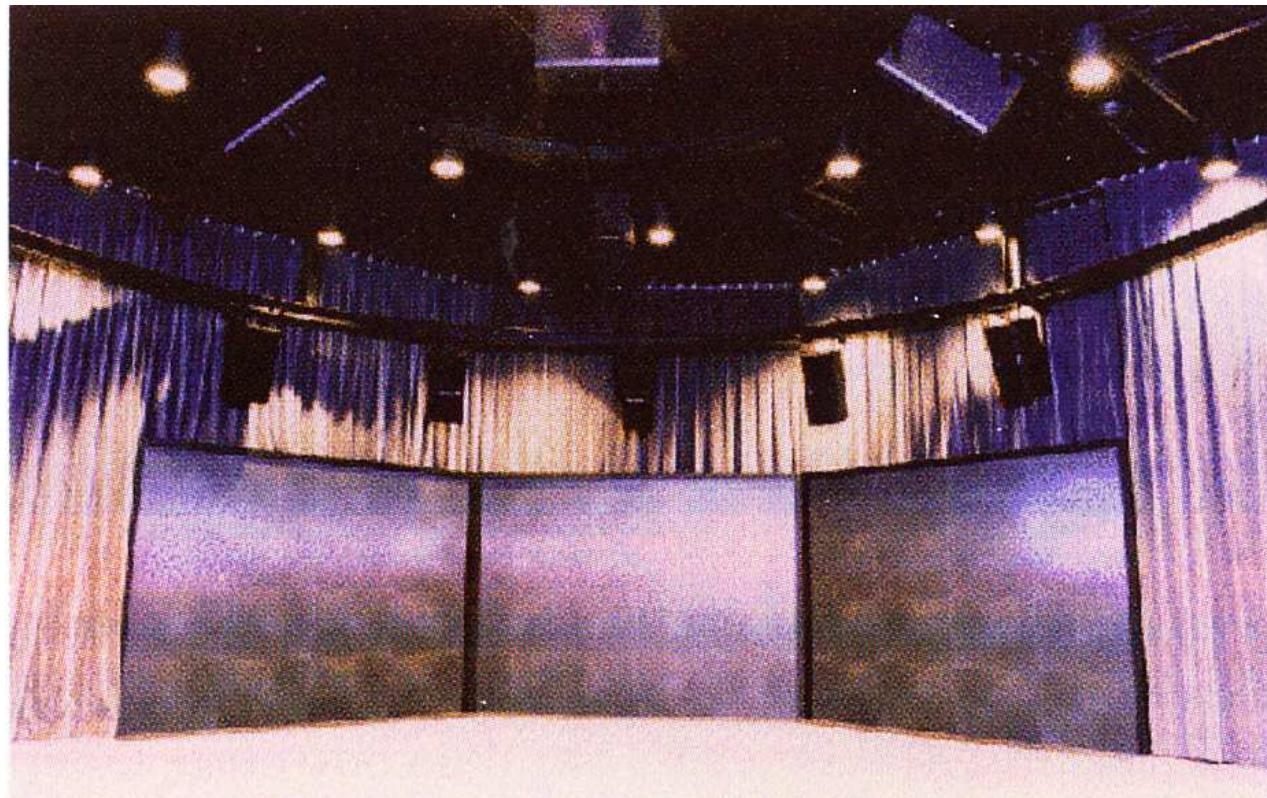
- Ohrhörer / Kopfhörer (geben Raumklang am besten wieder)
- komplexe Lautsprechersysteme

Akustische Wahrnehmung in virtuellen Welten



System zur Raumklangsteuerung (sound field control)

Akustische Wahrnehmung in virtuellen Welten



VR-System mit Lautsprecherausstattung

Haptische Wahrnehmung in virtuellen Welten

- haptisches System dient der Orientierung
- nutzt taktile Information (Berührung)
durch Sensoren in der Haut
- nutzt kinästhetische Information (Bewegungswahrnehmung)
durch Wahrnehmung der Gelenkwinkel, Muskelspannung,
Sehnenbelastung
- haptische Interfaces sind sehr anwendungsorientiert:
Force Feedback vermittel Gefühl für Härte, Gewicht, etc. von
Objekten
und Taktiles Feedback von Oberflächenbeschaffenheit, Form,
Temperatur, etc.

Haptische Wahrnehmung in virtuellen Welten

Haptische Schnittstellen

- Vibrotaktile Displays
bestehen aus einer einzelnen vibrierenden Komponente (z.B. in „Handys“)
- Taktile Displays
stimulieren die Haut, eingesetzt werden elektrotaktile, thermische und mechanische Displays
- Kinästhetische Displays
Haptische Oberflächen (Shape Displays): rechnergesteuerte, verformbare Oberfläche
Single Point Force Feedback: „Joystick mit Kraftrückkopplung“
Exoskelett: Menge von „Stelleinheiten“, die an Körper und/oder Gliedmaßen angebracht sind

Haptische Wahrnehmung in virtuellen Welten



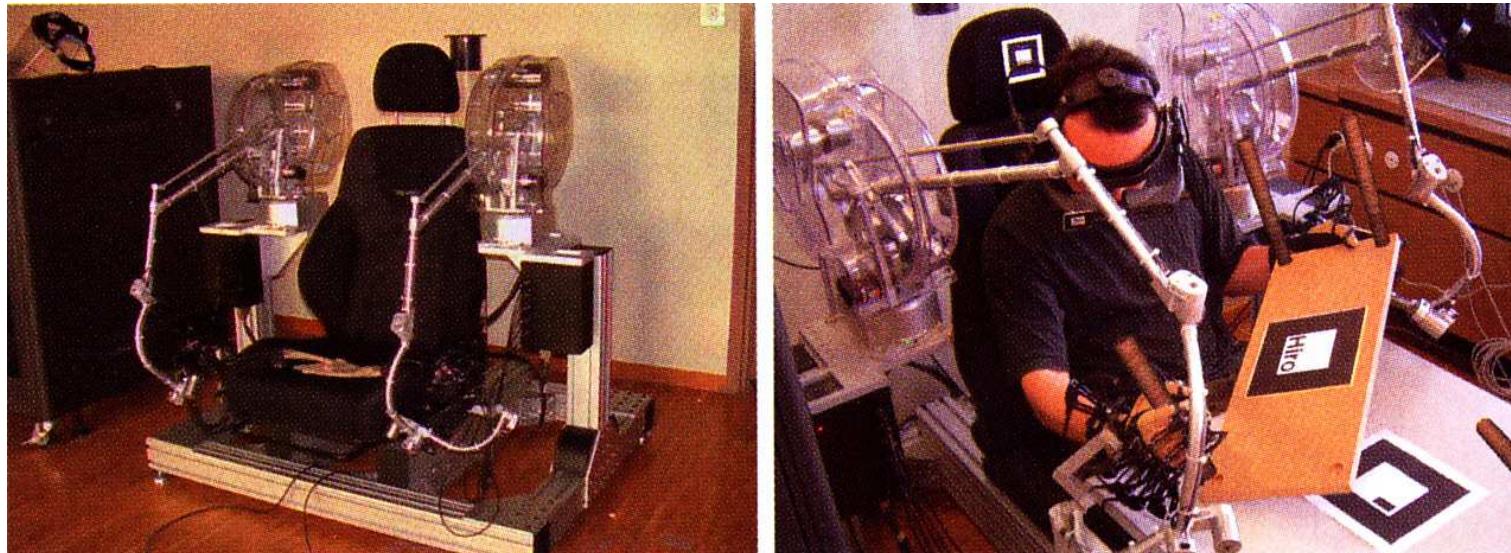
Haptisches Interface: Force-Feedback-Eingabegerät PHANTOM
(Personal HAptic iNterface Mechanism)

Haptische Wahrnehmung in virtuellen Welten



Haptisches Interface: Force-Feedback-Eingabegerät (bit-tech net)

Haptische Wahrnehmung in virtuellen Welten

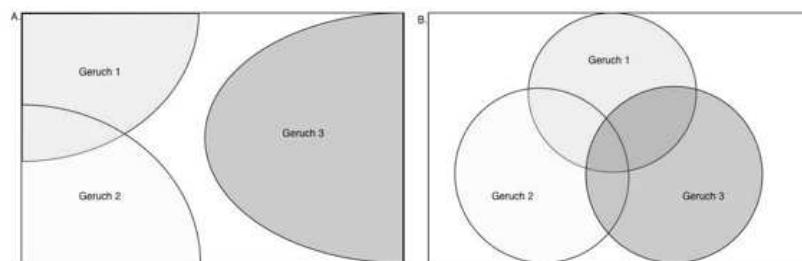


Komplexes Haptisches Interface:
The Haptic Workstation der Fa. Immersion

Geruch und Geschmack

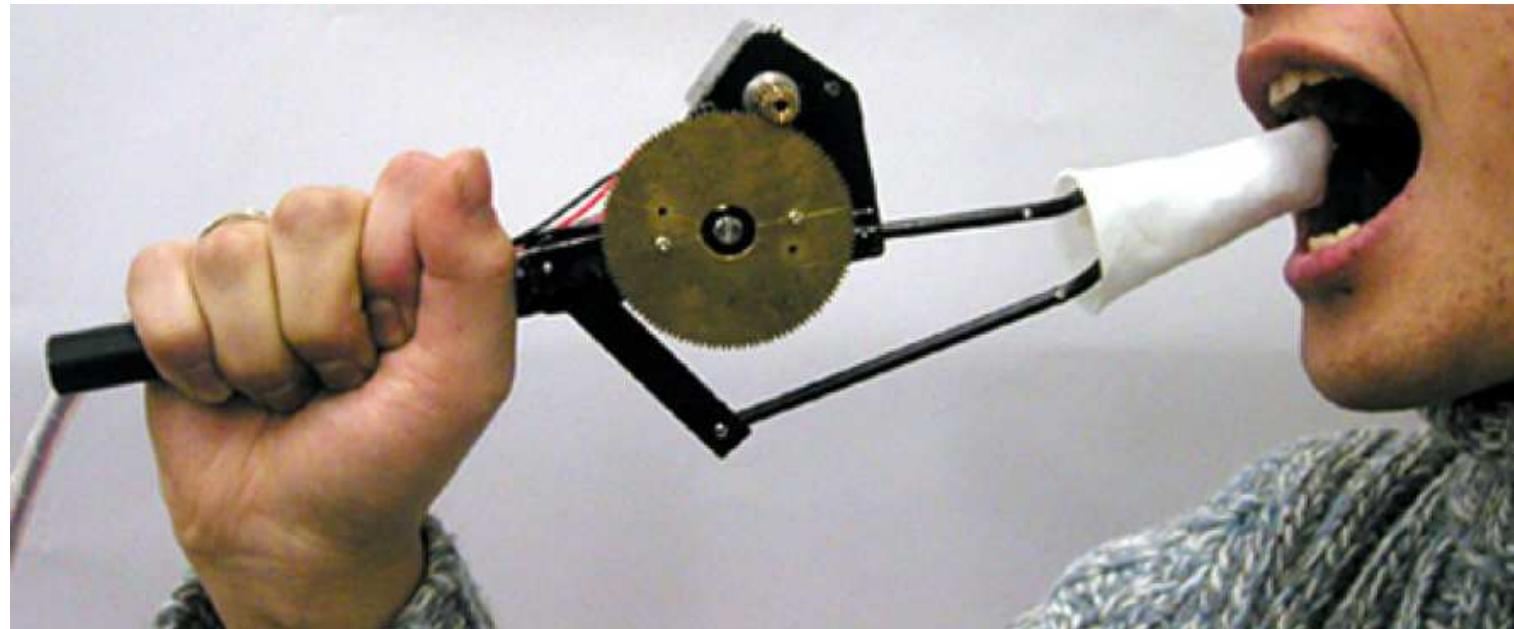
- Geruch und Geschmack dienen der Orientierung
- Geruch und Geschmack hinterlassen oft nachhaltige (emotionale) Erinnerungen
- die Einbeziehung der Geruchs- und Geschmackssinne in VR-Systeme steckt noch in den Anfängen; es gibt Prototypen für
 - Geruchsinterfaces (olfaktorische Interfaces)
 - Geschmacksinterfaces (in Form von Speisen- bzw. Getränkesimulatoren)

Geruch und Geschmack



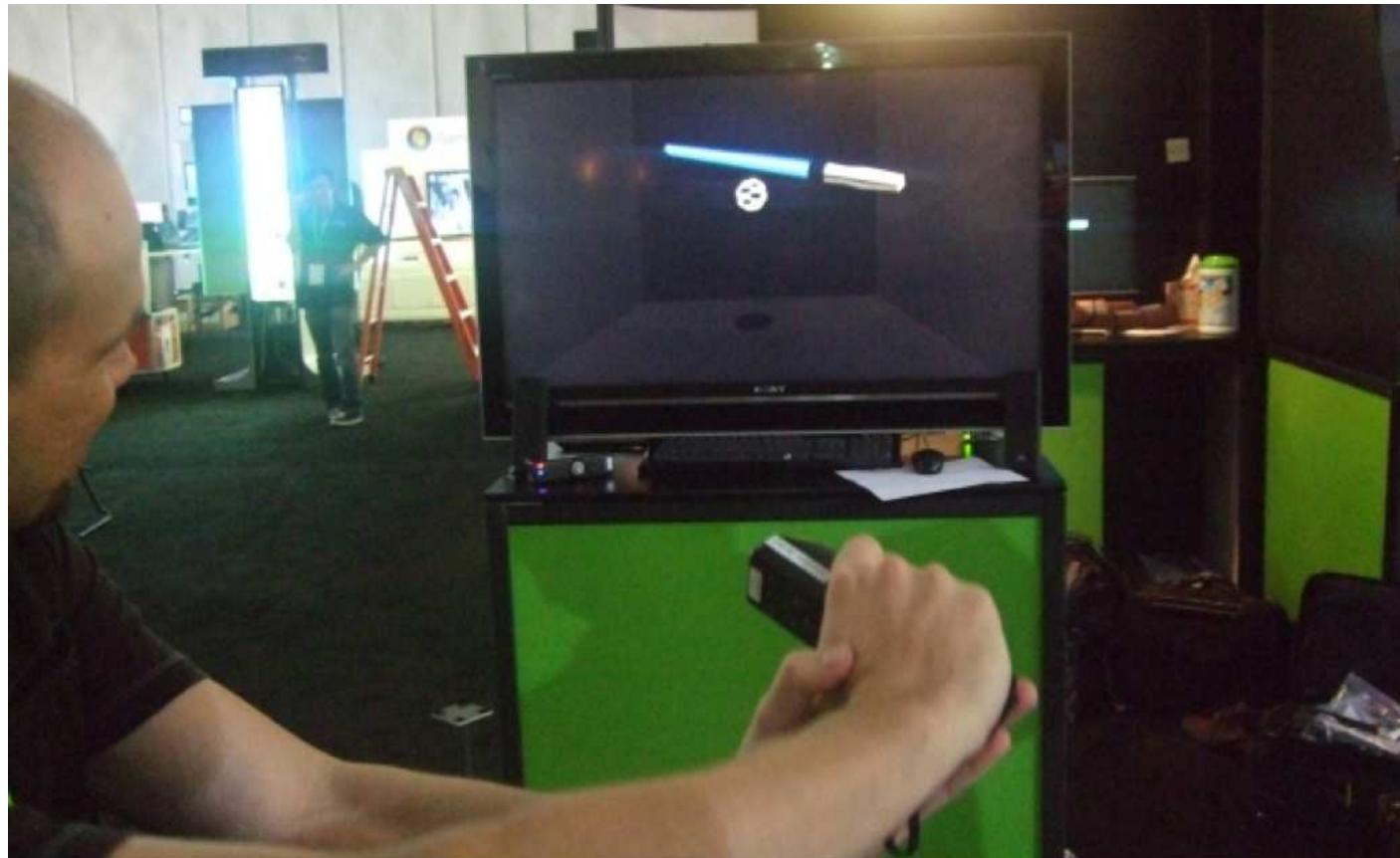
Sniffman – olfaktorisches Interface

Geruch und Geschmack



Geschmacksinterface

Bewegungserfassung



Sixense TrueMotion tracker

BCI: Brain Computer Interface



Schnittstelle zwischen Gehirn und Computer

BCI: Brain Computer Interface

- orientiert an Erkenntnissen aus den Neurowissenschaften
- die Hirnaktivität spiegelt bereits die Vorstellung eines Verhaltens wider
- an der Kopfhaut angebrachte Elektroden messen die hirnelektrischen Signale und übertragen sie an den Computer
- der Computer wandelt sie in Steuerungssignale für das VR-System um

Interactive Visual Computing (IVC)

bzw.

Computergrafik und Bildsynthese (CGB)

(Wintersemester 2010/11)

Leonie Dreschler-Fischer, Susanne Germer & Werner Hansmann

Übungen

Informationen zu den IVC/CGB-Übungen

- Für die IVC/CGB-Übungen stehen folgende Räume zur Verfügung:
immer montags von 10:00 h - 12:00 h:
D-011, D-012, D-013 (MacPool) , D-114, D-118/119 (Linux PCs),
außerdem zusätzliche Termine am 17. und 24. Januar 2011:
D-011, D-012 und D-013 von 12:00 h - 14:00 h
D-114 und D-118/119 von 12:00 h - 14:00 h
sowie am 19. und 26. Januar 2011:
D-011, D-012, D-013 und D-114 von 10:00 h - 12:00 h
- Jede Übungsgruppe erhält für dieses Semester ein gedrucktes Exemplar der „POV-Ray 3.6.1 User's Documentation“.
- Die POV-Ray 3.6.1 Documentation findet sich on-line unter <http://www.povray.org/documentation/>
- Kopien dieser Informationen sind in STiNE abgelegt

IVC/CGB-Übungsaufgabe

- Die Übungsaufgabe sollten Sie in Gruppenarbeit zu zweit oder dritt bearbeiten.
- Bitte tragen Sie sich in die Gruppenliste entsprechend ein.
- Auf den Computern im Informatikrechenzentrum steht Ihnen der Raytracer POV-Ray zur Verfügung, mit dem Sie ausgefeilte photorealistische Animationen erzeugen können.
- Für die Übungen sollten Sie POV-Ray verwenden, auch wenn wir Ihnen in der Vorlesung noch einfachere interaktive Modellierungswerkzeuge wie Maya und Cinema 4D zeigen werden.

Für den Übungsschein ist eine kurze Animationssequenz zu erstellen – das Thema ist Ihnen freigestellt – und zu dokumentieren, die folgende Elemente enthalten sollte:

1. Wenigstens ein Objekt in der Szene sollte bewegte Gliedmaßen haben.
2. Wenigstens eine Szene sollte die Kameraeinstellung variieren, z.B. in die Szene hineinfahren, schwenken oder zoomen.
3. Setzen Sie an wenigstens einer Stelle ein verarbeitetes Bild ein, z.B. als Höhenprofil, Kulisse oder Textur.
4. Erstellen sie einen animierten Titel.
5. Setzen Sie Überblendungen ein, um den Schnitt zwischen zwei Kamera-positionen oder anderen Bildwechseln zu betonen oder zu kaschieren. Schneiden können Sie Ihren Film am Video-Mac im KOGS-Labor (R131), bzw. auf den iMacs des Informatikrechenzentrums. Mit dem Programm Final Cut Pro bzw. iMovie-HD können Sie hier bzw. im „Musikstudio“ (R122) Ihr Meisterwerk vollenden.

Für den Übungsschein **abzugeben** sind
(versehen mit Ihren Namen und Matrikelnummern):

Bis zum 31. Januar 2011: -

- Eine CD mit dem fertigen Film (Quicktime-Movie oder Motion-JPG) und den POV-Ray-Skripten.
- Ein Poster für die Fachbereichs-EXPO.

Bis zum 02. Februar 2011: -

- Eine Dokumentation, in der Sie die Verwendung der eingesetzten Modellierungswerkzeuge, Transformationen usw. erläutern.

Bitte achten Sie bei den verwendeten Medien (Musik, Bilder) darauf, dass Sie keine Urheberrechte verletzen, denn wir möchten Ihr Werk gerne öffentlich aufführen können.

IVC/CGB-Übungsaufgabe

Themenvorschlag aus der *School of Life Science*:

Stoffwechsel in einer Zelle – oder anders formuliert:

Wie gewinnen wir aus Schokolade, Eiern bzw. Eisbein
(also aus Zucker, Eiweiß und Fett)

Energie und warum müssen wir dazu atmen?

Dazu könnte man die „Stoffwechsel-Autobahnen“ in einer (menschlichen) Zelle kreativ und lustig darstellen und dabei trotzdem eine „wahre“ Geschichte erzählen.

(Der Film soll natürlich hinterher auch den Schülerinnen und Schülern der School of Life Science gezeigt werden und wenn er gut ist, jedes Jahr in jedem neuen Jahrgang).

Ansprechpartner:

Werner Hansmann (FBI) und Stefan Marotzki (School of Life Science)

Informationen zur Nutzung von POV-Ray

Kopieren Sie aus STiNE die Datei CSG-demos.pov in Ihr Verzeichnis und teilen Sie die darin enthaltenen Beispiele wie folgt in einzelne Dateien:

- CSG-demo0.pov
- CSG-demo1.pov
- CSG-demo2.pov
- ..
- CSG-demo8.pov

Informationen zur Nutzung von POV-Ray (Forts.)

Arbeitsschritte zur Bildgenerierung unter **unix/linux**:

- Erzeugen Sie ein Bild von der in der Datei CSG-demo0.pov gespeicherten Szenenbeschreibung mit der Befehlsfolge

povray -I IVC-demo0.pov +W1024 +H716

Das Ergebnis wird automatisch in der Datei IVC-demo0.png abgelegt.
- Stellen Sie den Inhalt der Datei IVC-demo0.png mit Hilfe des Befehls xv IVC-demo0.png auf dem Bildschirm dar.
- Löschen Sie die Datei IVC-demo0.png wieder.

Informationen zur Nutzung von POV-Ray (Forts.)

Arbeitsschritte zur Generierung einer Bildsequenz (Animation):

- Erzeugen Sie eine Bildsequenz von der in der Datei IVC-demo2.pov gespeicherten Szenenbeschreibung mit povray -I IVC-demo2.pov +W512 +H400 +KFI0 +KFF11
Das Ergebnis wird automatisch in den Dateien IVC-demo200.png, IVC-demo201.png, ... , IVC-demo211.png abgelegt.
- Stellen Sie den Inhalt dieser Dateien mit Hilfe des Befehls animate CSG-demo2*.png auf dem Bildschirm dar.
- Drücken Sie die linke Maustaste und probieren Sie die im Pop-Up-Menü erscheinenden Befehle aus.
- Löschen Sie die Dateien CSG-demo2*.png wieder.

Informationen zur Nutzung von POV-Ray (Forts.)

Arbeitsschritte zur Bildgenerierung mit **MacMegaPOV**:

- unter *Programme* *MegaPov* starten
- in *MacMegaPOV* unter *File* *IVC-demo2.pov* aus Ihrem Verzeichnis laden
- in *MacMegaPOV* unter *Render* die Bildgenerierung starten
- ...

Zugriff vom Mac auf Verzeichnisse im Informatik-Netz:

im *Finder*, Pull-Down-Menü *Gehe zu* den Unterpunkt *Mit Server verbinden* wählen und im folgenden Pop-Up-Menü als Server- Adresse `smb://rzfilesrv1/infhome` eingeben, dort das gewünschte Verzeichnis aufsuchen