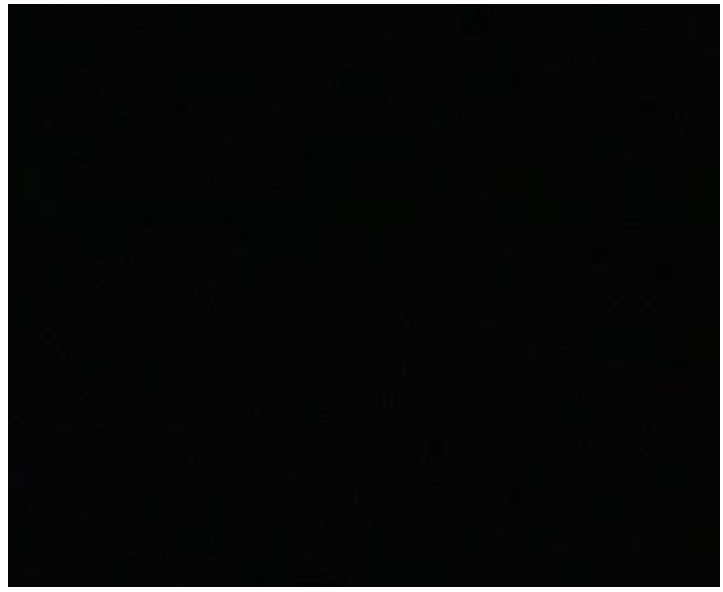
Verdeckungsrechnung 2

Bisher

- Back-face-Culling
- Min/Max Test
- Painters-Algorithm
- Z-Buffer
- Scan-Line-Algorithm (Watkins)
- A-Buffer

Was heißt schon effizient?



Was heißt schon effizient?

Es muß ganz klar unterschieden werden:

Wird eine Szene nur einmal gerendert,

oder

müssen permanent neue Ansichten der gleichen Szene erzeugt werden.

Bisherige Überlegungen (letzte Vorlesung) haben mehrmaliges Rendern nicht berücksichtigt.

Ein idealer Algorithmus müßte ...?

6

A) verdeckte Teile schnell aussortieren.

B) die zeitliche und räumliche Kohärenz nutzen.

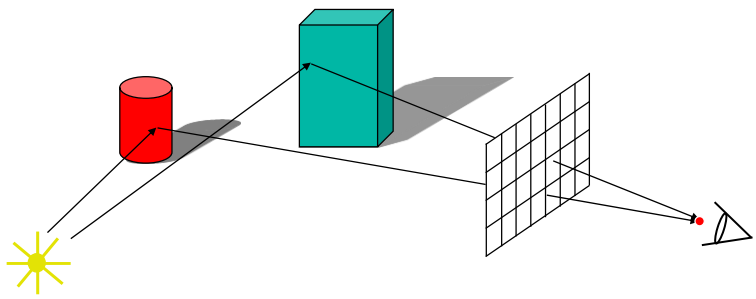
z.B. Z-Buffer: A - - -

B + +

Wir benötigen eine Repräsentation der räumlichen Organisation von Szenen!

Eine Motivation: Rendern durch „Ray Casting“

7



Die Farbe eines jeden Pixels in der Bildebene ist hängt von der Abstrahlung der sichtbaren Oberfläche ab.

„Ray Casting“

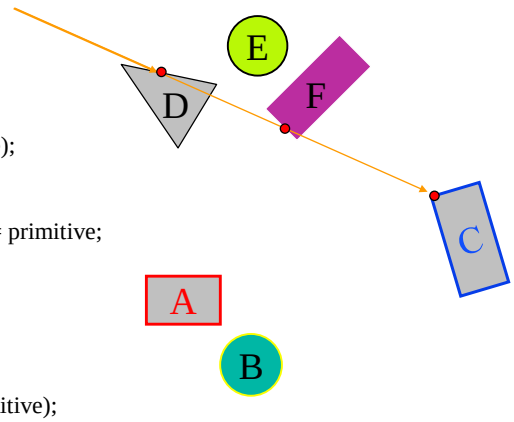
- Einfache Implementierung:

```
Image RayCast ( Camera camera, Scene scene, int wdth, int height)
 {
        Image image = New Image(width , heigth)
        for (int i  = 0; i < width; i++) {
                for (int j  = 0; j < heigth; j++) {
                        Ray ray = ConstructRayThroughPixel(camera, i,j );
                        Intersection hit = FindIntersection(ray,scene);
                        image[i][j] = GetColor(hit);
                 }
         }
        return image;
}
```

Strahl - Szene - Schnittpunkt

- Suche den vordersten Schnittpunkt mit allen Objekten (Dreiecke)

```
Intersection FindIntersection ( Ray ray, Scene scene)
 {
        min_t = infinity;
        min_primitive = Null;
        for each primitive in scene {
                t = Intersect(ray, primitive);
                if ( t < min_t ) then {
                        min_primitive = primitive;
                        min_t  = t;
                 }
         }
        return Intersection(min_t, min_primitive);
}
```



Strahl - Szene - Schnittpunkt

- Schnittpunkt mit geometrischen Grundbausteinen
  - Kugel
  - Dreieck
  - Gruppen von Bausteinen

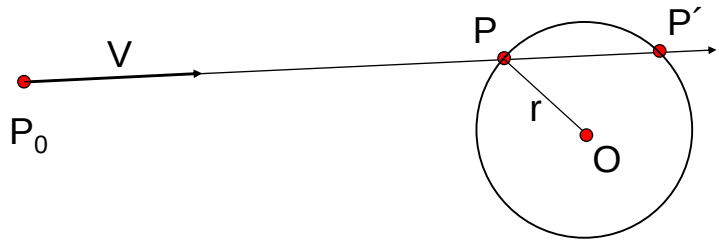
>> Beschleunigungsverfahren

- Hierarchie von Begrenzungsvolumina
- Raumaufteilung
  - gleichmäßige Raster
  - BSP trees
  - Octrees

Schnitt von Strahl mit Kugel

Ray: $P = P_0 + t\,V$

Kugel: $|P - O|^2 - r^2 = 0$



Schnitt von Strahl mit Kugel

Ray: $P = P_0 + t\,V$

Kugel: $| P - O |^2 - r^2 = 0$

Algebraische Methode

P ersetzen:

$$| \mathbf{P_0 + t\,V} - O |^2 - r^2 = 0$$

Löse Quadratischegeleichung

$$at^2 + bt + c = 0$$

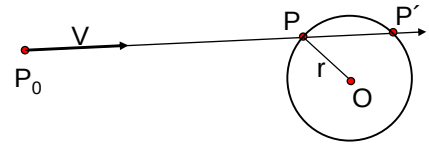
mit

$a = 1$

$b = 2V (P_0 - O)$

$c = | P_0 - O |^2 - r^2$

=> $P = P_0 + t\,V$



Schnitt von Strahl mit Kugel

Ray: $P = P_0 + t\,V$

Kugel: $| P - O |^2 - r^2 = 0$

Geometrische Methode

$L = O - P_0$

$t_{ca} = L \bullet V$

If ($t_{ca} < 0$) return 0

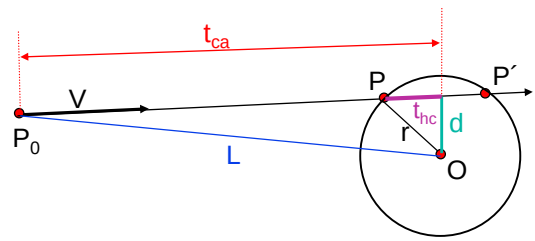
$d^2 = L \bullet L - t_{ca}^2$

if ($d^2 > r^2$) return 0

$t_{hc} = \mathrm{sqrt}(r^2 - d^2)$

$t = t_{ca} - t_{hc}$ und $t_{ca} + t_{hc}$

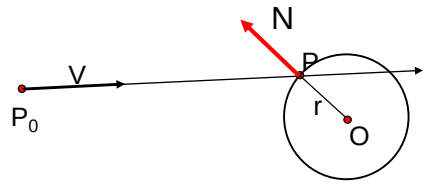
=> $P = P_0 + t\,V$



Schnitt von Strahl mit Kugel

- Normalen am Schnittpunkt werden für die Beleuchtung benötigt

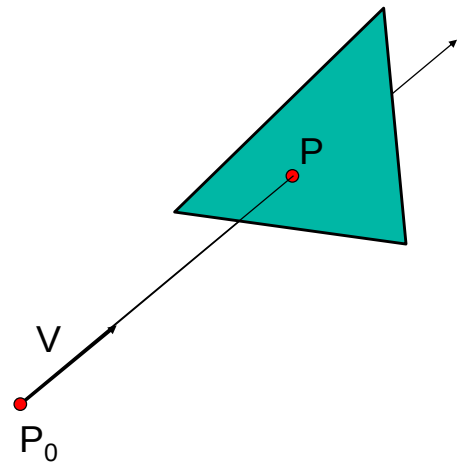
$$N = (P - O) / \| P - O \|$$



Schnitt von Strahl mit Dreieck

1. Schneide Strahl mit Ebene
2. Prüfe, ob Schnittpunkt innerhalb des Dreiecks liegt

Strahl: $P = P_0 + t\,V$



Schnitt von Strahl mit Ebene

Strahl: $P = P_0 + t\,V$

Ebene: $P \cdot N + d = 0$

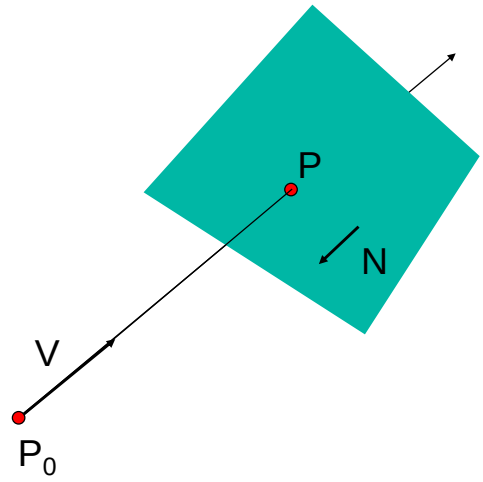
P einsetzen:

$$(\mathbf{P_0} + t\,\mathbf{V}) \cdot N + d = 0$$

Lösung

$$t = -(\mathbf{P_0} \cdot N + d) / (V \cdot N)$$

$$P = P_0 + t\,V$$



Schnitt von Strahl mit Dreieck I

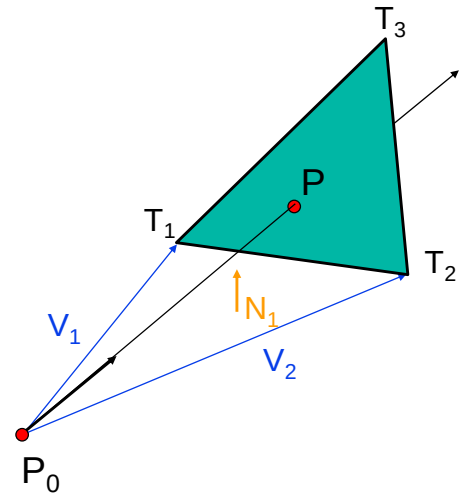
- Prüfe, ob Schnittpunkt innerhalb des Dreiecks liegt

For each side of triangle

$V_1 = T_1 - P_o$
$V_2 = T_2 - P_o$
$N_1 = V_2 \times V_1$
(normiere N_1)

if ($(P - P_o) \cdot N_1 < 0$)
return FALSE;

end;



Schnitt von Strahl mit Dreieck II

- Prüfe, ob Schnittpunkt innerhalb des Dreiecks liegt

Strahl: $P = P_0 + t\,V$

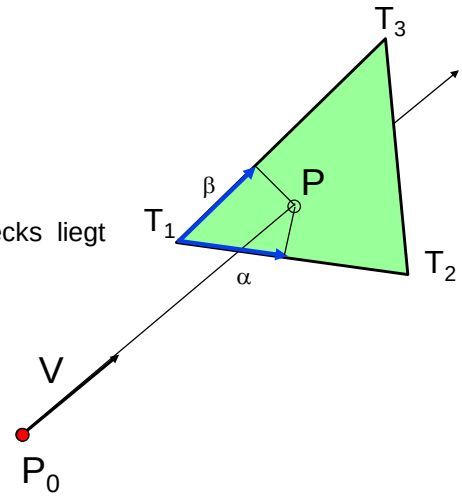
Berechne α, β:

$P = \alpha\,(T2-T1) + \beta\,(T3-T1)$

Prüfe ob der Punkt innerhalb des Dreiecks liegt

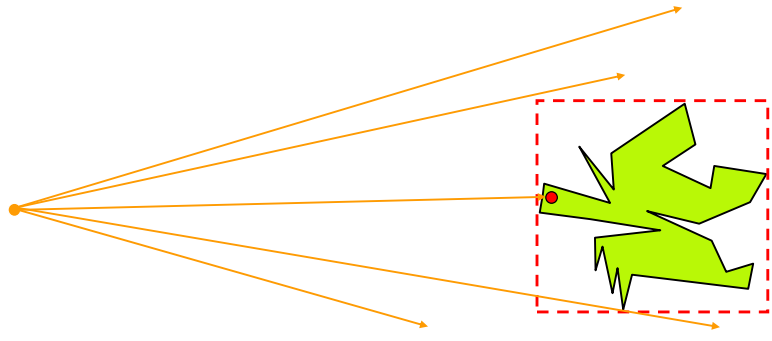
$0 \le \alpha \le 1$ und $0 \le \beta \le 1$

$\alpha + \beta \le 1$



Begrenzungsvolumen

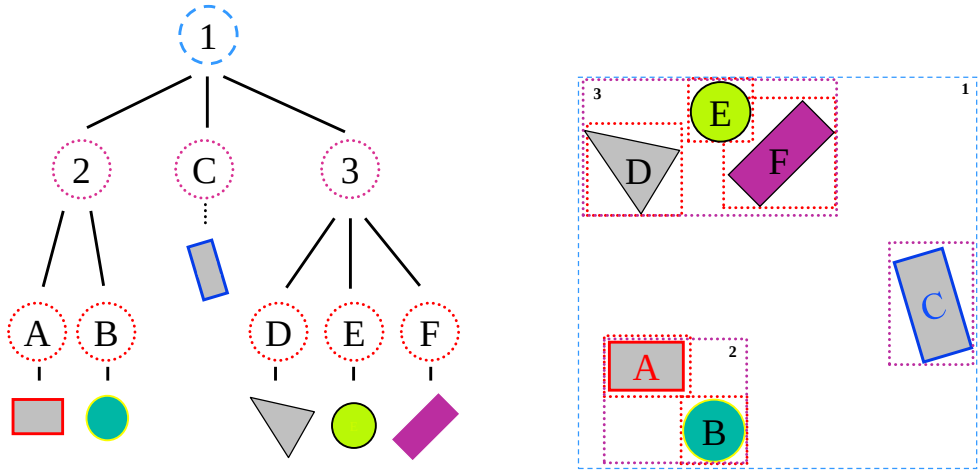
- Bestimme zuerst den Schnittpunkt mit einfachen Formen
  - Wenn der Strahl das Begrenzungsvolumen nicht schneidet, dann auch nicht seinen Inhalt!



Hierarchie von Begrenzungsvolumina

21

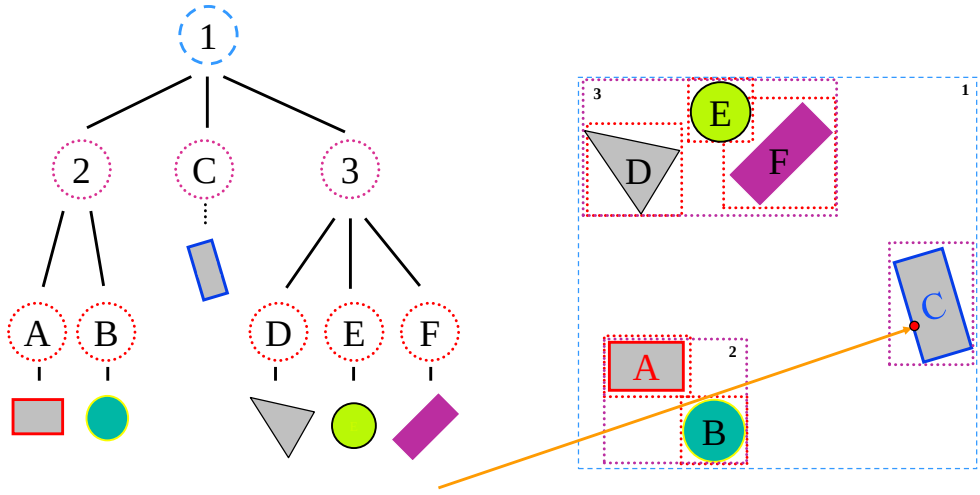
- Erzeuge ein Hierarchie von Volumina
  - Volumenknoten enthält alle Kindervolumina



Hierarchie von Begrenzungsvolumina

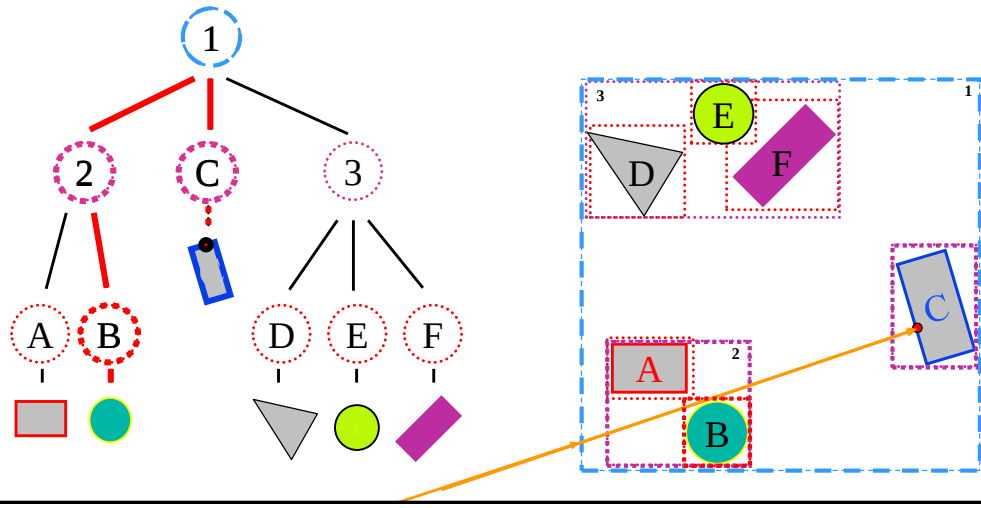
22

- Nutze die Hierarchie zur Beschleunigung des Strahl-Szene-Schnitts
  - Schnitt mit Knoteninhalt nur wenn Begrenzung geschnitten wird



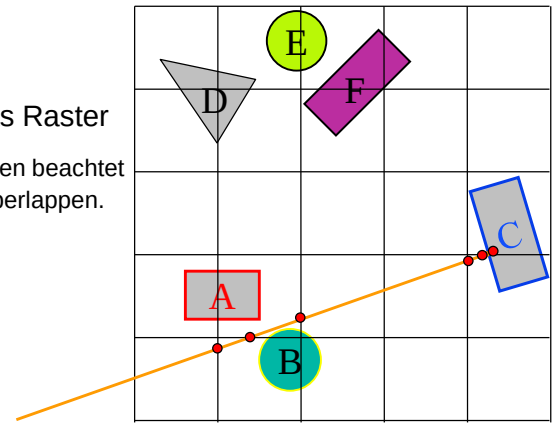
Hierarchie von Begrenzungsvolumina

- Nutze die Hierarchie zur Beschleunigung des Strahl-Szene-Schnitts
  - Schnitt mit Knoteninhalt nur wenn Begrenzung geschnitten wird



Gleichförmiges Raster

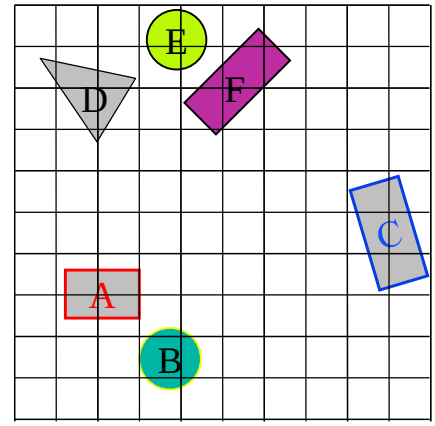
- Konstruiere ein gleichförmiges Raster über der Szene
  - Indiziere Objekte entsprechend ihrer Überlappung mit den Rasterzellen
- Verfolge die Strahlen durch das Raster
  - Schnell, es müssen nur die Zellen beachtet werden die mit den Objekten überlappen.



Gleichförmiges Raster

25

- Eventuell ein Problem: was ist die richtige Rasterauflösung ?
  - Zu grob: geringer Nutzen der Rasterung!
  - Zu fein: hohe Kosten!



Strahl - Szene - Schnittpunkt

26

- Schnittpunkt mit geometrischen Grundbausteinen
  - Kugel
  - Dreieck
  - Gruppen von Bausteinen

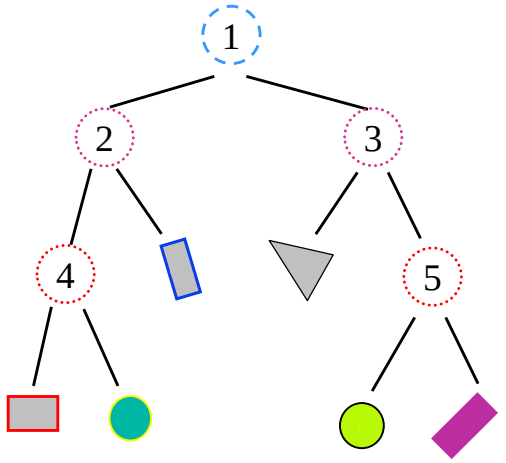
>> Beschleunigungsverfahren

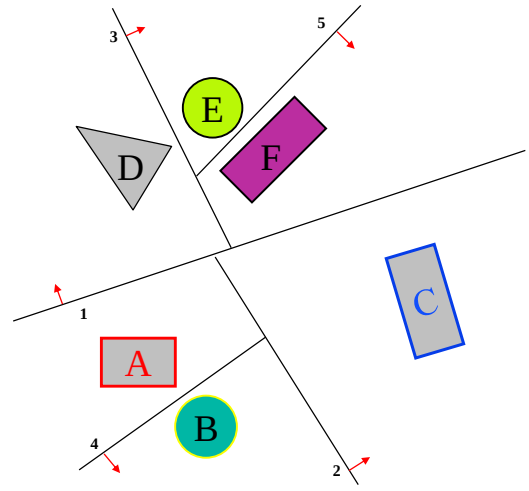
- Hierarchie von Begrenzungsvolumina
- Raumaufteilung
  - gleichmäßige Raster
  - BSP trees
  - Octrees

„Binary Space Partition (BSP) Tree“

27

- Einfacher rekursiver Algorithmus

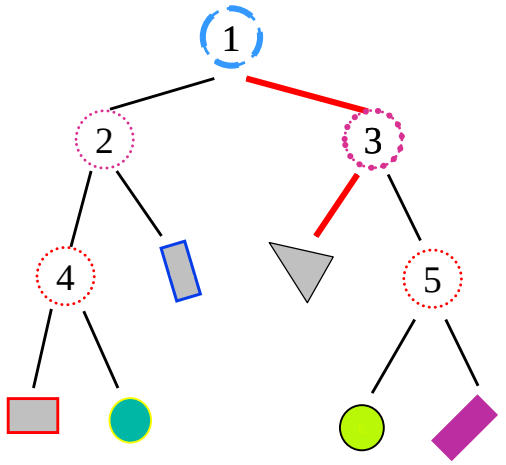




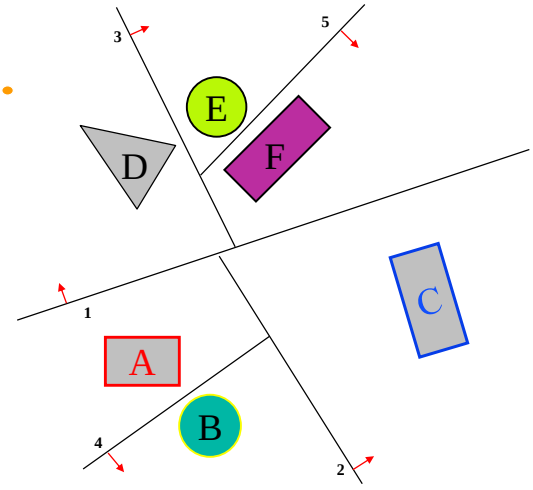
„Binary Space Partition (BSP) Tree“

28

- Einfache Suche eines Punktes



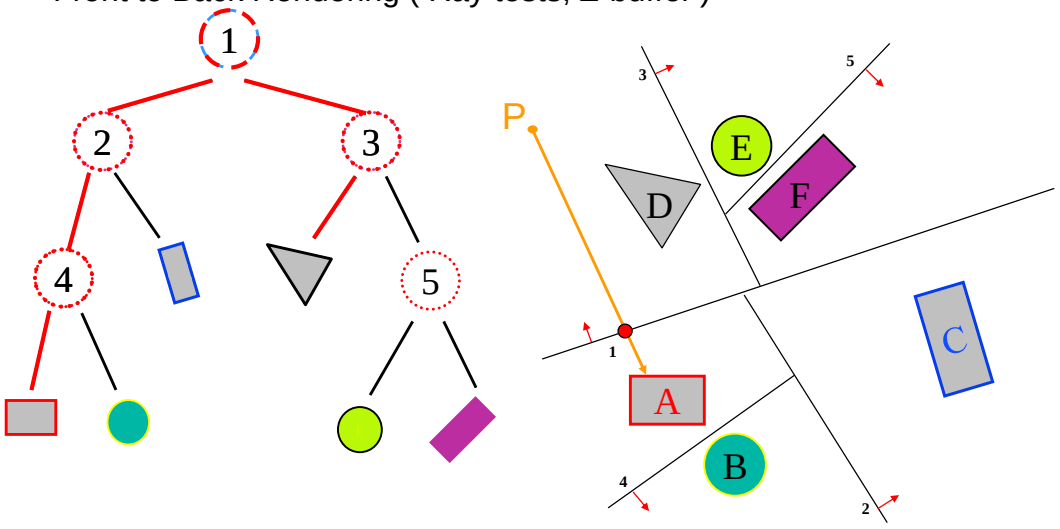
P.



Rendern von BSP-Trees

29

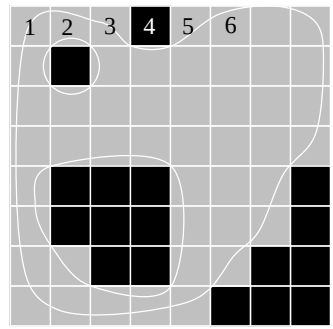
- Back to Front Sorting (Painters Algorithm)
- Front to Back Rendering (Ray tests, Z-buffer)



Repräsentation der Raumüberdeckung

30

Zerlegung des Raumes in seine Rasterfelder.



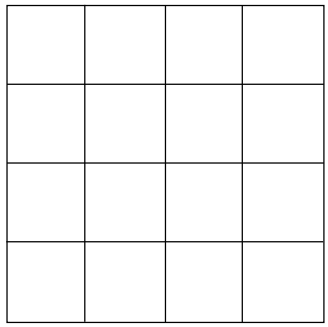
Nicht effizient,
da meist große Raumbereiche leer sind!

Wenig Information über die Struktur des Objekts!

Raumzerlegungen (Octrees)

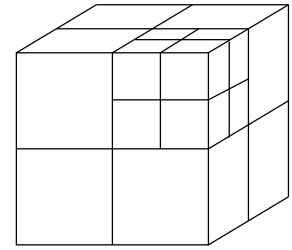
31

Hierarchische Zerlegung des Raumes durch binäre Unterteilungen.



2D Quadtrees

3D Octrees

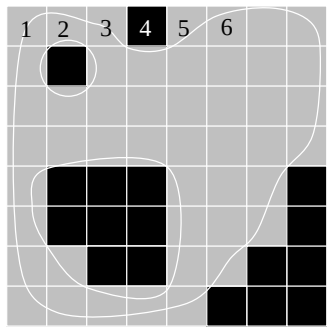


Quadtrees

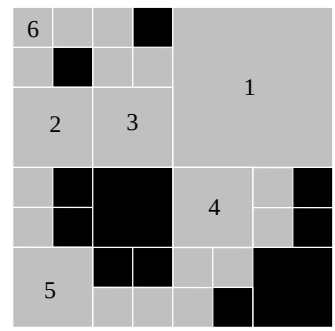
32

Es werden nur die Quadranten weiter unterteilt, deren innere Struktur nicht homogen ist.

Alle Raumpixel (Voxel)



Quadtree



Nicht jede Numerierung der Segmente ist sinnvoll ==========>

Quadtrees: Datenstruktur

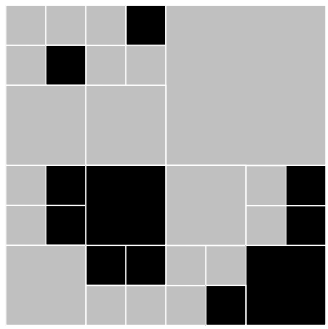
33

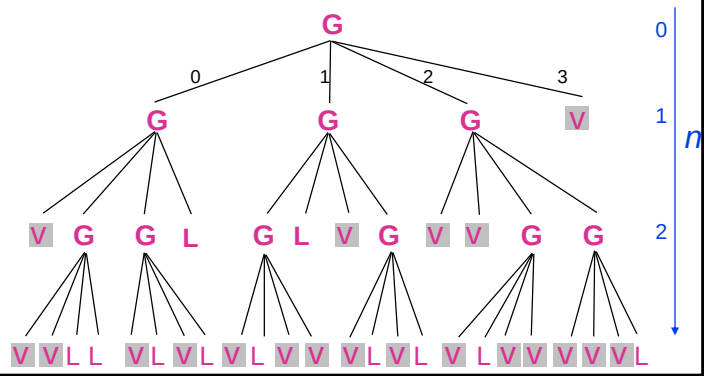
Segmente haben 3 Zustände: **V**oll **L**eer **G**emischt

Jeder Quadranten besitzt 4 Segmente.

Jeder Level n besitzt 4^n Segmente.

2	3
0	1



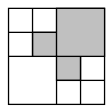


Arbeiten mit Quadtrees (Octrees)

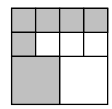
34

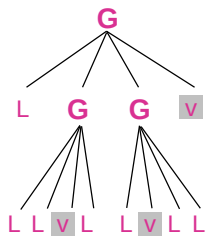
Boolsche Operationen lassen sich einfach implementieren.
Die Operationen werden für jeden Knoten durchgeführt.

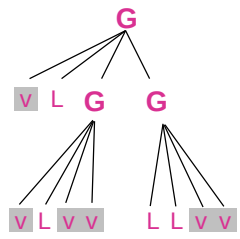
A)



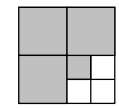
B)



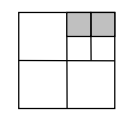




$A \cup B$



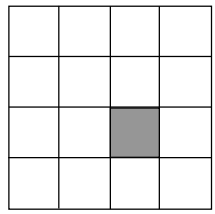
$A \cap B$



Nachbarschaften in Quadtrees (Octrees)

35

	Quadtree	Octree
Nachbarn:	8	26
an den Seiten	4	6
an den Ecken	4	8
an den Kanten	-	12

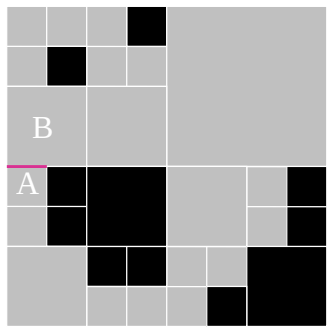


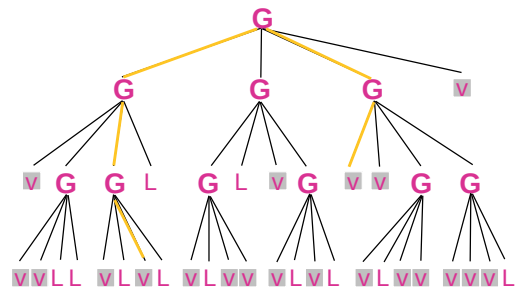
Wie finde ich die Nachbarn innerhalb den Datenstrukturen ?

Nachbarschaften in Quadtrees (Octrees)

36

Welches Segment grenzt oben an A?





Formal:
1. Gehe zurück zum gemeinsamen Knoten.
2. Suche aus allen Nachfolgern den Nachbar.

Je nach Kante und Ecke des Segments unterscheidet sich das Suchverhalten. Die einzelnen Suchschemata können in einer LUT gespeichert werden und müssen dann rekursiv kombiniert werden.

Octrees und polygonale Daten

37

1. Setze maximale Anzahl der Polygone pro Octreesegment.

2. Teile die gesamte Szene bis alle Würfel die Maximalzahl unterschreiten.

3. Falls bei der Teilung eines Würfels Objekte die Trennebenen schneiden, werden diese zu dem Würfel in der gröberen Auflösungsstufe assoziiert.

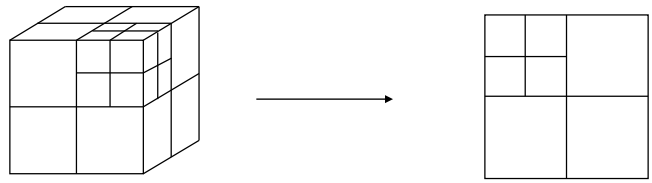
===> somit können kleine Polygone große Würfel belegen.

Octrees zur Verdeckungsrechnung

38

1. Octrees in Blickrichtung projizieren.

2. Projektion des Octrees auf einen Quadtree in der Bildebene. Sortiere für jedes Quadtreesegment die projizierten Octree-segmente der Tiefe nach.



3. Alle sich überlagernden Segmente des Octrees abarbeiten, meist von vorn nach hinten.

=== > alle Objekte rendern !!!!!

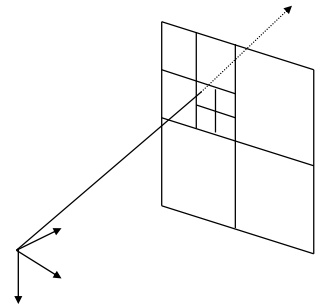
Mit welcher Methode **====>**

Polygone in Octrees rendern

39

Zwei gängige Methoden:

A) *Ray-casting*



1.) Bestimme für jeden Pixel den Sehstrahl und das vorderste Octreesegment.

2.) Schneide Strahl mit allen Objekten des Segments.

3a) Falls der Strahl ein Objekt trifft, bestimmt das vorderste den Wert des Pixels und für diesen Strahl existiert kein näheres Objekt.

3b) Falls der Strahl kein Objekt trifft, muss im nächst weiter entfernten Octreesegment gesucht werden

Polygone in Octrees rendern.

40

Zwei gängige Methoden:

B) *Z-buffer & scan-conversion*

1.) Sortiere alle Octreesegmente eines Quadtreesegments der Tiefe nach.

2.) Rendern der Octreesegmente von vorn nach hinten.

3a) Falls die Z-Position des Octreesegmente im Z-Buffer verdeckt ist, muß der Inhalt des Segments nicht gerendert werden.

3b) Falls nicht, werden alle Polygone des Segments gerendert (scan-convertiert)

Polygone in Octrees rendern. (3)

Z-buffer & scan-conversion

Wie läßt sich zeitliche Kohärenz ausnutzen ?

Annahme:

Von Bild zu Bild ändern sich weniger die Objekte als ihre Position!!

===> Die meisten sichtbaren Objekte des alten Bildes
werden auch im neuen Bild wiederum sichtbar sein!

===> Diese Objekte werden zuerst gerendert! Warum?