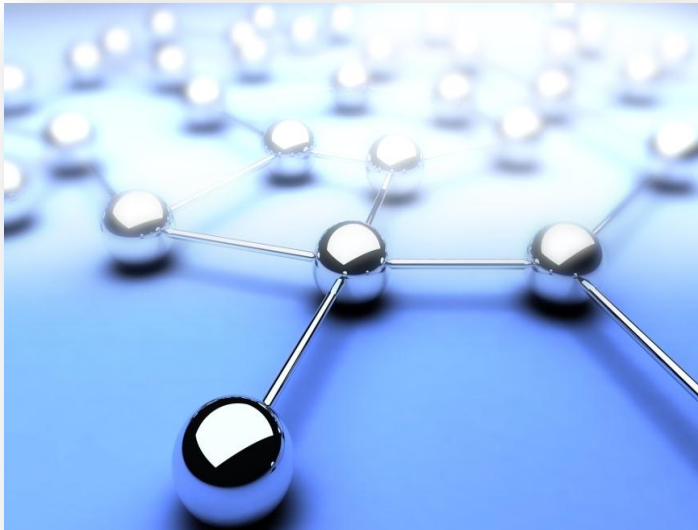


Vorlesung Algorithmen für hochkomplexe Virtuelle Szenen

Sommersemester 2012



Matthias Fischer
mafi@upb.de

Vorlesung 11
19.6.2012



Dynamische Berechnung von Potentially Visible Sets

- Motivation
- Idee
- Berechnungsbeispiel
- Algorithmus

- Tomas Akenine-Möller, Eric Haines
Real-Time Rendering
AK Peters, 2002
- Frédo Durand
A multidisciplinary survey of visibility
SIGGRAPH 2000 course notes: Visibility, Problems, Techniques, and Applications
auch zu finden im 2. Teil der Dissertation von Durand:
Frédo Durand. 3D Visibility: Analytical Study and Applications. PhD thesis, Université Joseph Fourier, 1999 (siehe Web)
- Daniel Cohen-Or, Yiorgos L. Chrysanthou, Cláudio T. Silva, Frédo Durand
A survey of visibility for walkthrough applications
IEEE Transactions on Visualization and Computer Graphics, 9(3):412–431, 2003
- David P. Luebke and Chris Georges
Portals and Mirrors: Simple, Fast Evaluation of Potentially Visible Sets
Proc. of the Symposium on Interactive 3D Graphics, S. 105-106, 1995
<http://doi.acm.org/10.1145/199404.199422>

Das bereits vorgestellte statische PVS-Verfahren

PVS-Berechnung wurde im Preprocessing durchgeführt:

„Cell-to-Cell-Visibility“

- Die Berechnung der PVS-Mengen erfolgte durch Überprüfung, ob Sichtlinien sich durch aufeinander folgende Portale legen lassen (Problemreduktion).
- Berechnung ist sichtpunktunabhängig.

Das PVS wurde sichtpunktabhängig zur Laufzeit verkleinert:

„Eye-to-Cell-Visibility“

Frage

Lassen sich beide Schritte sichtpunktabhängig zur Laufzeit berechnen?



Welchen Vorteil bietet die Berechnung zur Laufzeit?

- Speicherplatz für Stabtrees entfällt, nur Adjazenzgraph wird benötigt (Stabtrees können groß werden, es kann sehr viele geben)
- Verkürzung der Preprocessingzeiten
- Dynamische Szenenänderungen und Modellierung ist in begrenzten Bereichen möglich: z.B. Verschieben von Fenstern



Berechnung in zwei Schritten

1. Aufbau einer räumlichen Unterteilung mit einem BSP-Baum

- Unterteilung in Zellen und Portale
- es gibt nur den Adjazenzgraphen
(so wie bei dem bisherigem Verfahren)

2. Walkthrough mit sichtpunktabhängiger Berechnung der PVS

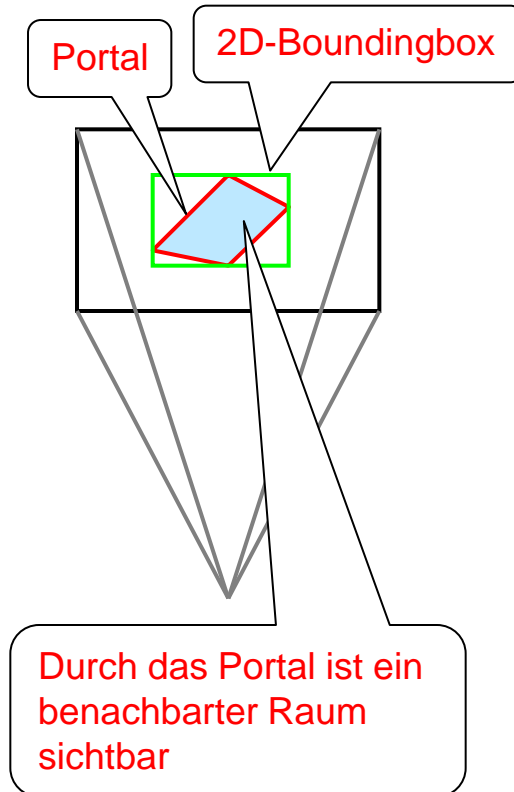
Wir benötigen keine Stabtrees mehr !!

Idee

- Rendere die Zelle des Betrachters,
- und von dort aus nur noch benachbarte Zellen, die durch Portale zu sehen sind.
- Baue auf diese Weise Portalsequenzen auf, bis in jeder Portalsequenz keine weiteren Portale mehr sichtbar sind.

Beachte dabei:

- Portale sind spezielle Polygonzüge.
- Verwende für die auf dem Bildschirm projizierten Portale die 2D-Boundingbox (BB) im Bildraum.



Dynamische Berechnung von PVS

Berechnungsbeispiel



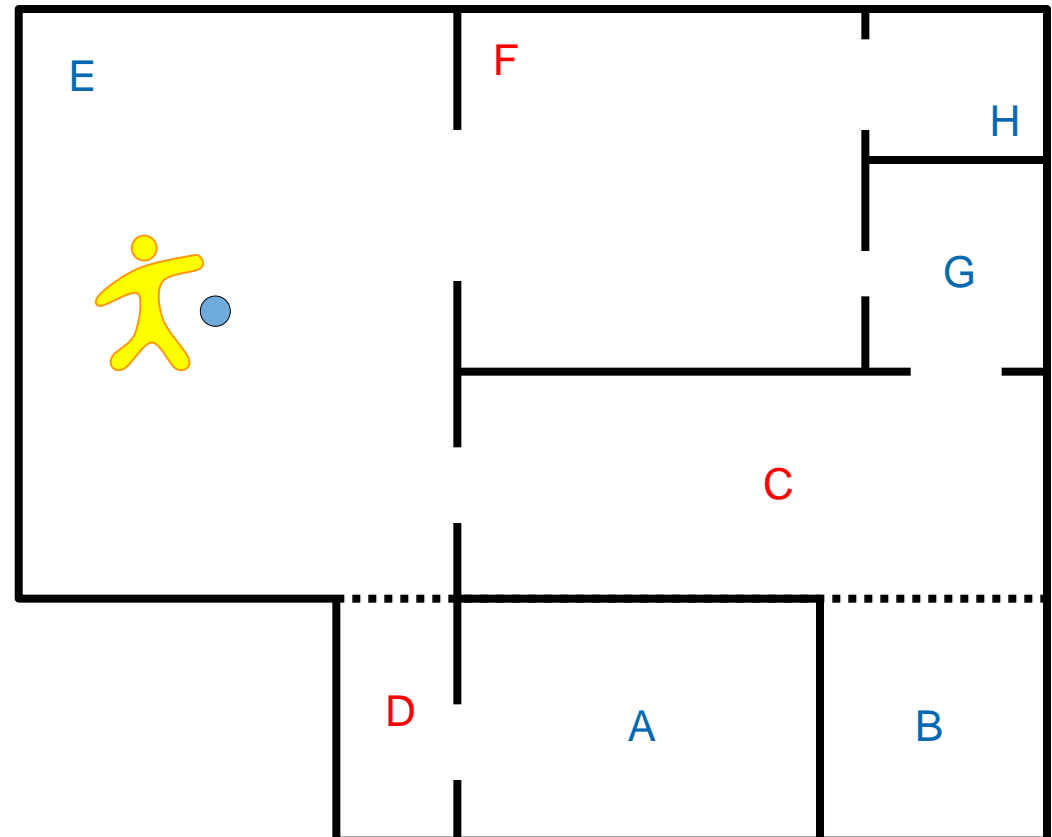
Beispiel

Der Betrachter steht in Zelle

E

Benachbarte Zellen sind:

F, C, D



Dynamische Berechnung von PVS

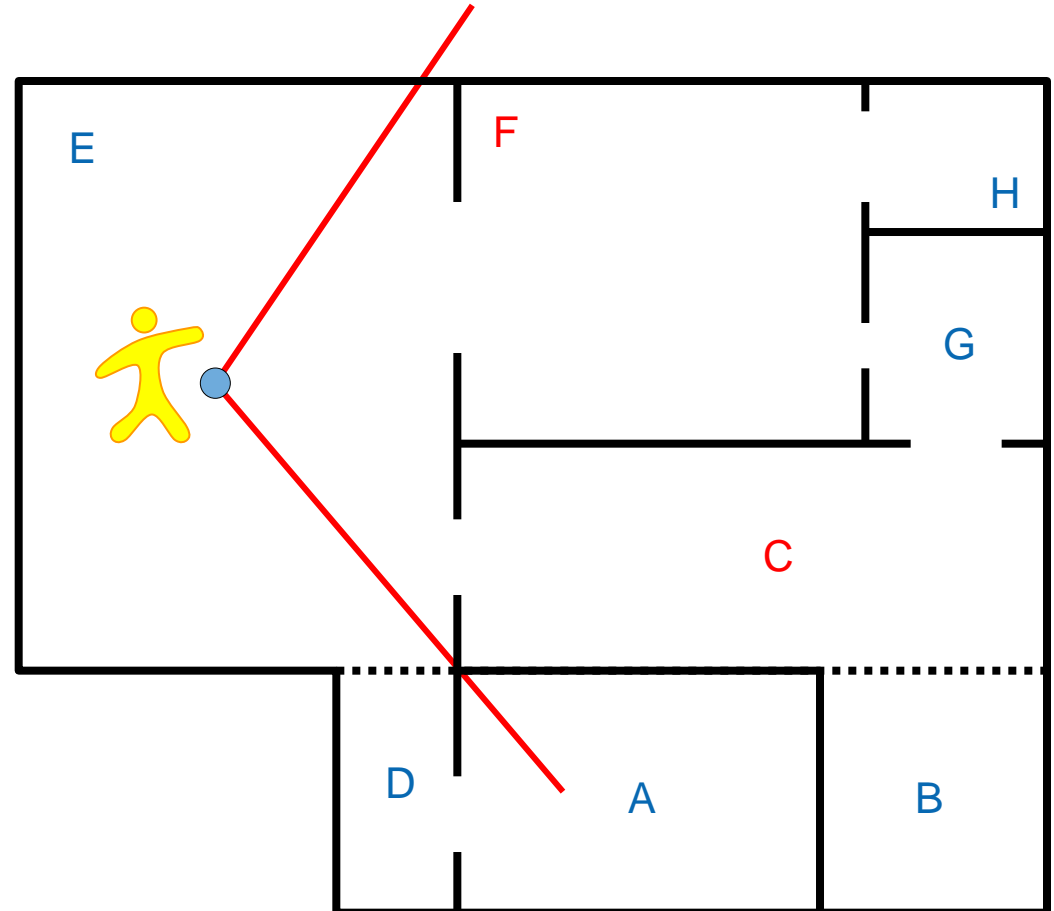
Berechnungsbeispiel



Beispiel

Im Sichtkegel des
Betrachters sind nur noch
Zellen

C und **F**



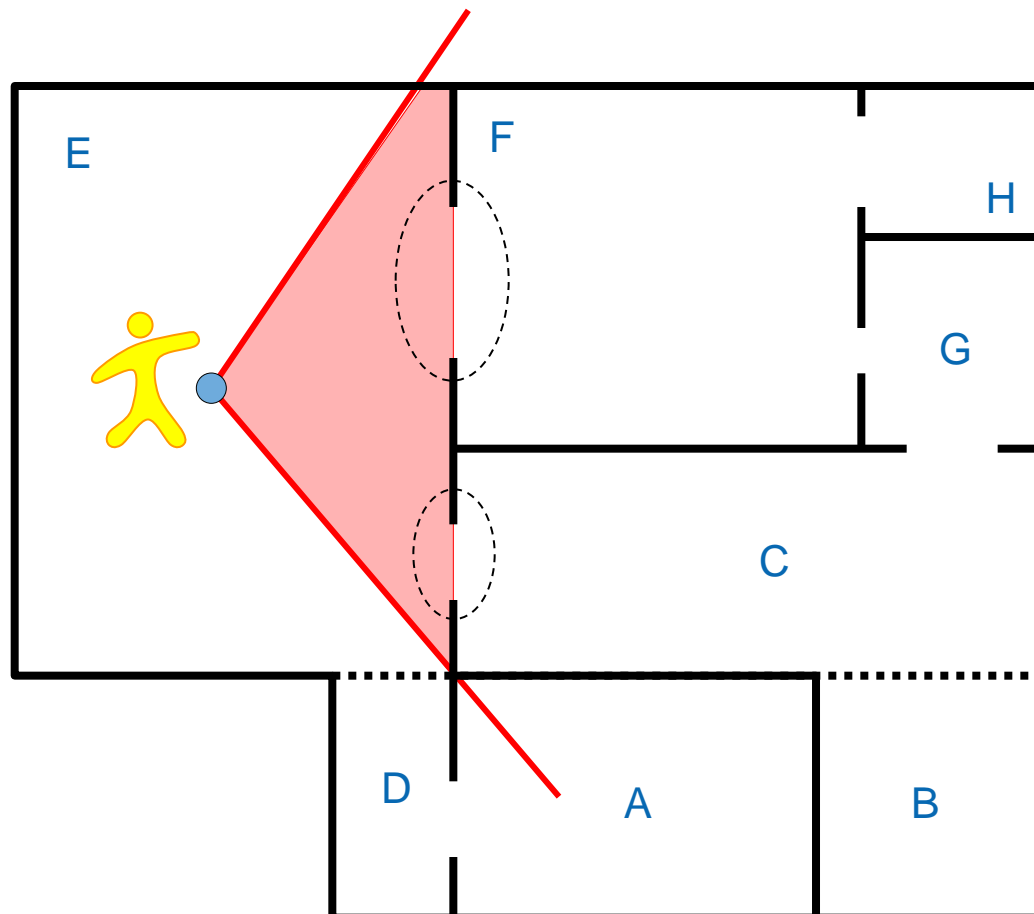
Dynamische Berechnung von PVS

Berechnungsbeispiel



Beispiel

Von Zelle **C** und **F** werden jeweils die 2D Cullboxen berechnet.



Dynamische Berechnung von PVS

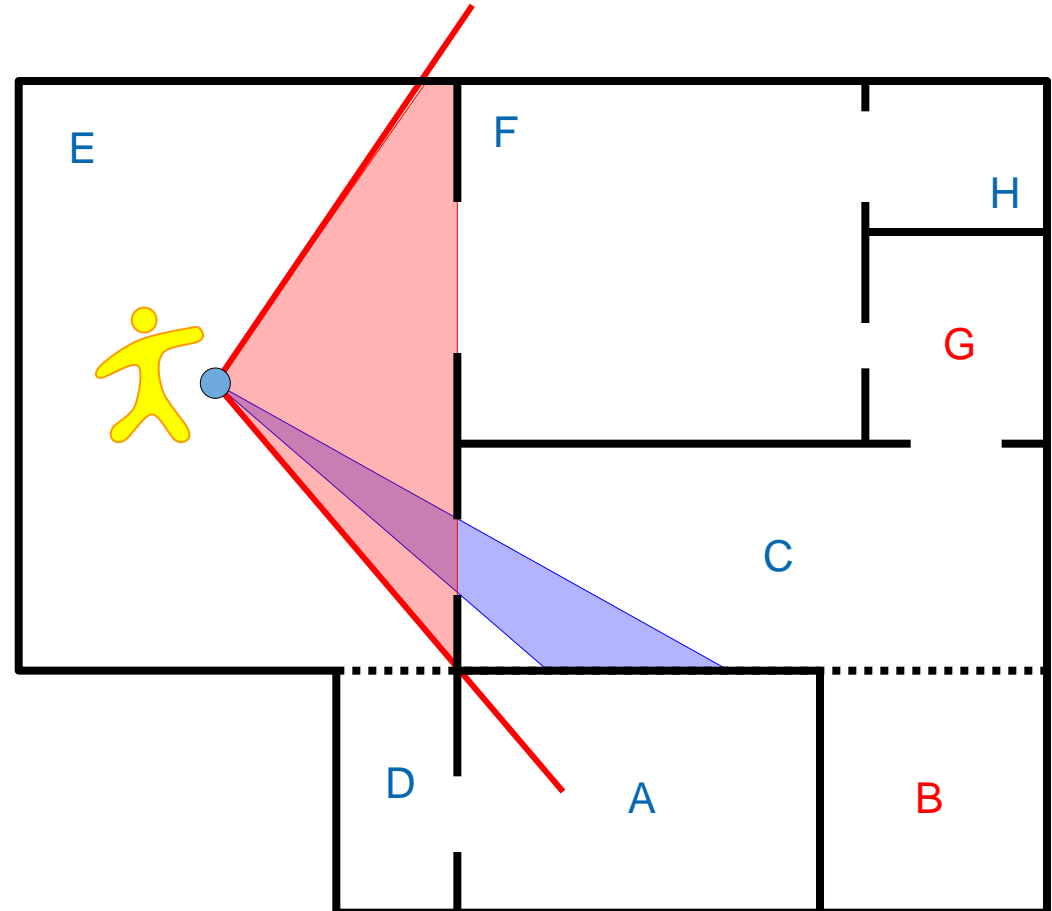
Berechnungsbeispiel



Beispiel

Nachbarzelle **C** wird gerendert:

- In der Cullbox **E-C** befindet sich kein Portal zu **G** und **B**.
- Hier stoppt die Suche im Adjazenzgraphen.



Dynamische Berechnung von PVS

Berechnungsbeispiel

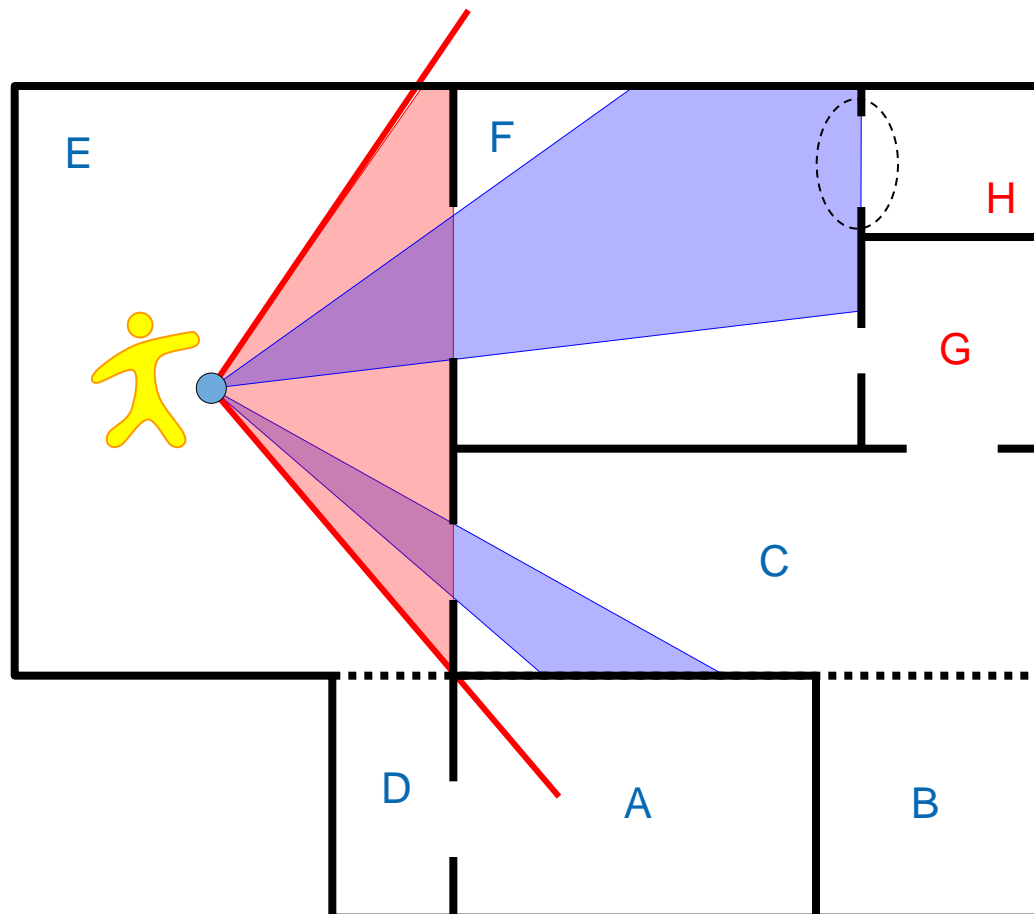


Beispiel

Die Nachbarzelle **F** wird gerendert:

In der Cullbox **E-F** ist das Portal zu **G** nicht zu sehen, aber **H** ist über ein weiteres Portal sichtbar.

Darum stoppt die Suche im Adjanzenzgraphen jetzt bei **G**.



Dynamische Berechnung von PVS

Berechnungsbeispiel

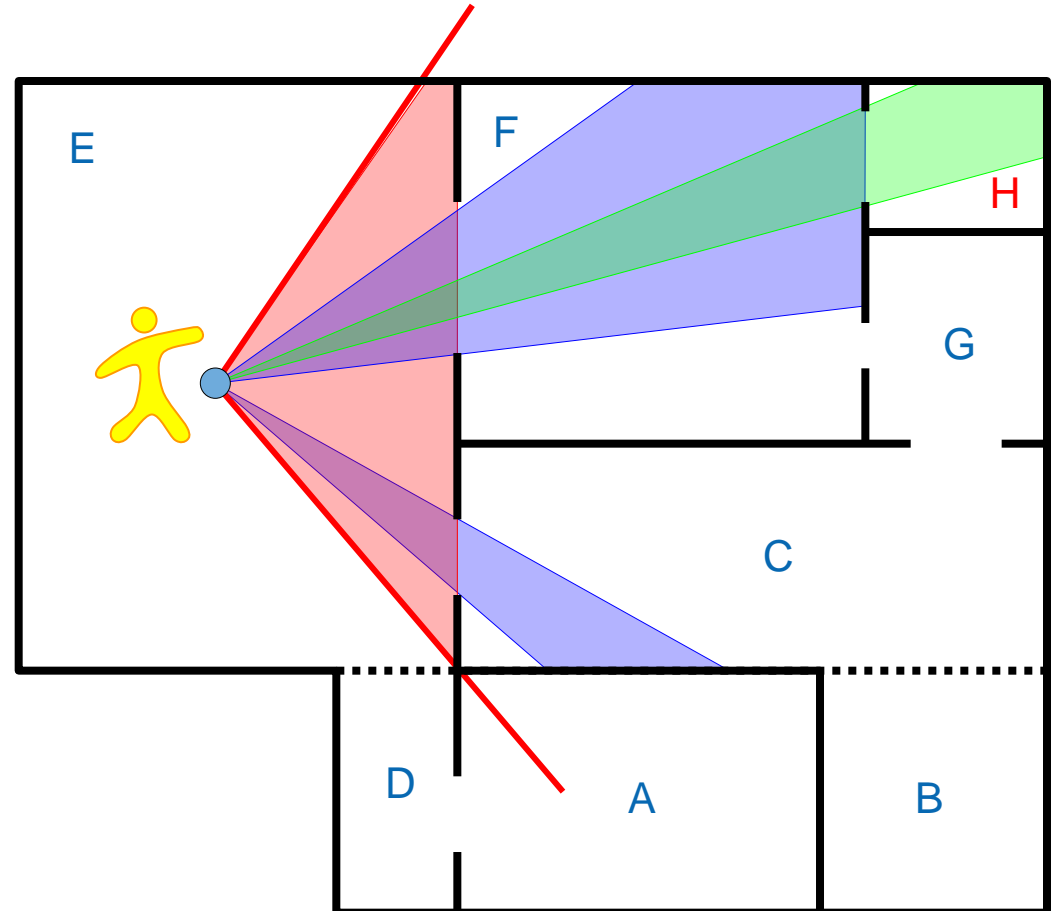


Beispiel

In der Cullbox **E-F** ist jedoch das Portal zu **H** zu sehen!

Darum wird Zelle **H** gerendert.

Auch bei **H** stoppt die Suche im Adjazenzgraphen, da es in **H** keine weiteren Portale gibt.

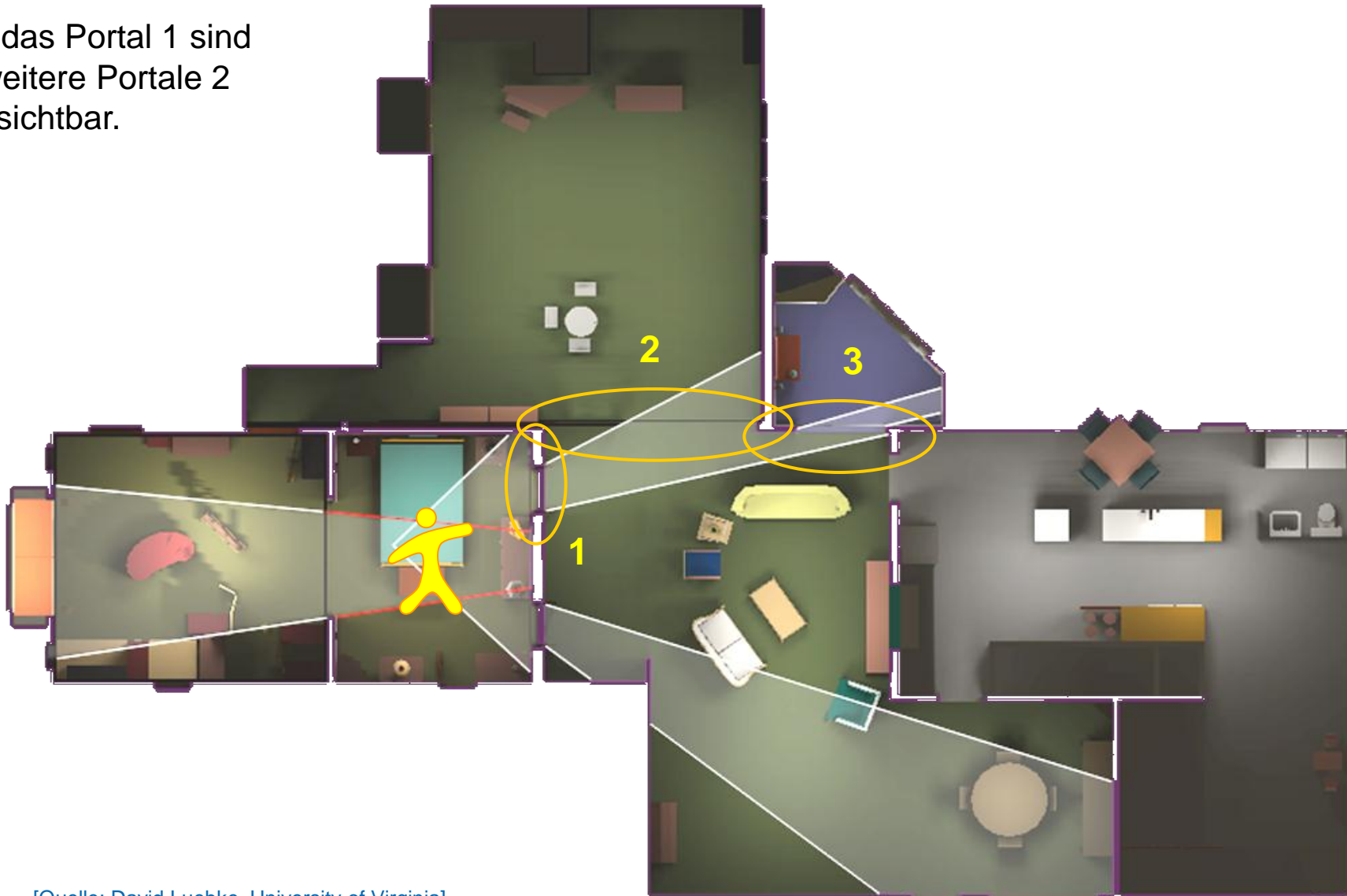


Dynamische Berechnung von PVS

Beispiel



Durch das Portal 1 sind zwei weitere Portale 2 und 3 sichtbar.



[Quelle: David Luebke, University of Virginia]

Dynamische Berechnung von PVS

Beispiel



Die weißen
Rahmen sind die
achsen-
parallelen
Boundingboxen
der Portale.

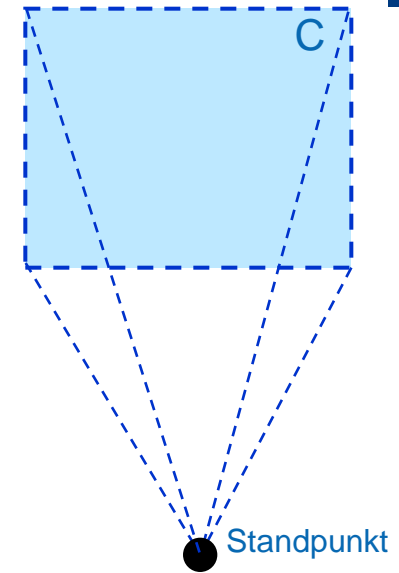


[Quelle: David Luebke, University of Virginia]

Sichtpunktabhängige Berechnung der PVS

Algorithmus (Initialisierung)

1. Bestimme die Zelle **V**, in der der Betrachter steht (Verwende dazu den Adjazenzgraphen)
2. Starte mit einer leeren Portalsequenz **P**
3. Initialisiere eine 2D-Boundingbox **C** in der Größe des Bildschirms



Dynamische Berechnung von PVS

Algorithmus

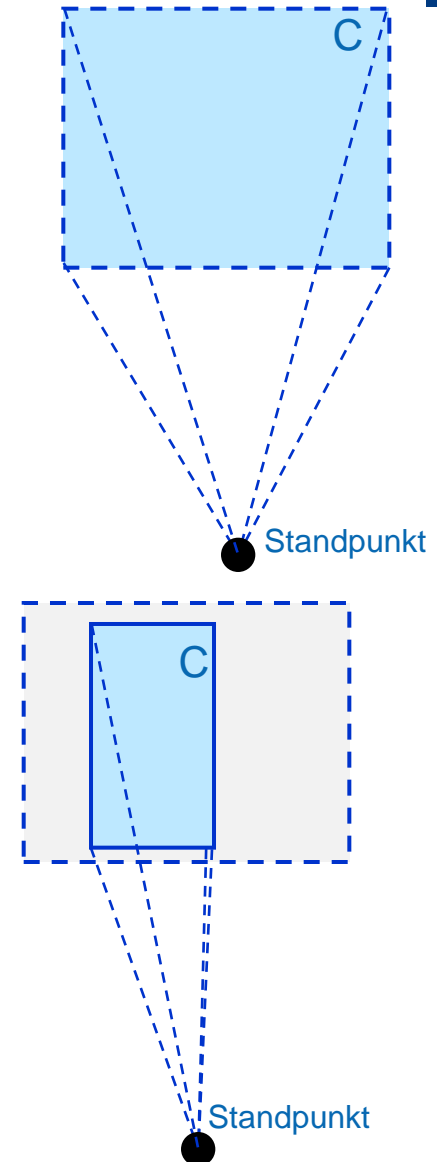


Algorithmus

4. Rendere die **Geometrie der Zelle V**, führe dabei **Frustum Culling** mit folgendem Frustum (Cullbox) durch:

Das Frustum (=Cullbox) ist definiert durch:

- die Spitze ist der Standpunkt des Betrachters,
- einen Querschnitt des Pyramidenstumpfs bildet das Rechteck C im Bildraum



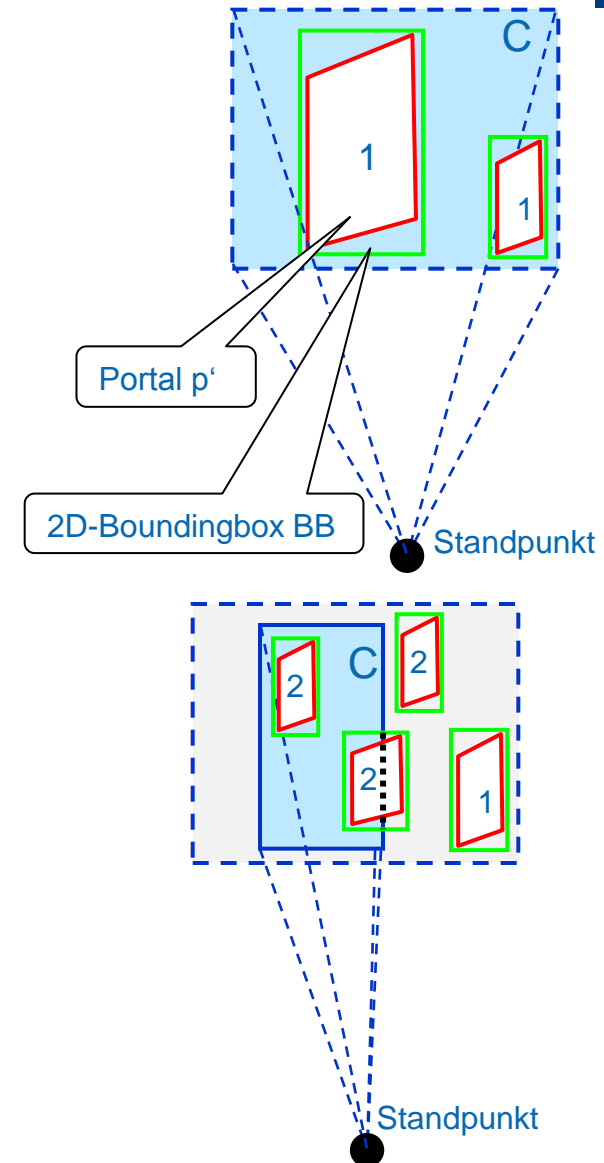
Dynamische Berechnung von PVS

Algorithmus

Algorithmus

5. Für alle direkten Nachbarn V' der Zelle V :
Für alle Portale p' der aktuellen Zelle V , die zu V' führen:
 - projiziere das Portal p' in den Bildraum (Eckpunkte) und bestimme die **2D-Boundingbox BB**.
 - berechne im Bildraum die **Schnittmenge** von **C** und **BB** $:= C'$

Die Nachbarn V' werden über den Adjazenzgraphen bestimmt.



Dynamische Berechnung von PVS

Algorithmus

Algorithmus

6. Für alle in Schritt 5 berechneten **Schnittmengen C'**

Falls Schnittmenge **C'** leer

- benachbarte Zelle **V'** durch die Portalsequenz **P** unsichtbar
- Traversierung und Suche im Adjazenzgraph endet hier

sonst

// Teile der Nachbarzelle **V'** sind sichtbar

- setze $C := C'$,
 $V := V'$,
 $P := P + p'$
- verfähre rekursiv mit Schritt 4 fort

