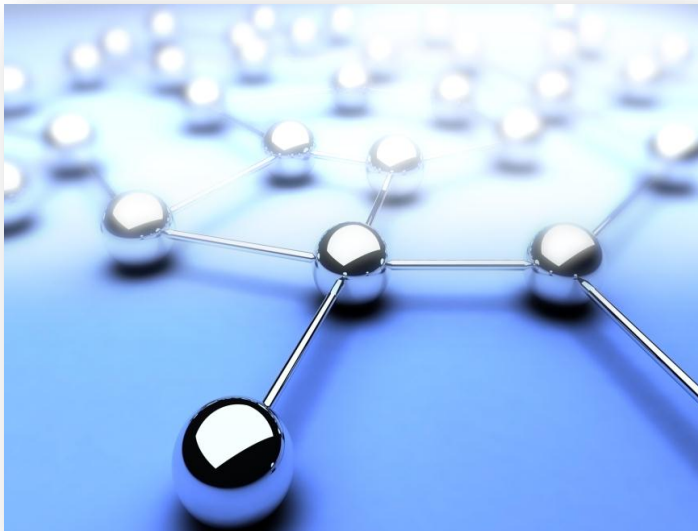


Vorlesung Algorithmen für hochkomplexe Virtuelle Szenen

Sommersemester 2012



Matthias Fischer
mafi@upb.de

Vorlesung 11
19.6.2012

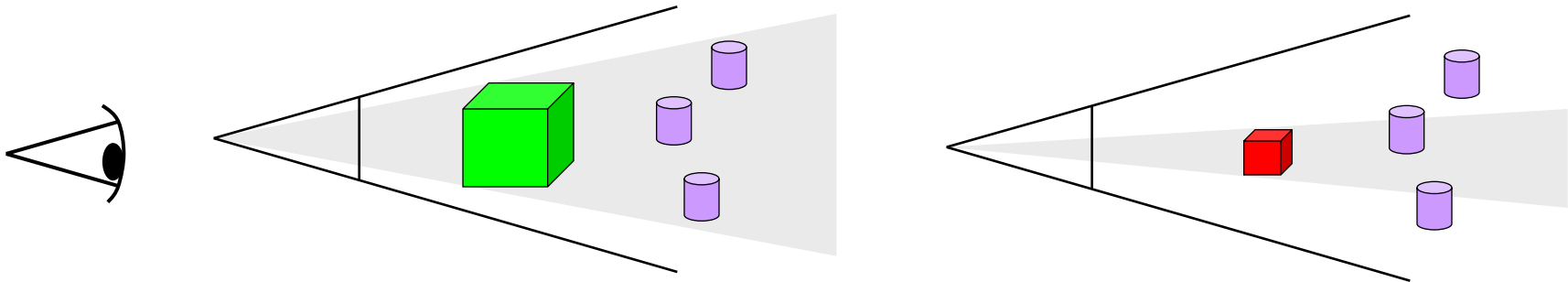
Hierarchical Occlusion Maps (HOM)

- Beobachtung und Ausgangspunkt
- Idee
- HOM-Algorithmus
- Occluder-Datenbank
- Occlusion-Map-Hierarchie
- Visibility-Culling
- Undurchsichtigkeit
- Occlusion-Map
- Bildpyramide
- Eigenschaften der Occlusion Maps
- Überlappungstest
- Auswahl der Occluder
- Tiefentest
- Approximatives Visibility Culling

- 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
- Hansong Zhang, Dinesh Manocha, Tom Hudson, Kenneth E. Hoff, III
Visibility culling using hierarchical occlusion maps
Proc. 24th Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97),
p. 77-88, 1997
<http://doi.acm.org/10.1145/258734.258781>

Ausgangspunkt

- Große Occluder lassen sich gut verwenden, um effizient Occlusion-Culling durchzuführen.
- Kleine Occluder verdecken „zu wenig“ und eignen sich daher weniger.

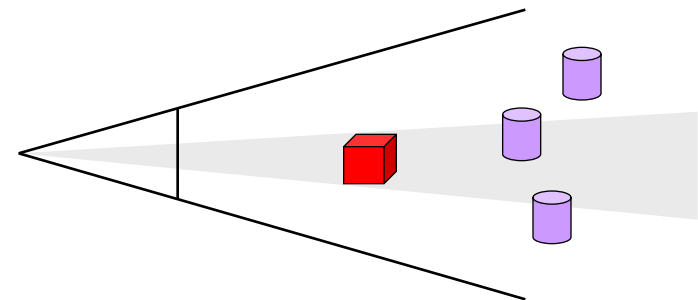
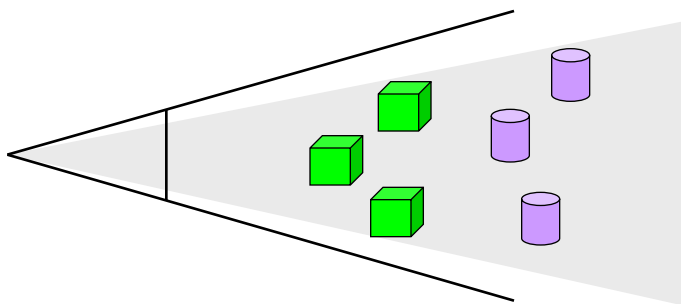


Idee des Algorithmus

- Viele kleine Occluder ergeben sichtpunktabhängig einen großen Occluder.
- Verwende einen „großen“ Occluder zum Culling.

Fragen

1. Wie werden „große“ Occluder erkannt und berechnet?
2. Wie verwenden wir die berechneten „großen“ Occluder?



Hauptmerkmale des Algorithmus

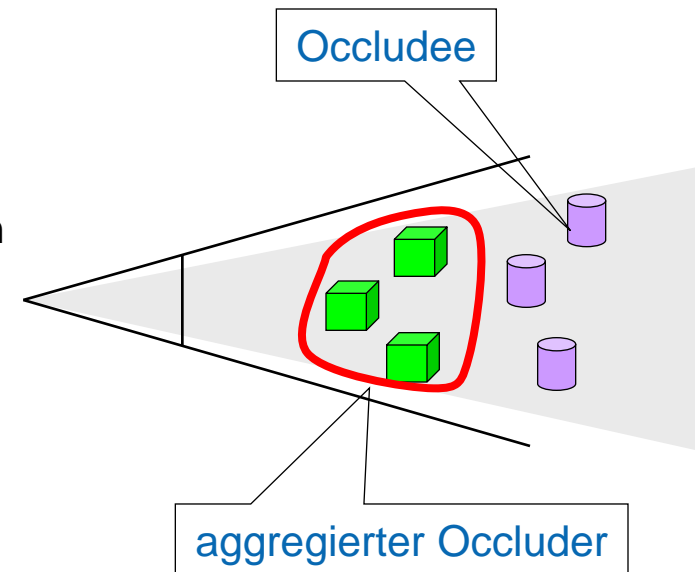
Zusammensetzung einer „Gruppe“ aus kleinen Occludern, die enger Nahbarschaft stehen.

Occluder Fusion

- Vereinigung einer Occludermenge.
- Es entsteht ein aggregierter Occluder.
- Ein aggregierter Occluder verdeckt mehr, als die einzelnen kleinen Occluder separat verdecken können.

Allgemeingültigkeit

- keine speziellen Strukturen (wie bei PVS) im Modell notwendig
- keine Restriktionen an die Occludertypen
- daher gut geeignet für CAD- und Outdoor-Szenen



Hierarchical Occlusion Maps

HOM-Algorithmus

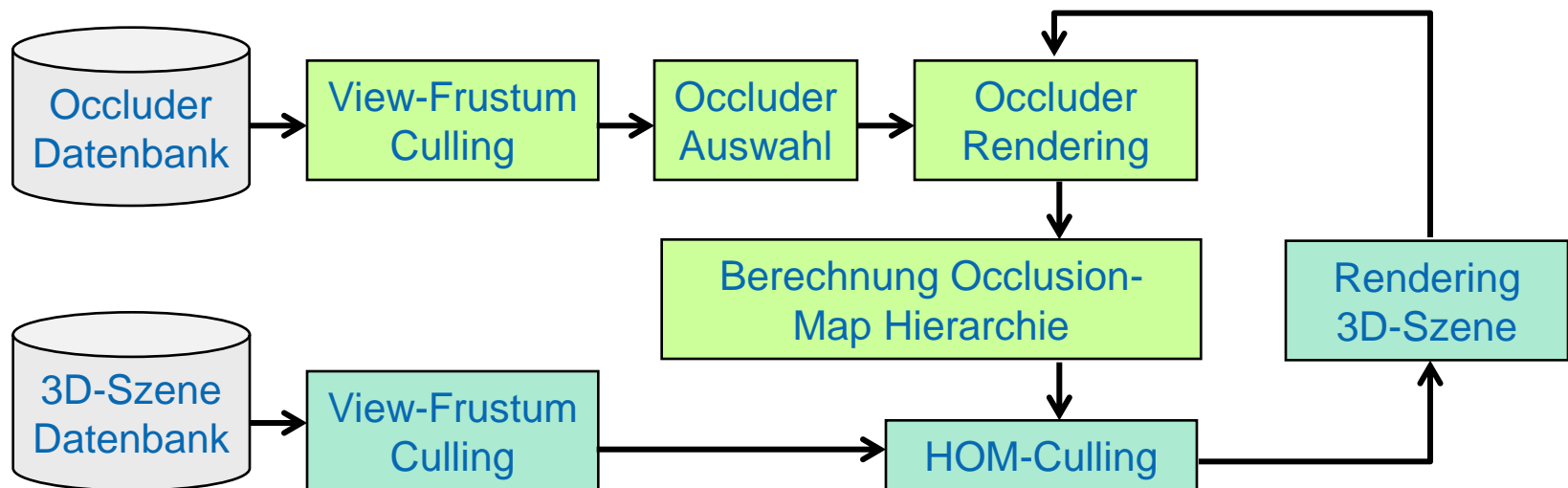


HOM-Algorithmus

Der Algorithmus arbeitet in zwei Schritten;
die beiden Schritte werden für jedes Frame berechnet:

1. Berechnung der Occlusion-Map Hierarchie
2. Visibility Culling und Rendering

HOM = Hierarchical Occlusion Maps



Hierarchical Occlusion Maps

Occluder-Datenbank

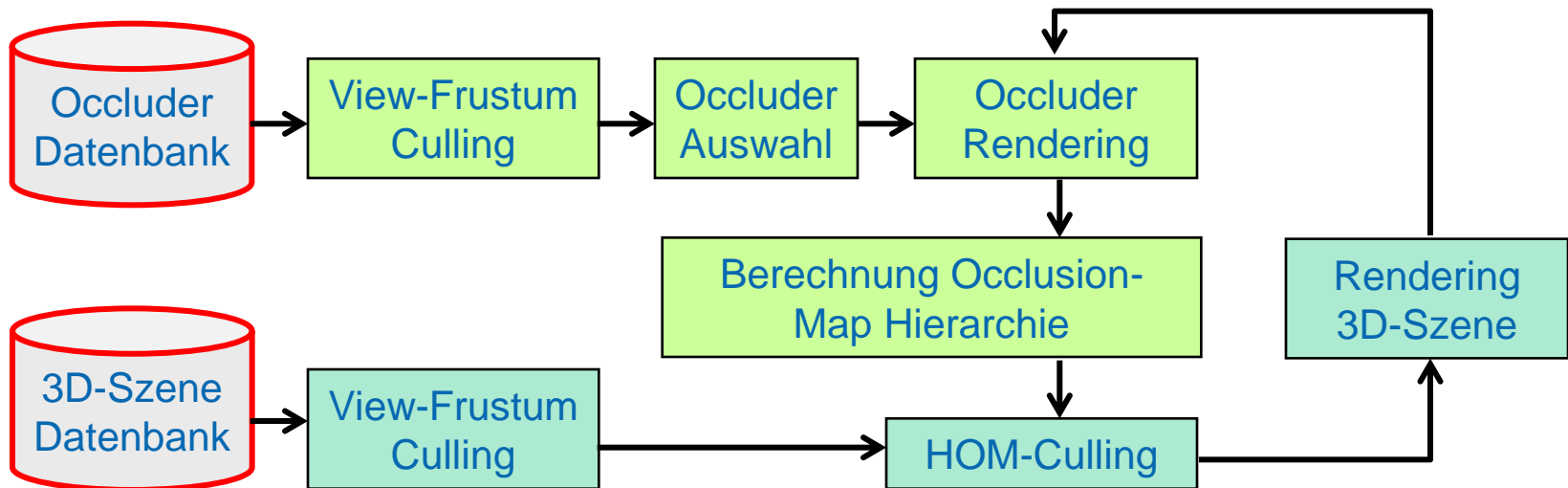


Occluder-Datenbank

Zusätzlich zur 3D-Szene gibt es eine **Occluder-Datenbank**:

- Die Occluder-Datenbank wird im Preprocessing berechnet.
- Die Occluder werden als Bounding-Volume-Hierarchie gespeichert.

Auch die **3D-Szene** wird als **Bounding-Volume-Hierarchie** gespeichert.



Hierarchical Occlusion Maps

Occlusion-Map-Hierarchie

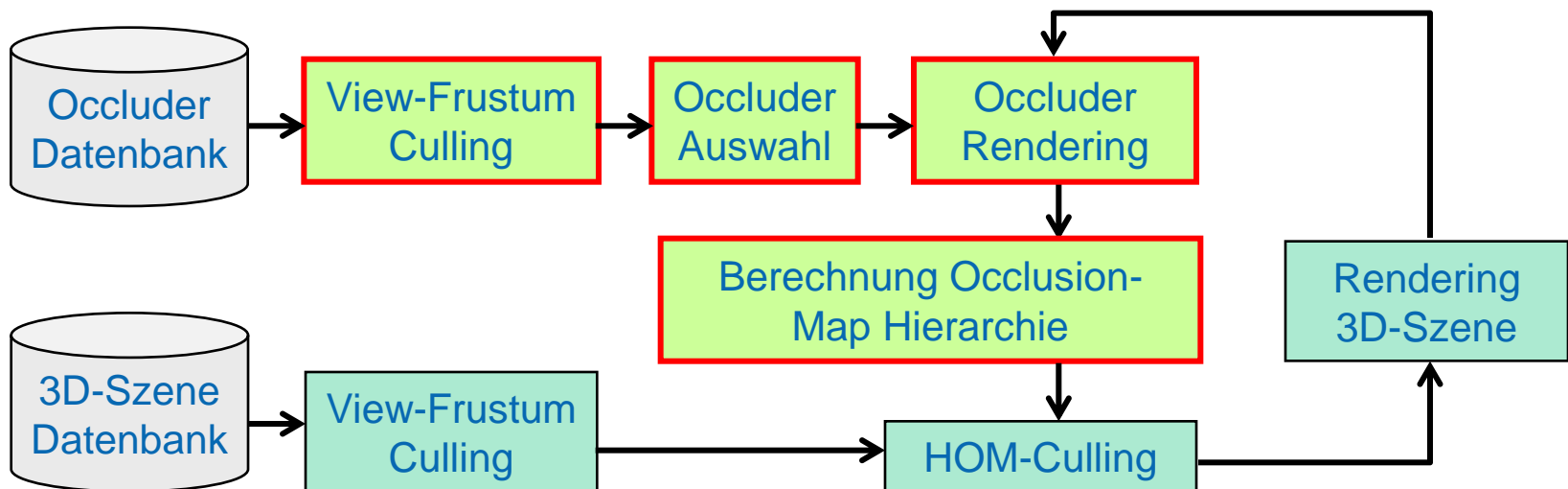


Berechnung der Occlusion-Map-Hierarchie

Aus der Occluder-Datenbank werden passende Occluder gewählt und anschließend eine Hierarchie von Occlusion-Maps aufgebaut:

1. Mit **View-Frustum-Culling** werden nur **Occluder** gewählt, die im Frustum liegen.
2. Von den Occludern des Frustums wird nur eine Teilmenge ausgewählt. (s.u.)
3. Alle gewählten Occluder werden als weiße Objekte in den Framebuffer gerendert (Schwarz-Weiß Bild), dieses Bild hat die höchste Auflösung der Occlusion-Map.

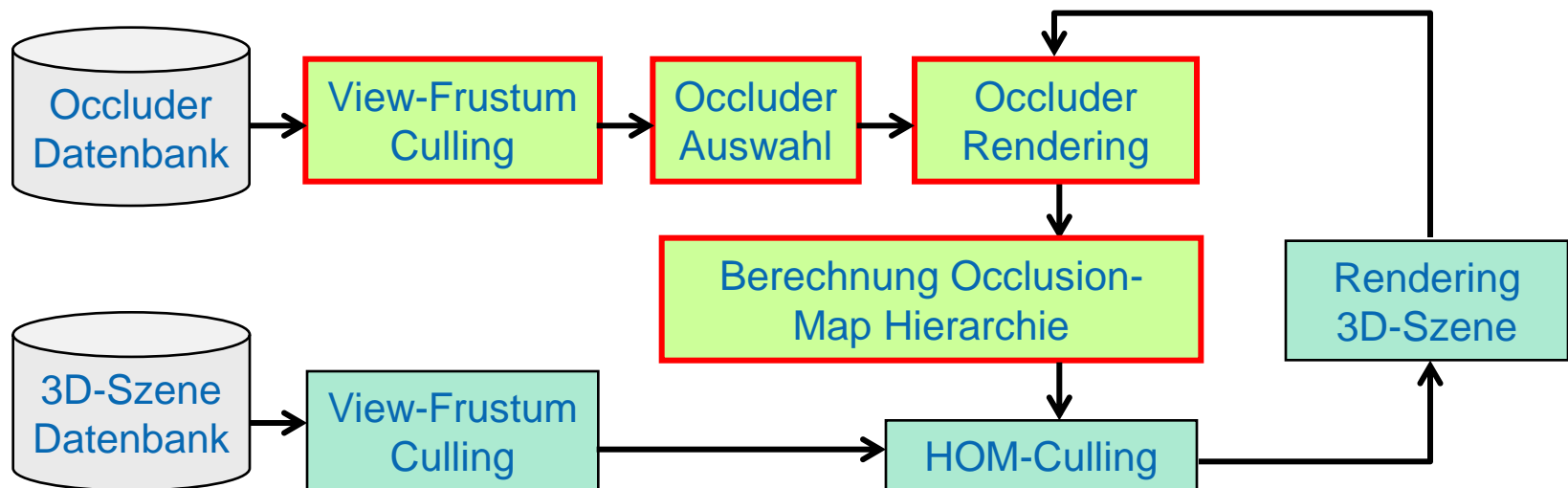
.....



Berechnung der Occlusion-Map-Hierarchie

.....

4. Berechnung eines Tiefenpuffers zur Tiefenschätzung der Occluder.
5. Aufbau der Hierarchiestufen: die Occlusion-Map wird sukzessive vergrößert, indem mehrere Pixel zusammengefasst und ein Mittelwert berechnet wird.



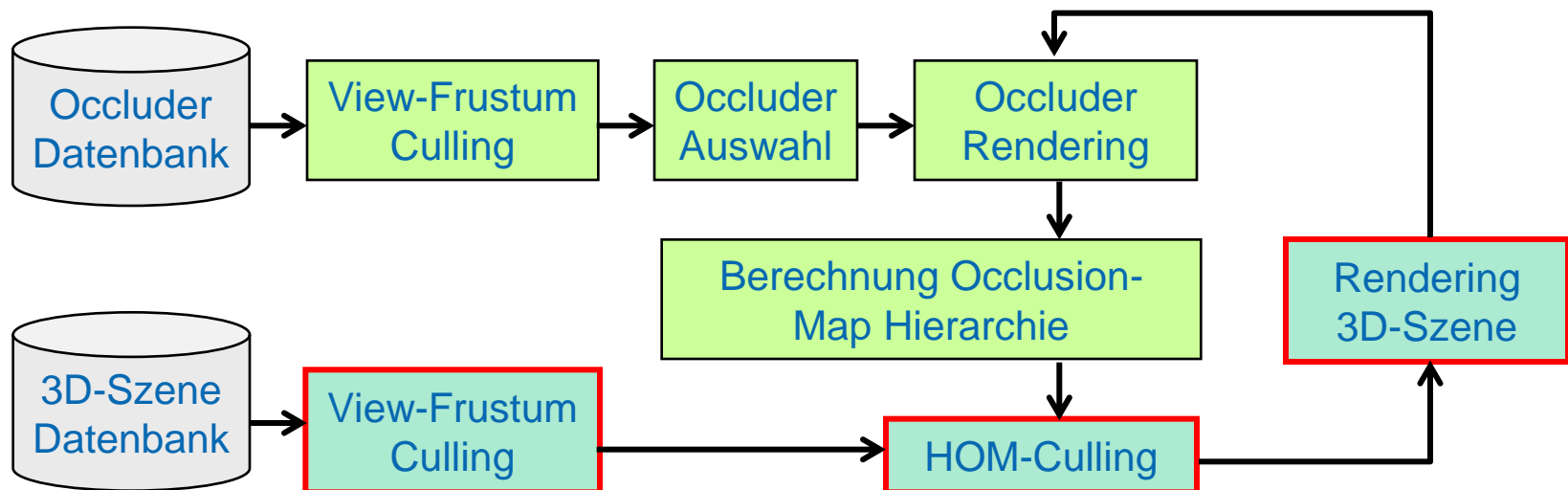
Visibility Culling und Rendering

Die 3D-Szene wird traversiert und mit Hilfe der Occlusion-Map-Hierarchie das Visibility-Culling durchgeführt;

alle nicht als unsichtbar deklarierten Objekte werden gerendert:

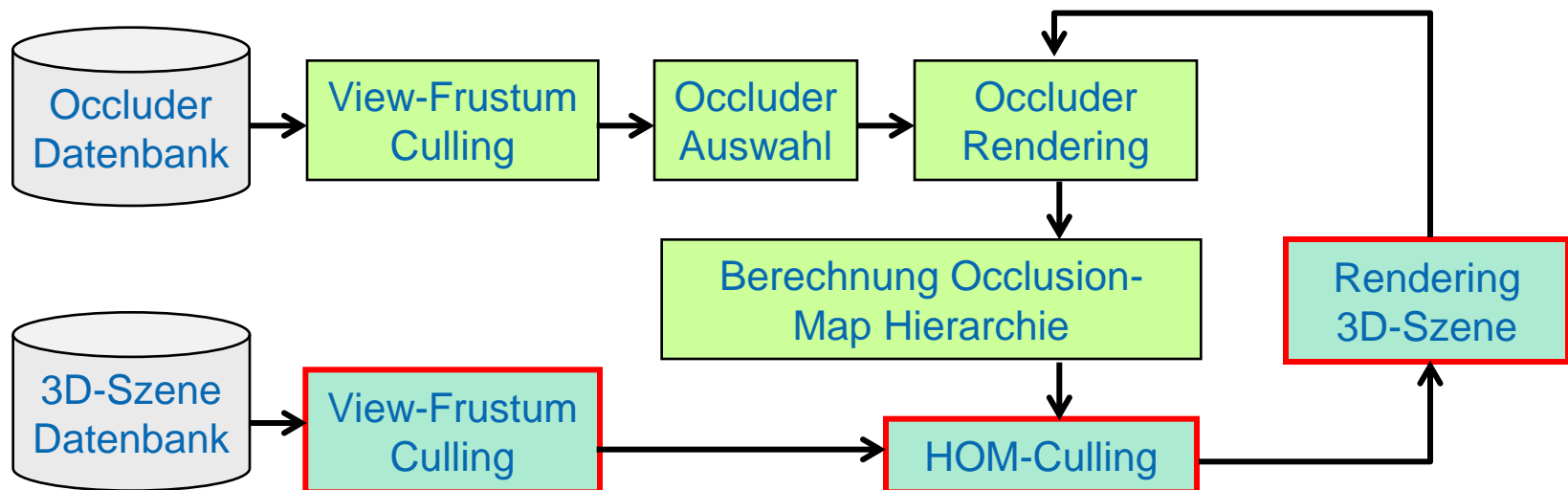
1. Mit View-Frustum-Culling wird alles außerhalb des Frustums verworfen.
2. Prüfe mit dem Tiefenpuffer, ob die Occludees hinter den Occludern liegen.

.....



Visibility Culling und Rendering

-
- Überlappungstest: „Was wird vom aggregierten Occluder verdeckt?“
 - Traversiere die Occlusion-Map-Hierarchie: von oben „geringe Auflösung“ nach unten „hohe Auflösung“,
 - überprüfe, ob die 2D-Projektion eines Occludee im Bildraum durch „undurchsichtige“ (s.u.) Teile der Occlusion-Map verdeckt ist.

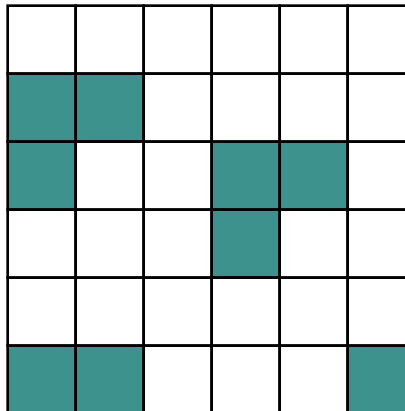


Undurchsichtigkeit

- Wir projizieren Occluder in den 2D-Raum,
- dazu rendern wir Objekte in den Frame-Buffer,
- einige Pixel werden eingefärbt, andere bleiben farblos („leer“),
- als Undurchsichtigkeit definieren wir das Verhältnis von eingefärbten Pixeln zur gesamten Anzahl der Pixel.

Beispiel:

Undurchsichtigkeit = 9/36



Hierarchical Occlusion Maps

Occlusion-Map

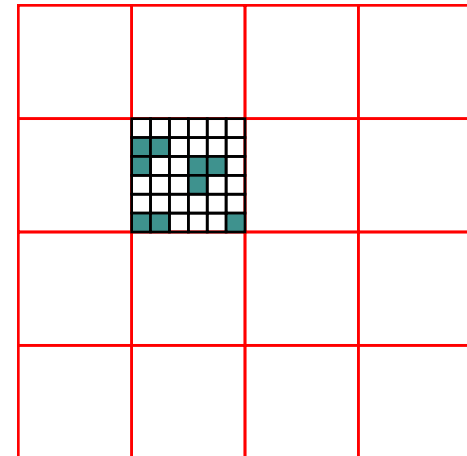


Occlusion Map

- Eine Occlusion Map ist ein 2-dimensionales Array.
- Jeder Eintrag enthält den Undurchsichtigkeitswert für einen rechteckigen Bereich des 2D-Bildraums.

Beispiel:

- Occlusion-Map mit 16 Einträgen.
- Jedes der 16 Pixel fasst einen Bereich von 36 Subpixeln zusammen.
- Das dargestellte Pixel hat eine Undurchsichtigkeit von $9/36$.



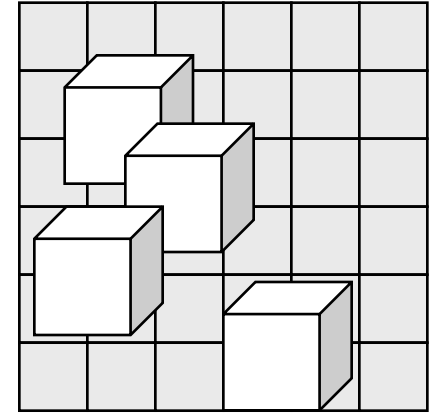
Hierarchical Occlusion Maps

Occlusion-Map



Wie wird eine Occlusion-Map berechnet?

- Wir rendern die Occluder als vollständig weiße Objekte in den Frame-Buffer.
- Wir erhalten eine Occlusion-Map in der höchsten Auflösung.
- Die Occlusion-Map aggregiert mehrere Occluder im 2D-Bildraum.



Hierarchical Occlusion Maps

Bildpyramide



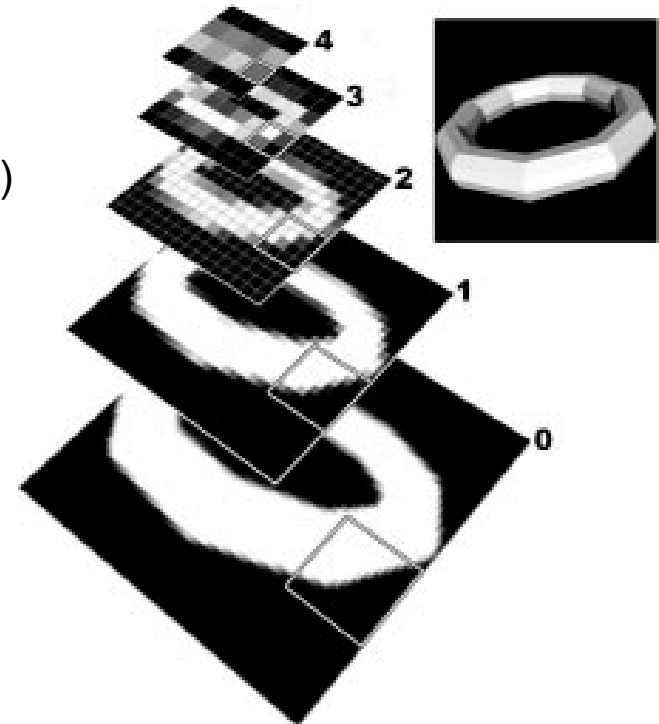
Bildpyramide Hierarchie von Occlusion-Maps (HOM)

Aus der Occlusion-Map wird eine Hierarchie von Occlusion-Maps berechnet:

- wir fassen rechteckige Bereiche des 2D-Bildraums (Pixel) zusammen,
- jeder Wert einer höheren Hierarchiestufe errechnet sich aus dem Mittelwert von tieferen Ebenen.

Wir müssen unterscheiden zwischen:

- der Berechnung der Occlusion-Map auf der untersten Hierarchieebene; berechnet aus dem Rendering weißer Occluder (s. vorherige Folie),
- und der Berechnung von höheren Hierarchiestufen.



Hierarchical Occlusion Maps

Bildpyramide



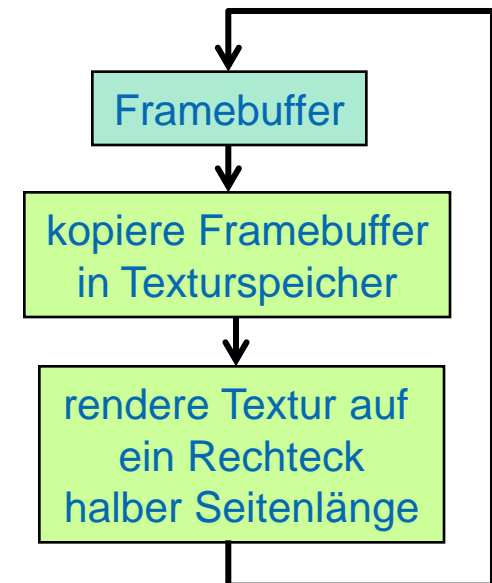
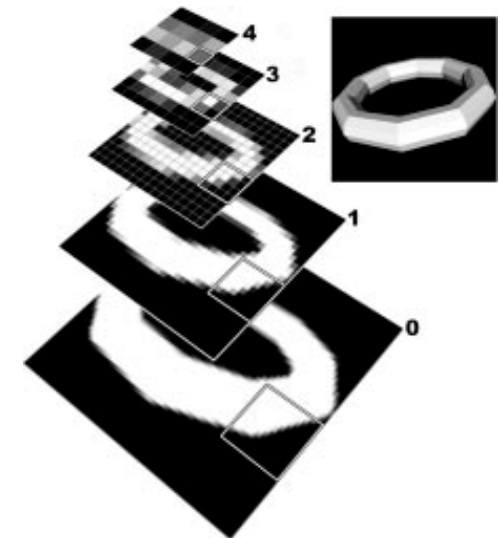
Wie bilden wir höhere Hierarchiestufen?

- Wir fassen immer einen Block von 2x2 Pixeln in einer höheren Hierarchiestufe zusammen.

Wie berechnen wir höhere Hierarchiestufen?

- Wir rendern die Occlusion-Map der höchsten Auflösung $2N * 2N$ in den Framebuffer,
- anschließend laden wir das gerenderte Bild in den Texturspeicher,
- die Textur wird auf ein Rechteck der Größe $N * N$ gerendert (bspw. durch bilineare Interpolation),
- die vier zusammengefassten Pixel werden gleichmäßig gewichtet.

Wir benötigen zur Vergrößerung der Occlusion-Maps Hardware mit Textur-Mapping-Unterstützung !



Hierarchical Occlusion Maps

Eigenschaften der Occlusion Maps

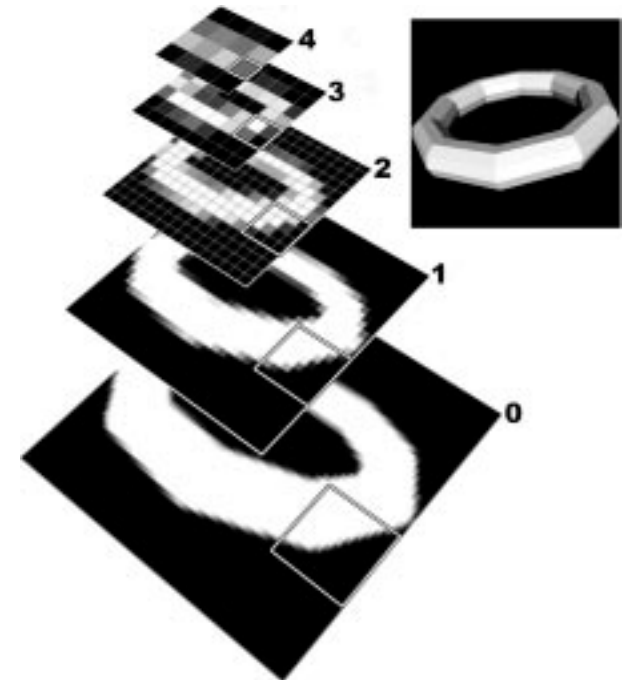


Die Occlusion-Map aggregiert eine Gruppe von Occludern.

Wie wird die Sichtbarkeit eines Objektes bestimmt?

Was können wir mit einer Occlusion-Map anfangen?

- Wir können berechnen, ob ein Objekt vollständig oder teilweise von aggregierten Occludern „überlappt“ wird.
- Wir wissen aber noch nicht, ob es durch die aggregierten Occluder vollständig verdeckt wird.
- Es fehlt noch die Tiefeninformation der aggregierten Occluder.



Hierarchical Occlusion Maps

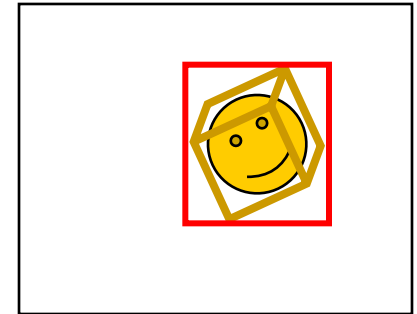
Überlappungstest



Wie berechnen wir, ob ein Objekt von der Occlusion-Map vollständig oder teilweise „überlappt“ wird?

Schritt 1: Projektion des Objektes in den 2D-Bildraum

- Wir betrachten die 3D-Boundbox (Bbox) des Objektes.
- Die Ecken der 3D-Bbox werden in den Bildraum projiziert.
- Von den projizierten Eckpunkten wird die 2D-Boundingbox im Bildraum berechnet.

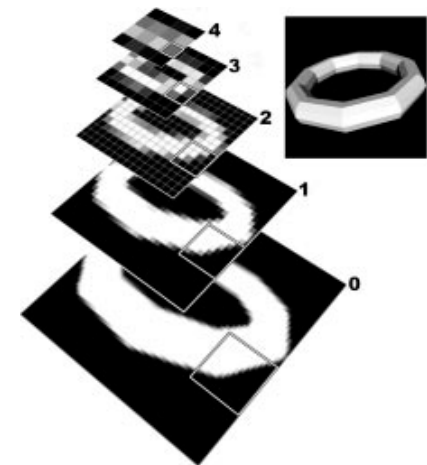
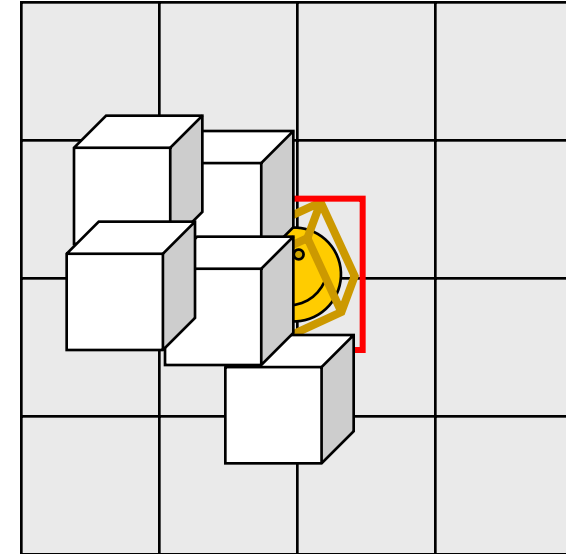


2D-Bildraum

Schritt 2: Überlappungstest mit der 2D-Bbox

Wir benutzen die Occlusion-Map Hierarchie (HOM)

- Wir starten in der HOM in dem Level, bei dem die Größe des „Pixels“ der Größe der 2D-BBox entspricht.
- Prüfe alle Pixel in der Map, die von der 2D-Bbox geschnitten werden.
- Falls ein geschnittenes Pixel der Map die 2D-Box nicht vollständig „verdeckt“, betrachte das Pixel rekursiv eine Stufe tiefer in der Hierarchie. („verdeckt“ bedeutet bei konservativem Culling ein Schwellwert =1).
- Prüfe alle Teilpixel, die die 2D-Bbox schneiden.
- Sind alle überprüften Pixel vollständig undurchsichtig, liegt das Objekt vollständig „innerhalb“ eines aggregierten Occluders.



Occluderauswahl im Preprocessing

Die Occluder werden im Preprocessing ausgewählt und in Form einer Bounding-Volume Hierarchie in einer Datenbank gespeichert.

Alle Objekte der 3D-Szene werden untersucht, in die Occluder-Datenbank kommen nur Objekte

- die eine gewisse Größe nicht unterschreiten (kleine Objekte eignen sich als Occluder weniger),
- die nicht zu komplex sind: Occluder müssen schnell gerendert werden, komplexe Objekte nehmen viel Zeit in Anspruch.

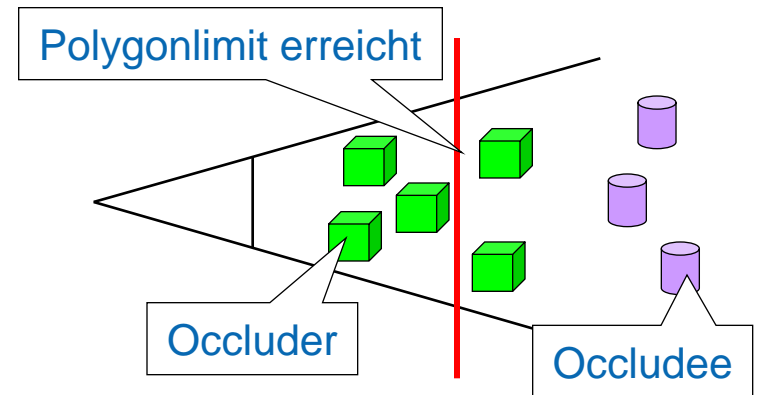
Occluderauswahl zur Laufzeit

Zur Laufzeit wählen wir Occluder, die

- möglichst nah an der Kamera stehen,
- wenig Polygone haben.

Vorgehensweise

- Bestimme alle Occluder des Frustums.
- Sortiere alle nach der Entfernung (3D-Bbox-Mittelpunkt).
- Startend von der Kameraposition, wähle sukzessive Occluder bis ein maximaler Wert an Polygonkomplexität erreicht ist.
- Die Polygonkomplexität der Occluder ist die Summe aller ihrer Polygone.

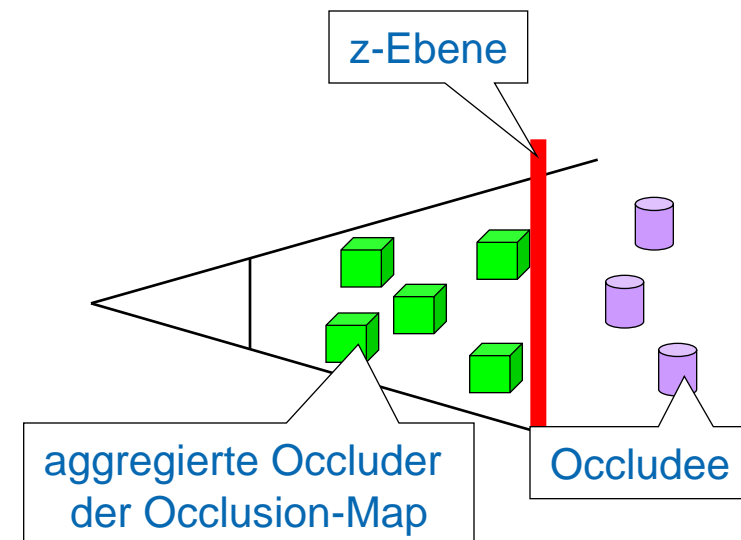


Tiefentest eines Objektes mit der Occlusion-Map

Nachdem festgestellt wurde, dass sich ein Objekt mit der Occlusion-Map vollständig überlappt, müssen wir prüfen, ob es hinter den aggregierten Occludern liegt.

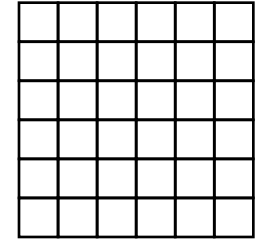
Variante 1

- Wir speichern für die Occlusion-Map den z-Wert einer z-Ebene, die die Occluder von möglichen Occludees trennt.
- Der Wert der z-Ebene ist der z-Wert des am weitesten entfernten Occluders.
Genauer: der weiteste Eckpunkt der Boundingbox.



Variante 2

Verfeinere die Idee mit einer z-Ebene und verwende mehrere z-Ebenen!

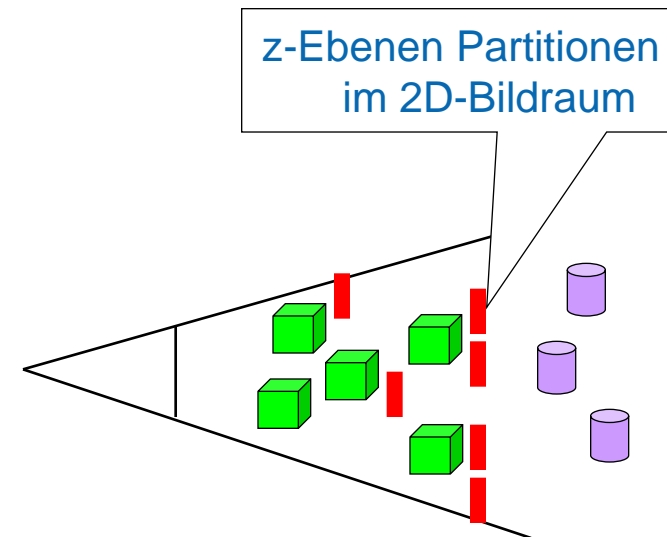


Tiefenpuffer

- Wir partitionieren den 2D-Bildraum, bspw. in 64x64 gleich große Kacheln.
- Bilde für jede Kachel des Tiefenpuffers eine z-Ebene.
- Jede Kachel enthält den z-Wert des am weitesten entfernten Occluders, der in diese Kachel projiziert wird.

Wie berechnen wir den Tiefenpuffer?

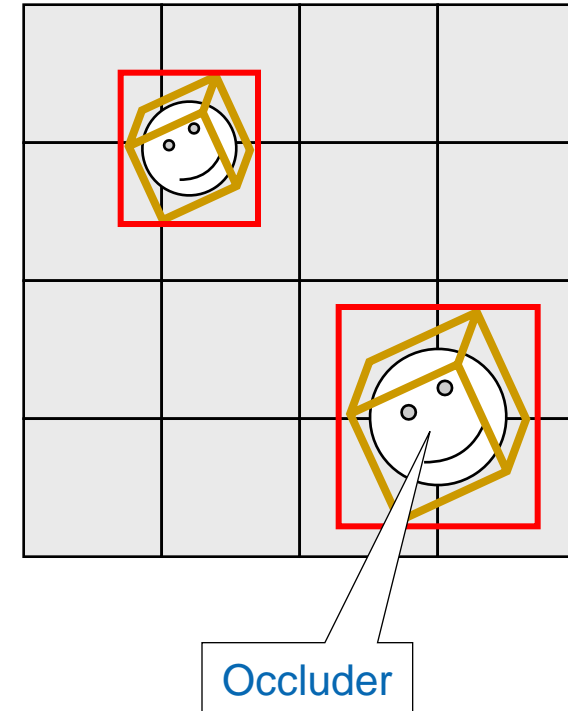
Wie wird er verwendet?



Variante 2

Berechnung des Tiefenpuffer

- Wir projizieren die Ecken der 3D-Bbox in den 2D-Bildraum.
- Von den Ecken bilden wir die 2D-Bbox.
- Zur Aktualisierung verwenden wir den Z-Wert der am weitesten entfernten Ecke der 3D-Bbox.
- Aktualisiert werden alle Einträge des Tiefenpuffer, die von der 2D-Bbox geschnitten werden.



Hierarchical Occlusion Maps

Tiefentest

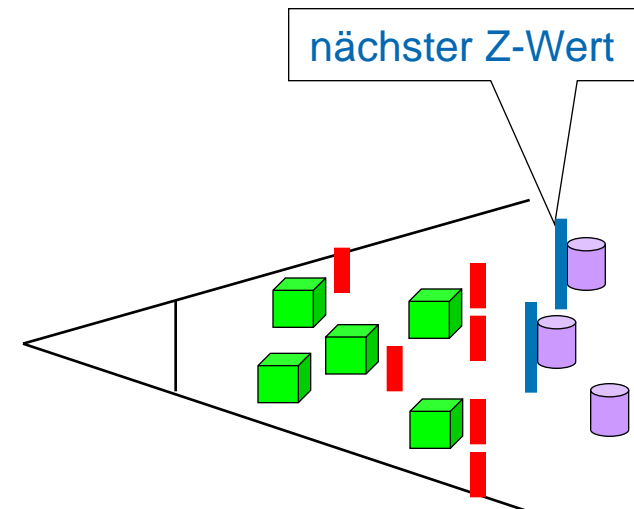
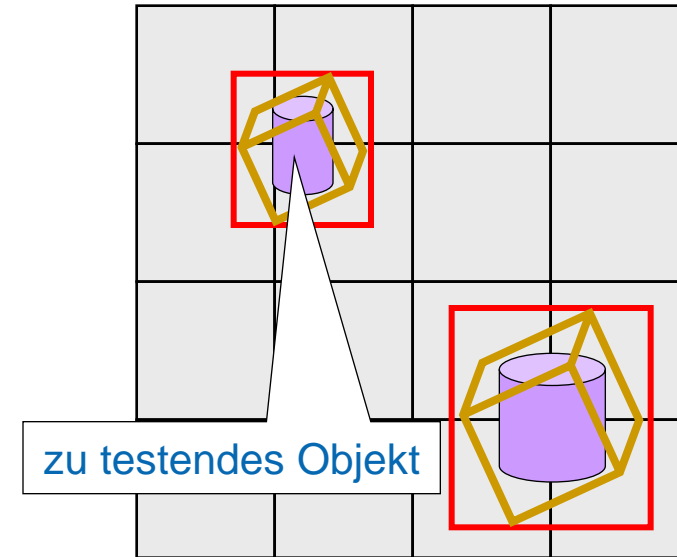


Variante 2

Verwendung des Tiefenpuffer

„Occludee-Test“

- Wir projizieren die Ecken der 3D-Bbox des zu testenden Objektes in den 2D-Bildraum.
- Von den Ecken bilden wir die 2D-Bbox.
- Wir berechnen den nächsten Z-Wert aller Ecken der 3D-Bbox.
- Wir prüfen diesen Wert mit allen Einträgen des Tiefenpuffers, die von der 2D-Bbox geschnitten werden.



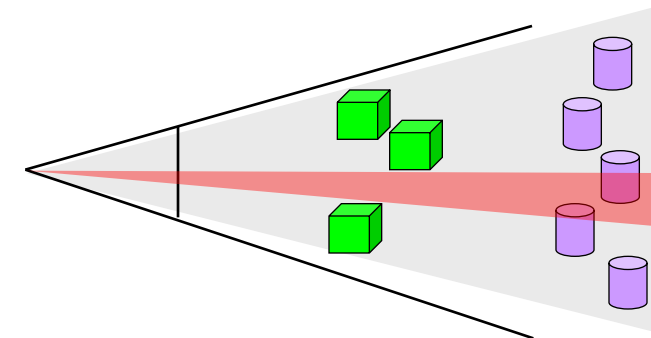
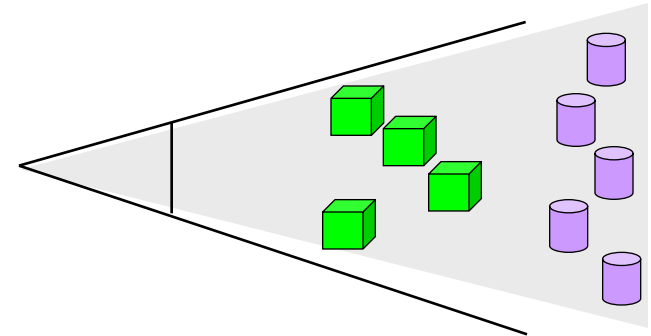
Approximatives Visibility Culling

Was bedeutet hier approximatives Visibility Culling?

- Wir betrachten kleine Löcher in der Occlusion-Map.
- Diese Löcher behandeln wir, als wären sie durch Occluder „geschlossen“.
- Wir werden folglich Objekte nicht darstellen („cullen“), die durch diese Löcher sichtbar sind.

Warum tun wir das?

- Beschleunigung des Rendering
- kleine Löcher fallen in der Darstellung kaum auf



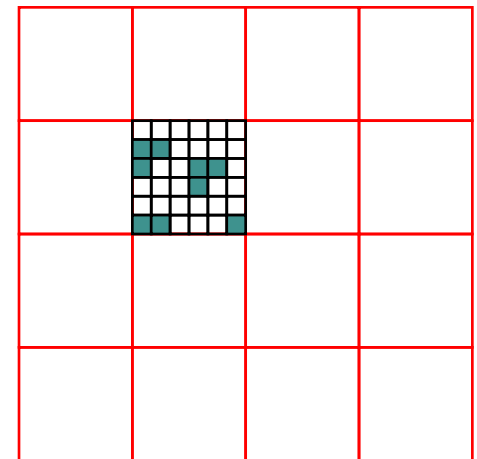
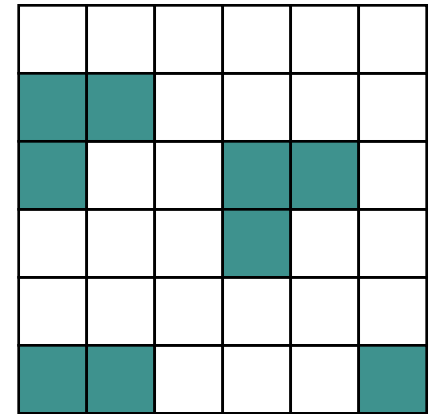
Hierarchical Occlusion Maps

Approximatives Culling



Wie führen wir mit HOM approximatives Culling durch?

- Bisher verwenden wir immer einen Undurchsichtigkeitswert von 1.
- Wir lassen nun einen Undurchsichtigkeitswert < 1 als Schwellwert zu.
- Je kleiner der Schwellwert, desto größer der zugelassene Fehler.



Hierarchical Occlusion Maps

Approximatives Culling



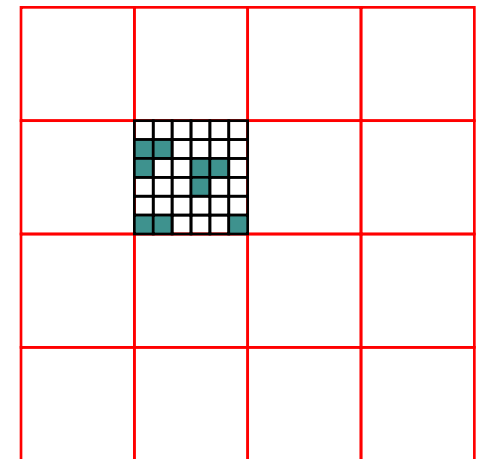
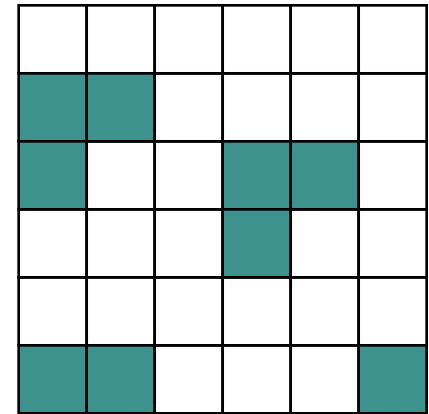
Vorgehensweise

- Wir überlegen uns zuerst, wie groß ein Loch werden darf.
- Wir berechnen den Schwellwert für die Ebenen der Hierarchie.

Beispiel

- Bildschirmauflösung $1024 * 1024$ Pixel,
Occlusion-Map $64 * 64$ Pixel
- jedes Pixel der Occlusion-Map entspricht $16 * 16$ Pixeln im gerenderten Bild
- Wir wollen auf einer Fläche von $16 * 16$ Pixel ein Loch von 25 Pixeln zulassen

$$\rightarrow \text{Schwellwert} = 1 - \frac{25}{16*16} = 0,9$$



Hierarchical Occlusion Maps

Approximatives Culling



Allgemein gilt:

- Lässt man n fehlerhafte Pixel bei insgesamt m Pixeln einer Hierarchie der Ebene k zu, dann erhält man als Schwellwert

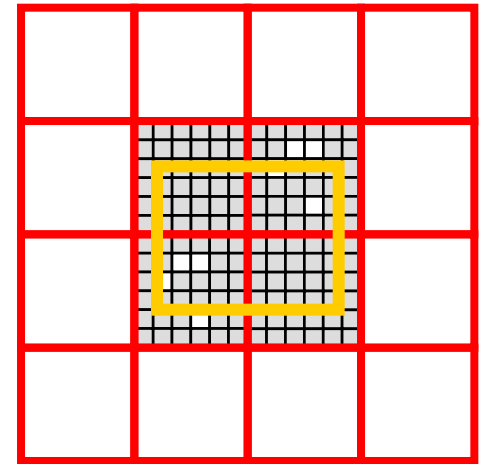
$$O_k = 1 - \frac{n}{m}.$$

- Auf Hierarchieebene $k + 1$ ergibt sich dann die vierfache Menge an Pixeln, der Schwellwert ergibt sich zu:

$$O_{k+1} := 1 - \frac{n}{4m} = 1 - \frac{1 - O_k}{4} = \frac{3 + O_k}{4}$$

Zur Laufzeit

- Beim Überlappungstest testen wir die 2D-Bbox des Objektes mit einem Pixel der Occlusion-Map.
- Liegt der Undurchsichtigkeitswert über dem gewählten Schwellwert, brechen wir die Berechnung ab.
- Wir betrachten das Objekt so, als würde es vollständig von den Occludern überlappt.



Hierarchical Occlusion Maps

Approximatives Culling



Beispiel

- blau:
Occluder
- weiß:
gerendert
- rot:
durch Culling weggefallen

