Rest des Kurses

- Heute: Gouraud+Phong-Interpolation; Schatten
- Freitag 28.6.2024: Schattenfilterung
- Montag 1.7.2024: Monte-Carlo-Rendering 1 (Integralschätzer, Path Tracing)
- Freitag 5.7.2024: Monte-Carlo-Rendering 2 (Importance Sampling)
- Montag 8.7.2024: Storytelling Workshop
- Freitag 12.7.2024: Storytelling Reprise; tbd
- Montag 15.7.2024: Zusammenfassung + Enacom Vorstellung
- Dienstag 16.7.2024, 23:59: Vorschauvideo fällig
- Donnerstag 18.7.2024, 23:59: Abgabe der finalen Animation
- Freitag 19.7.2024: Computer Animation Festival

- Mittwoch 7.8.2024, 9:00: Klausur 1, Ort wird noch bekanntgegeben
- Montag 9.9.2024: Klausur 2, Ort wird noch bekanntgegeben

UNIVERSITÄT BONN

RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT BONN

INSTITUT FÜR INFORMATIK DER UNIVERSITÄT BONN

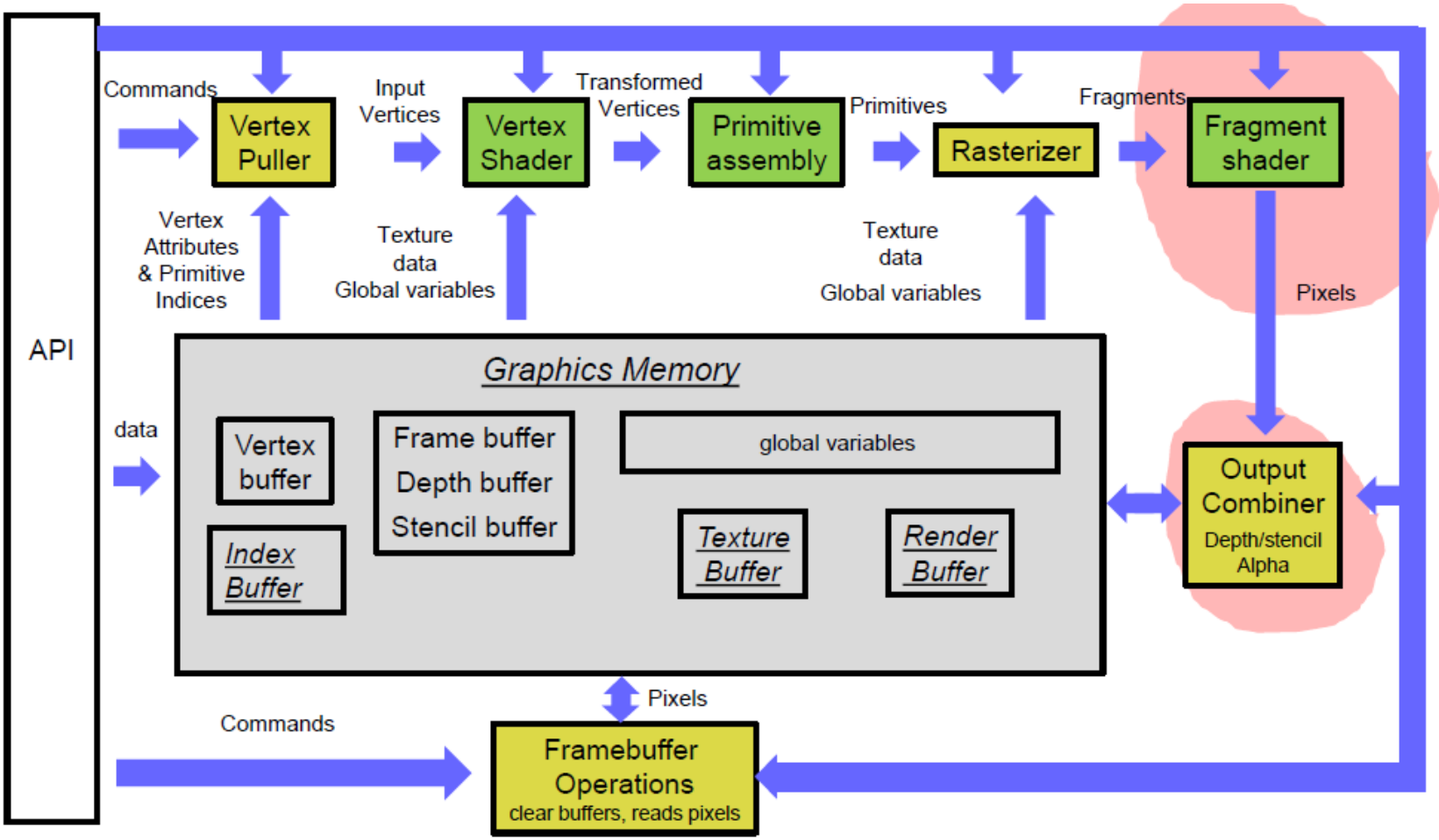
Einführung in die Computergrafik

Kapitel 17: Schatten

Prof. Dr. Matthias Hullin

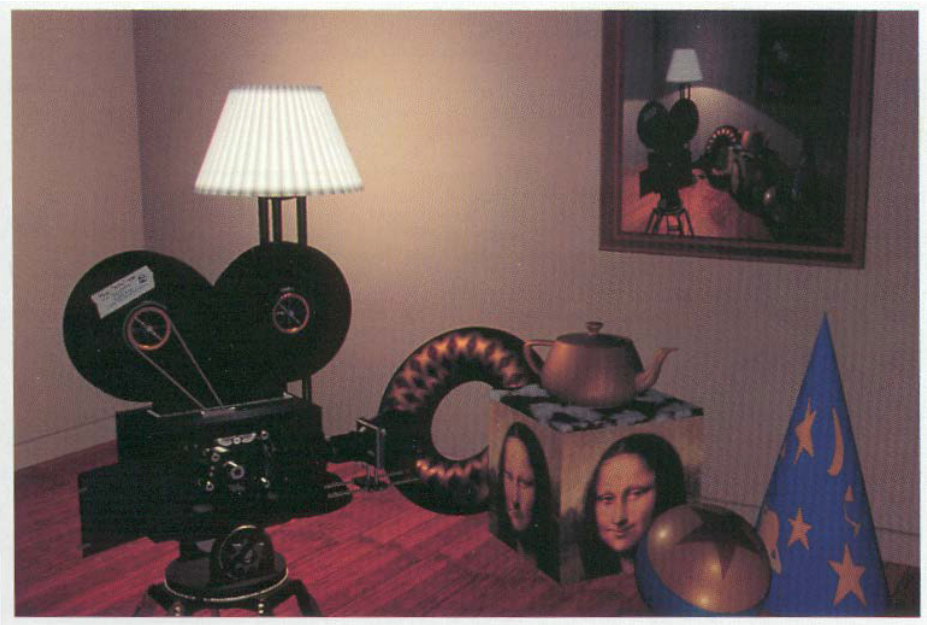
Institut für Informatik
Abteilung 2: Visual Computing
Universität Bonn

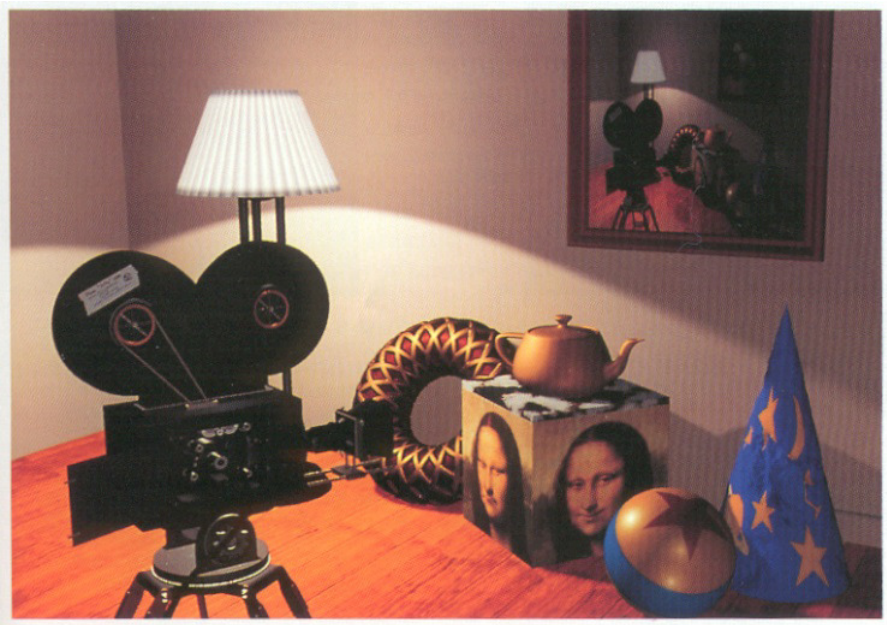
24. Juni 2024


API
Commands
Vertex Puller
Input Vertices
Vertex Shader
Transformed Vertices
Primitive assembly
Primitives
Rasterizer
Fragments
Fragment shader
Vertex Attributes & Primitive Indices
Texture data Global variables
Texture data Global variables
Pixels
Graphics Memory
data
Vertex buffer
Index Buffer
Frame buffer
Depth buffer
Stencil buffer
global variables
Texture Buffer
Render Buffer
Output Combiner
Depth/stencil Alpha
Pixels
Commands
Framebuffer Operations
clear buffers, reads pixels

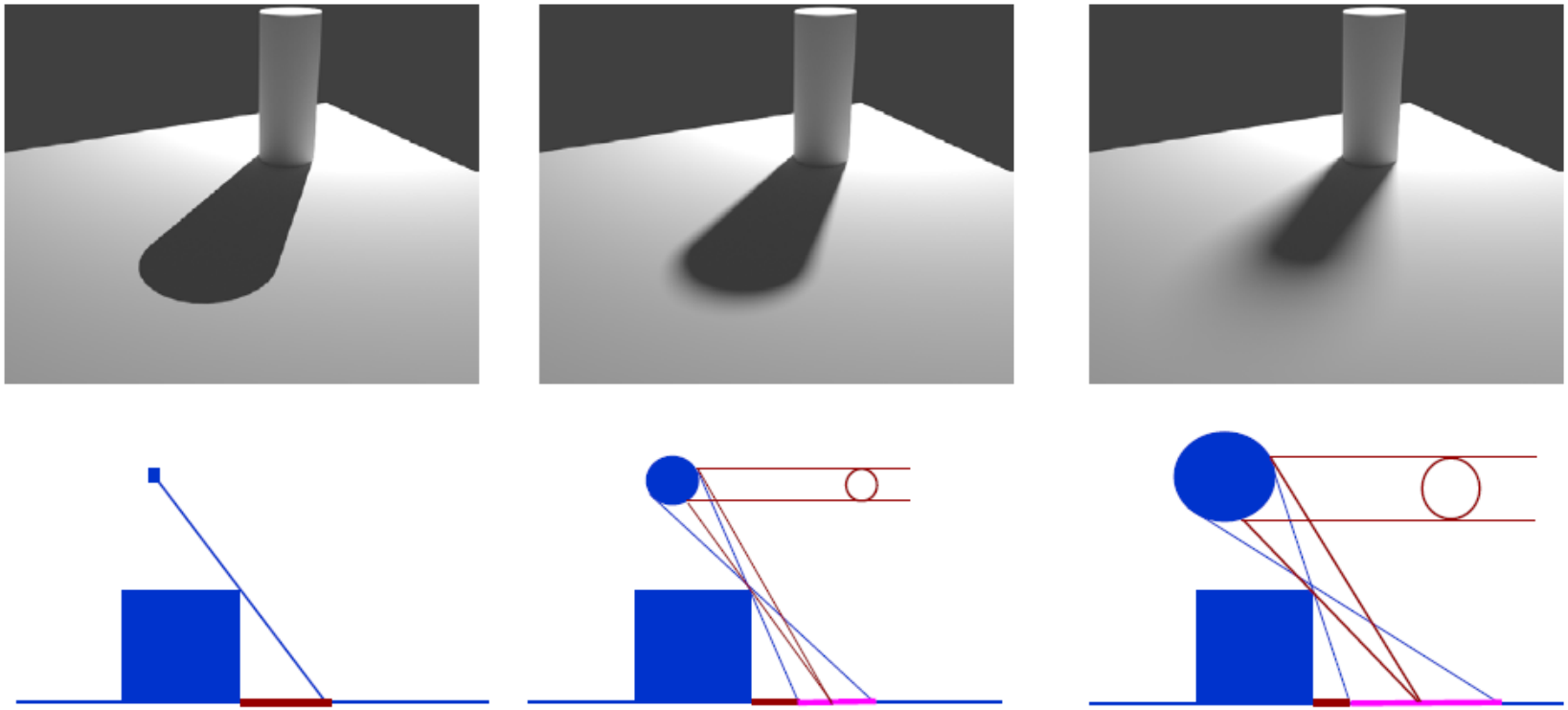
Wozu sind Schatten gut?

- Szenen ohne Schatten sehen künstlich aus
- Schatten helfen die räumliche Lage von Objekten abzuschätzen
- Schatten geben Hinweise auf die Lichtquellen-position





Harte und weiche Schatten



Geometrische Schatten

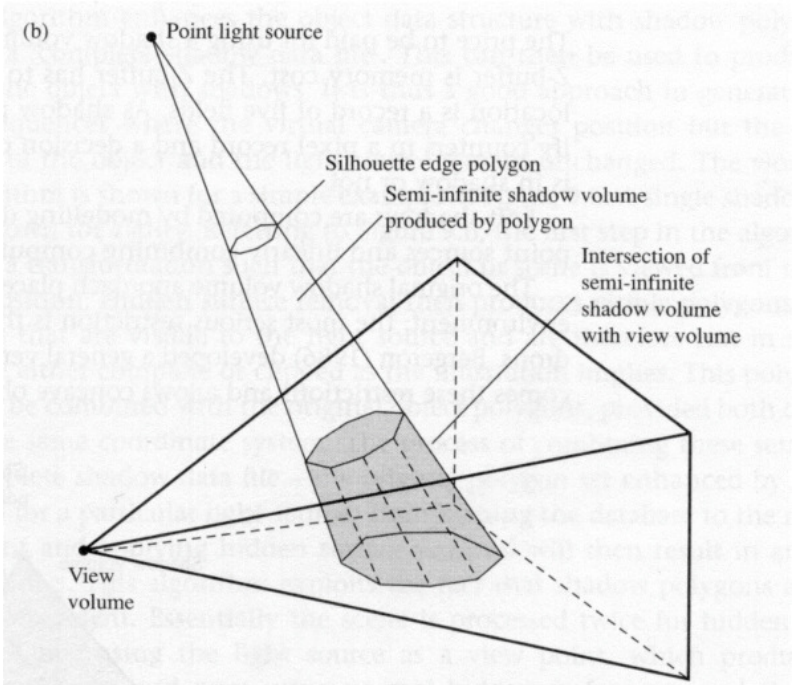
Folgende Schatteneigenschaften sind wichtig für CG:

- Punktlichtquellen erzeugen harte Schatten.
- Wenn Aug- und Punktlichtposition gleich sind, sind keine Schatten sichtbar.
  - Sichtbarkeitsalgorithmen zur Schattenberechnung verwendbar.
- Schatten in statischen Szenen sind unabhängig vom Betrachter.

Theorem (Schattenvolumen)

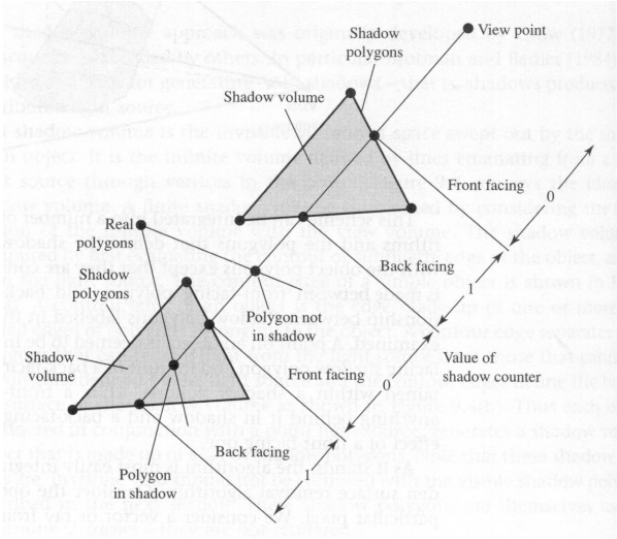
*Das **Schattenvolumen** von einem Objekt bzgl. einer Punktlichtquelle q enthält alle Punkte p, deren Verbindungslinien pq das Objekt schneiden und sich innerhalb des Sichtfrustums befinden.*

- Das Schattenvolumen wird von der **Silhouette** des Objektes bzgl. der Lichtquellenposition definiert.
- Jedes Objekt erzeugt bzgl. jeder Lichtquelle ein Schattenvolumen.
- Die Randpolygone des Schattenvolumens nennt man **Schattenpolygone**.
- Ein Punkt auf einem Objekt liegt im Schatten, wenn er innerhalb des Schattenvolumens ist.
- In diesem Fall liegt er hinter einem dem Betrachter zugewandten Schattenpolygon und vor einem abgewandten



Schattenvolumen

- Diese Konstellation kann leicht mit Sichtbarkeitsalgorithmen überprüft werden. Die Berechnung erfolgt pro Pixel.
- Jedes Pixel besitzt einen Zähler, der 0 ist, genau dann wenn der zu sehende Punkt im Licht liegt.
- Der Zähler wird um 1 erhöht, wenn ein zugewandtes Schattenpolygon geschnitten wird und bei abgewandten um 1 erniedrigt.



Dieser Algorithmus kann z.B. mit dem z-Buffer kombiniert werden (Brotman, Badler, IEEE Computer Graphics and Applications 1984).

Nachteile:

- Schattentest aufwändig (zusätzliche Schattenpolygone)
- auf geschlossene Geometrien beschränkt
- Silhouetten schwer zu bestimmen

Stencil-Buffer Algorithmus (Heidmann 1991, mehrere Durchläufe)

1. Lösche Farb-, z- und Stencil-Buffer.
2. Zeichne Szene nur mit ambienter und emissiver Belechtung (Farb- und z-Buffer enthalten Werte des vordersten Fragments).
3. Zeichne Schattenpolygone, aber aktualisiere nur den Stencil-Buffer:
   - zuerst mit Backface Culling um Wert für vordere Polygone zu inkrementieren;
   - dann mit Frontface Culling um den Wert zu dekrementieren. (Danach ist der Stencilwert gleich Null für beleuchtete Pixel und größer Null für Pixel im Schatten.)
4. Zeichne Szene erneut mit:
   - der entsprechenden Lichtquelle,
   - aktiviertem Stencil-Test, um nur Pixel mit Stencil-Wert Null zu zeichnen (also Pixel, die nicht im Schatten liegen),
   - eingeschaltetem z-Test (um nur sichtbare Fragmente zu zeichnen).
   - Der Beitrag der Lichtquelle kann mit Blending aufaddiert werden.
5. Lösche Stencil Buffer und wiederhole 3. und 4.
   - $2n + 1$ Durchläufe für n Lichtquellen

Hauptproblem:

- Schattenvolumen können große Flächen des Bildschirms vielfach (z.B. 100-fach) überdecken.
- Erfordert sehr hohe Füllrate.
- Laufzeit hängt stark von geometrischer Komplexität ab.

Andere (lösbare) Probleme:

- Near und Far-Clippingebene dürfen Schattenvolumen nicht abschneiden.
- Silhouetten von nicht wasserdichter Geometrie ermitteln ist schwierig.

Vorteil:

- nicht Bild-basiert, daher kein Aliasing

Shadow Mapping

Shadow Mapping ist der am weitesten verbreitete Ansatz zur Darstellung dynamischer Schatten.

Vorteil gegenüber Schattenvolumen:

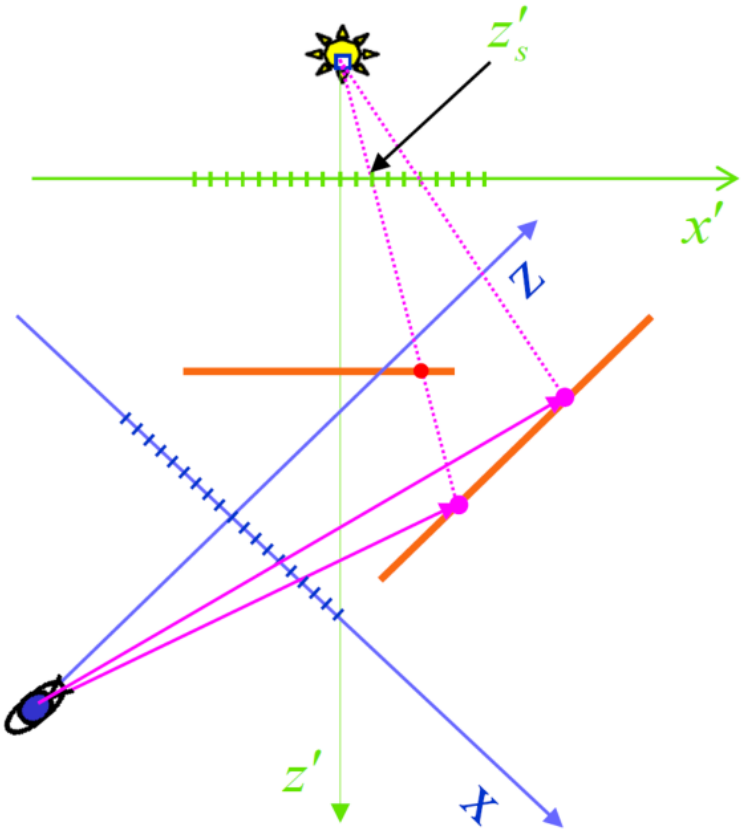
- Die Laufzeit ist weitgehend unabhängig von der geometrischen Komplexität.
- Jede Geometrie, die man rendern kann, kann (ohne Weiteres) Schatten werfen.

Nachteil gegenüber Schattenvolumen:

- Es treten Diskretisierungsartefakte auf.

Algorithmus benötigt zusätzlichen z-Buffer pro Lichtquelle (Shadow Map genannt). Die Schritte sind wie folgt:

1. Die Szene wird aus der Perspektive der Lichtquelle in die Shadow Map gezeichnet. Gespeichert wird die Tiefe.
2. Die Szene wird vom Augpunkt aus gezeichnet. Für jedes Fragment, werden seine Koordinaten (x, y, z) aus dem Bildschirmkoordinatensystem in Koordinaten (x', y', z') bzgl. der Lichtquelle transformiert.
3. Die x', y' Koordinaten werden zur Addressierung der Shadow Map benutzt. Wenn z' größer als der gespeicherte Wert z'_s ist, liegt der Punkt im Schatten.



Beispiel:

Abbildung: Berechnete Tiefe z'

Abbildung: Shadow Map (Sicht von Lichtquelle)

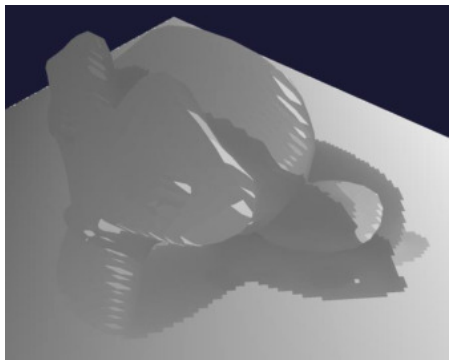


Abbildung: Tiefe aus Shadow Map z'_s

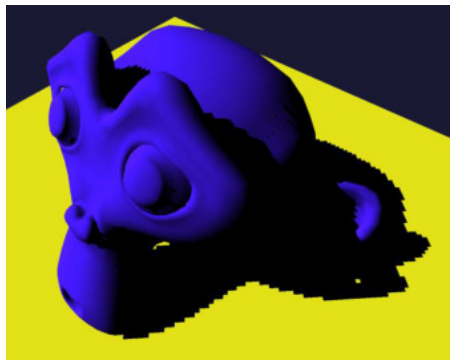
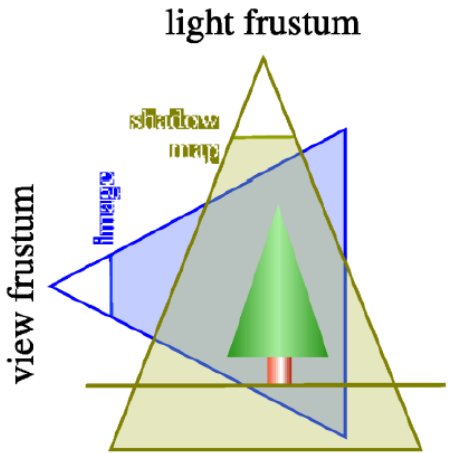
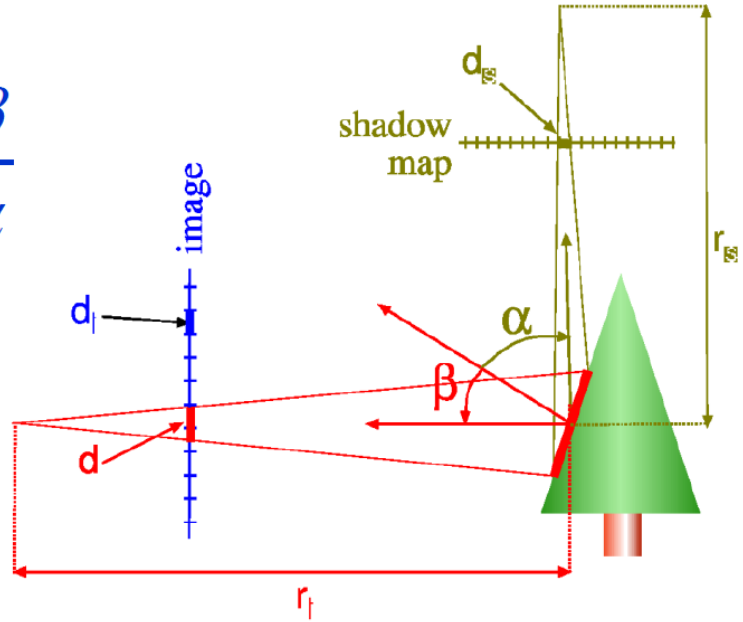


Abbildung: Szene mit Schatten



$$d = d_s \frac{r_s}{r_i} \frac{\cos\beta}{\cos\alpha}$$

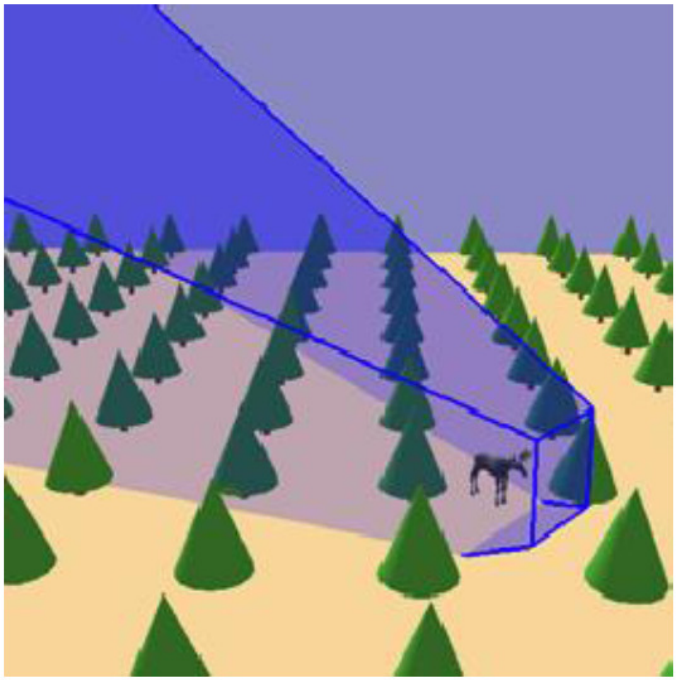


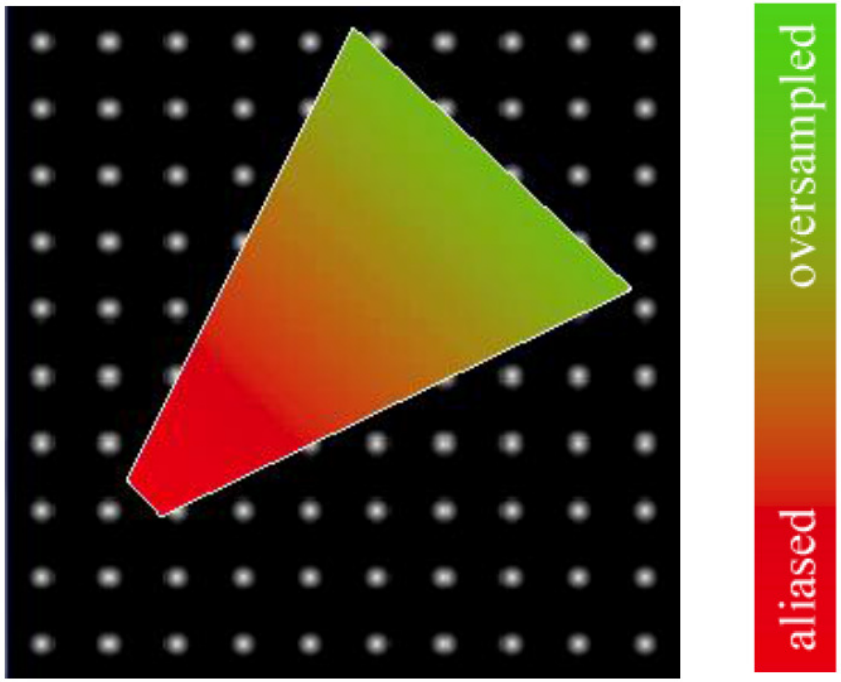
Die Auflösung der Shadow Map ist unzureichend, wenn $d \geq d_i$:

- **Perspektivisches Aliasing:** $d_s \frac{r_s}{r_i}$ groß, wenn Betrachter nahe am Objekt
- **Projektives Aliasing:** $\frac{\cos(\beta)}{\cos(\alpha)}$ groß, wenn das Licht nahezu parallel zur Fläche ist

Perspektivisches Aliasing:

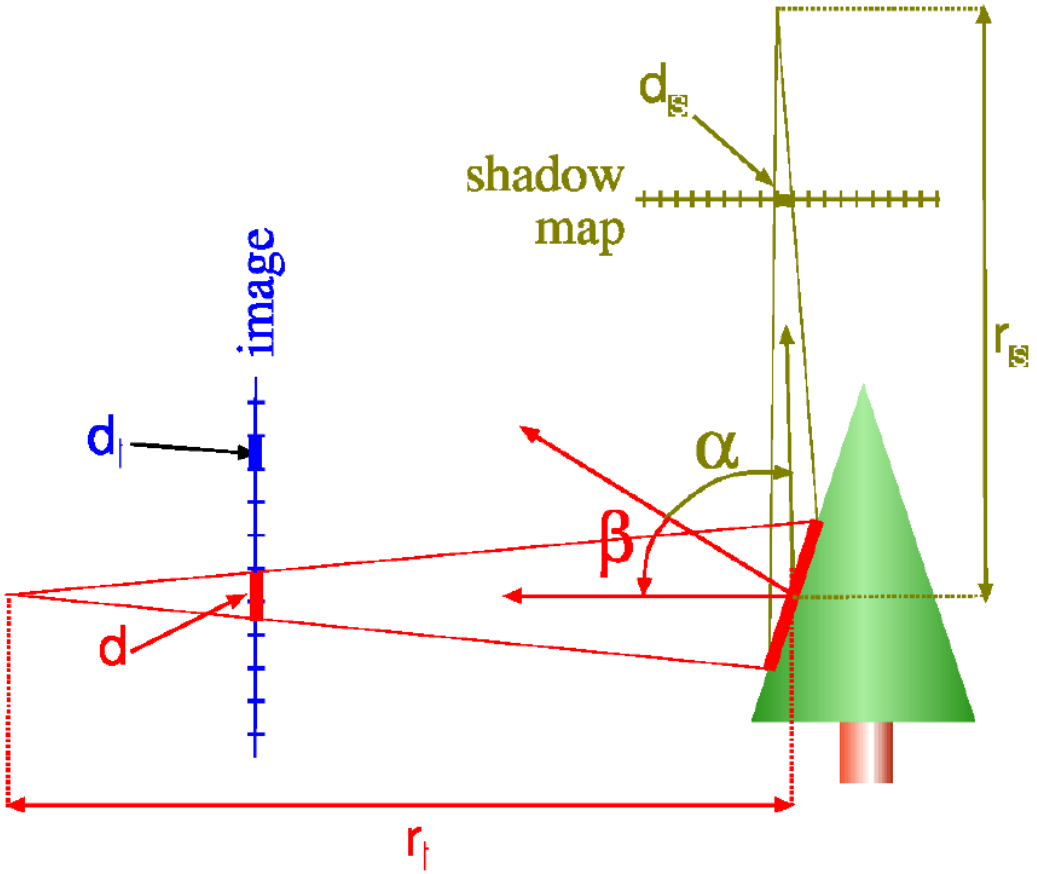
- Blickpunktabhängig
- Nahe beim Betrachter: Aliasing
- In der Entfernung: Oversampling





Projektives Aliasing:

- Passiert überall, wenn das Licht fast parallel einfällt
- Hängt vom Winkel des einfallenden Lichts ab
- Sehr lokal



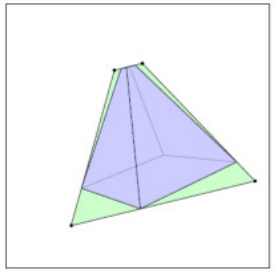
Trapezoidal Shadow Maps beseitigen perspektivisches Aliasing wie folgt:

1. Generiere eine naive Projektion für die Shadow Map und wende sie auf die Eckpunkte des Sicht-Frustums an.

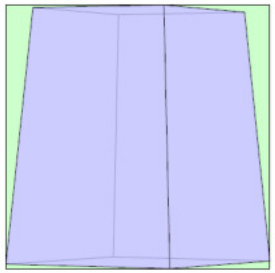
2. Erzeuge ein Trapez, das das projizierte Sicht-Frustum einschließt.

3. Wende eine weitere perspektivische Projektion an damit das Trapez die komplette Shadow Map überdeckt (einfache Matrixmultiplikation).

Sicht-Frustum und Trapez
vom Licht aus gesehen

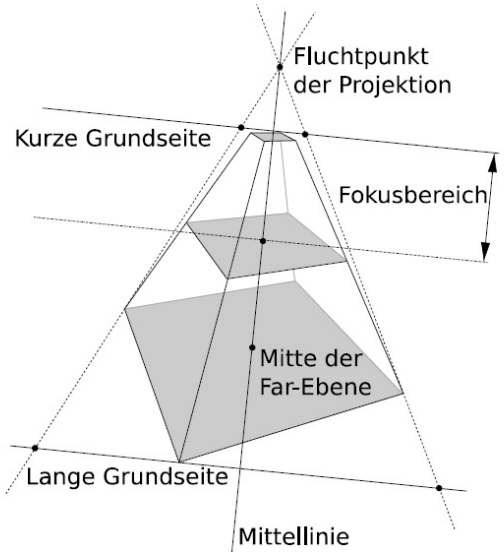


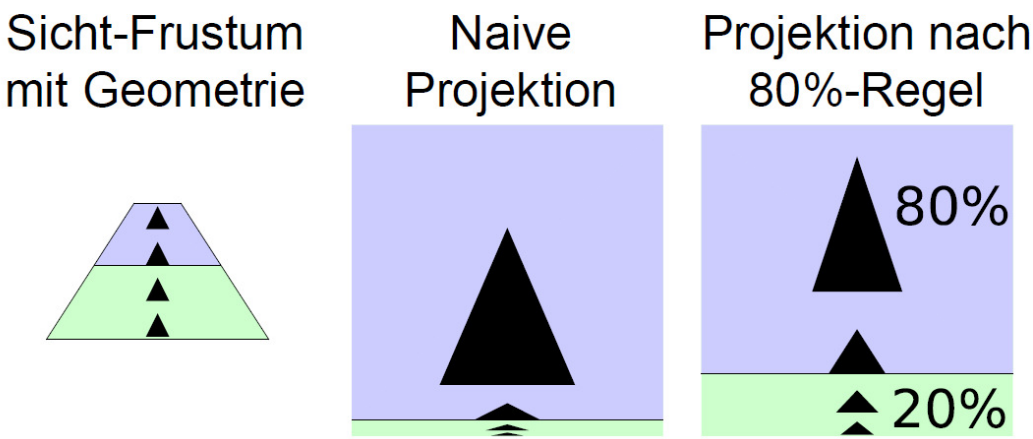
Sicht-Frustum und Trapez
nach Projektion



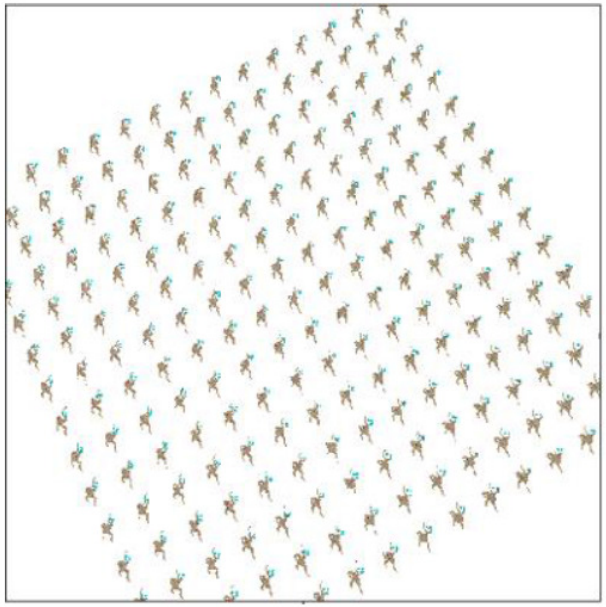
Das Trapez hat:

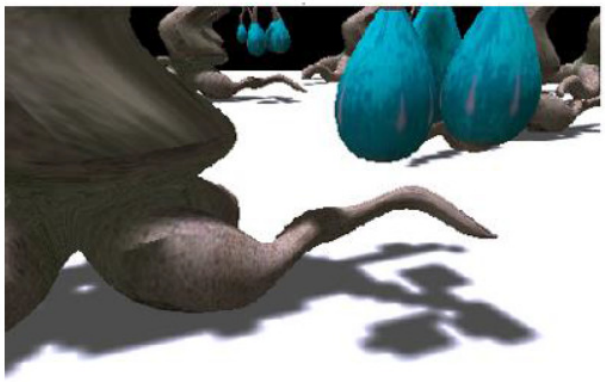
- eine Mittellinie, die durch die Kamera und die Mitte der Far-Clippingebene geht.
- Grundseiten, die das Sicht-Frustum berühren.
- einen Fluchtpunkt durch den 80% der Shadow Map einen Fokusbereich nahe der Kamera überdecken.



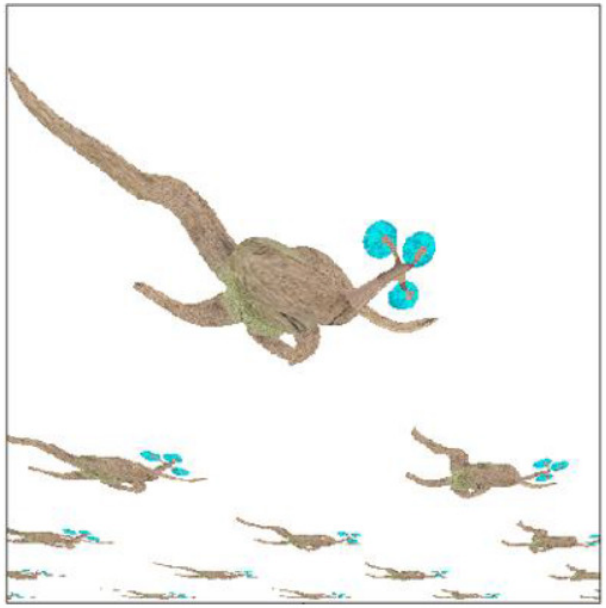


Normale Projektion



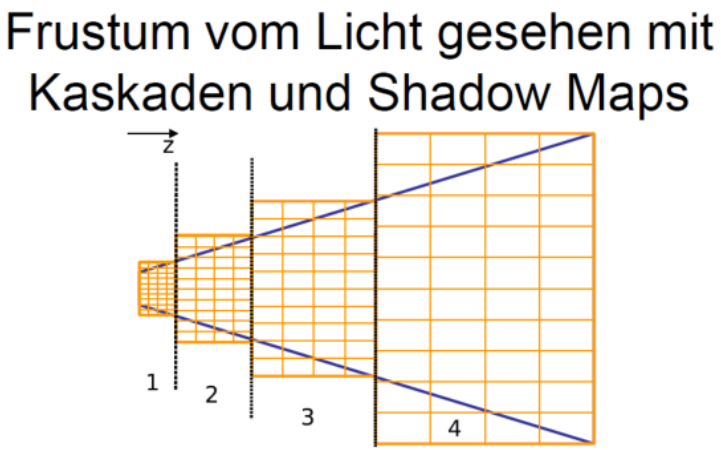


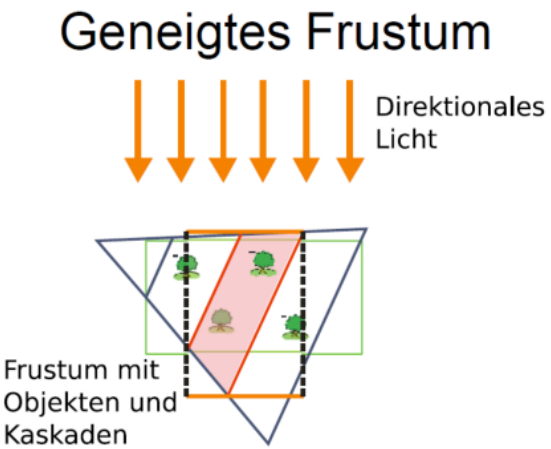
Trapezoidal Shadow Maps



Trapezoidal Shadow Maps helfen nicht wenn das Licht von hinten kommt (Sicht-Frustum wird zu Rechteck). Gerade für große Außenareale sind dann **Cascaded Shadow Maps** besser geeignet.

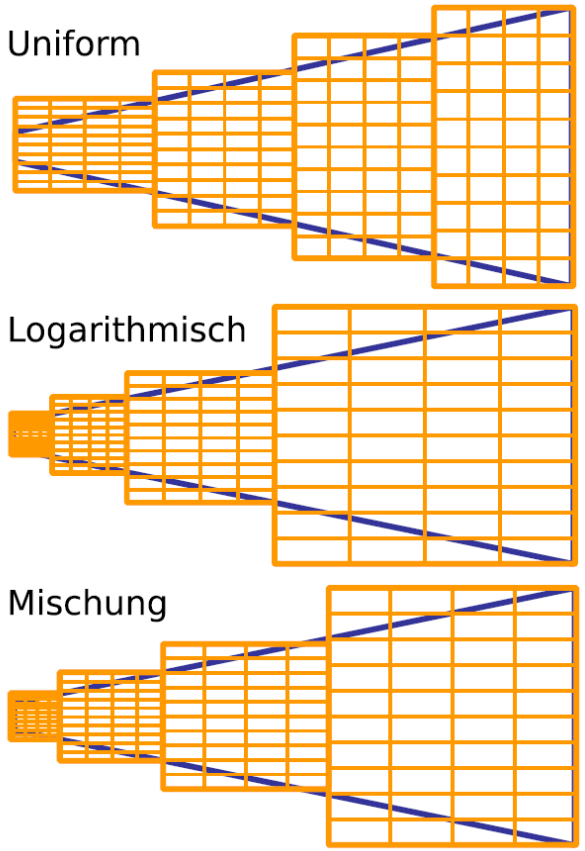
Cascaded Shadow Maps: Das Sicht-Frustum wird entlang Ebenen mit konstantem z in Kaskaden aufgeschnitten. Jede Kaskade erhält ihre eigene Shadow Map.





Tiefen für Kaskaden können wie folgt gewählt werden:

- **Uniform:** Alle Kaskaden haben die gleiche Länge entlang der z-Achse.
  - Zu geringe Auflösung für nahe Bereiche.
- **Logarithmisch:** Die i-te Kaskade hat Länge $L \cdot c_i$.
  - Zu hohe Auflösung für nahe Bereiche.
- **Mischung:** Die Längen aus den obigen Verfahren werden linear kombiniert.
- **Adaptiv:** Der z-Buffer wird in jedem Frame analysiert um die Kaskaden optimal an die sichtbare Geometrie anzupassen.
  - Sehr gut aber kompliziert.



Cascaded Shadow Maps

Auf modernen GPUs können alle Shadow Maps simultan gerendert werden. Beim Rendern der Szene wird im Fragment Shader berechnet welche Shadow Map benutzt werden sollte und mit dieser normales Shadow Mapping durchgeführt.

Szene mit Cascaded Shadow Maps

Farbkodierte Kaskaden

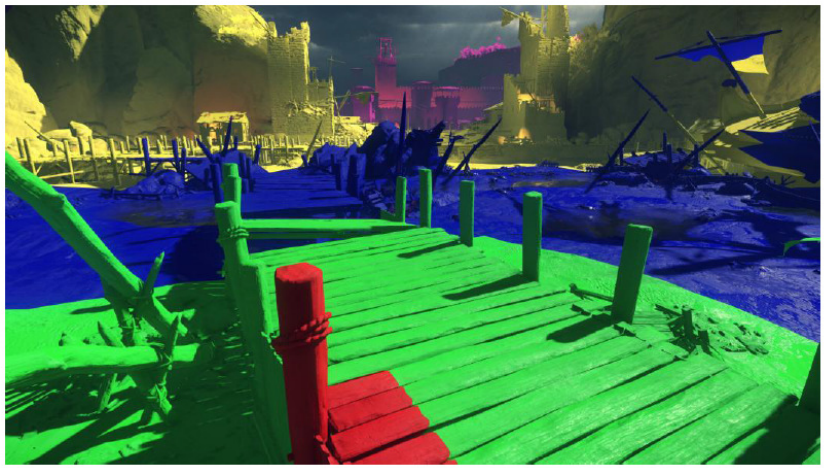
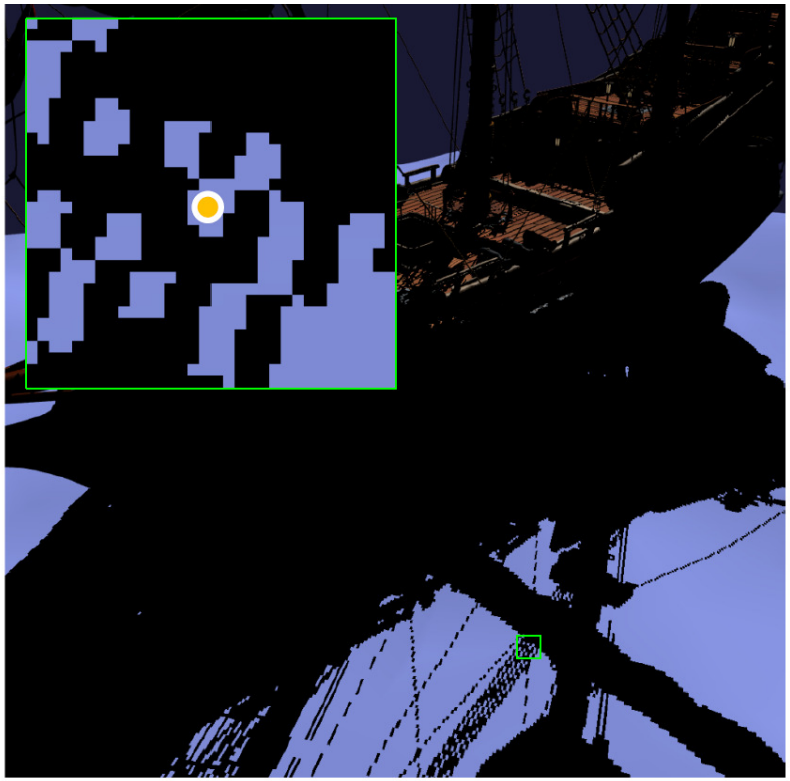


Bild von Crytek GmbH

Problem: Die Auflösung der Shadow Map kann oft nicht hoch genug sein, um die Szenengeometrie abzubilden. Zu rendernde Fragmente liegen im Allgemeinen *teilweise* im Schatten. Berücksichtigt man dies nicht, erhält man eine weitere Form von Aliasing-Artefakten:



Percentage Closer Filtering

Idee: Wende Texturfilterung auf die Shadow Map an.

Aber: Die Shadow Map direkt zu filtern würde Geometrie glätten statt Schattenränder zu glätten. Man würde:

1. Tiefenwerte zu Filterkernel holen.
2. Tiefenwerte filtern.
3. Schattentest mit gefilterter Tiefe durchführen.

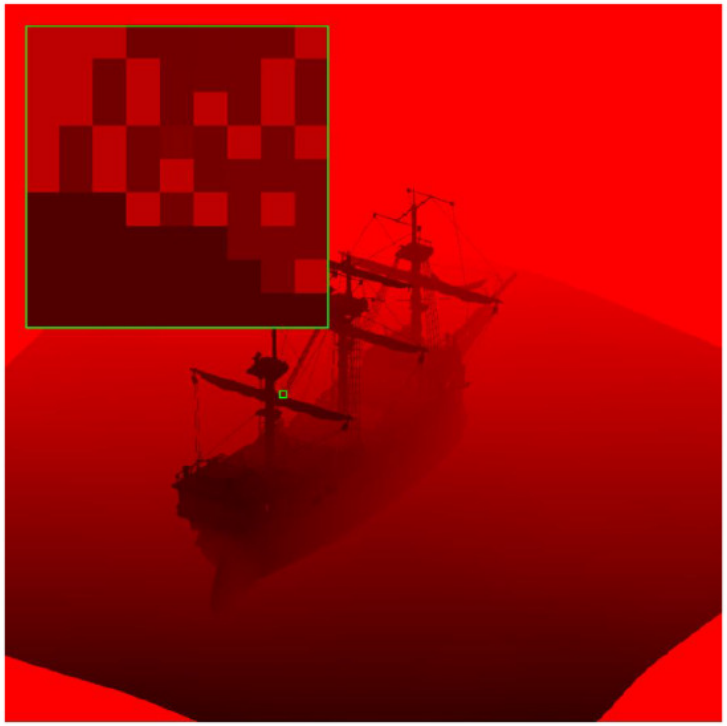
Percentage Closer Filtering (PCF) arbeitet wie folgt:

1. Tiefenwerte zu Filterkernel holen.
2. Schattentest mit allen Tiefenwerten durchführen.
3. Binäre Schattenwerte filtern.

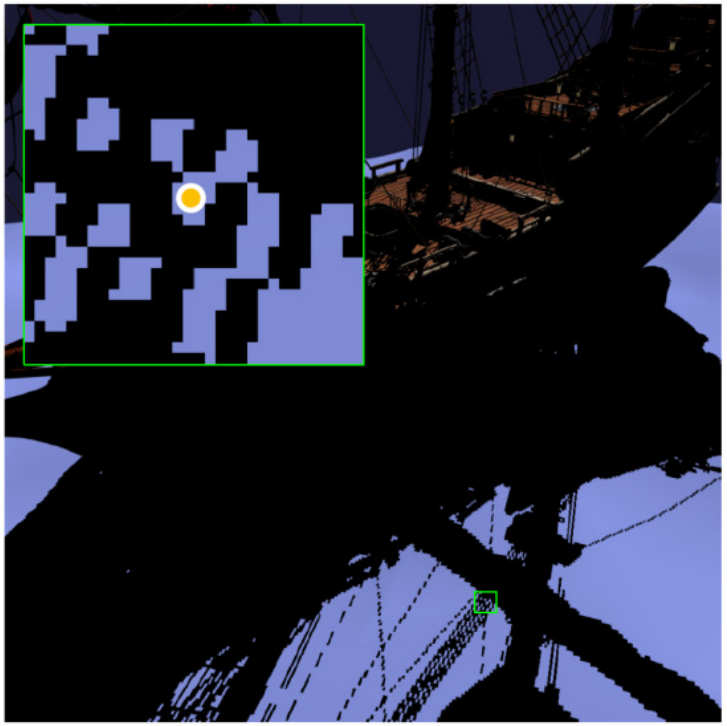
Percentage Closer Filtering (PCF):

1. Tiefenwerte zu Filterkernel holen.

Shadow Map und Filterbereich



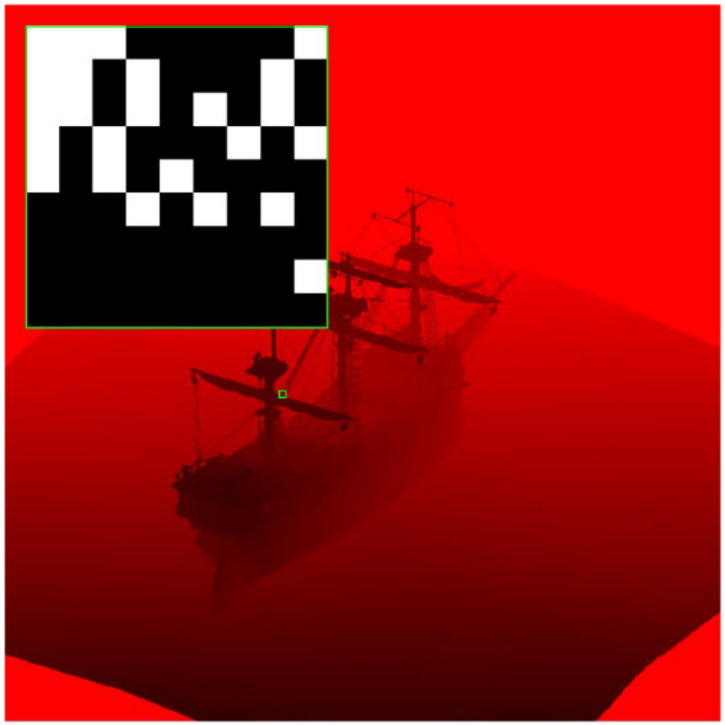
Szene und Fragment



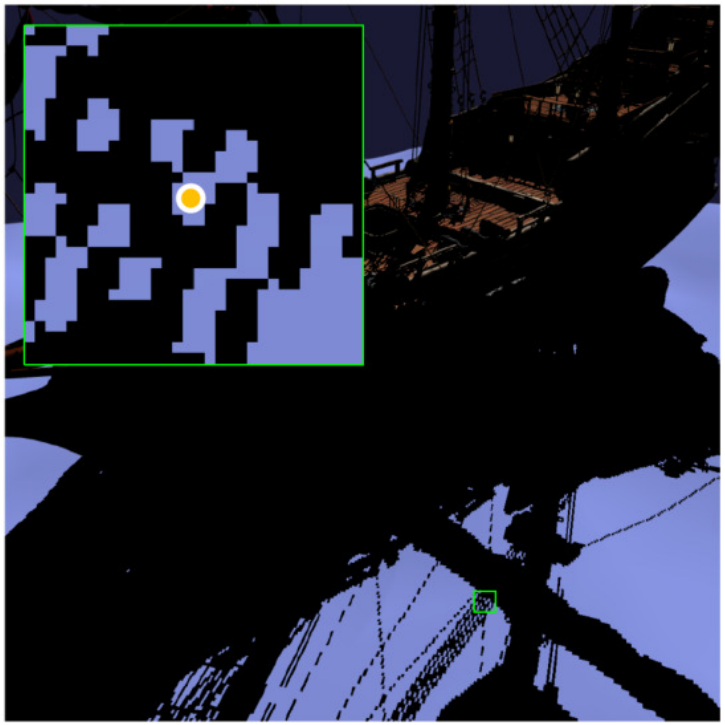
Percentage Closer Filtering (PCF):

2. Schattentest mit allen Tiefenwerten durchführen.

Shadow Map und Filterbereich

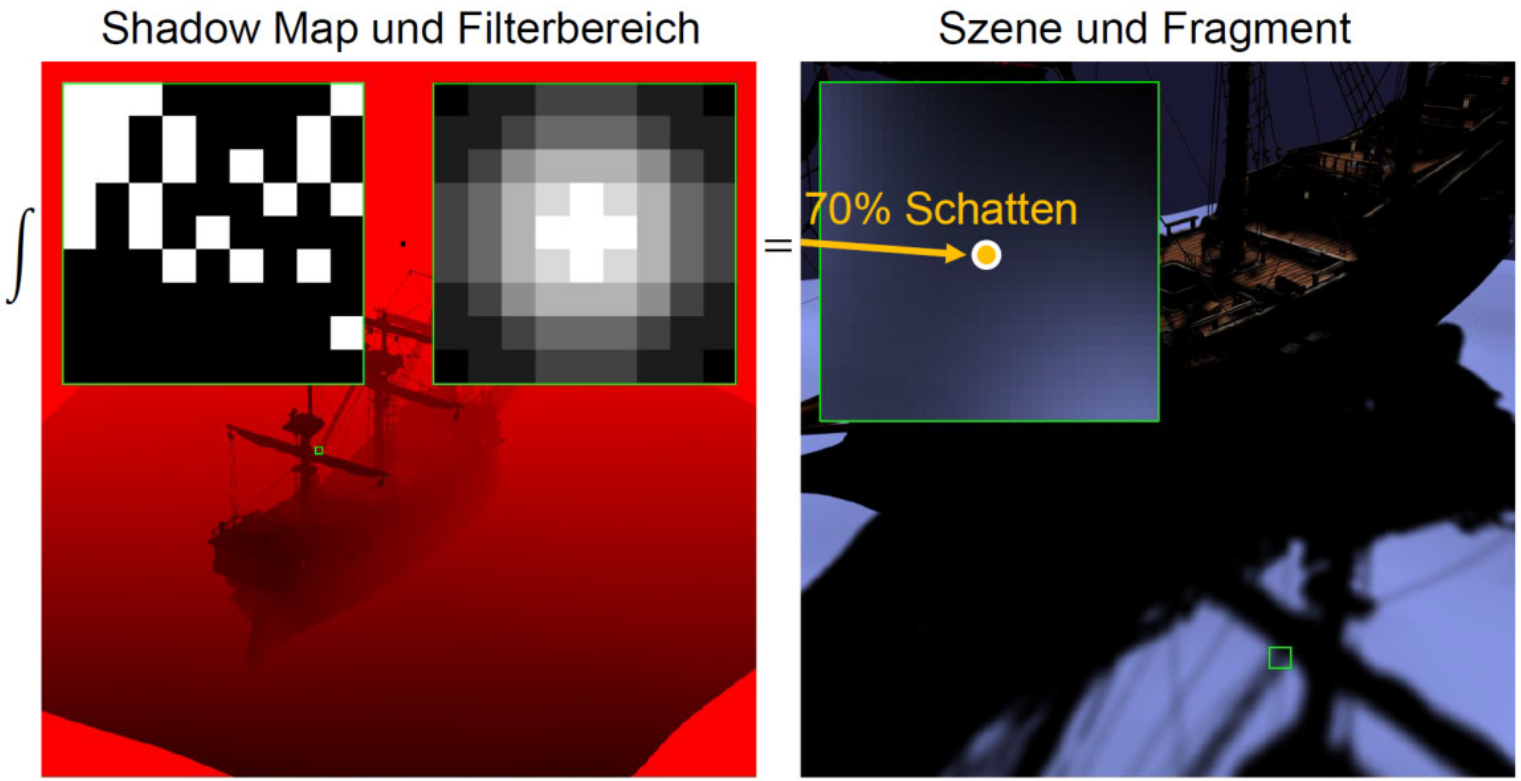


Szene und Fragment



Percentage Closer Filtering (PCF):

3. Binäre Schattenwerte filtern.



Percentage Closer Filtering

Implementierung von PCF erfolgt im Fragment Shader. Für 2×2 Kernel (bilineare Filterung) ist PCF in Hardware implementiert.

Probleme:

- Pro Fragment muss man viele Tiefenwerte abfragen.
- Man kann Filter nicht direkt auf die Shadow Map anwenden:
  Kein Mip Mapping, anisotrope Filterung, Konvolution, Multisample Antialiasing, etc.

Trotzdem ist PCF weit verbreitet.

Variance Shadow Maps

Variance Shadow Maps (VSMs) sind modifizierte Shadow Maps, die man direkt filtern kann. Eine VSM hat zwei Kanäle:

- Rot (R) speichert Tiefe wie normale Shadow Map.
- Grün (G) speichert quadrierte Tiefe: $G = R^2$

Zunächst sind die beiden Kanäle redundant aber nach dem Filtern gestatten sie Rückschlüsse auf die Verteilung der eingegangenen Tiefenwerte. Der gefilterte Schattenwert kann dann geschätzt werden.

Vorteil:

- Geringere Kosten pro Fragment

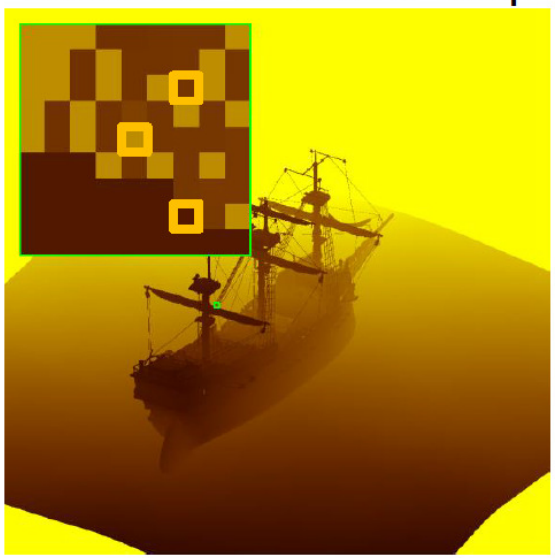
Nachteil:

- Mehr Speicherbedarf

Für jeden Texel speichert die VSM z und z^2.

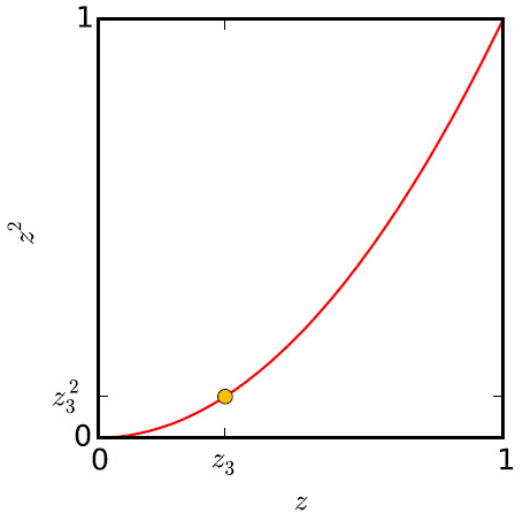
So lange man nur einzelne Texel betrachtet ist diese Information redundant und das Ergebnis der Schattenberechnung binär.

Variance Shadow Map

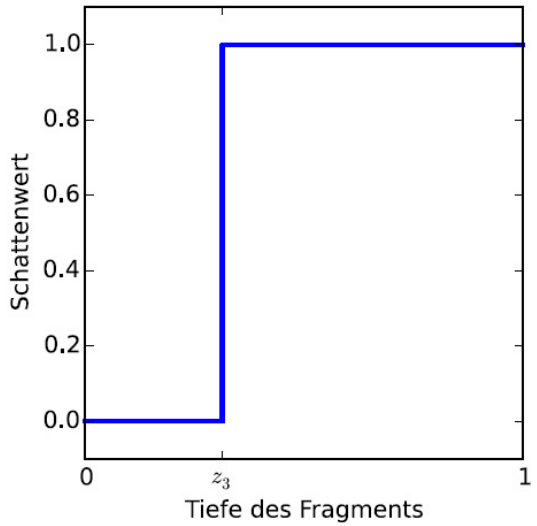


R=z_3, G=z_3^2

Werte in VSM



Schattenwerte bei PCF



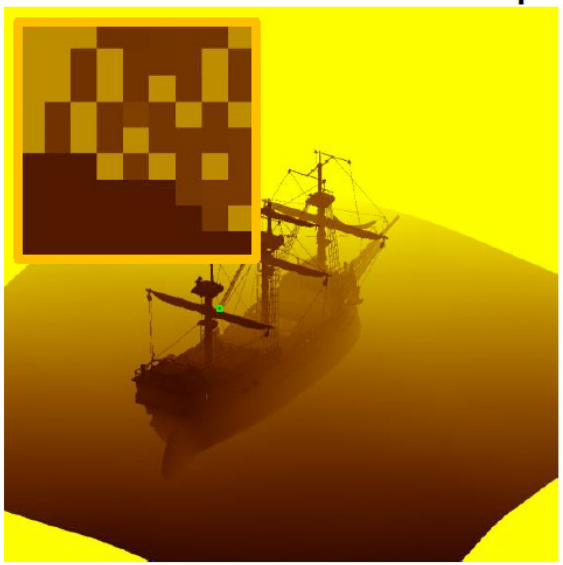
Nach dem Filtern sind R und G nicht mehr redundant.

Chebyshev's Ungleichung (unter)schätzt den Schattenwert:

$$\mu = R, \quad \sigma^2 = G - \mu^2, \quad \text{Schattenwert} \geq \begin{cases} \frac{\sigma^2}{\sigma^2 + (z-\mu)^2} & : z \geq \mu \\ 0 & : \text{sonst} \end{cases}$$

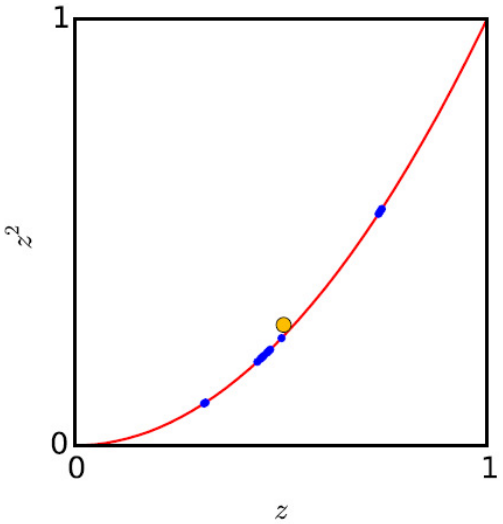
Wenn man nur R und G kennt ist dies die beste untere Schranke.

Variance Shadow Map

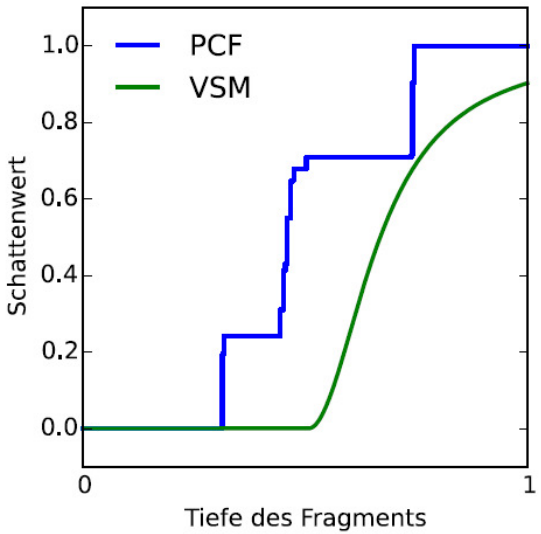


R=$\sum_{i=1}^{n} w_i \cdot z_i$, G=$\sum_{i=1}^{n} w_i \cdot z_i^2$

Werte in VSM



Schattenwerte



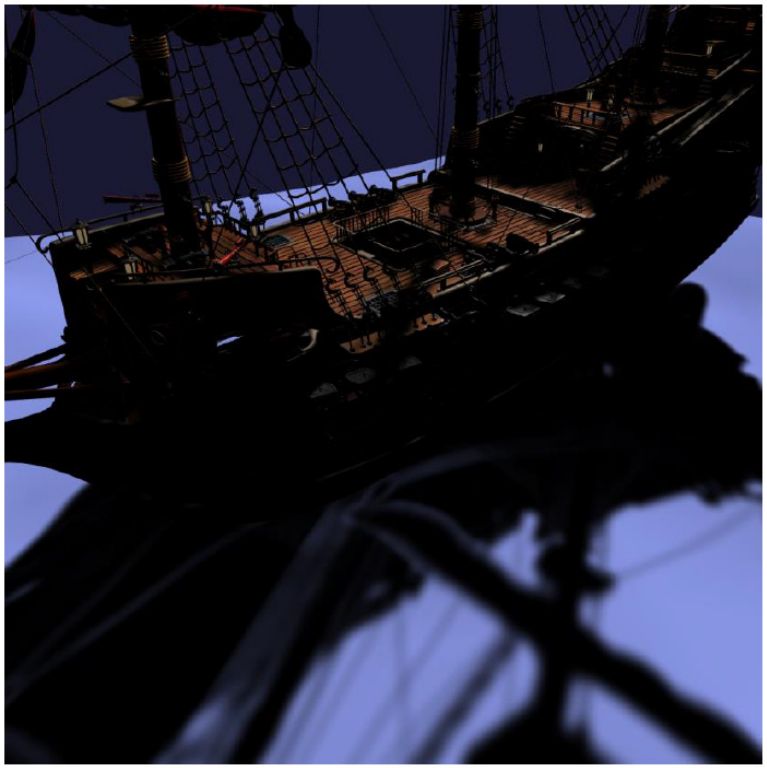
Die untere Schranke wird als Schätzwert genutzt.

Problem:

- Light leaking (helle Streifen im Schatten)

Gefilterte VSM

Szene mit VSM-Schatten



Moment Shadow Maps (MSMs)

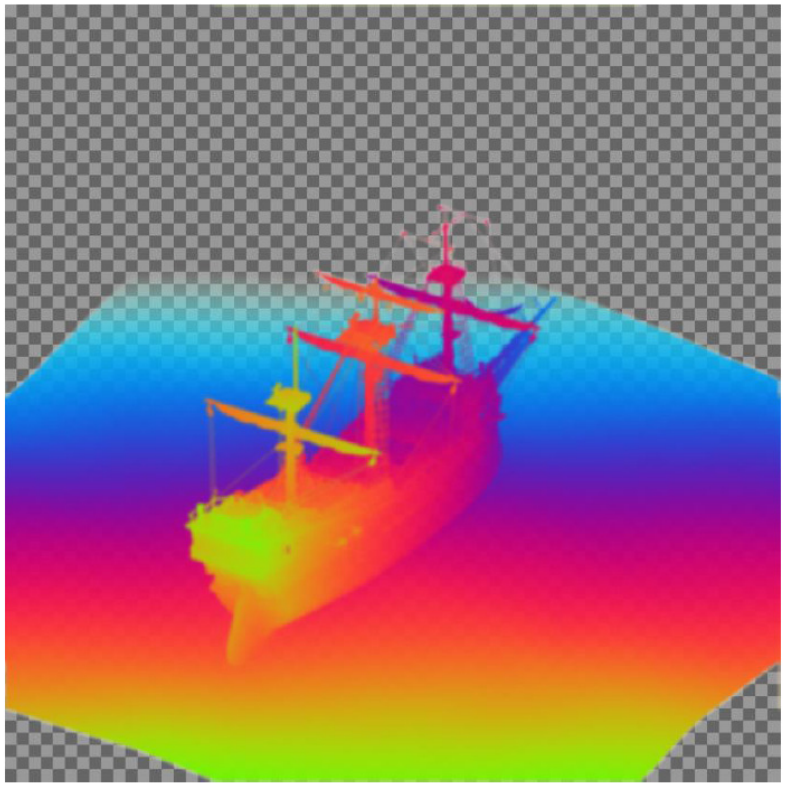
Für Ergebnisse ohne Light Leaking liefert eine VSM schlicht zu wenig Information. Moment Shadow Mapping (MSM) speichert daher mehr. Eine MSM hat vier Kanäle (Rot, Grün, Blau und Alpha).

- Rot (R) speichert Tiefe wie normale Shadow Map.
- Grün (G) speichert quadrierte Tiefe: $G = R^2$
- Blau (B) speichert dritte Potenz: $B = R^3$
- Alpha (A) speichert vierte Potenz: $A = R^4$

Wie bei VSM wird die beste untere Schranke als Schätzwert für den Schattenwert berechnet.

Die untere Schranke wird als Schätzwert genutzt.

Gefilterte MSM



Szene mit MSM-Schatten

Mehr dazu: Moment Shadow Mapping, Christoph Peters and Reinhard Klein In *Proceedings of the 19th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, San Francisco, California, ACM, 2015