Interaktive Computergrafik

Vorlesung im Sommersemester 2017
Kapitel 2: Echtzeit-Schattenverfahren (Teil 1)

Prof. Dr.-Ing. Carsten Dachsbacher
Lehrstuhl für Computergrafik
Karlsruher Institut für Technologie

Karlsruhe Institute of Technology

Schatten

- wichtiger Aspekt für die Wahrnehmung
  - wie ist die räumliche Anordnung von Objekten?
  - woher kommt das Licht?

Schatten

- vermittelt zusätzliche Informationen über Objekte, die Schatten werfen…

Schatten

- … und Informationen über Objekte, auf die Schatten geworfen wird…

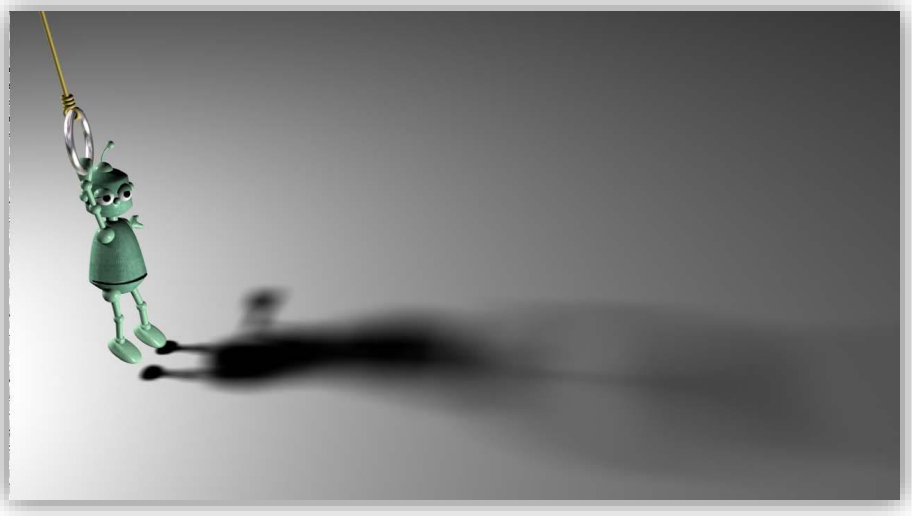
Schatten

- … und Informationen über Objekte, auf die Schatten geworfen wird…

Schatten

- … und Informationen über die Lichtquellen:
  - Punktlichtquelle: harte Schatten
    - Shadow Volumes [Crow77]
    - Shadow Maps [Williams78]

  - Flächenlichtquelle
    - Ray Casting und Sampling (u.a. mit Voxelisierung)
    - spezielle Echtzeitverfahren



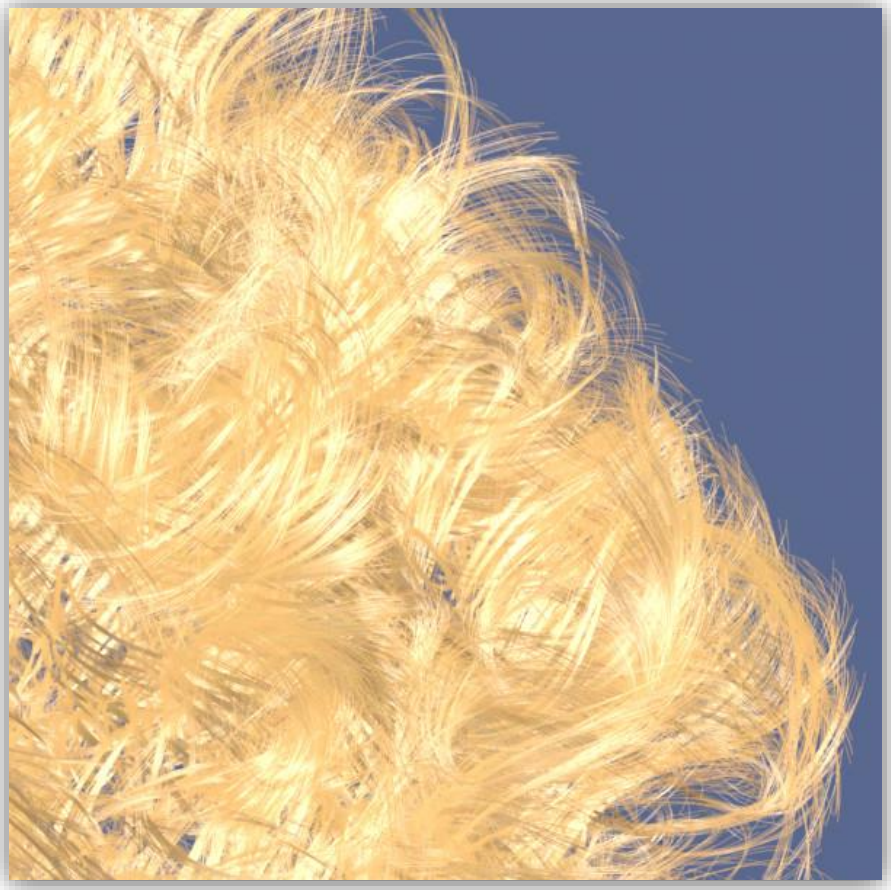
Schatten

- … und Informationen über die Lichtquellen:

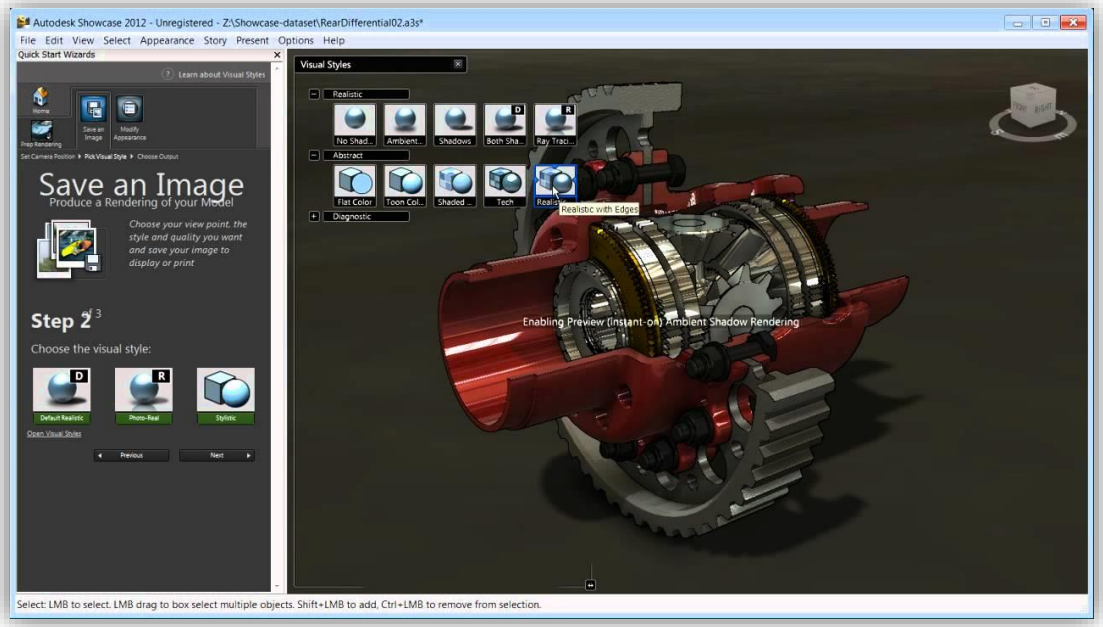
Bilder:
A survey of real-time soft-shadow algorithms, Hasenfratz et al.
http://hal.inria.fr/inria-00441603

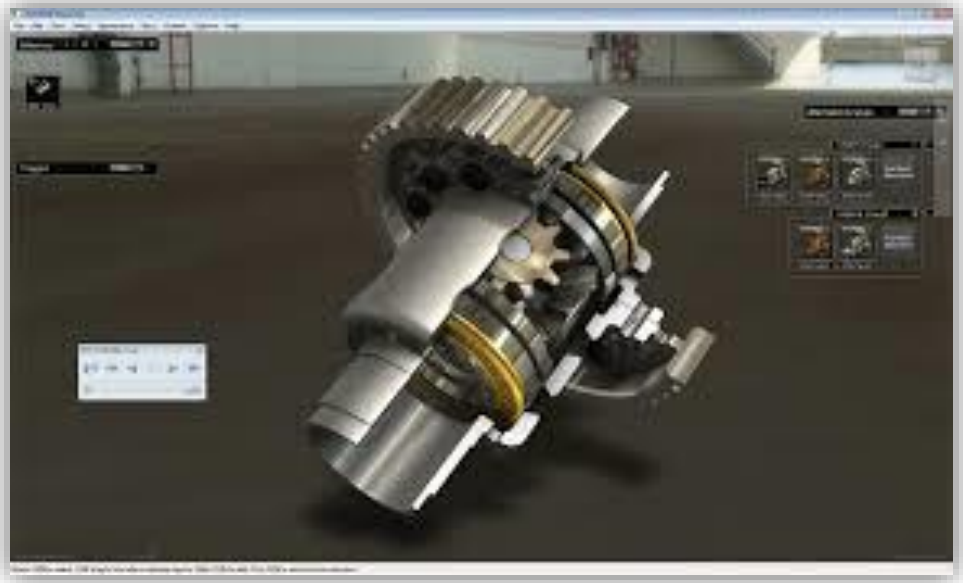
Schatten

- es existieren ebenfalls eine Reihe spezieller Verfahren für „volumetrische“ Objekte, wie z.B. Haare, Rauch, …

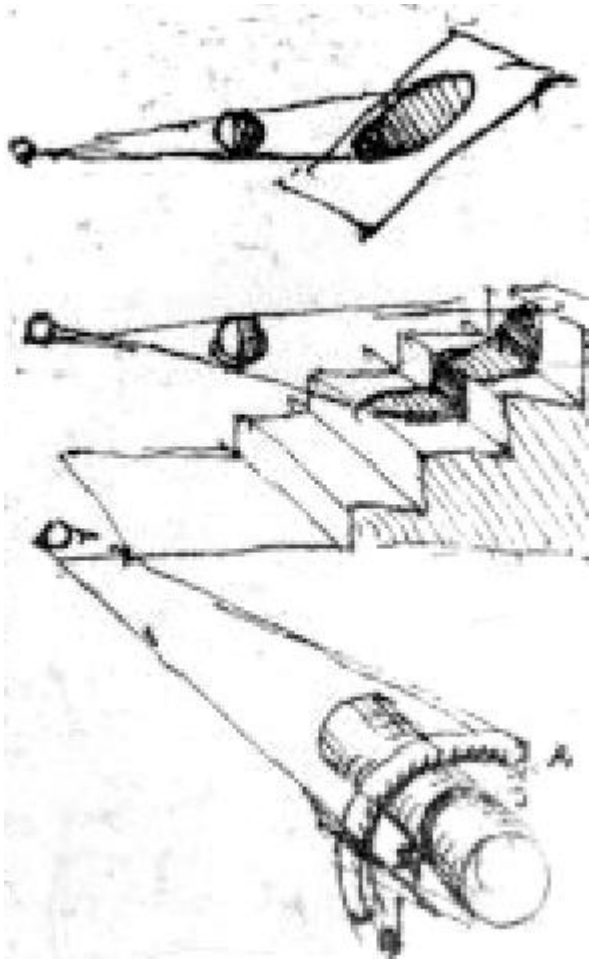


Beispiele (CAD-Modelle)

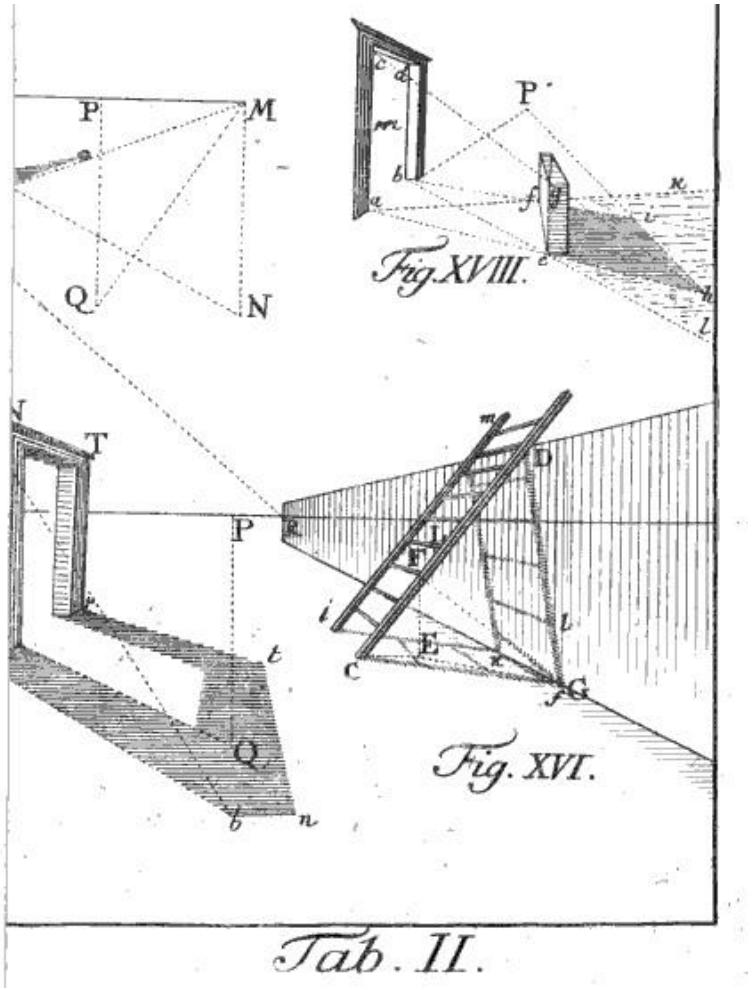




Schatten

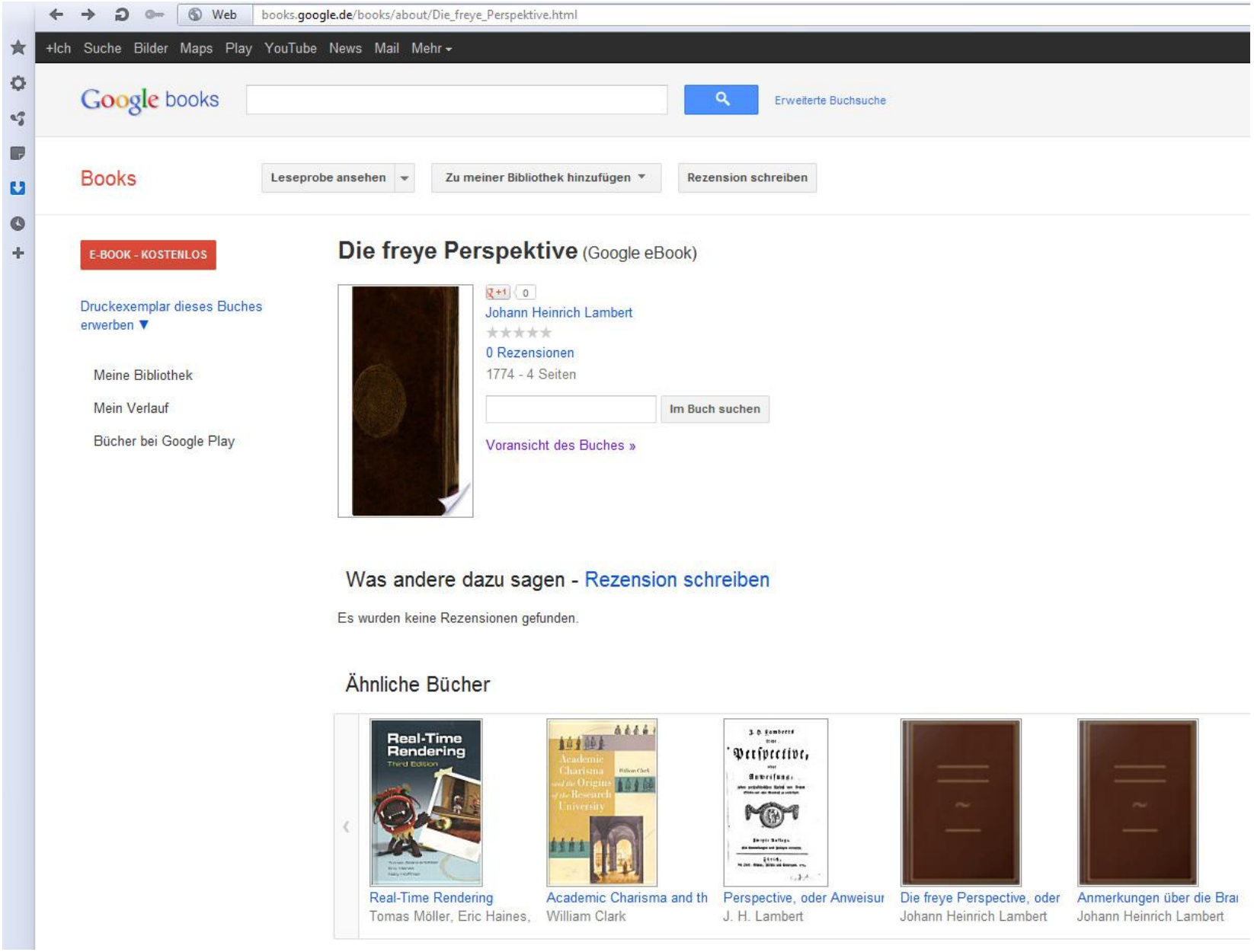


Leonardo Da Vinci, Trattato della pittura
(Codex Urbinas), 1490
Traktat über u.a. technische Probleme der Malerei
https://it.wikisource.org/wiki/Trattato_della_Pittura_(da_Vinci)

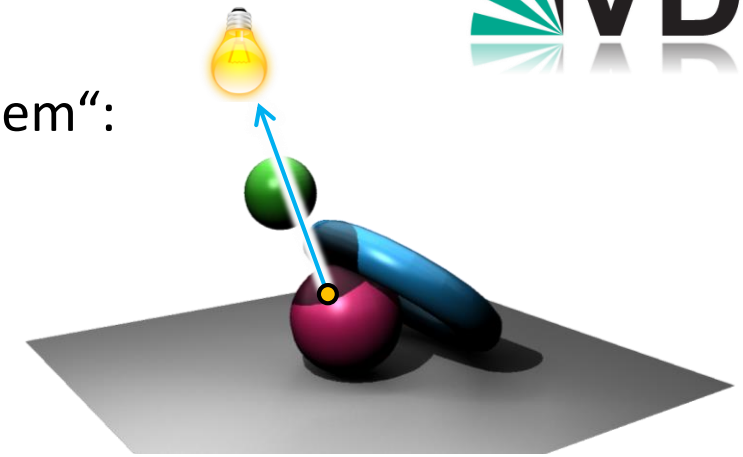


Johann Heinrich Lambert,
Die freye Perspektive, 1759

Schatten


Web
books.google.de/books/about/Die_freye_Perspektive.html
+Ich Suche Bilder Maps Play YouTube News Mail Mehr
Google books
Erweiterte Buchsuche
Books
Leseprobe ansehen
Zu meiner Bibliothek hinzufügen
Rezension schreiben
E-BOOK - KOSTENLOS
Druckexemplar dieses Buches erwerben
Meine Bibliothek
Mein Verlauf
Bücher bei Google Play
Die freye Perspektive (Google eBook)
Johann Heinrich Lambert
0 Rezensionen
1774 - 4 Seiten
Im Buch suchen
Voransicht des Buches »
Was andere dazu sagen - Rezension schreiben
Es wurden keine Rezensionen gefunden.
Ähnliche Bücher
Real-Time Rendering
Third Edition
Real-Time Rendering
Tomas Möller, Eric Haines,
Academic Charisma and the Origins of the Research University
Academic Charisma and th
William Clark
Perspective, oder Anweisur
J. H. Lambert
Die freye Perspective, oder
Johann Heinrich Lambert
Anmerkungen über die Brai
Johann Heinrich Lambert

Schatten



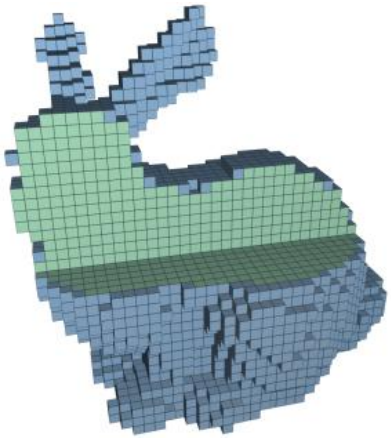
- Schattenberechnung ist ein „globales Problem“: man muss die gesamte Szene kennen, um festzustellen, ob ein Punkt im Schatten liegt, also ein „Schattenstrahl“ eine andere Fläche schneidet

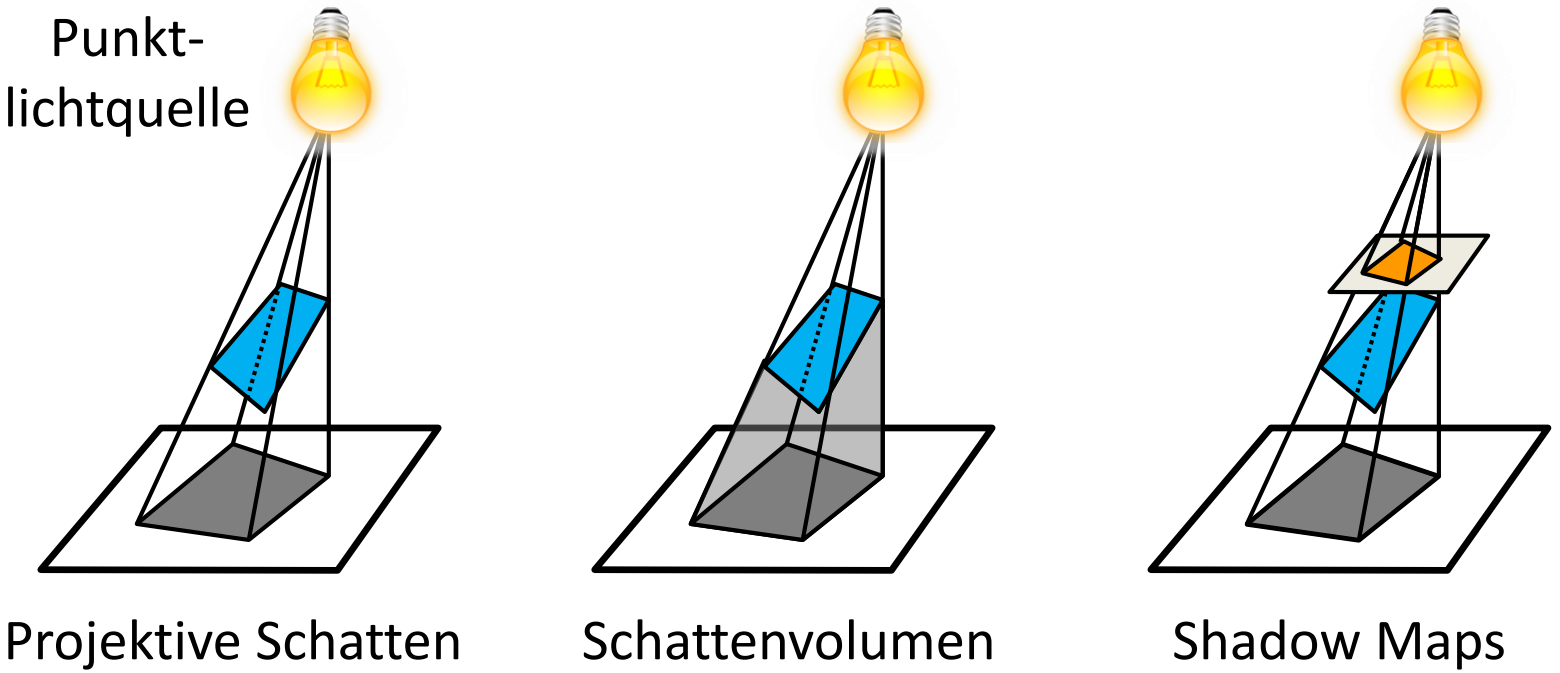
- in der Grafik-Pipeline werden alle Primitive unabhängig voneinander (Vorstellung: „nacheinander“) gezeichnet
  - offensichtlich: jedes Primitiv kann Schatten auf jedes vorherige oder spätere Primitiv werfen und umgekehrt
  - diese Information steht während der Verarbeitung eines Primitivs nicht zur Verfügung
  - deswegen kann/wird es kein `glEnable( GL_SHADOWS )` geben

- siehe Real-Time Rendering, 3rd Ed., Kapitel 9.1 (Shadow) und Kapitel 9.9 (Precomputed Lighting)

Klassifikation Schattenverfahren

- **projektive Schatten**: nur für Schatten auf Ebenen, keine Selbstverschattung
- **vorberechnete Schatten/Beleuchtung** (sog. Light Maps)
- **Schattenvolumen**: Objektraum-Verfahren
- **Shadow Maps**: Bildraum-Verfahren
- **Ray Casting / Ray Marching**: Voxelisierung

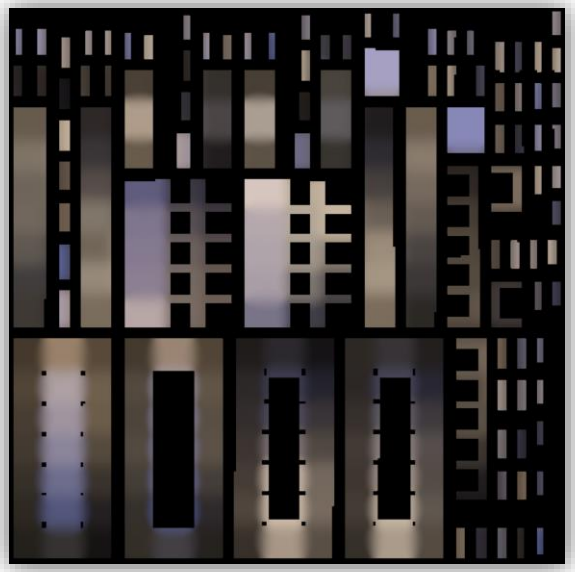




Vorberechnete Schatten

Light Maps: Texturen mit vorberechneter Beleuchtung und Schatten

- benötigt eine Parametrisierung (z.B. einen Textur-Atlas)
- nur für statische Szenen(teile), oder begrenzte Variation, möglich
- vorberechnet mit Raytracing, Path Tracing, …
- Light Maps meist niedrig(er) aufgelöst, Detail durch andere Texturen (diffuse Farbe, Normal/Gloss Map, …)
- manchmal wird auch einfallendes Licht richtungsabhängig gespeichert, z.B. für Normal Mapping (siehe z.B. www.valvesoftware.com/.../GDC2004_Half-Life2_Shading.pdf)



Bilder, Demo: http://codeflow.org/entries/2012/aug/25/webgl-deferred-irradiance-volumes/

Lightmaps und „Baked Lighting“

- verwendet in: Quake 1 ☺, mobilen Anwendungen, bis hin zu AAA-Games

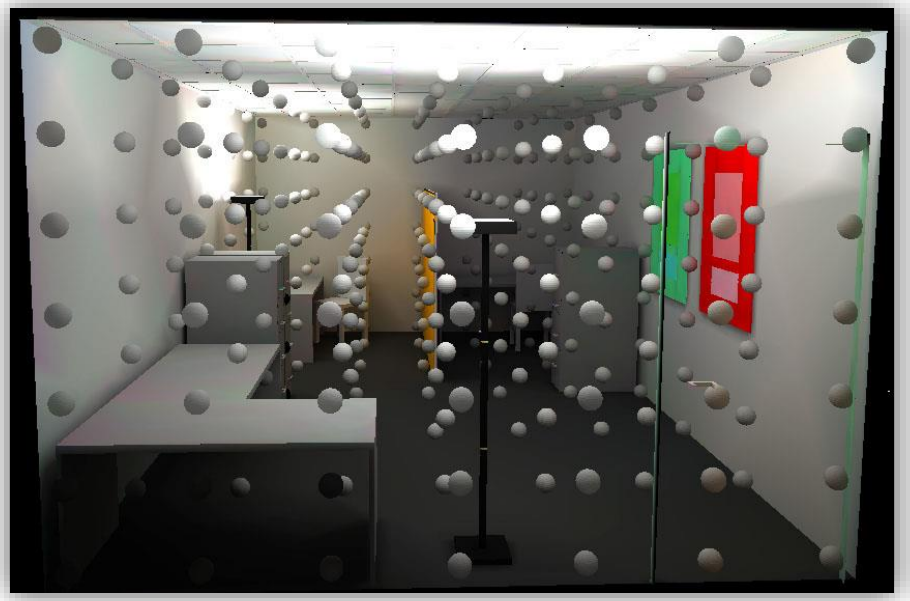
Quake 1

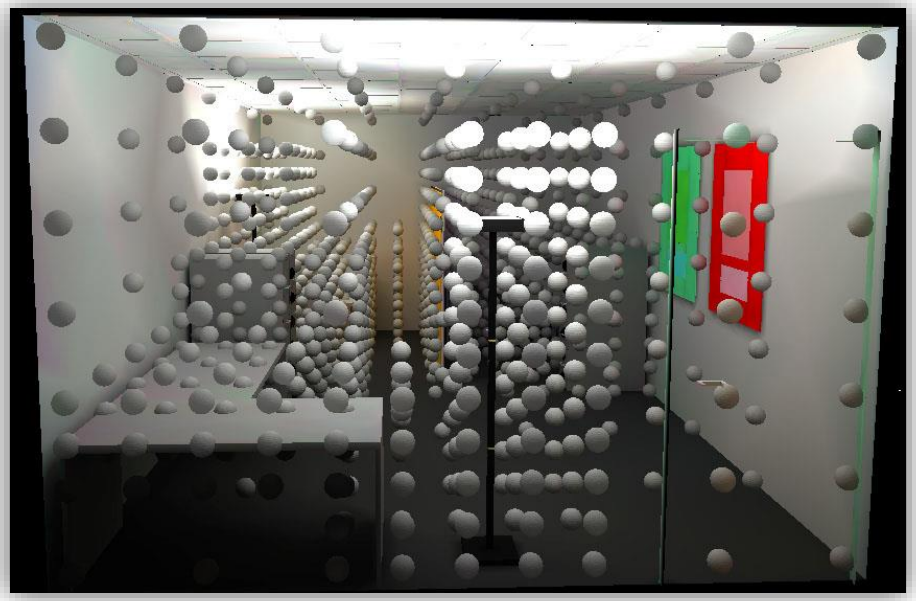
Mirror‘s Edge EA Games

Half Life 2

Irradiance Volumes

- speichere einfallendes Licht (Irradiance) in 3D, d.h. für viele Punkte im Raum: regelmäßigen Gitter (3D-Textur), Octree, adaptiv
- Irradiance $E(\boldsymbol{x}, \boldsymbol{n}) = \int_{\Omega} L(\boldsymbol{x}, \boldsymbol{\omega}_i) \langle \boldsymbol{n}, \boldsymbol{\omega}_i \rangle^{+} d\boldsymbol{\omega}_i$
- Lookup für eine Oberfläche bei $\boldsymbol{x}$ mit Normale $\boldsymbol{n}$
  - lokalisiere die nahsten Gitterpunkte (z.B. für trilineare Interpolation)
  - interpoliere Irradiance(darstellung)
  - werte die Irradiance für Richtung $\boldsymbol{n}$ aus

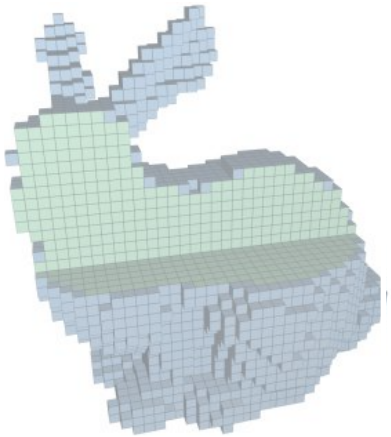


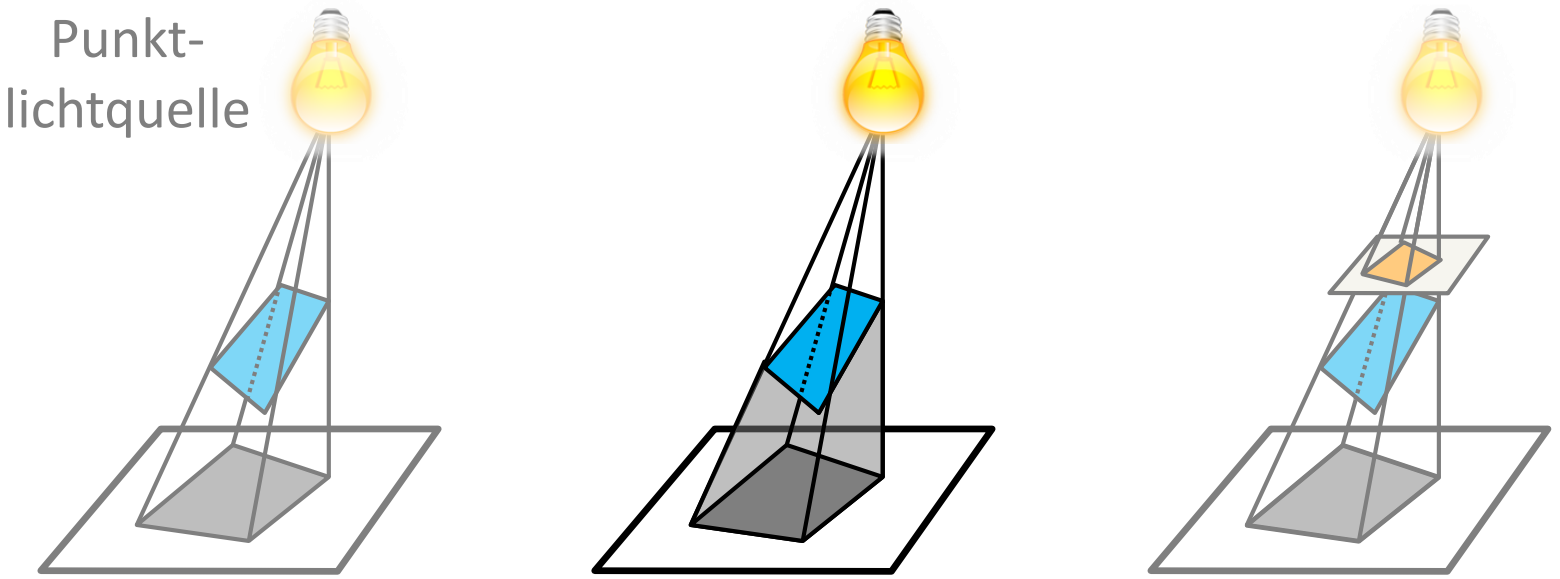


Bilder: The Irradiance Volume, Gene S. Greger, MSc Thesis

Klassifikation Schattenverfahren

- **projektive Schatten**: nur für Schatten auf Ebenen, keine Selbstverschattung
- **vorberechnete Schatten/Beleuchtung** (sog. Light Maps)
- **Schattenvolumen**: Objektraum-Verfahren
- **Shadow Maps**: Bildraum-Verfahren
- **Ray Casting / Ray Marching**: Voxelisierung

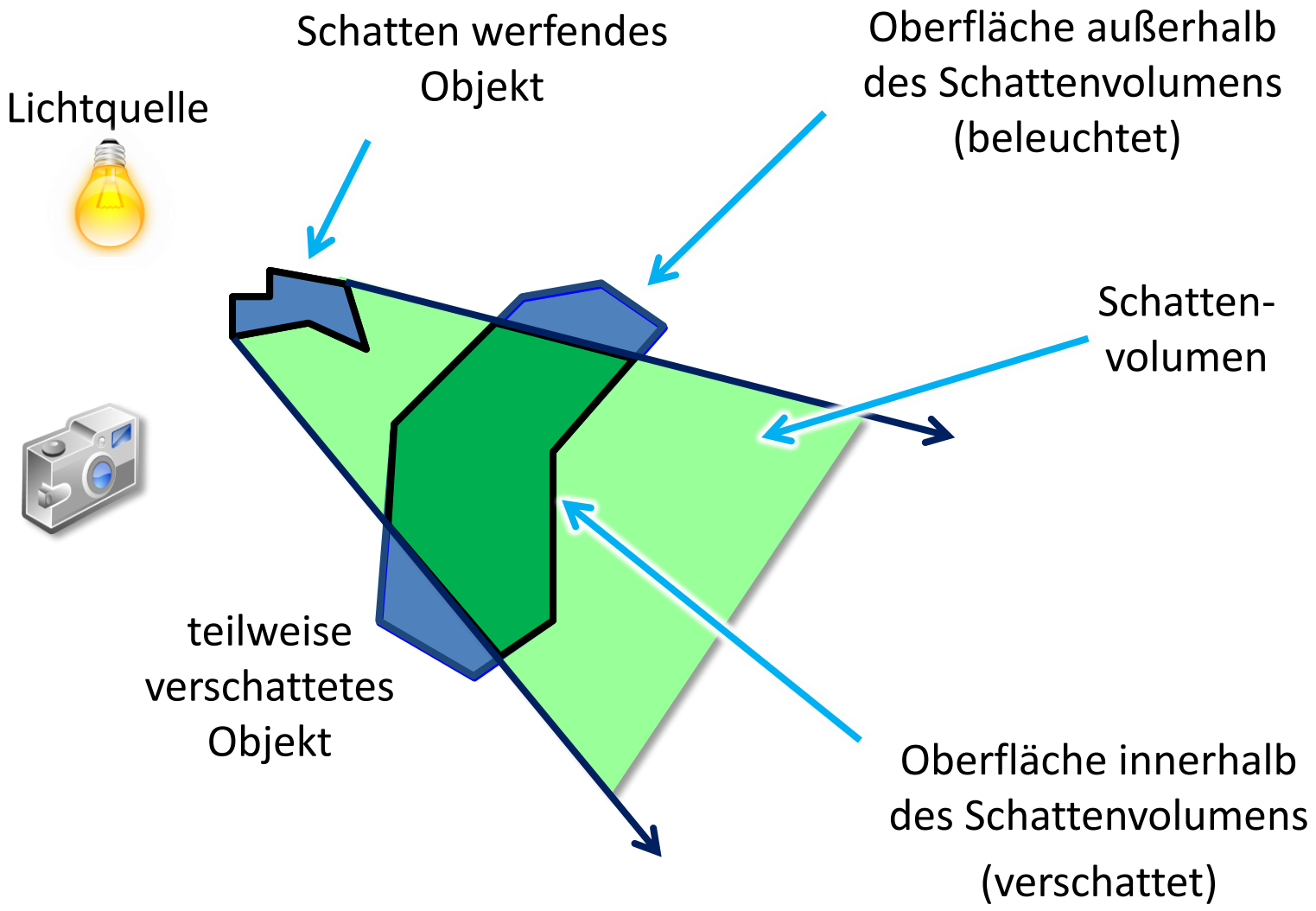




Projektive Schatten | Schattenvolumen | Shadow Maps

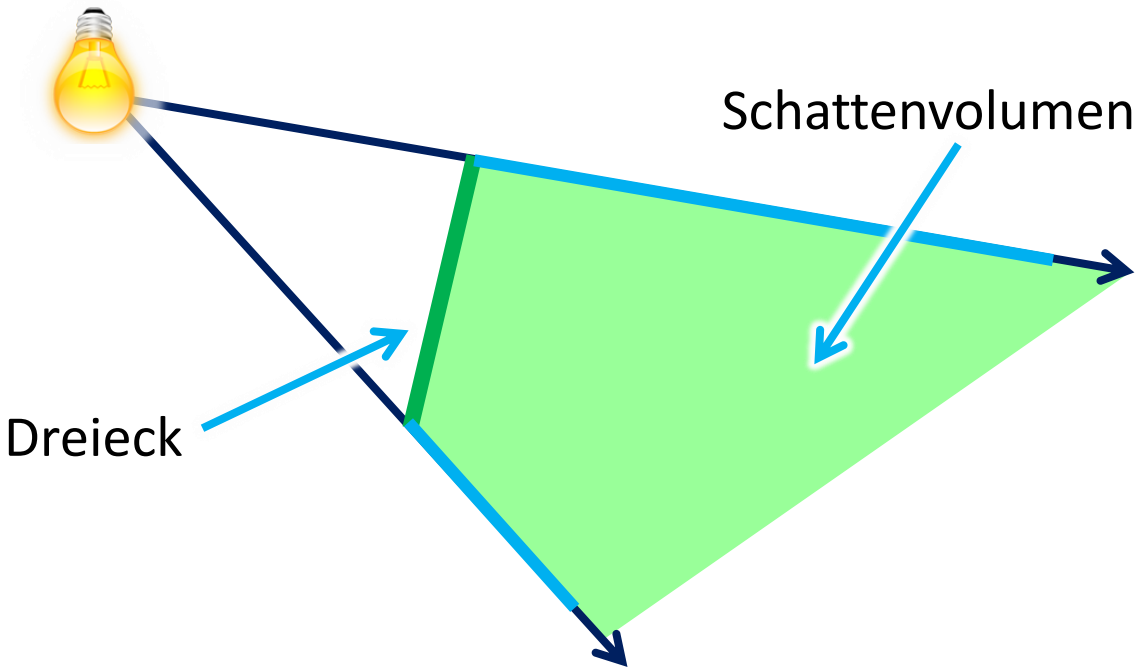
Schattenvolumen (Crow '77)

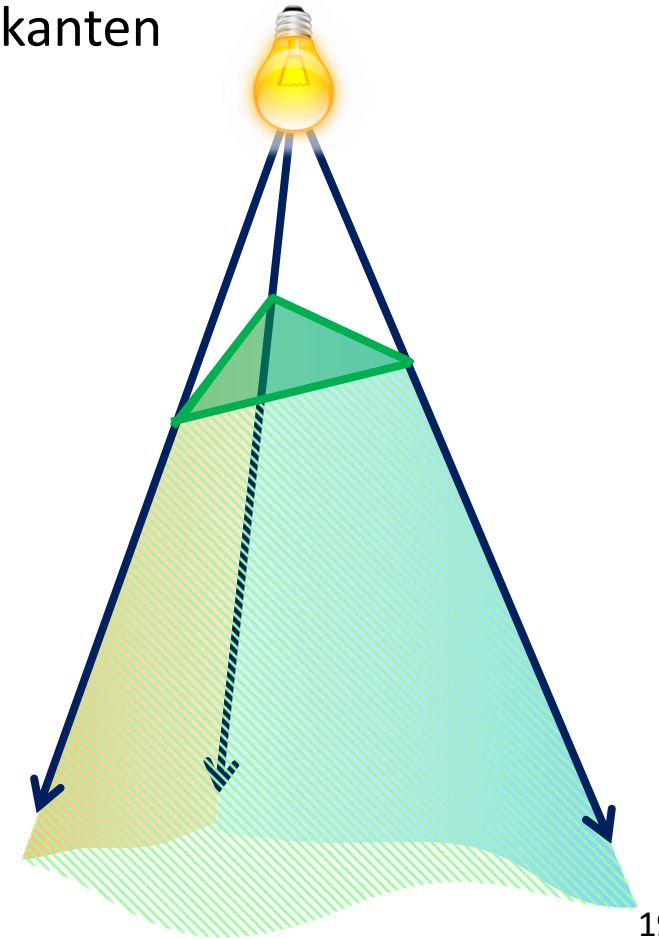
- ein Schattenvolumen umschließt den Raum, der durch ein Objekt von verschattet wird (meist von einer Punktlichtquelle aus gesehen)



Schattenvolumen eines Dreiecks

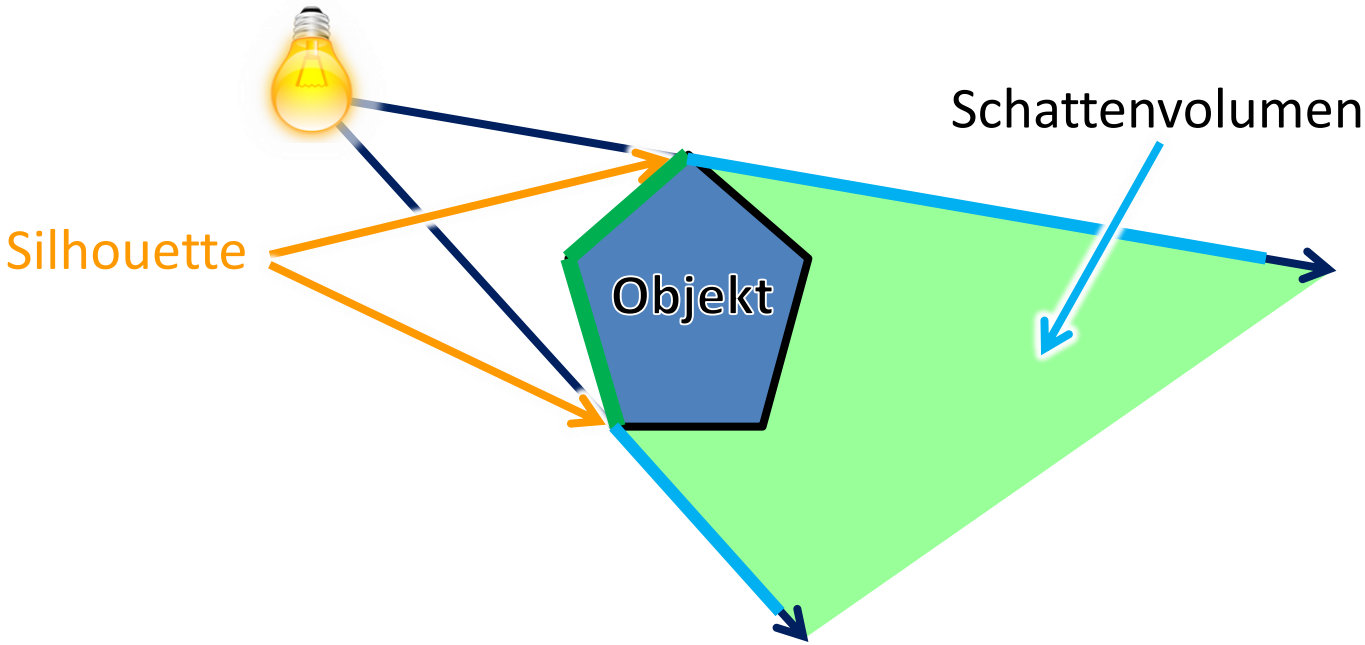
- für jedes Paar aus Lichtquelle und Fläche:
  bestimme das Schattenvolumen durch seine **Begrenzungsflächen**
- Schattenvolumen eines einzelnen Dreiecks besteht
  - aus dem Dreieck selbst und
  - den Seitenflächen gebildet aus den Dreieckskanten
    extrudiert von der Lichtquelle weg





Schattenvolumen eines Dreiecksnetzes

- die Oberfläche des Schattenvolumens eines **geschlossenen** Dreiecksnetzes besteht aus
  - Dreiecken, die zur Lichtquelle hin orientiert sind und
  - Seitenflächen gebildet aus der Objektsilhouette
- nicht geschlossenes Netz: zusätzlich Randkanten und daraus extrudierte Seitenflächen



Schattenvolumen eines Dreiecksnetzes

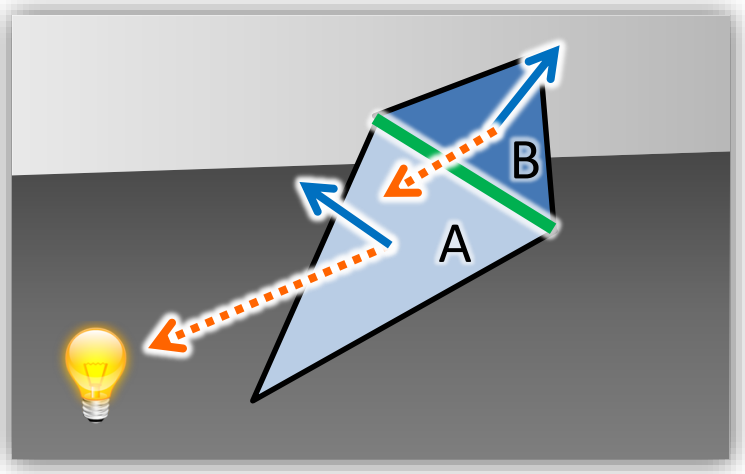
- die Oberfläche des Schattenvolumens eines **geschlossenen** Dreiecksnetzes besteht aus
  - Dreiecken, die zur Lichtquelle hin orientiert sind und
  - Seitenflächen gebildet aus der Objektsilhouette

- Schattenvolumen unterschiedlicher Objekte oder Objektteile können sich überlappen:

Schattenvolumen eines Dreiecksnetzes

Bestimmung der Objektsilhouette

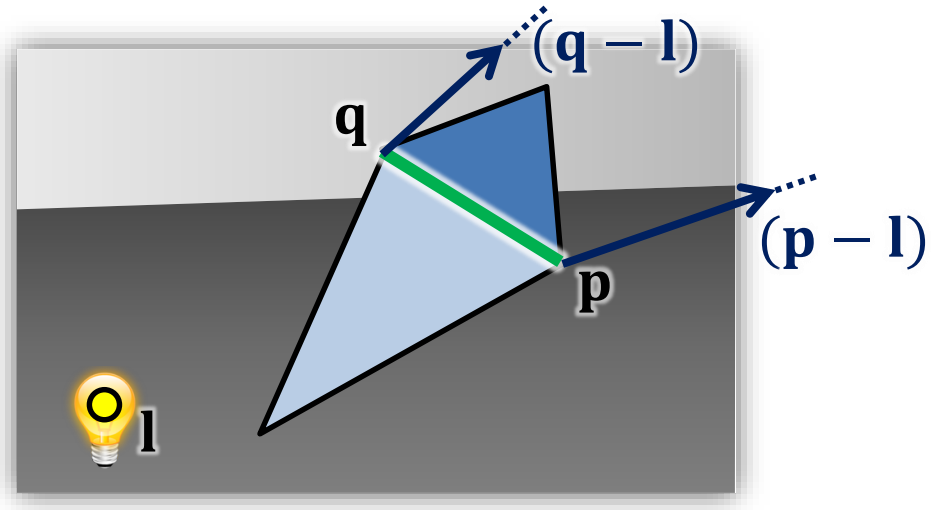
- … für geschlossene Teile von Dreiecksnetzen („watertight“, 2-manifold)
- zur Silhouette gehören Kanten von deren angrenzenden Dreiecken eines **zur Lichtquelle zeigt** und eines **von der Lichtquelle abgewandt** ist
- Vorgehen:
  - bestimme für jedes Dreieck, ob es zur LQ oder davon weg zeigt (Skalarprodukt zwischen der Dreiecksnormale und dem Vektor zur LQ)
  - betrachte jede Kante und ihre zwei benachbarten Dreiecke A und B:
    - zeigt A zur LQ und B zeigt weg (oder umgekehrt)
      → Kante ist Teil der Silhouette



Schattenvolumen eines Dreiecksnetzes

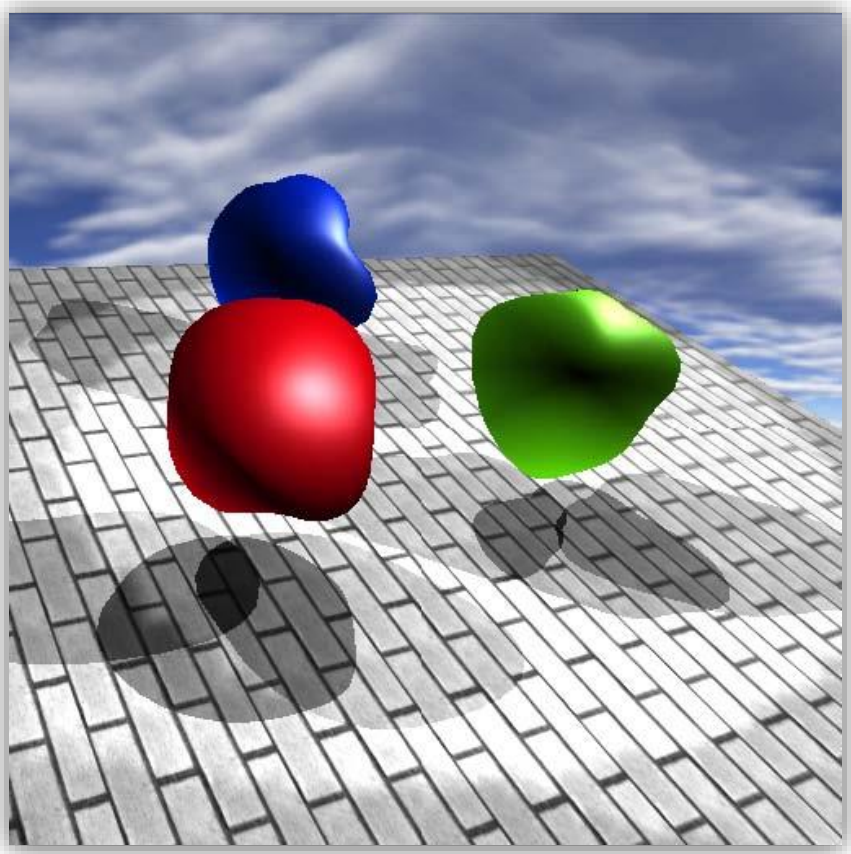
Oberfläche des Schattenvolumens

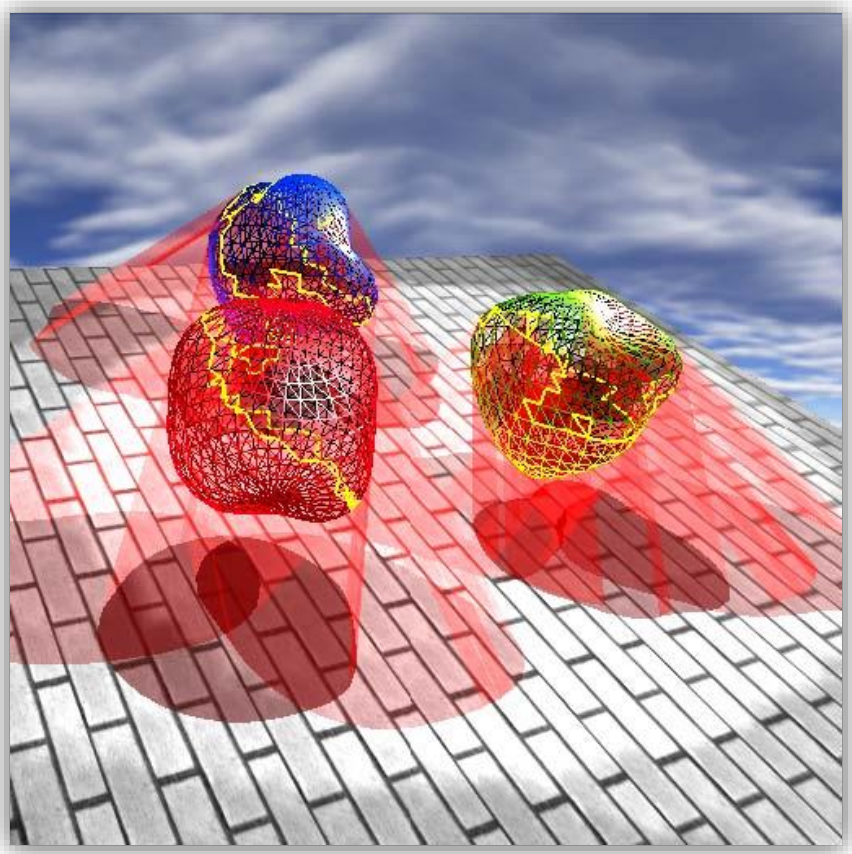
- sie die Kante $\overline{\mathbf{pq}}$ Teil der Silhouette
  - extrudiere $\overline{\mathbf{pq}}$ weg von der Lichtquelle ins Unendliche
  - dadurch entsteht eine **viereckige Seitenfläche des Schattenvolumen** mit den Eckpunkten $\mathbf{p}$, $\mathbf{p} + \infty \cdot (\mathbf{p} - \mathbf{l})$, $\mathbf{q} + \infty \cdot (\mathbf{q} - \mathbf{l})$ und $\mathbf{q}$
  - Punkte im Unendlichen durch homogene Koordinaten dargestellt: Vertex (v_x, v_y, v_z) und Lichtquellenposition (l_x, l_y, l_z) → Punkt im Unendlichen: $\mathbf{v}'_\mathbf{h} = (v_x - l_x, v_y - l_y, v_z - l_z, 0)$
- und: **alle zur LQ gewandten Dreiecke sind ebenfalls Teil des Schattenvolumen**



Schattenvolumen eines Dreiecksnetzes

- Beispiel: Schattenvolumen für 3 Lichtquellen
  - die gelben Kanten zeigen die Objektsilhouetten bzgl. einer der Lichtquellen



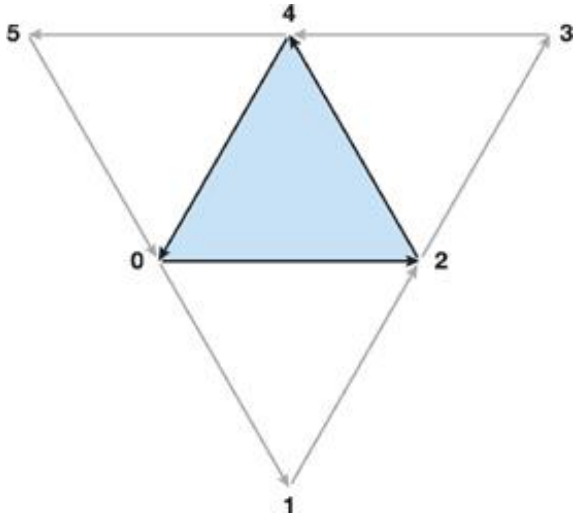


Bilder: Stefan Brabec

Berechnung der Silhouette auf der GPU

Berechnung mittels Geometry Shader

- Eingabe: Dreieck mit Nachbardreiecken
- Ausgabe (wenn es zur LQ zeigt):
  Seitenflächen des Schattenvolumen und das Dreieck selbst
- im GS ist es möglich auf Nachbardreiecke zuzugreifen, wenn die entsprechenden Indizes der Vertices angegeben werden
  - dazu zeichnet man Primitive vom Typ **`GL_TRIANGLES_ADJACENCY`** und übergibt pro Dreieck 6 Vertex-Indices statt 3
  - Zugriff mittels **`gl_PositionIn[]`**



  - detaillierte Beschreibung dieser Technik:
    http://http.developer.nvidia.com/GPUGems3/gpugems3_ch11.html

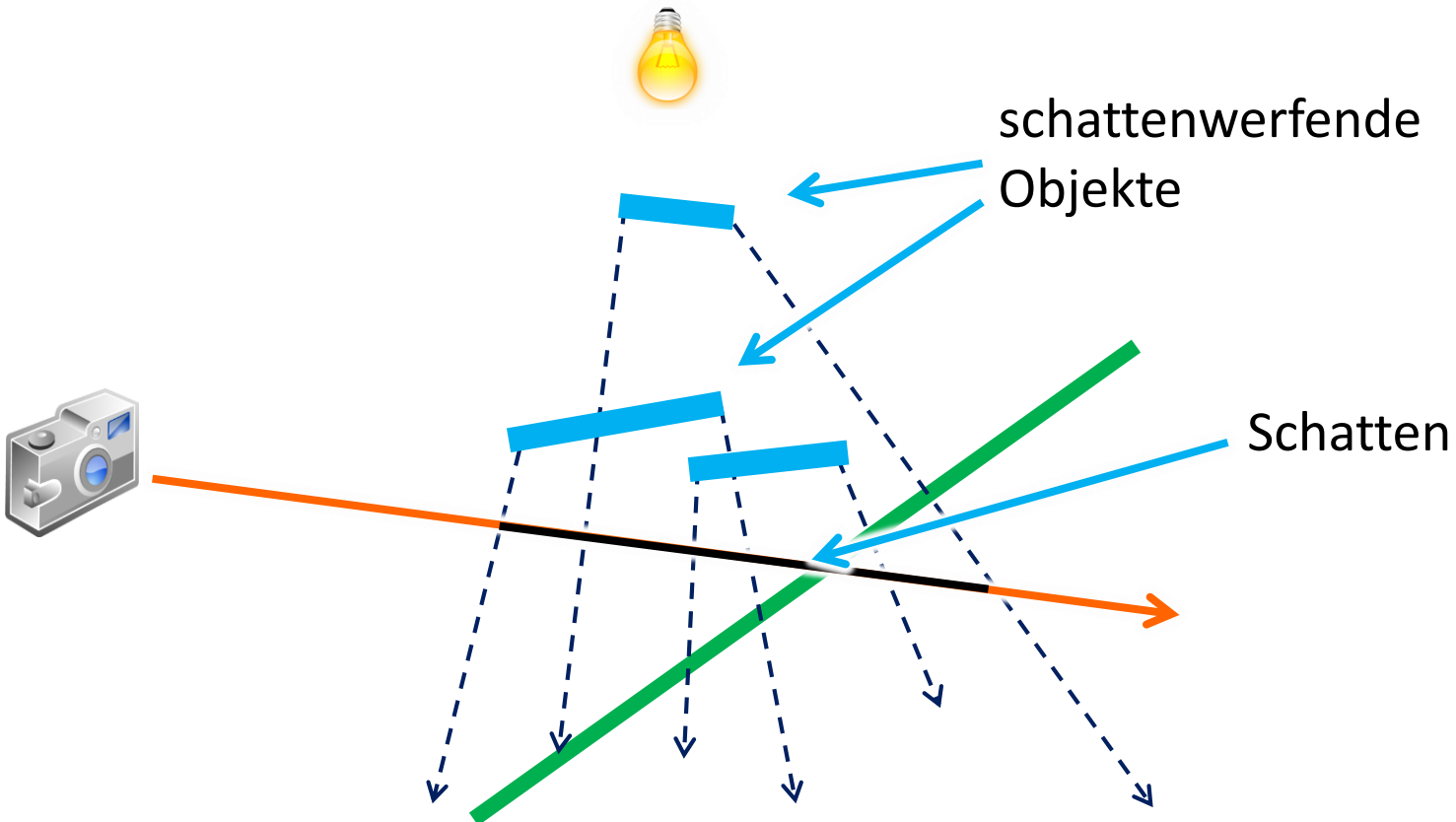
Schattenvolumen

- bis zu diesem Punkt: bestimme die Schattenvolumen für alle Kombinationen von Objekten und (Punkt-)Lichtquellen
- stelle dann für jeden Pixel fest, ob er von einer LQ beleuchtet wird, oder sich in einem entsprechenden Schattenvolumen befindet
- es handelt sich um ein Objektraumverfahren: wir beschreiben die Oberfläche der Schattenvolumen geometrisch
  → akkurate Schatten **für Punktlichtquellen**

Schattenvolumen in Doom 3

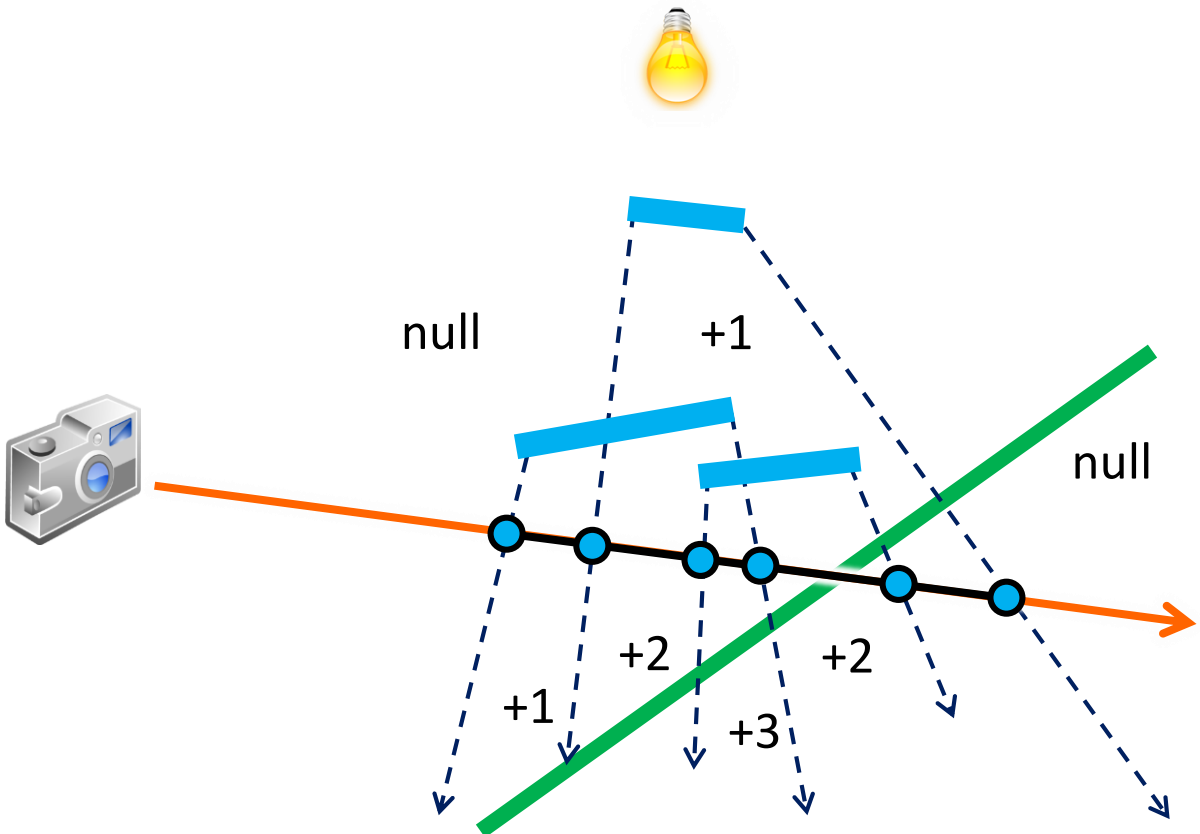
Schattenvolumen: Schattentest

- mehrere, teils überlappende, Schattenvolumen entstehen durch mehrere Objekte und/oder nicht-konvexe Oberflächen
- wie stellt man fest, ob sich ein Punkt in einem SV befindet?
  - eine Möglichkeit: Strahlschuss und Odd-Even Test (viel zu teuer!)
  - Odd-Even Test für viele Pixel gleichzeitig mit dem Stencil Buffer (Strahl Kamera-Oberfläche)



Schattenvolumen: Schattentest

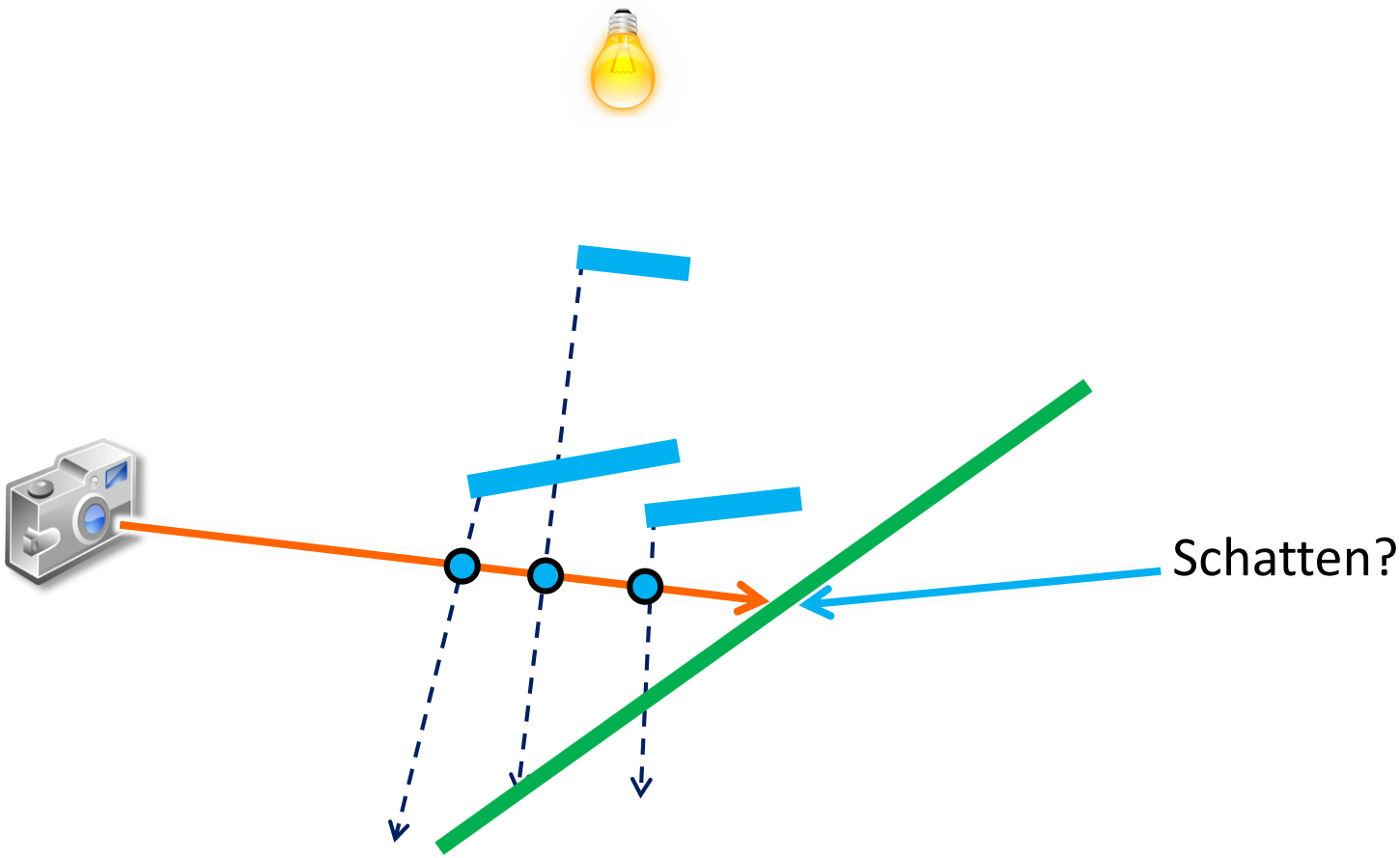
- verfolge einen Strahl vom Betrachter zum Oberflächenpunkt
  - setze einen Zähler zu Beginn auf Null
  - beim Eintritt in ein SV: inkrementiere Zähler
  - beim Verlassen eines SV: dekrementiere Zähler
  - an der Oberfläche: Punkt befindet sich im Schatten, wenn Zähler > 0



Beispiel: Zählen mit dem Stencil Buffer

Weiteres Beispiel

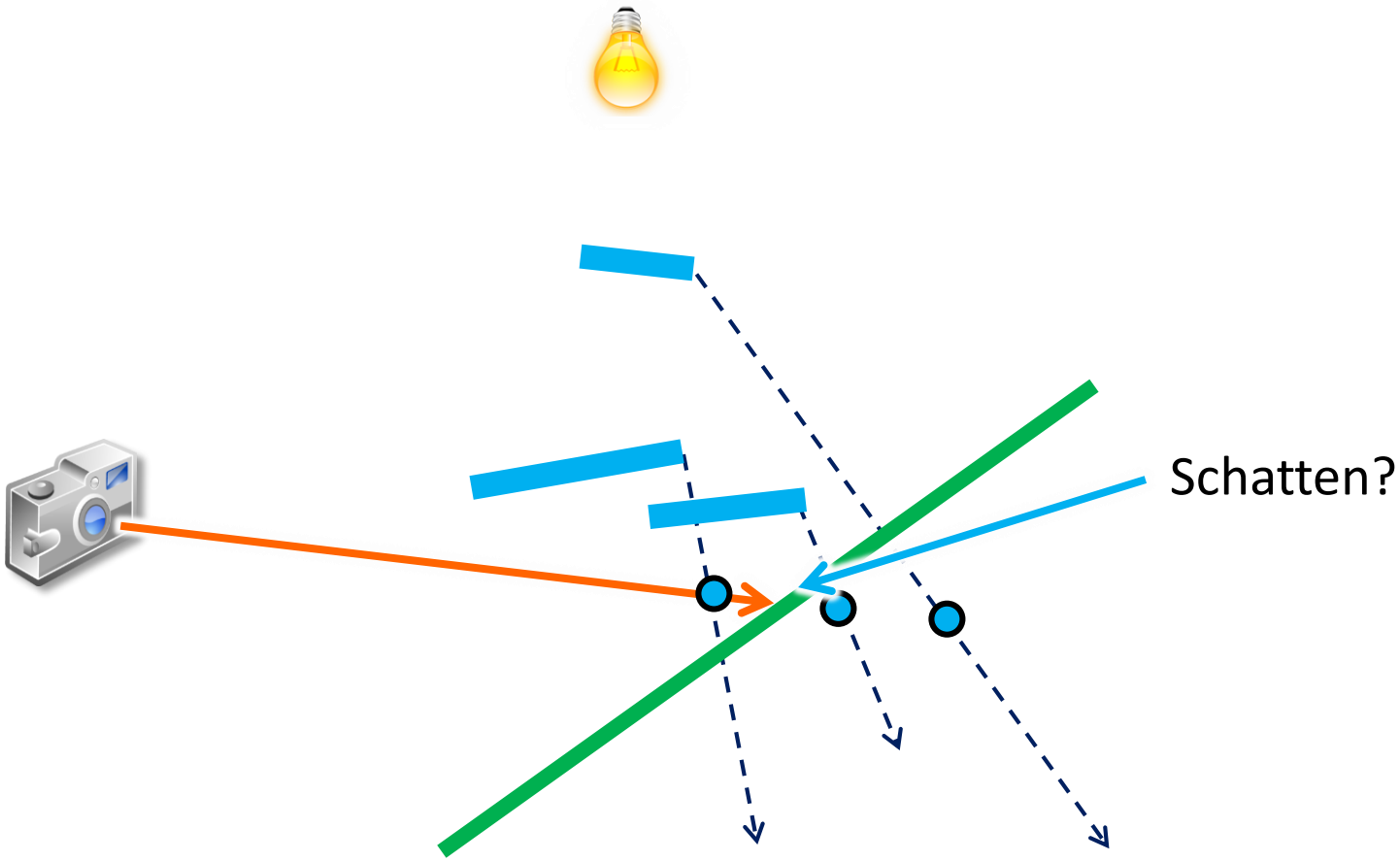
- 3 Eintrittspunkte in die Schattenvolumen (Vorderseiten)
- Zähler am betrachteten Punkt $= 3$



Beispiel: Zählen mit dem Stencil Buffer

Weiteres Beispiel

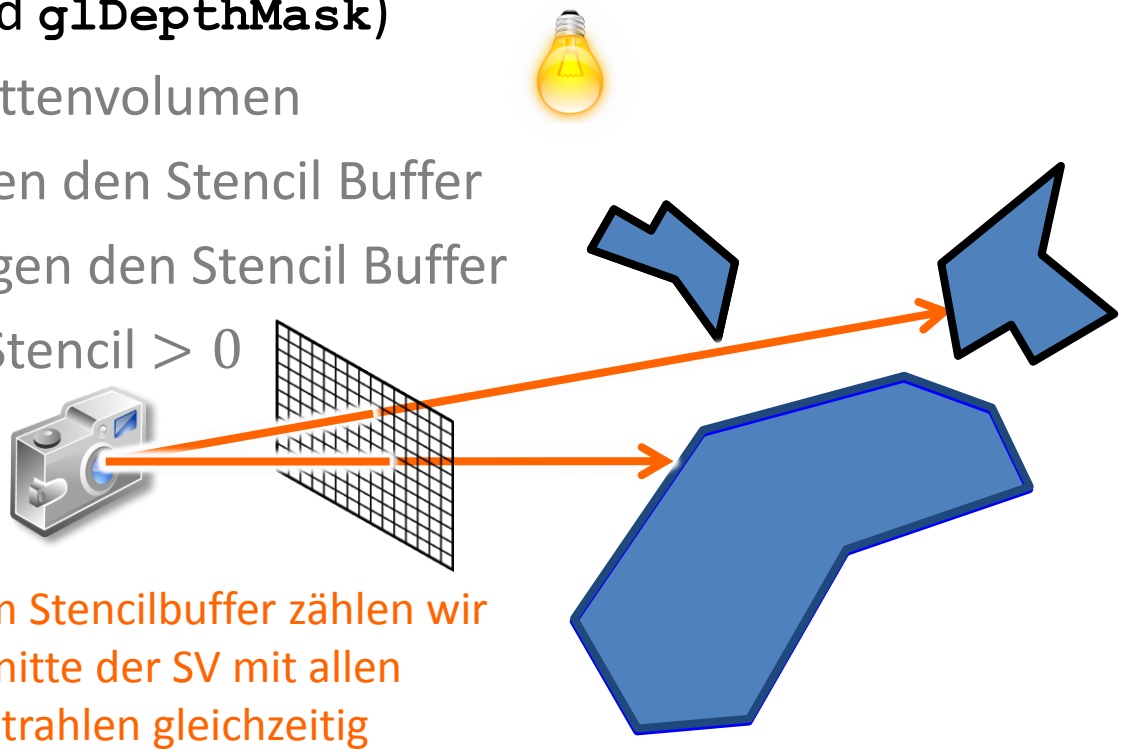
- 1 Austrittspunkt aus dem/den Schattenvolumen (Rückseiten)
- Zähler am betrachteten Punkt $= 2$
  - die zwei weiteren Schnittpunkte fallen beim Tiefentest durch



Schattenvolumen: Schattentest

Stencil Buffer zum Zählen der Schnitte mit Schattenvolumen

- zuerst werden die Objekte gezeichnet
- anschließend wird Schreiben in den Farb- und Tiefenpuffer deaktiviert (mit **`glColorMask`** und **`glDepthMask`**)
- zeichne dann die Schattenvolumen
  - Vorderseiten erhöhen den Stencil Buffer
  - Rückseiten erniedrigen den Stencil Buffer
- Schatten ist dort, wo Stencil > 0

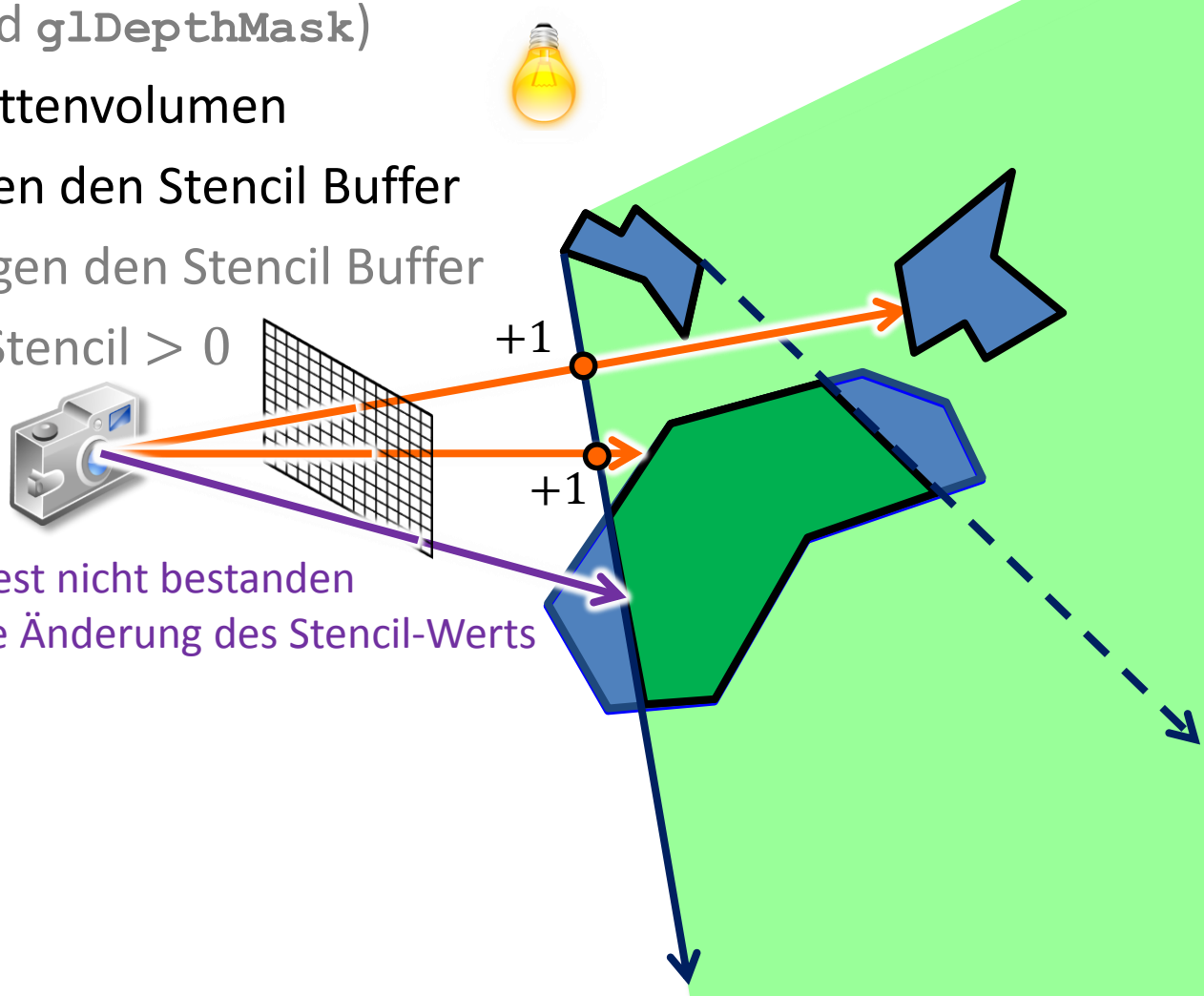


mit dem Stencilbuffer zählen wir die Schnitte der SV mit allen Primärstrahlen gleichzeitig

Schattenvolumen: Schattentest

Stencil Buffer zum Zählen der Schnitte mit Schattenvolumen

- zuerst werden die Objekte gezeichnet
- anschließend wird Schreiben in den Farb- und Tiefenpuffer deaktiviert (mit `glColorMask` und `glDepthMask`)
- zeichne dann die Schattenvolumen
  - Vorderseiten erhöhen den Stencil Buffer
  - Rückseiten erniedrigen den Stencil Buffer
- Schatten ist dort, wo Stencil > 0



Schattenvolumen: Schattentest

Einschub: Zählen mit dem Stencil Buffer (OpenGL)

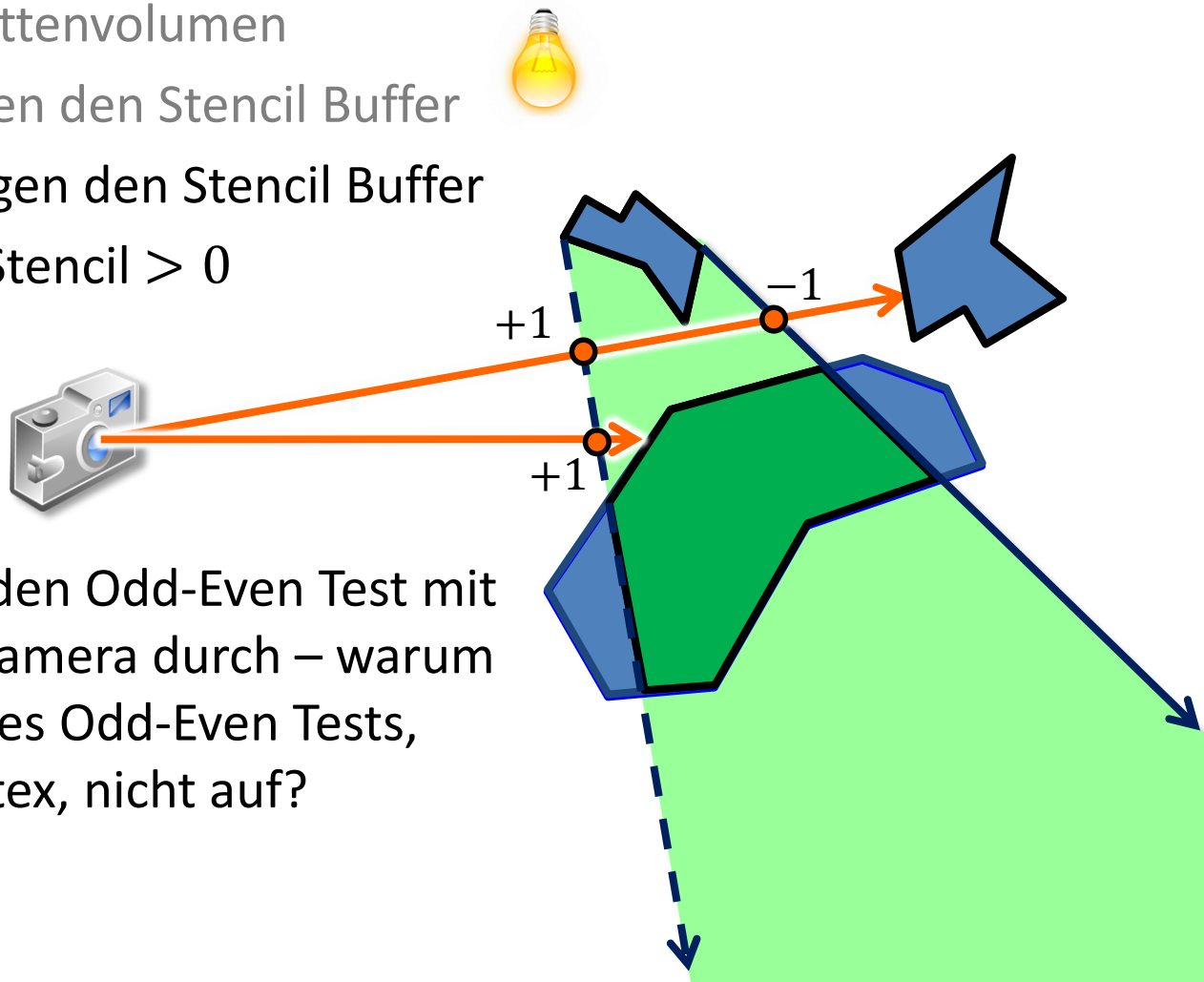
- zeichne Seitenflächen der Schattenvolumen, die zum Betrachter zeigen (= Eintrittspunkte des Strahls in das SV)
  - Auswahl überlässt man OpenGL durch Backface Culling
  - mit `glEnable( GL_CULL_FACE )` und `glFrontFace(...)` wird bestimmt, ob Vorder- oder Rückseiten gezeichnet werden (bei konsistenter Orientierung der Eckpunkte)

- inkrementiere Stencil-Wert, wenn der Tiefentest bestanden wird:
  `glStencilOp( GL_KEEP, GL_KEEP, GL_INCR )`
  - definiert die Operationen für:
    - (1) Stencil Test fehlgeschlagen: `GL_KEEP`
    - (2) Tiefentest fehlgeschlagen: `GL_KEEP`
    - (3) Tiefentest bestanden: `GL_INCR`

Schattenvolumen: Schattentest

Stencil Buffer zum Zählen der Schnitte mit Schattenvolumen

- zuerst werden die Objekte gezeichnet
- anschließend wird Schreiben in den Farb- und Tiefenpuffer deaktiviert
- zeichne dann die Schattenvolumen
  - Vorderseiten erhöhen den Stencil Buffer
  - Rückseiten erniedrigen den Stencil Buffer
- Schatten ist dort, wo Stencil > 0

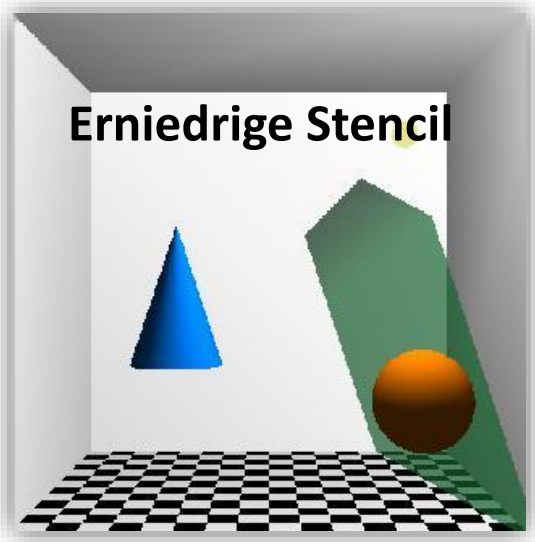


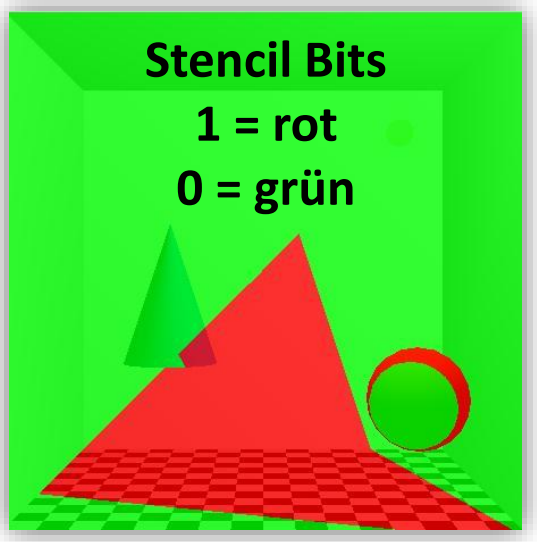
- im Prinzip führen wir den Odd-Even Test mit Strahlen in Richtung Kamera durch – warum treten die Probleme des Odd-Even Tests, z.B. Schnitt Strahl-Vertex, nicht auf?

Schattenvolumen: Zusammenfassung

Verwendung mit Beleuchtungsberechnung

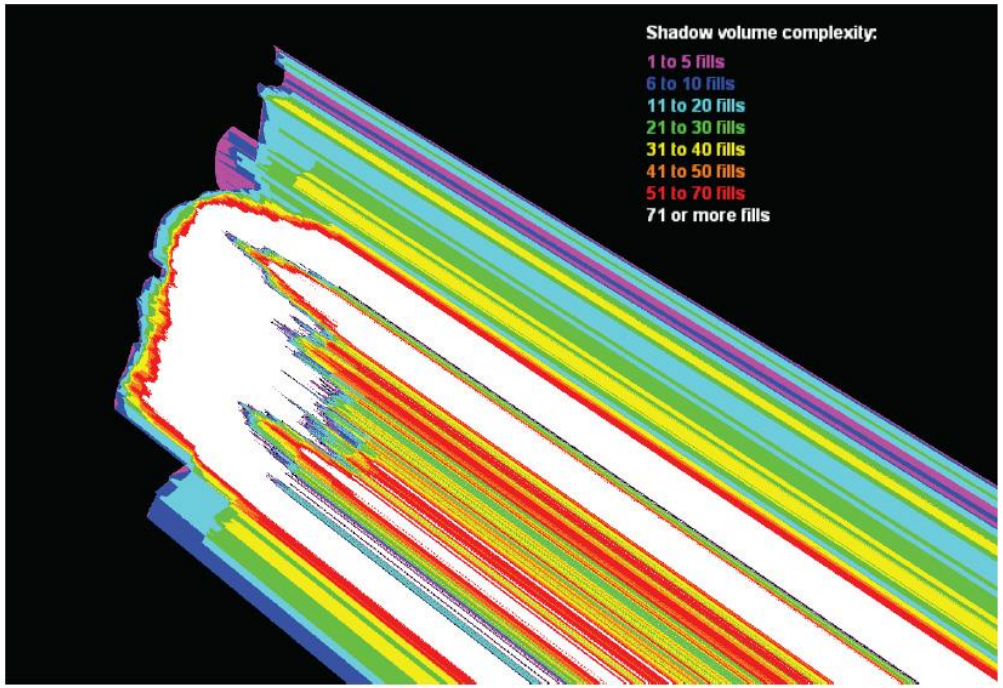
- ein Nachteil des SV-Verfahrens: der Schattentest kann nicht direkt im Fragment Shader (= direkt beim Shading) stattfinden, daher:
  - zeichne die Szene nur mit ambienter Beleuchtung (und Tiefenpuffer)
  - deaktiviere Schreibzugriffe auf Tiefen- und Farb-Puffer
  - zeichne die Vorder- und Rückseiten des SV mit Stencil-Buffer
  - zeichne die Szene mit Beleuchtung dort, wo der Stencilwert $= 0$ ist





Schattenvolumen: Problem Tiefenkomplexität

- selbst einfache Objekte können komplexe Schattenvolumen erzeugen
- Tiefenkomplexität = Anzahl der Schnitte des Augstrahls mit SV



Schattenvolumen: Technische Details…

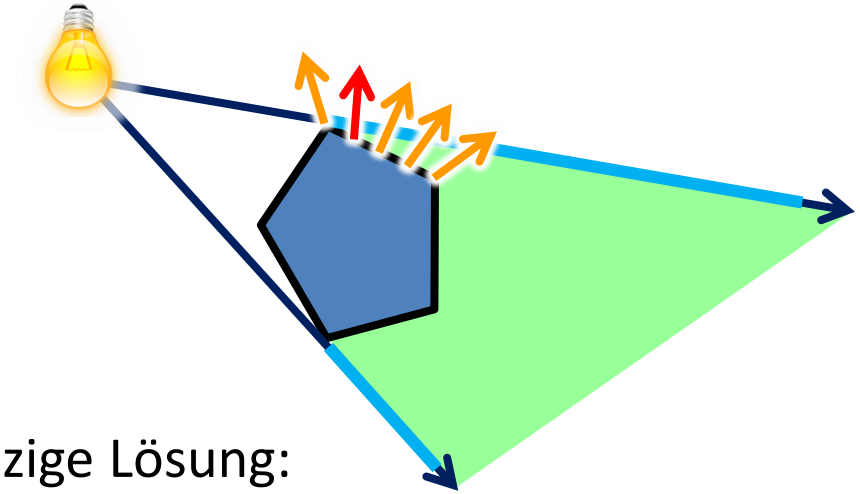
Unterscheidung der Vorder- und Rückseiten

- Backface-Culling (wie beschrieben) ist ineffizient: zweimaliges Zeichnen
- OpenGL "two-sided stencil test"
  - unterschiedliche Stencil-Buffer Operationen können für Vorder- und Rückseiten festgelegt werden:
    `glActiveStencilFace( GL_FRONT ); glStencilOp( ... );`
  - damit genügt ein einmaliges Zeichnen der Schattenvolumen
- Bittiefe des Stencil Buffers ist beschränkt (meist 8 Bit)
  - d.h. der Zähler läuft bei einer Tiefe von 256 über (!)
  - kann zu Problemen bei komplexen Szenen führen
  - deswegen: zeichne abwechselnd Vorder- und Rückseiten, z.B. durch Zeichnen der SV einzelner Objekte und nicht der ganzen Szene oder two-sided Test

Schattenvolumen: Problem Beleuchtung

Inkonsistenz zwischen Verschattung und Beleuchtung

- Entscheidung, ob ein Punkt beleuchtet ist erfolgt durch das SV, berechnet aus der Silhouette, also auf Basis der **Geometrie/Normalen**
- Problem: Beleuchtungsberechnung erfolgt mit interpolierter Normale, was zu sichtbaren Artefakten nahe der Silhouetten führen kann
- Beispiel: interpolierte Normale
  noch zur Lichtquelle gewandt,
  aber im Schatten



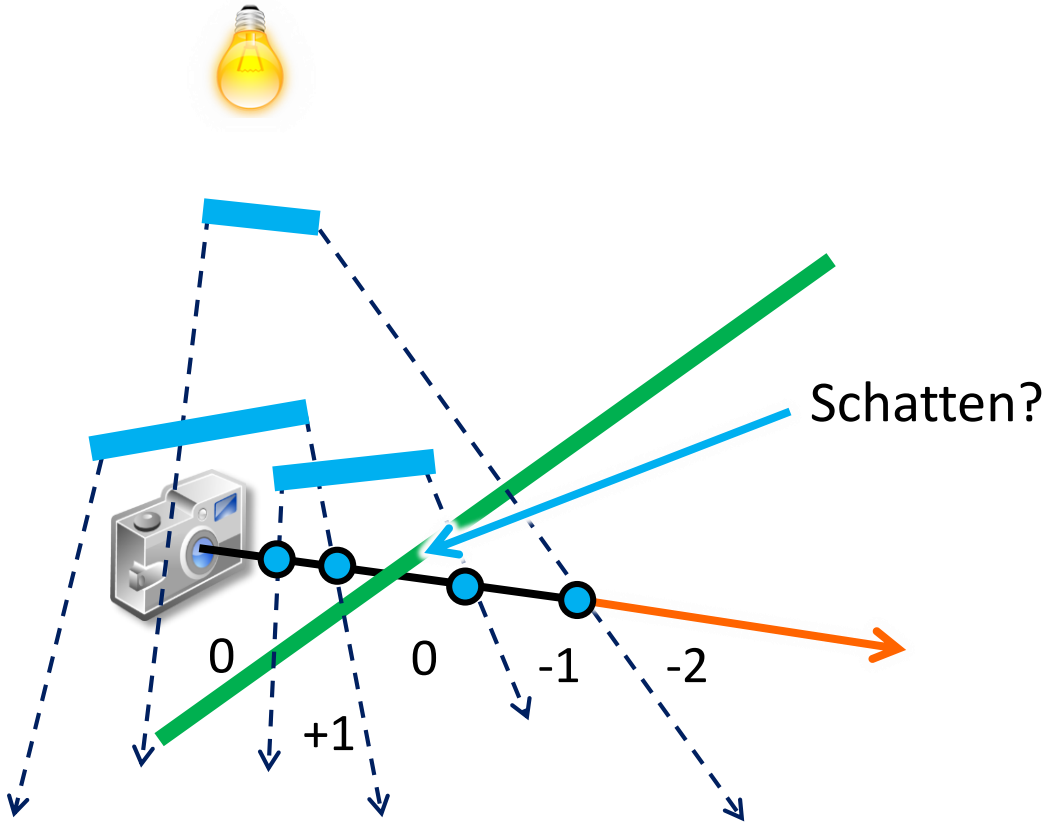
- einzige Lösung:
  feiner tessellierte Netze
  (analoges Problem bei anderen Verfahren!)

Schattenvolumen: Probleme des Stencil Buffer

Probleme durch nicht gefundene Ein- und Austrittspunkte

- Ein- und Austrittspunkte werden nur gezählt, wo das SV rasterisiert wird
- was passiert – wie in diesem Beispiel – wenn sich der Betrachter in einem Schattenvolumen befindet?

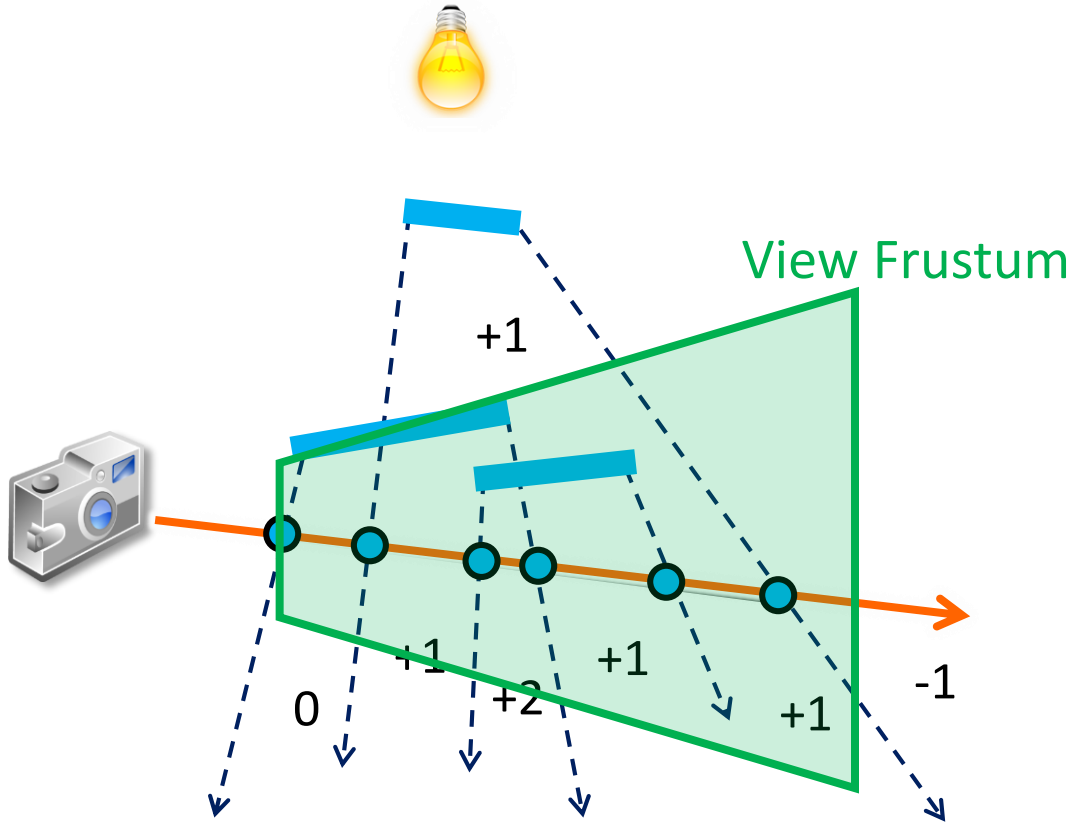
  → der Startwert des Zählers ist falsch!



Schattenvolumen: Probleme des Stencil Buffer

Probleme durch nicht gefundene Ein- und Austrittspunkte

- Ein- und Austrittspunkte werden nur gezählt, wo das SV rasterisiert wird
- die Near Plane kann Eintrittspunkte entfernen und unterschiedliche Pixel müssten eigentlich unterschiedliche Startwerte für den Zähler besitzen
- Lösung: der z-Fail Algorithmus (→ Übung)

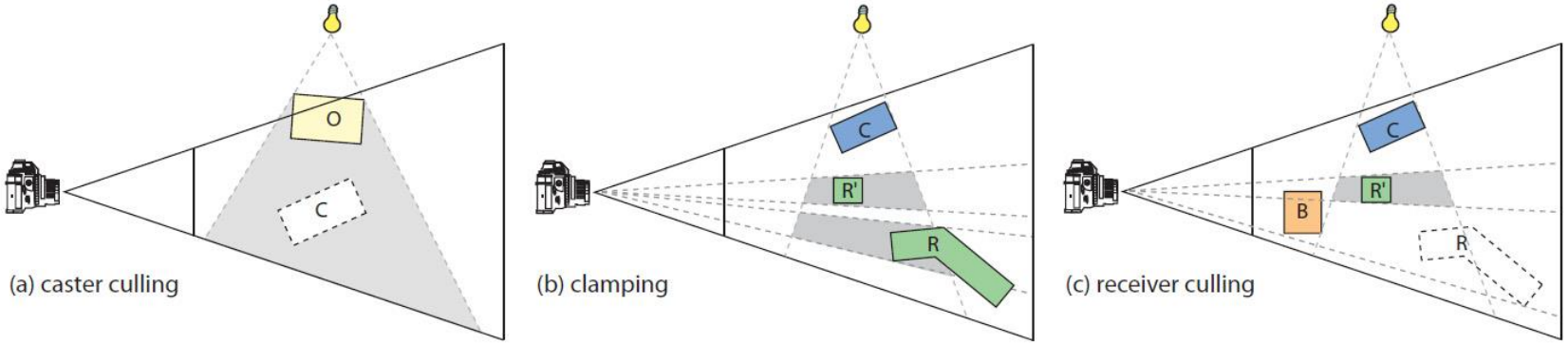


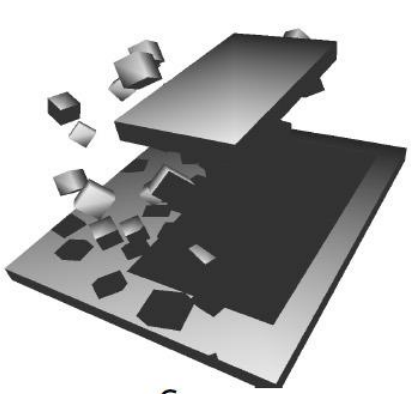
Schattenvolumen: Zusammenfassung

- Eigenschaften / Vorteile
  - Objektraum-Verfahren, d.h. Schatten sind „pixelgenau“
  - Schattenvolumen sind unabhängig von Kamera
    (erlaubt fixe Schattenvolumen bei statischen Objekten/Lichtquellen)
- Nachteile
  - hoher Rasterisierungsaufwand für das Zeichnen der Schattenvolumen
  - mehrere Rendering-Durchgänge notwendig: Zählen der Ein-/Austritte, Rendering von ambientem Licht und Beleuchtung separat, …
  - mehrere Durchgänge bei mehreren Lichtquellen
  - anfällig gegenüber topologischen Problemen (geschlossene Netze etc.)
- Fazit: werden verwendet, wenn **akkurate harte Schatten** wichtig sind, beispielsweise in CAD Anwendungen

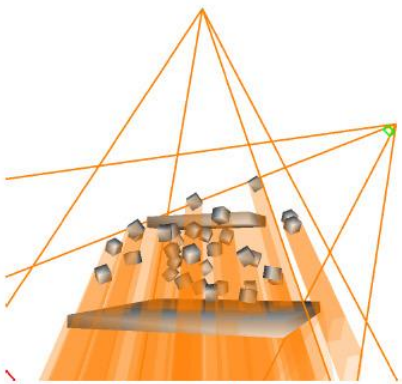
Optimierung von Schattenvolumen

- Verwerfen von Schattenvolumen für Objekte die selbst vollständig im Schatten liegen (links)
- Begrenzen von Schattenvolumen auf Bereiche in denen Objekte liegen (mitte) und nicht von der Kamera verdeckte Bereiche (rechts)
- nur der Blick über den Tellerrand, keine praktische Relevanz…

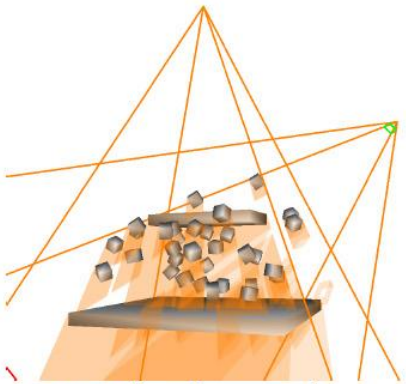




Scene



Shadow volumes



CC Shadow volumes

CC Shadow Volumes, http://gamma.cs.unc.edu/CCSV/