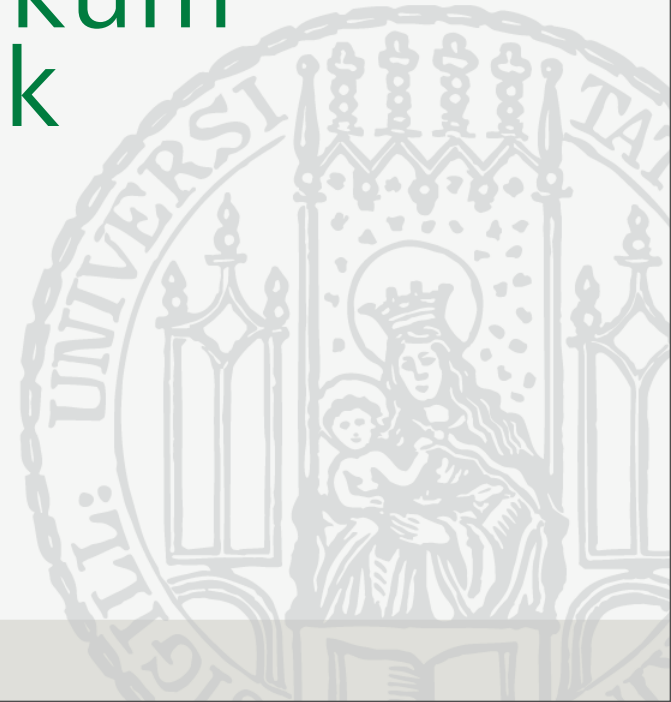


Prof. Andreas Butz, Dipl.Inf. Otmar Hilliges

# Programmierpraktikum 3D Computer Grafik

Dynamische Schattenberechnung





- Der Stencil-Puffer
- Der 1-bit Stencil-Puffer
- Der 8-bit Stencil-Puffer
- Volumetrische Echtzeitschatten
- Soft-Shadows
- Reflexionen
- Weitere Anwendungen



- Weiterer Puffer der Grafikkarte, neben Tiefen- und Framebuffer
- Stencil-Werte werden zusammen mit den Tiefenwerten gespeichert
- Stencil bedeutet Schablone, d.h. einzelne Pixel des Bildes können bestimmte Werte gegeben werden



## OpenGL-Funktionen:

```
gl{Enable|Disable}(GL_STENCIL_TEST);  
glClearStencil(GLint s);  
glStencilMask(GLuint mask);  
glStencilFunc(GLenum func, GLint ref, GLuint mask);  
glStencilOp(GLenum fail, GLenum zfail, GLenum zpass);
```

## Weitere Informationen zu den Funktionen:

- OpenGL Reference Page:  
<http://www.mevis.de/opengl/glStencilOp.html>



- Zwei mögliche Zustände: 0 und 1, z.B. sichtbar vs. unsichtbar
- Mögliche Anwendung:
  - Masken-Effekt



Ganzes Umgebungsbild



Das Cockpit muss nicht zwingend dauernd neu gezeichnet werden, nur das was man durch das Fenster sieht und evtl. einige Instrumente



Stencil Maskierung: Nur die weissen Bereiche müssen dauernd erneuert werden, die restlichen Pixel (dunkel) müssen nicht gerendert werden



das fertige Bild



## Weitere Vorteile des 1-bit Stencil-Puffers:

- Unnötige Arbeit verhindern:
  - Feste Bereiche im Bild nicht bei jedem Frame neu berechnen → Entlastet die Hardware
  - Beispiel: Siehe Folie 5
- Fade-Effekt zweier bewegter Szenen:
  - Sinnvoll bei Animationen (selten bei Spielen)
  - Sichtbares Bild hat alle Stencil-Werte auf 1, unsichtbares auf 0
  - Übergang zwischen beiden Bildern durch Vertauschen einzelner Stencil-Werte zwischen beiden Bildern
- Erzeugen von (Stereo) 3D Bildern für spezielle Hardware
  - HMD, Shutterbrillen, etc.



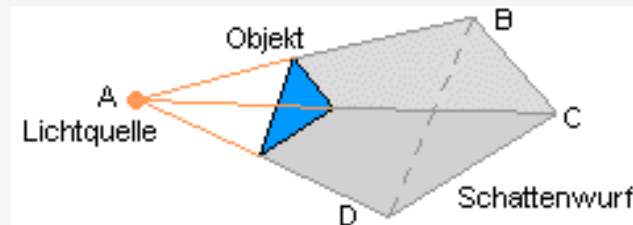
- Ermöglicht 256 Unterschiede pro Pixel, statt der 2 beim 1-bit Puffer
- Üblicherweise werden die 8 Bit auf verschiedene Effekte aufgeteilt:
  - 1 Bit für den Maskierungseffekt
  - 4 Bits für die Schattenberechnung
  - Restliche Bits für einen weiteren Effekt
- Mögliche Effekte:
  - Volumetrische Schatten
  - Reflexionseffekte



- Erfolgt meist am Schluss der Rendering-Pipeline, d.h. wenn alles fertig gerendert und texturiert ist
- Benötigte Parameter:
  - z-Wert jedes Pixels im Tiefen-Puffer
  - Standort der Lichtquelle
- Vorgang:
  - Betrachten des Tiefen-Puffers aus Sicht der Lichtquelle und nicht von der Kameraposition



- Schatten beginnen an den Kanten eines Objekts und folgen der Gerade, die von der Lichtquelle ausgeht
- Beispiel (Schattendreiecke):







- Nun folgt die Tiefenwert-Berechnung der Dreiecke → Test, ob die Pixel im Framebuffer innerhalb dieser Schattenzone liegen



- Nun folgt die Tiefenwert-Berechnung der Dreiecke → Test, ob die Pixel im Framebuffer innerhalb dieser Schattenzone liegen
- Liegen diese innerhalb, werden die entsprechenden Stencil-Bits erhöht

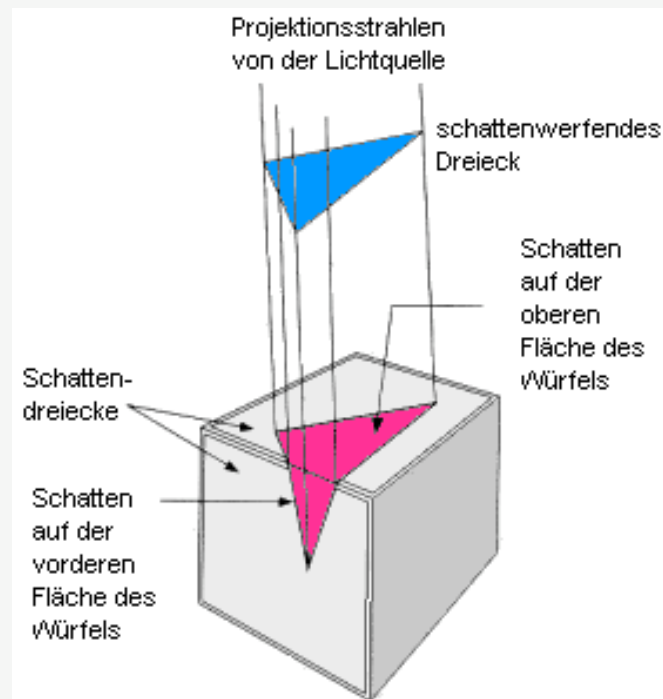


- Nun folgt die Tiefenwert-Berechnung der Dreiecke → Test, ob die Pixel im Framebuffer innerhalb dieser Schattenzone liegen
- Liegen diese innerhalb, werden die entsprechenden Stencil-Bits erhöht
- Nun blendet man ein abgedunkeltes Viereck über die gerenderte Szene



- Nun folgt die Tiefenwert-Berechnung der Dreiecke → Test, ob die Pixel im Framebuffer innerhalb dieser Schattenzone liegen
- Liegen diese innerhalb, werden die entsprechenden Stencil-Bits erhöht
- Nun blendet man ein abgedunkeltes Viereck über die gerenderte Szene
- Dieses Viereck ist nur an den Stellen sichtbar, an denen Stencil-Flags sind

## Veranschaulichung:





- Weitere Form der Schattenberechnung mit dem Stencil-Puffer:
  - Entsprechende (markierte) Pixel neu rendern
  - Material- und Nebeneffekte wirken besser
- Etwas aufwendigerer Ansatz, aber auf moderner Hardware immer noch sehr schnell







- Neuer Ansatz:



- Neuer Ansatz:
  - Szene untexturiert und unschattiert rendern



- Neuer Ansatz:
  - Szene untexturiert und unshattiert rendern
  - Nur Tiefenwerte speichern und Schattendreiecke bestimmen



- Neuer Ansatz:
  - Szene untexturiert und unshattiert rendern
  - Nur Tiefenwerte speichern und Schattendreiecke bestimmen
  - Mit Stencil-Bit markierte Bereiche mit **deaktivierten** Lichtquellen berechnen



## ■ Neuer Ansatz:

- Szene untexturiert und unshattiert rendern
- Nur Tiefenwerte speichern und Schattendreiecke bestimmen
- Mit Stencil-Bit markierte Bereiche mit **deaktivierten** Lichtquellen berechnen
- Nicht-markierte Bereiche mit **aktivierten** Lichtquellen berechnen

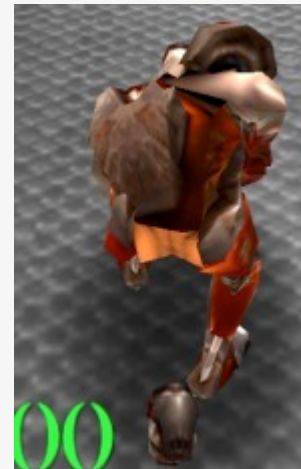


- Neuer Ansatz:
  - Szene untexturiert und unshattiert rendern
  - Nur Tiefenwerte speichern und Schattendreiecke bestimmen
  - Mit Stencil-Bit markierte Bereiche mit **deaktivierten** Lichtquellen berechnen
  - Nicht-markierte Bereiche mit **aktivierten** Lichtquellen berechnen
- Liefert beste Resultate, ist aber auch am aufwendigsten

- Alle Verfahren funktionieren am besten, solange das Objekt einfach gehalten ist
- In Spielen wird daher die einfachste Variante des entsprechenden Objekts gewählt, falls mehrere verfügbar sind



Volumetrischer  
Echtzeitschatten



Standardschatten  
als grauer Fleck







- Reine Echtzeitschatten sind nicht realistisch genug



- Reine Echtzeitschatten sind nicht realistisch genug
- Schatten besitzen normalerweise keine kantigen Ränder → Verlauf



- Reine Echtzeitschatten sind nicht realistisch genug
- Schatten besitzen normalerweise keine kantigen Ränder → Verlauf
- Ansatzweise sind Soft-Shadows durch den Stencil-Puffer realisierbar





- Verfahren:



- Verfahren:
  - Zusätzlich größeres Schattendreieck mit höherem Transparenzwert berechnen



## ■ Verfahren:

- Zusätzlich größeres Schattendreieck mit höherem Transparenzwert berechnen
- Dieses Schattendreieck über das originale Schattendreieck legen





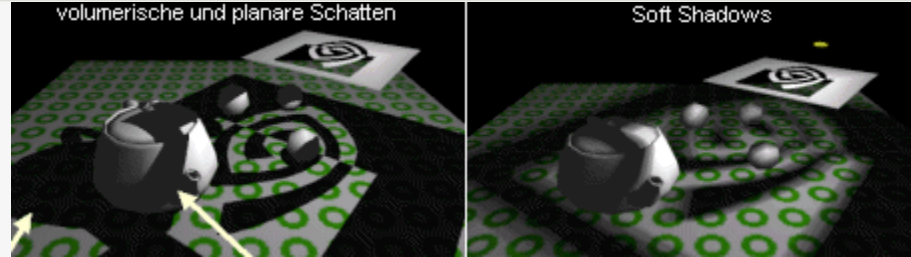
- Verfahren:
  - Zusätzlich größeres Schattendreieck mit höherem Transparenzwert berechnen
  - Dieses Schattendreieck über das originale Schattendreieck legen
  - Funktioniert nur bei Schattenwürfen auf Böden und Mauern
- Weitere Möglichkeit:



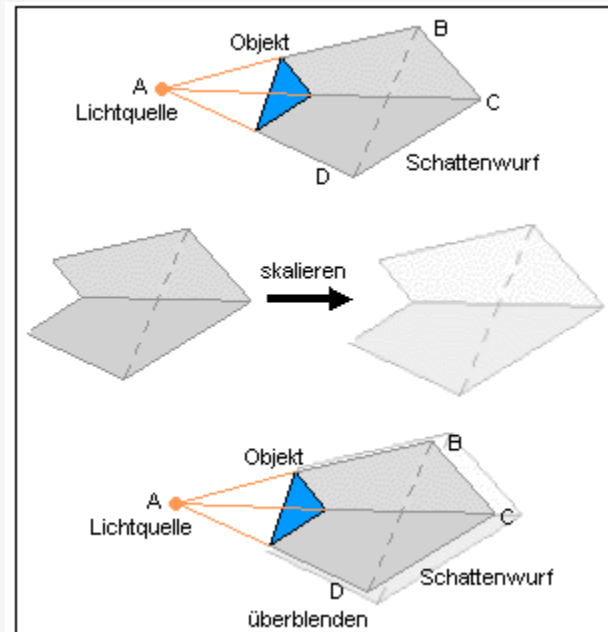
- **Verfahren:**
  - Zusätzlich größeres Schattendreieck mit höherem Transparenzwert berechnen
  - Dieses Schattendreieck über das originale Schattendreieck legen
  - Funktioniert nur bei Schattenwürfen auf Böden und Mauern
- **Weitere Möglichkeit:**
  - Minimale Verschiebung der Lichtquelle und Ändern des Transparenzwerts und jeweils eine Neuberechnung der Szene



Beispiel:

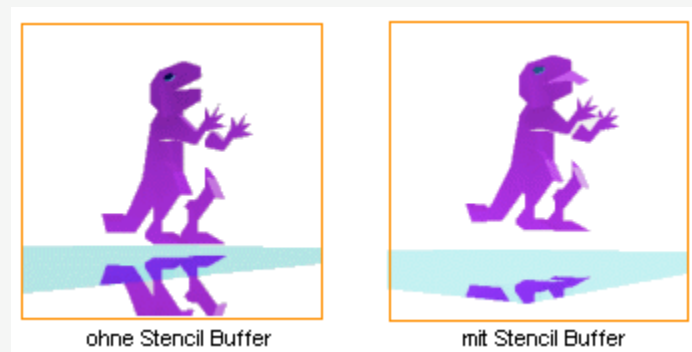


Vergleich von Echtzeitschatten und Soft-Shadows



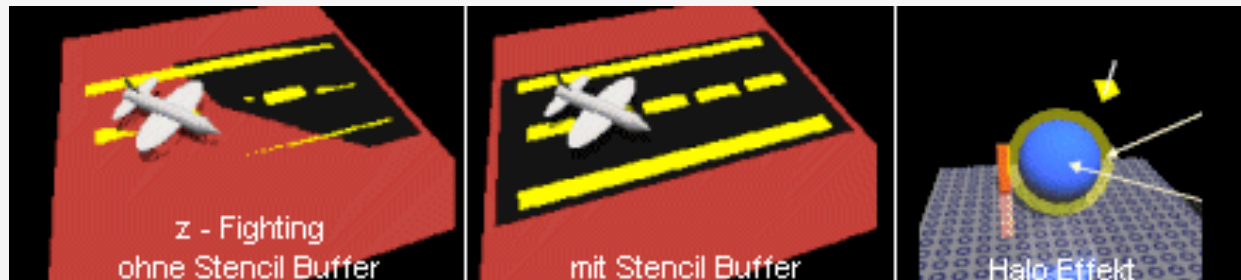
Approximierte Soft-Shadows

- Stencil-Puffer ist ein hilfreiches Mittel für Reflexionen
- Objekte spiegeln sich eventuell nicht an allen Oberflächen
- Es werden daher nur Polygone markiert, die auch reflektieren sollen





- Verhinderung von Polygonflackern (z-Fighting), d.h. Co-planare Dreiecke flackern nicht
- Leucht-Effekt (Halo): Objekt wird mit Stencil-Puffer gezeichnet, vergrößert und dort halbtransparent gezeichnet, wo keine Stencil-Werte gesetzt sind





OpenGL Reference Page: <http://www.mevis.de/opengl/opengl.html>

GLUT Man Pages:

<http://www.cs.uccs.edu/~semwal/man.html>

3D Concepts:

<http://www.3dconcept.ch/artikel/stencilPuffer/>