

Übung 6: Gruppe 28

Niclas Kusenbach, 360227 Alicia Bayerl, 2633336

Mohamed Naceur Hedhili, 2957151 Selma Naz Öner, 2662640

December 18, 2025

7.1 Schatten

a) Phong-Beleuchtungsmodell (vgl. Slide 66)

Die Lichtintensität I_{total} an einem Punkt setzt sich zusammen aus:

$$I_{total} = I_{amb} + I_{diff} + I_{spec}$$

1. **Ambientes Licht** (I_{amb}): Grundhelligkeit der Szene (indirekte Beleuchtung simuliert durch Konstante). Richtungsunabhängig.

$$I_{amb} = k_{amb} \cdot C_{amb}$$

2. **Diffuse Reflexion** (I_{diff}): Matte Oberflächen (Lambert-Reflexion). Helligkeit hängt vom Winkel zwischen Lichtvektor L und Oberflächennormale N ab.

$$I_{diff} = k_{diff} \cdot C_{light} \cdot (\vec{N} \cdot \vec{L})$$

3. **Spiegelnde Reflexion** (I_{spec}): Glanzpunkte (Highlights). Hängt vom Winkel zwischen Reflexionsvektor R und Blickvektor V ab. Der Exponent m bestimmt die Rauigkeit (je höher m , desto kleiner und schärfer der Glanzpunkt).

$$I_{spec} = k_{spec} \cdot C_{light} \cdot (\vec{R} \cdot \vec{V})^m$$

b) weitere Schatten (vgl. Slide 75)

1. Self-Shadowing: Wenn ein Objekt Schatten auf sich selbst wirft.
2. Cast Shadows: Der Schatten, den ein Objekt auf andere Objekte wirft.

7.2 Rasterisierung (vgl. Slide 92)

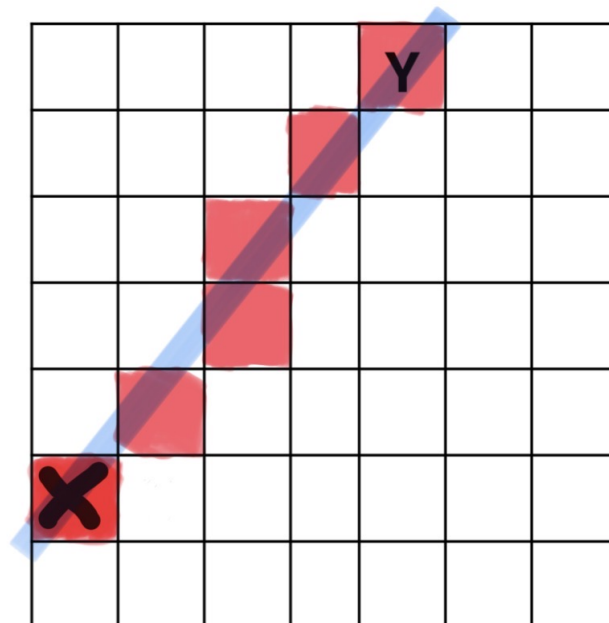
Gegeben: Startpunkt $(x_1, y_1) = (1, 2)$, Endpunkt $(x_2, y_2) = (5, 7)$. Berechnung der Differenzen: $\Delta x = 5 - 1 = 4$, $\Delta y = 7 - 2 = 5$

Da $\Delta y > \Delta x$, handelt es sich um eine steile Linie (Steigung > 1). Der Algorithmus muss daher entlang der y-Achse schreiten und die x-Koordinate anpassen. Wir vertauschen die Logik von x und y im Vergleich zum Standard-Algorithmus auf Folie 92.

1. $\delta x \leftarrow 7 - 2 = 5$ (entspricht real Δy)
2. $\delta y \leftarrow 5 - 1 = 4$ (entspricht real Δx)
3. $x \leftarrow 2$
4. $y \leftarrow 1$
5. **Setze Pixel**(y, x) $\Rightarrow (1, 2)$
6. $\xi \leftarrow \lfloor \delta x / 2 \rfloor = \lfloor 5 / 2 \rfloor = 2$ (Integer-Division)

Die Schleife läuft solange $x < x_2$ (also $x < 7$).

Iter.	x (Start)	Schritt 9: $\xi \leftarrow \xi - \delta y$	Test $\xi < 0$	y (neu)	Schritt 12: ξ (Korr.)	Pixel (y, x)
1	2 \rightarrow 3	2 - 4 = -2	Ja	1 + 1 = 2	-2 + 5 = 3	(2, 3)
2	3 \rightarrow 4	3 - 4 = -1	Ja	2 + 1 = 3	-1 + 5 = 4	(3, 4)
3	4 \rightarrow 5	4 - 4 = 0	Nein	3 (bleibt)	- (keine Korr.)	(3, 5)
4	5 \rightarrow 6	0 - 4 = -4	Ja	3 + 1 = 4	-4 + 5 = 1	(4, 6)
5	6 \rightarrow 7	1 - 4 = -3	Ja	4 + 1 = 5	-3 + 5 = 2	(5, 7)



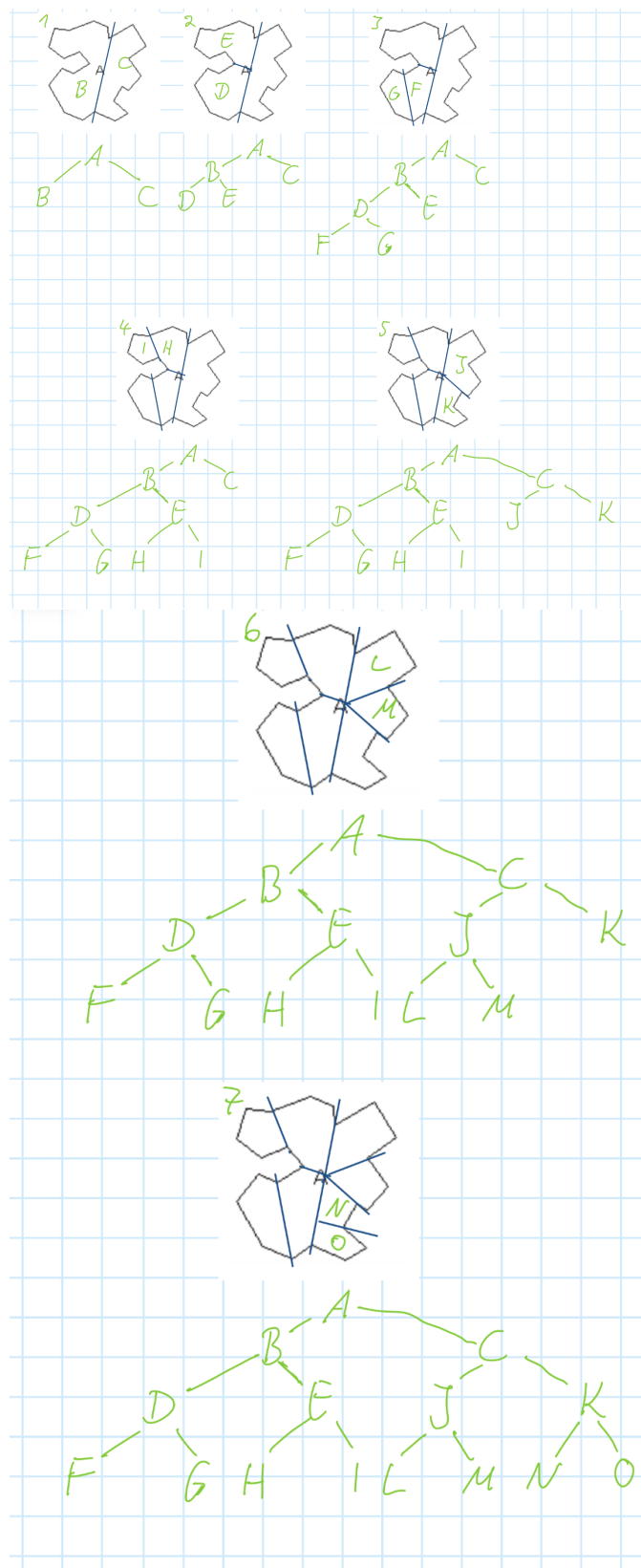
1 7.3 Räumliche Datenstrukturen

a) Binary Space Partitioning (vgl. Slide 61, Moodle Forum)

Theorie: Definition der Unterteilungsebenen

- Die Schnittebenen werden in einem BSP-Tree üblicherweise durch die Polygone der Szene selbst definiert (sogenanntes Auto-Partitioning).
- *Zusatzinfo aus Forum:* Der Unterteilungsprozess wird fortgesetzt, bis alle resultierenden Regionen (Zellen) konvex sind. Da einzelne Liniensegmente/Polygone stets konvex sind, endet der Prozess spätestens, wenn jeder Blattknoten nur noch ein Fragment enthält.

Anwendung: Fortführung des Algorithmus



b) Backface Culling (vgl. Slide 84)

Gegeben: $\vec{n} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \vec{s} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$

Aus der Vorlesung wissen wir, dass wenn $\vec{n} \cdot \vec{s} > 0$, dann ist es eine Rückseite.
Also rechnen wir:

$$\vec{n} \cdot \vec{s} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = 0 \cdot 0 + 0 \cdot 1 + 1 \cdot 1 = 1$$

Damit ist $\vec{n} \cdot \vec{s} = 1 > 0$ und das Dreieck mit der Normalen \vec{n} ist eine Rückseite.

c) (vgl. Slide 83, 98)

Bei konvexen Körpern entfernt Backface Culling alle unsichtbaren Flächen und macht damit z-Buffer theoretisch überflüssig.

Ein Nachteil ist, dass Transparenz nicht korrekt realisierbar ist, da der z-Buffer nur einen Tiefenwert pro Pixel speichert. Es kann immer nur das vorderste Objekt dargestellt werden, wodurch dahinterliegende Objekte verworfen werden.