

Computergrafik 1

Blatt 11

Picking

- Raycasting
 - Strahl in die Szene „schicken“ und Schnittpunkte mit Objekten berechnen
 - Rechenaufwändig
- Color-Coding
 - Eindeutige Farbe pro Objekt (unabhängig von der tatsächlich ausgegebenen Farbe)
 - Rendering mit diesen Farben in einem speziellen Buffer
 - Bestimmung des Objekt über die Pixelfarbe
- OpenGL: GL_SELECT
 - Spezieller Rendering-Modus (keine Ausgabe auf dem Display)
 - Eigener „Pick“-Buffer und eindeutige IDs
 - Im GL_SELECT-Modus werden alle Objekte in einem kleinen Fenster um die Mausposition gerendert
 - Identifizierung der dort gerenderten Objekte

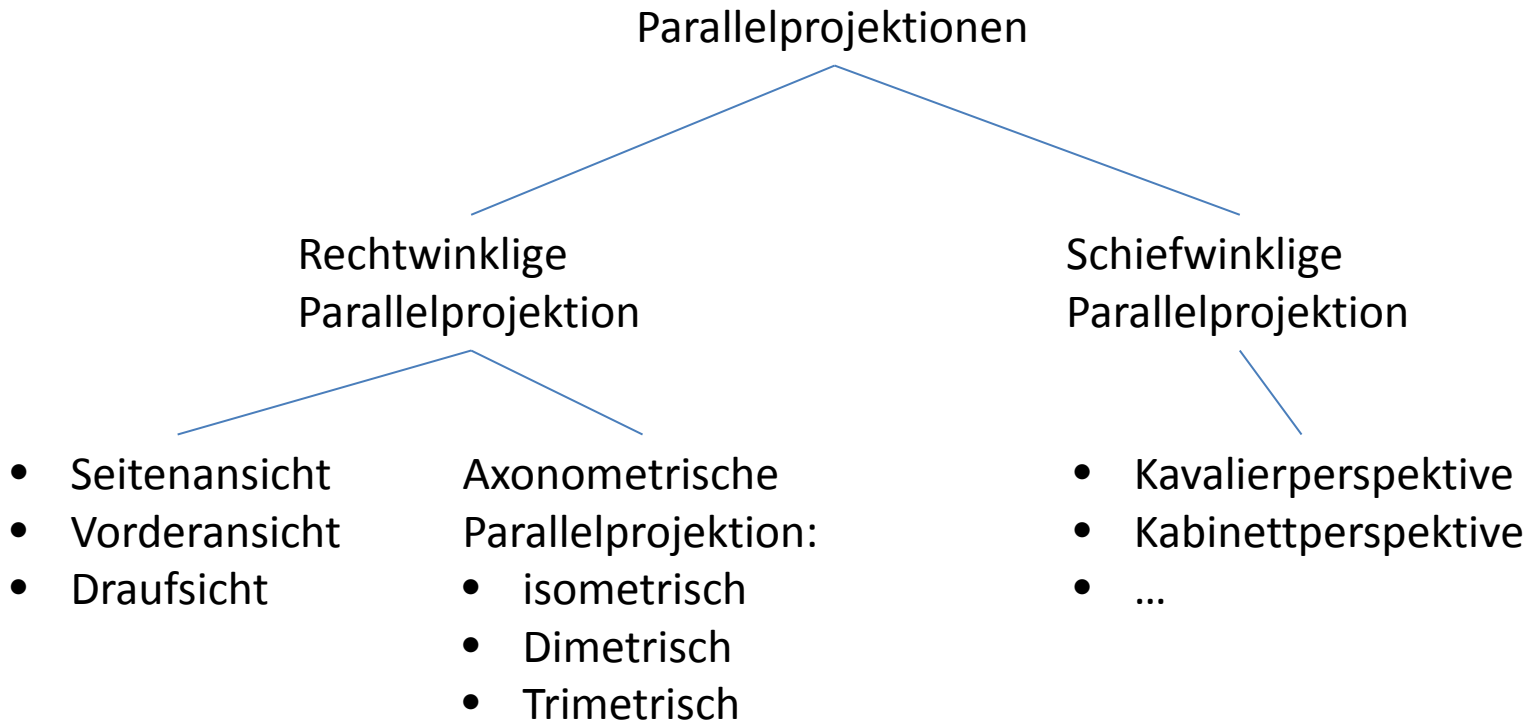
Transformationen

(Skalierungsmatrix*Translationsmatrix*Rotationsmatrix)*Punkt

$$\text{Transformationsmatrix } T = \begin{pmatrix} 5 & 0 & 0 & 25 \\ 0 & \frac{707}{200} & -\frac{707}{200} & 10 \\ 0 & \frac{707}{200} & \frac{707}{200} & -15 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T^* (1,1,1,1) = \begin{pmatrix} 30 \\ 10 \\ -7,93 \\ 1 \end{pmatrix}$$

Projektionen: Parallelprojektion

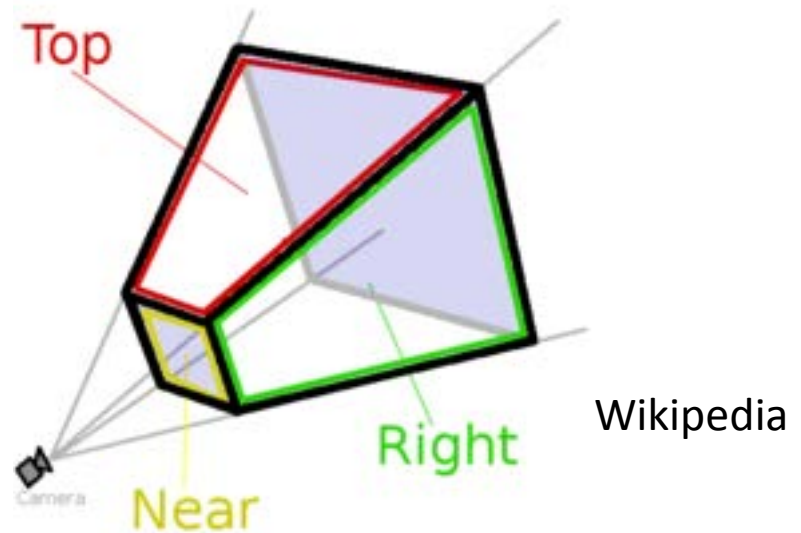


Projektionen: Parallelprojektion

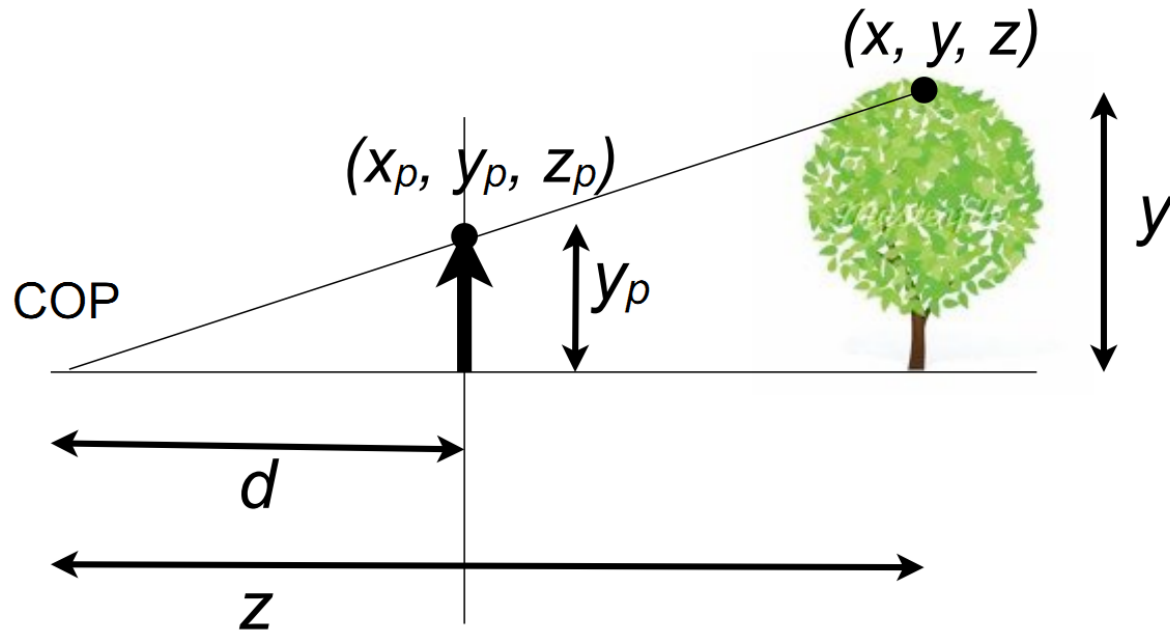
- Rechtwinklige Parallelprojektion:
 - Normale der Sichte ebene und Richtung der Projektion stimmen überein
- Schiefwinklige Parallelprojektion:
 - Normale der Sichte ebene und Richtung der Projektion stimmen nicht überein
- Projektionszentrum: im Unendlichen → Projektoren verlaufen parallel
- Axonometrische Parallelprojektion:
 - Normale der Sichte ebene nicht parallel zu einer der Koordinatenachsen
 - Isometrisch: Winkel zwischen v_{pn} und allen drei Koordinatenachsen gleich
 - Dimetrisch: Winkel zwischen v_{pn} und zwei Koordinatenachsen gleich
 - Trimetrisch: Winkel zwischen v_{pn} und allen drei Koordinatenachsen unterschiedlich

Projektionen: Perspektivische Projektion

- Perspektivische Verkürzung
- Projektionsstrahlen laufen in einem Projektionszentrum zusammen



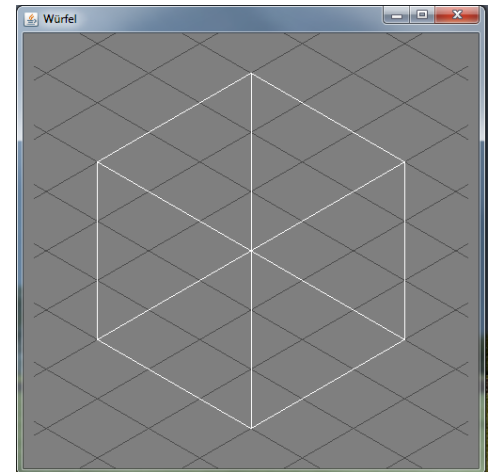
Projektionen: Perspektivische Projektion



Isometrische Projektion

- Spezialfall der rechtwinkligen Parallelprojektion

```
gl.glMatrixMode(GL2.GL_PROJECTION);  
gl.glLoadIdentity();  
gl.glOrtho(-3, 3, -3, 3, 0, 100);  
gl.glMatrixMode(GL2.GL_MODELVIEW);  
gl.glLoadIdentity();  
glu.gluLookat(3, 3, 3, 0, 0, 0, 1, 0);
```

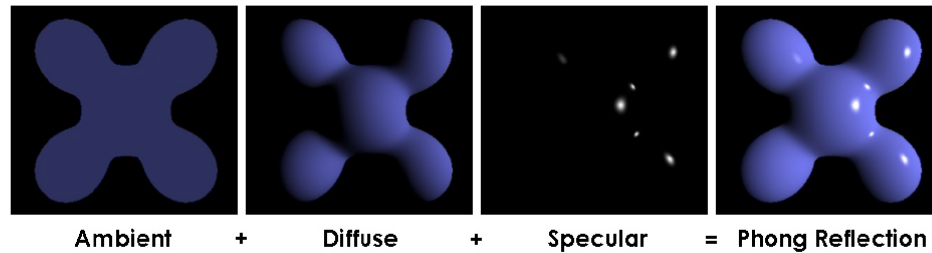


Lichter

- **Punktlicht**
 - Position im Raum
 - Unendlich klein
 - Licht wird gleichmäßig in alle Richtungen ausgestrahlt
 - Intensitätsabnahme kann unterschiedlich definiert werden
 - Harte Objekt- und Schlagschatten
- **Spot**
 - Position und Orientierung im Raum
 - Öffnungswinkel
 - Weicher Übergang an den Grenzen des Lichtstrahls
- **Gerichtetes Licht**
 - Annahme: Licht ist für alle Objekte gleich weit entfernt
 - Keine Position im Raum, keine Intensitätsabnahme
 - Parameter: Richtung und Intensität
- **Ambientes Licht**
 - Gleiche Intensität aus allen Richtungen
 - Simuliert indirekten Lichteinfall

Beleuchtungmodell nach Phong

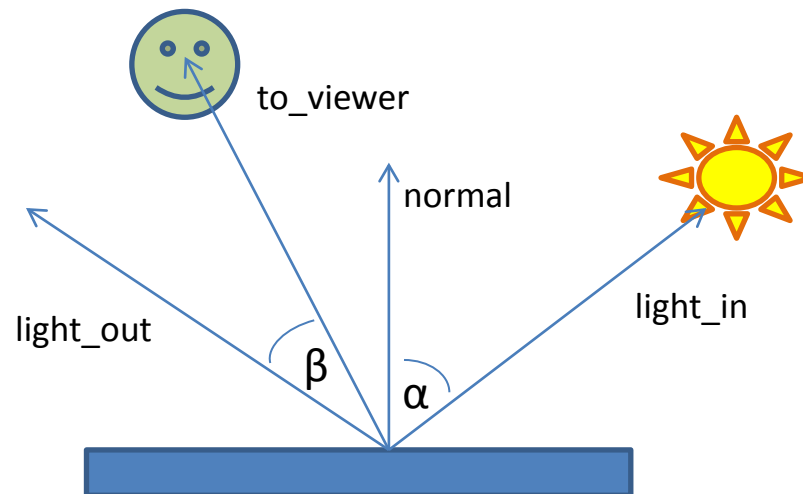
$$I_o = I_{amb} + I_{diff} + I_{spec}$$



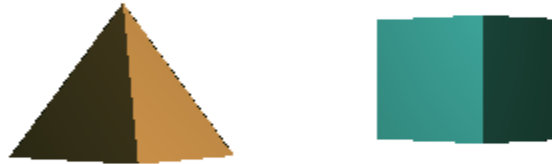
$$I_{amb} = I_a * k_a$$

$$I_{diff} = I_{diff} * k_{diff} * \cos\alpha$$

$$I_{sp} = I_{sp} * k_{sp} * \cos^n\beta$$



Animation



Animation

Schritte:

- Einfache Beleuchtung definieren
- Pyramide und Würfel erstellen
 - Geometrie
 - Normalvektoren (warum?)
- Animation
- Transformationen
 - Rotation der Pyramide
 - Rotation des Würfels