Theorieübung 1

Graphische Datenverarbeitung I
WiSe 2024/25
Team 01

1 Pipeline

1.1 Input

Woraus besteht der Input der Pipeline?

Der Input einer Pipeline besteht aus Beleuchtungsalgorithmen, den Koordinaten und Texturen der Objekte, Positionsdaten von Lichtquellen und der Kamera.

1.2 Objekte

Zum Input gehören unter anderem ,,Objekte". In welcher Form sind ,,Objekte" im Input konkret gegeben?

Die Objekte werde durch die Koordinaten ihrer Vertices beschrieben, die Kanten erzeugen, wenn Sie verbunden werden. Dadurch werden die Faces der Objekte repräsentiert. Welche Vertuces mit welchen anderen Vertices verbunden sind, wird ebenfalls mitgegeben. Außerdem sind die Texturen der Objektseiten relevant, wie auch Normal Maps, UV Maps, Displacement Maps, etc.

1.3 Output

Was ist der Output der Pipeline?

Die Ausgabe der Pipeline ist ein zwei-dimensionales Bild, welches eine vollständig gerenderte Ansicht der Kamera der 3D-Szene ist.

1.4 Pipelining

Weshalb ist eine Pipeline, die aus n-Abschnitten besteht, (theoretisch) n-mal schneller als eine Pipeline mit nur einem Abschnitt?

Weil die Pipeline gewisse Abschnitte theoretisch parallel durchlaufen kann. So wird simultan an verschiedenen Schritten gearbeitet, wodurch die Gesamtberechnungszeit verringert wird.

1.5 Bottleneck

Weshalb ist die Pipeline Geschwindigkeit vom Bottleneck abhängig? Wieso warten die anderen Pipeline Abschnitte bis der Bottleneck-Abschnitt fertig ist?

Sodass die Abschnitte der Pipeline arbeiten können, benötigen Sie die Ausgabedaten vorheriger Schritte. Wenn ein Bottleneck-Abschnitt länger braucht um deren Ausgabe zu errechnen, müssen die nachfolgenden Abschnitte darauf warten, bis der Bottleneck-Abschnitt fertig ist. Das gleiche gilt für vorherige Abschnitte, die darauf warten müssen, bis der Bottleneck Abschnitt fertig ist, und mit deren Ausgabe weiterrechnet. Diese Abschnitte sind im 'Idle' bis der Bottleneck-Abschnitt wieder bereit ist.

2 Model & View Transformation

2.1 Globale und lokale Koordinaten

Stellen Sie die Gleichung $(x, z)^T = f(u, v)$ auf, die die u, v Koordinaten in das Weltkoordinatensystem transformiert.

$$f(u,v) = \begin{pmatrix} x_{\text{Planet}} \\ z_{\text{Planet}} \end{pmatrix} + \begin{pmatrix} \cos(350°) & -\sin(350°) \\ \sin(350°) & \cos(350°) \end{pmatrix} * \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} u \\ v \end{pmatrix}$$
$$\begin{pmatrix} x_{\text{Planet}} \\ z_{\text{Planet}} \end{pmatrix} = \begin{pmatrix} 4 * \cos(100°) \\ -4 * \sin(100°) \end{pmatrix}$$

Bestimmen Sie nun die Positionen der Szenenobjekte bezüglich des Weltkoordinatensystems

- Sonne: $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
- Planet: $\begin{pmatrix} 4 * \cos(100°) \\ 4 * \sin(100°) \end{pmatrix} \equiv \begin{pmatrix} -0.69 \\ 3.94 \end{pmatrix}$
- Mond: $\begin{pmatrix} -0.69459 \\ -3.939 \end{pmatrix} + \begin{pmatrix} cos(350°) & -\sin(350°) \\ \sin(350°) & \cos(350°) \end{pmatrix} * \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} -2 \\ 0 \end{pmatrix} = \begin{pmatrix} 1.279 \\ -4.287 \end{pmatrix}$

2.2 Ausrichtung der Kamera

Bestimmen Sie, welche Translation und welche Rotation auf die Szene ausgeübt werden müssen, um die Kamera in den Ursprung zu verschieben und anschließend die Blickrichtung nach z zu rotieren

1. Translation der Szene
   Die Kamera muss um $-(2, 2\sqrt{3})$ verschoben werden, sodass dieser im Ursprung des Koordinatensystems ist.
   Translation $T = \begin{pmatrix} -2 \\ -2\sqrt{3} \end{pmatrix}$

2. Rotation der Szene
   Die Kamera ist nun im Ursprung, muss jedoch noch so rotiert werden, sodass der Blickvektor mit der z-Achse in negative Richtung übereinstimmt. Der aktuelle Blickvektor der Kamera ist $\begin{pmatrix} -2 \\ -2\sqrt{3} \end{pmatrix}$
   Der Zielvektor ist $\vec{z}_- = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$

   Winkel zwischen Blickrichtung Kamera und $\vec{z}_-$ kann wie folgt berechnet werden:

$$\cos(\theta) = \frac{\vec{k} \circ \vec{z}_-}{||\vec{k}|| * ||\vec{z}_-||}$$

$$\vec{k} \circ \vec{z}_- = \begin{pmatrix} -2 \\ -2\sqrt{3} \end{pmatrix} * \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -2 * 0 + -2\sqrt{3} * -1 = 2\sqrt{3}$$

$$||\vec{k}|| = \sqrt{(-2)^2 + (-2\sqrt{3})2} = 4$$

$$||\vec{z}_-|| = \sqrt{(0)^2 + (-1} = 1$$

$$\cos(\theta) = \frac{2\sqrt{3}}{4} = \frac{\sqrt{3}}{2}$$

$$\theta = 30^\circ$$

   Rotationsmatrix 2D für 30°: $R = \begin{pmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix}$

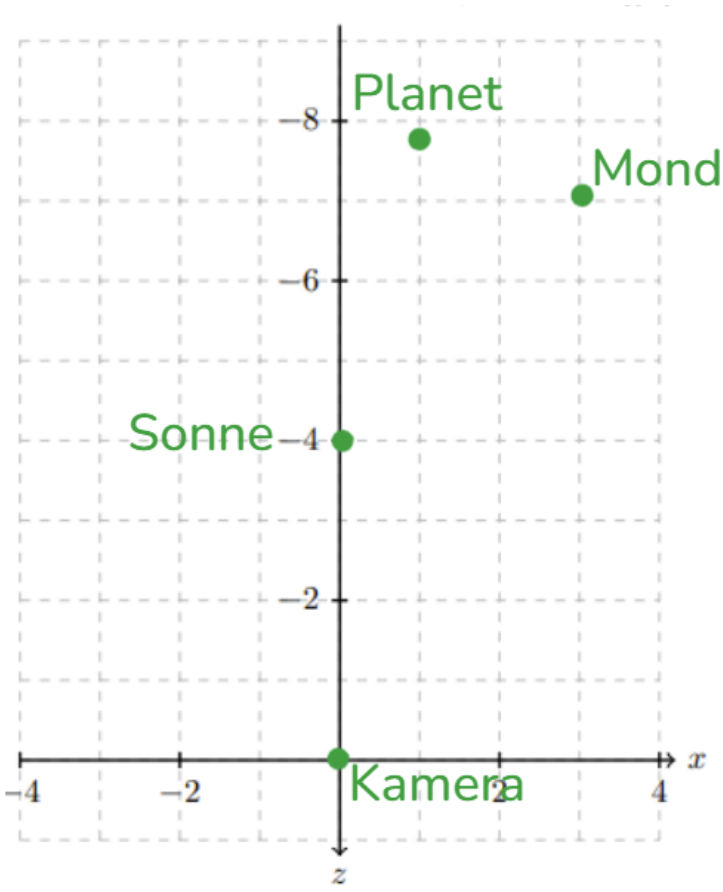
2.3 Model View Transformation

Berechnen Sie die Positionen der Szenenobjekte nach der Model- und View-Transformation. Fertigen Sie eine Skizze an

$$g \begin{pmatrix} x \\ z \end{pmatrix} = \begin{pmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{pmatrix} * (\begin{pmatrix} x \\ z \end{pmatrix} - \begin{pmatrix} 2 \\ 2\sqrt{3} \end{pmatrix})$$

- Sonne: $g \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{pmatrix} * (\begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2\sqrt{3} \end{pmatrix}) = \begin{pmatrix} 0 \\ -4 \end{pmatrix}$
- Planet: $g \begin{pmatrix} -0.69 \\ 3.94 \end{pmatrix} = \begin{pmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{pmatrix} * \begin{pmatrix} -2.69 \\ 0.4759 \end{pmatrix} = \begin{pmatrix} 1,37 \\ -7,76 \end{pmatrix}$

- Mond: $g\begin{pmatrix}1.279\\-4.287\end{pmatrix} = \begin{pmatrix}\cos(30°) & -\sin(30°)\\ \sin(30°) & \cos(30°)\end{pmatrix} * (\begin{pmatrix}1.279\\-4.287\end{pmatrix} - \begin{pmatrix}2\\2\sqrt{3}\end{pmatrix}) = \begin{pmatrix}3.25\\-7.07\end{pmatrix}$
- Kamera: $g\begin{pmatrix}2\\2\sqrt{3}\end{pmatrix} = \begin{pmatrix}\cos(30°) & -\sin(30°)\\ \sin(30°) & \cos(30°)\end{pmatrix} * (\begin{pmatrix}2\\2\sqrt{3}\end{pmatrix} - \begin{pmatrix}2\\2\sqrt{3}\end{pmatrix}) = \begin{pmatrix}0\\0\end{pmatrix}$



3 Optische Triangulation

3.1 Pixel zu Mittelpunkt

Stellen Sie die Gleichung $\beta = f_1(p)$ auf, um aus einer Pixelposition p (Ganzzahl zwischen 0 und 1023) den Winkel β zu berechnen. Verwenden Sie dabei die Koordinate des Pixelmittelpunktes!

$$f_1(p) = \arctan(\frac{(p - 511.5) * 48mm}{1024 * 24mm})$$

Wie groß ist β, wenn der Laserpunkt in der Mitte von Pixel 522 registriert wird?

$$f_1(522) = \arctan(\frac{(522 - 511.5) * 48mm}{1024 * 24mm}) \approx 1.17^\circ$$

3.2 Spiegelwinkel zu Winkel

Stellen Sie die Gleichung $\alpha = f_2(\gamma)$ auf, um aus dem Spiegelwinkel γ den Winkel α zu berechnen.

$$f_2(\gamma) = \gamma - (90^\circ - \gamma)$$

Berechnen Sie $f_2(45^\circ)$ und $f_2(77^\circ)$

$$f_2(45^\circ) = 45^\circ - (90^\circ - 45^\circ) = 0^\circ$$

$$f_2(77^\circ) = 77^\circ - (90^\circ - 77^\circ) = 64^\circ$$

3.3 Winkel zu Tiefenwert

Stellen Sie die Gleichung $z = f_3(\alpha, \beta)$ auf, um aus den Winkeln α und β den Tiefenwert z zu berechnen.

$$f_3(\alpha, \beta) = \frac{b}{\tan(\alpha) + \tan(\beta)}$$

Welcher Tiefenwert gehört zu den Winkeln $\alpha = 15^\circ$ und $\beta = 30^\circ$

$$f_3(15^\circ, 30^\circ) = \frac{90cm}{\tan(15^\circ) + \tan(30^\circ)} \approx 105,47cm$$

3.4 Winkel und Z-Koordinate zu X-Koordinate

Stellen Sie die Gleichung $x = f_4(\beta, z)$ auf, um aus dem Winkel β und z die x-Koordinate zu berechnen.

$$f_4(\beta, z) = z * \tan(\beta)$$

Berechnen Sie $f_4(40^\circ, 100cm)$.

$$f_4(40°, 100cm) = 100cm * \tan(40°) \approx 83,9cm$$

3.5 Gesamtgleichung

Stellen Sie nun die Gesamtgleichung $(x,z)^T = f_5(p,\gamma)$ auf, um aus der Pixelposition p und dem Winkel γ die Koordinaten $(x,z)^T$ des abgetasteten Punktes zu berechnen. Kürzen Sie sich aufhebende $\tan$ und $\arctan$ Funktion.

$$f_5(p,\gamma) = (x,z)^T \tag{1}$$

$$x = f_4(\beta, z) = \frac{z}{\tan(90° - \beta)} \tag{2}$$

$$z = f_3(\alpha, \beta) = \frac{b}{\tan(\alpha) + \tan(\beta)} \tag{3}$$

$$\beta = f_1(p) = \arctan(\frac{(p - 511.5) * 48mm}{1024 * 24mm}) \tag{4}$$

$$f_5(p,\gamma) = (f_4(\beta, z), f_3(\alpha, \beta))^T \tag{5}$$

$$f_5(p,\gamma) = (f_4(f_1(p), f_3(\alpha, \beta)) \tag{6}$$

$$x = z * \frac{(p - 511.5) * 48mm}{1024 * 24mm} \tag{7}$$

$$z = \frac{b}{\tan(2\gamma - 90°) + \frac{(p-511.5)*48mm}{1024*24mm}} \tag{8}$$

$$\tag{9}$$

3.6 Beispielaufbau

Zum Spiegelwinkel $\gamma = 67°$ wird ein Laserpunkt im Mittelpunkt von Pixel 377 registriert. Welche Koordinaten hat der abgetastete Punkt mit oben beschriebenem Aufbau?

$$f_5(377, 67°) = (x, z)^T \tag{10}$$

$$z = \frac{b}{\tan(2\gamma - 90°) + \frac{(p-511.5)*48mm}{1024*24mm}} \tag{11}$$

$$z = \frac{90cm}{\tan(2 * 67° - 90°) + \frac{(377-511.5)*48mm}{1024*24mm}} \tag{12}$$

$$z = \frac{90cm}{\tan(44°) + \frac{-6456}{24576}} \tag{13}$$

$$z = \frac{90cm}{0.7029934623} \approx 128,02cm \tag{14}$$

$$x = z * \frac{(p - 511.5) * 48mm}{1024 * 24mm} \tag{15}$$

$$x = 128,02cm * \frac{(377 - 511.5) * 48mm}{1024 * 24mm} \tag{16}$$

$$x = 128,02cm * \frac{-6456}{24576} \approx -33,63cm \tag{17}$$

$$\Rightarrow (-33,63cm, 128,02cm) \tag{18}$$

Theorieübung 2

Graphische Datenverarbeitung I
WiSe 2024/25
Team 01

1 Transformationen

Berechnen Sie die vollständigen 4x4 Transformationsmatrizen $M^A(\alpha)$ und $M^B(\alpha)$ für die Zahnräder A und B in Abhängigkeit von α.

Die Rotation wird direkt auf das Zahnrad A angewendet. Aufgrunddessen, dass die Zahnräder nicht im Ursprung des Koordinatensystems sind, müssen diese nicht nur rotiert werden, sondern auch transliert. Jedes Zahnrad wird wie folgt Transformiert:

1. Zahnrad wird in den Ursprung transformiert
2. Zahnrad wird rotiert
3. Zahnrad wird zurück transformiert

Daraus folgt folgendes:

$$M(\alpha) = T * R(\alpha) * T^{-1}$$

T : Transformierung in den Ursprung
$R(\alpha)$: Rotationsmatrix
T^{-1} : Transformierung zurück in die ursprüngliche Position

1.1 Bestimmung von T und T^{-1} für A und B

Zahnrad A

$$T_A = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix}; T_A^{-1} = \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Zahnrad B

$$T_B = \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix}; T_B^{-1} = \begin{pmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.2 Bestimmung der Rotationsmatrix $R(\alpha)$ für A und B

Die Rotationsmatrix für eine Rotation um die z-Achse ist wie folgt:

$$R(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Diese entspricht der Rotationsmatrix für Zahnrad A. Für Zahnrad B müssen zwei Anpassungen durchgeführt werden.

- Rotationsrichtung umdrehen: Aufgrunddessen, dass Zahnrad B durch die Rotation von Zahnrad A beeinflusst wird, ändert sich die Rotationsrichtung. Wenn Zahnrad A sich um den Uhrzeigersinn rotiert, rotiert Zahnrad B sich gegen den Uhrzeigersinn. Somit ist der gegebene Rotationswinkel für B negativ.
- Zahnradverhältnis: Die Rotation auf Zahnrad A äußert sich nicht mit einer 1:1 Rotation auf Zahnrad B aus. Zahnrad A hat 20 Zähne und Zahnrad B hat 50. Somit rotiert sich Zahnrad B um $-\frac{20}{50}\alpha$, wenn sich Zahnrad A um α rotiert.

Somit ist die Rotationsmatrix für Zahnrad B diese:

$$R_B(\alpha) = \begin{pmatrix} \cos(-0.4\alpha) & -\sin(-0.4\alpha) & 0 & 0 \\ \sin(-0.4\alpha) & \cos(-0.4\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aufgrund dessen, dass $\cos(\alpha) = \cos(-\alpha)$ und $\sin(\alpha) = -\sin(-\alpha)$ gelten, wird bei der Rechnung später der Winkel im cos positiv bleiben und die sin Funktion ebenfalls entsprechend angepasst bei $M^B(\alpha)$.

1.3 Zusammensetzung in $M^A(\alpha)$ und $M^B(\alpha)$

$$M^A(\alpha) = T_A \cdot R_A(\alpha) \cdot T_A^{-1}$$

$$M^A(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M^A(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & 2 \\ \sin(\alpha) & \cos(\alpha) & 0 & 7 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M^A(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & -2 \cdot \cos(\alpha) + 7 \cdot \sin(\alpha) + 2 \\ \sin(\alpha) & \cos(\alpha) & 0 & -7 \cdot \cos(\alpha) - 2 \cdot \sin(\alpha) + 7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M^B(\alpha) = T_B \cdot R_B(\alpha) \cdot T_B^{-1}$$

$$M^B(\alpha) = \begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 7 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos(0.4\alpha) & \sin(0.4\alpha) & 0 & 0 \\ -\sin(0.4\alpha) & \cos(0.4\alpha) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M^B(\alpha) = \begin{pmatrix} \cos(0.4\alpha) & \sin(0.4\alpha) & 0 & 5 \\ -\sin(0.4\alpha) & \cos(0.4\alpha) & 0 & 7 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M^B(\alpha) = \begin{pmatrix} \cos(0.4\alpha) & \sin(0.4\alpha) & 0 & -5\cos(0.4\alpha) - 7\sin(0.4\alpha) + 5 \\ -\sin(0.4\alpha) & \cos(0.4\alpha) & 0 & -7\cos(0.4\alpha) + 5\sin(0.4\alpha) + 7 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2 Aufgabe 2: Baryzentrische Koordinaten

2.1 Berechnung am Tetraeder

$$\det M = \det \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \det \begin{pmatrix} 0 & -2 & 0 & -1 \\ 0 & 1 & 5 & 2 \\ 0 & 0 & 1 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 37$$

Nun wird jede Spalte der Matrix jeweils mit dem Punkt $p = (-1, 2, 2)^T$ ersetzt und erneut die Determinante bestimmt.

$$\det M_0 = \det \begin{pmatrix} -1 & -2 & 0 & -1 \\ 2 & 1 & 5 & 2 \\ 2 & 0 & 1 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 4$$

$$\det M_1 = \det \begin{pmatrix} 0 & -1 & 0 & -1 \\ 0 & 2 & 5 & 2 \\ 0 & 2 & 1 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 10$$

$$\det M_2 = \det \begin{pmatrix} 0 & -2 & -1 & -1 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 2 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 6$$

$$\det M_2 = \det \begin{pmatrix} 0 & -2 & 0 & -1 \\ 0 & 1 & 5 & 2 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 17$$

Aus diesen Determinanten und der Determinante von M, folgen diese baryzentrischen Koordinaten:

$$\lambda_0 = \frac{\det M_0}{\det M} = \frac{4}{37} = 0,\overline{108} \approx 0.11$$

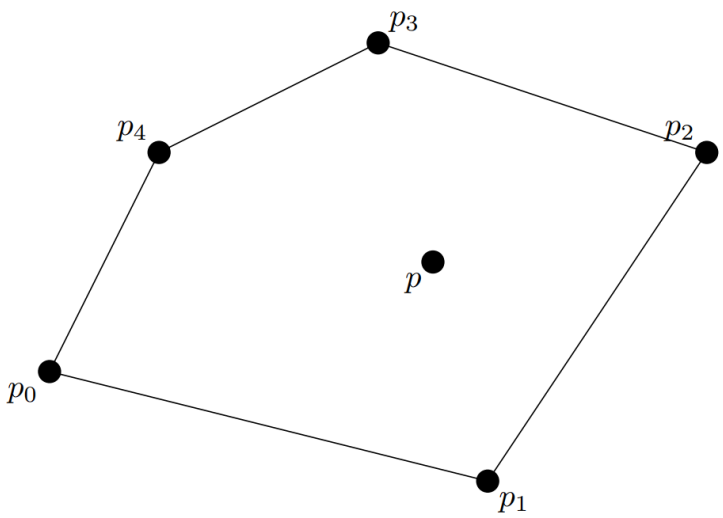
$$\lambda_1 = \frac{\det M_1}{\det M} = \frac{10}{37} = 0,\overline{270} \approx 0.27$$

$$\lambda_2 = \frac{\det M_2}{\det M} = \frac{6}{37} = 0,\overline{162} \approx 0.16$$

$$\lambda_3 = \frac{\det M_3}{\det M} = \frac{17}{37} = 0,\overline{459} \approx 0.46$$

2.2 Verallgemeinerte baryzentrische Koordinaten

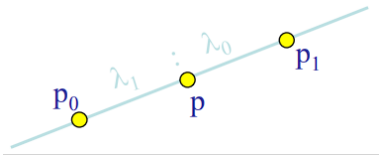
Gegeben Sei ein Pentagon



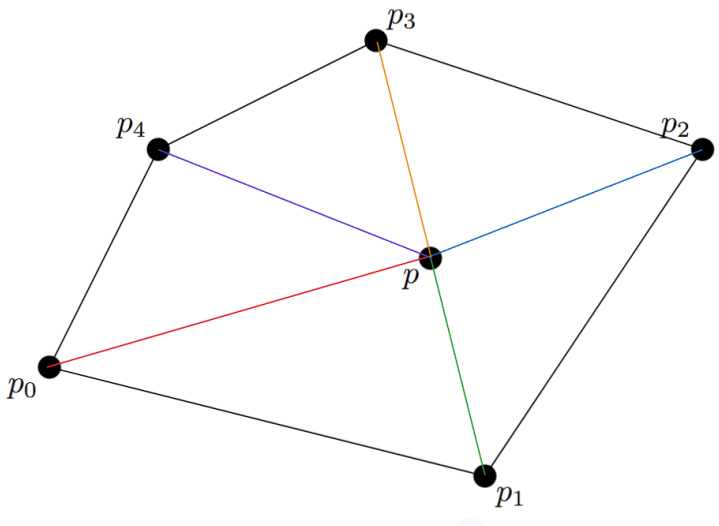
mit $p_0 = (0,1)^T, p_1 = (4,0)^T, p_2 = (6,3)^T, p_3 = (3,4)^T, p_4 = (1,3)^T$**.**
Berechnen sie zwei Tupel $(\lambda_0, \lambda_1, \lambda_2, \lambda_3)$ **der verallgemeinerte baryzentrische Koordinaten für den Punkt** $p = (3.5, 2)^T$**, wobei gelten soll:** $\lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$**.**
Für das erste Tupel soll gelten: $\lambda_i \geq 0, i = 0,1,2,3,4$**. Es ist also erlaubt dass ein oder mehrere Lambdas gleich null sind.**
Für das zweite Tupel soll die strengere Annahme $\lambda_i > 0, i = 0,1,2,3,4$ **gelten.**

2.2.1 Tupel mit $\lambda_i \geq 0, i = 0,1,2,3,4$

Rückblick Vorlesung:



Wenn p also auf der Verbindungslinie zweischen zwei Eckpunkten des Pentagons liegt, würde die Berechnung sehr vereinfacht werden. Verbindet man eine Linie von jedem Eckpunkt zu p erkennt, man, dass p in der Verbindungslinie zwischen p_3 und p_1 liegt.



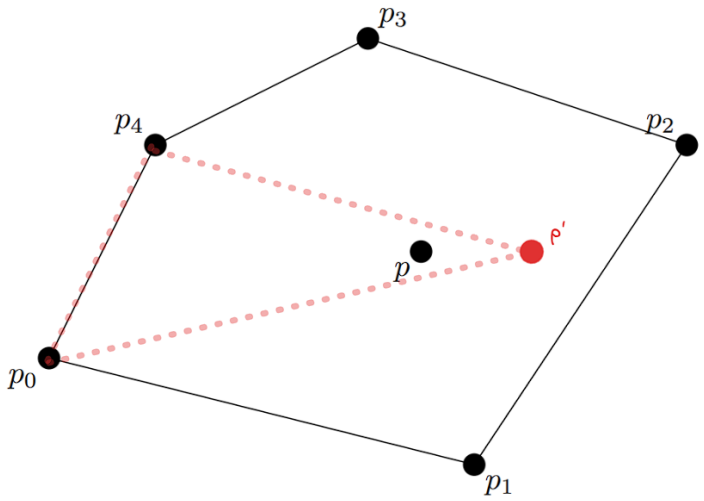
Also könnte man λ_0, λ_4 und λ_2 auf 0 setzen, somit erhält man:

$$
\begin{aligned}
p &= 0 \cdot p_0 + \lambda_1 \cdot p_1 + 0 \cdot p_2 + \lambda_3 \cdot p_3 + 0 \cdot p_4 \\
p &= \lambda_1 \cdot p_1 + \lambda_3 \cdot p_3 \\
\begin{pmatrix} 3.5 \\ 2 \end{pmatrix} &= \lambda_1 \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} + \lambda_3 \cdot \begin{pmatrix} 3 \\ 4 \end{pmatrix} \\
\lambda_1 &= \frac{1}{2} \\
\lambda_3 &= \frac{1}{2}
\end{aligned}
$$

Somit erhält man für den ersten Tupel: $(0, \frac{1}{2}, 0, \frac{1}{2}, 0)$

2.2.2 Tupel mit $\lambda_i > 0, i = 0, 1, 2, 3, 4$

Damit gewährleistet wird, dass alle baryzentrischen Koordinaten größer als 0 sind, muss sichergestellt werden, dass der Punkt p nicht entlang der Kante eines Dreiecks verläuft. Denn sobald der Punkt p als Eckpunkt eines Dreiecks fungiert, ist mind. eine baryzentrische Koordinate 0. Aufgrund dessen muss ein Dreieck konstruiert werden, welches dafür sorgt, dass p innerhalb dieses Dreiecks ist. Hierfür wird ein zusätzlicher Punkt p' erzeugt. Für p' wird die Koordinate $(5, 2)^T$ gewählt.



Um mit p', p zu bestimmen, müssen die baryzentrischen Koordinaten von p' bestimmt werden. Hierfür wird das Dreieck genutzt, welches durch p_1, p_2 und p_3 dargestellt wird. [Die Zahlen im Exponenten sollen nicht Quadratzahlen oder so darstellen. Es dient nur zur Nummerierung]

$$\det M_{p_1p_2p_3} = \det \begin{pmatrix} 4 & 6 & 3 \\ 0 & 3 & 4 \\ 1 & 1 & 1 \end{pmatrix} = 11$$

$$\det M^0_{p_1p_2p_3} = \det \begin{pmatrix} 5 & 6 & 3 \\ 2 & 3 & 4 \\ 1 & 1 & 1 \end{pmatrix} = 4 \Rightarrow \lambda_0 = \frac{4}{11}$$

$$\det M^1_{p_1p_2p_3} = \det \begin{pmatrix} 4 & 5 & 3 \\ 0 & 2 & 4 \\ 1 & 1 & 1 \end{pmatrix} = 6 \Rightarrow \lambda_1 = \frac{6}{11}$$

$$\det M^2_{p_1p_2p_3} = \det \begin{pmatrix} 4 & 6 & 5 \\ 0 & 3 & 2 \\ 1 & 1 & 1 \end{pmatrix} = 1 \Rightarrow \lambda_2 = \frac{1}{11}$$

Nun werden die Baryzentrischen Koordinaten für das Dreieck $p_0, p', p4$ berechnet.

$$\det M_{p_0p'p_4} = \det \begin{pmatrix} 0 & 5 & 1 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} = 9$$

$$\det M^0_{p_0p'p_4} = \det \begin{pmatrix} 3.5 & 5 & 1 \\ 2 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} = 1.5 \Rightarrow = \frac{1}{6}$$

$$\det M^1_{p_0 p' p_4} = \det \begin{pmatrix} 0 & 3.5 & 1 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} = 6 \Rightarrow = \frac{2}{3}$$

$$\det M^2_{p_0 p' p_4} = \det \begin{pmatrix} 0 & 5 & 3.5 \\ 1 & 2 & 2 \\ 1 & 1 & 1 \end{pmatrix} = 1.5 \Rightarrow = \frac{1}{6}$$

$$\begin{aligned} \lambda_0 &= \frac{1}{6} \\ \lambda_1 &= \frac{2}{3} * \frac{4}{11} = \frac{8}{33} \\ \lambda_2 &= \frac{2}{3} * \frac{6}{11} = \frac{4}{11} \\ \lambda_3 &= \frac{2}{3} * \frac{1}{11} = \frac{2}{33} \\ \lambda_4 &= \frac{1}{6} \end{aligned}$$

Nun um zu kontrollieren, ob diese baryzentrischen Koordinaten stimmen, muss die Summe all dieser 1 ergeben.

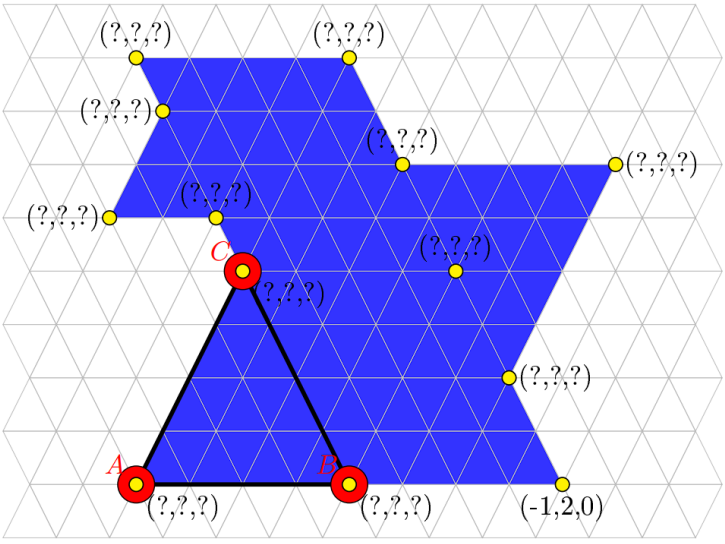
$$\frac{1}{6} + \frac{8}{33} + \frac{4}{11} + \frac{2}{33} + \frac{1}{6} = \frac{11}{66} + \frac{16}{66} + \frac{24}{66} + \frac{4}{66} + \frac{11}{66} = \frac{2 * 11 + 16 + 24 + 4}{66} = \frac{66}{66} = 1$$

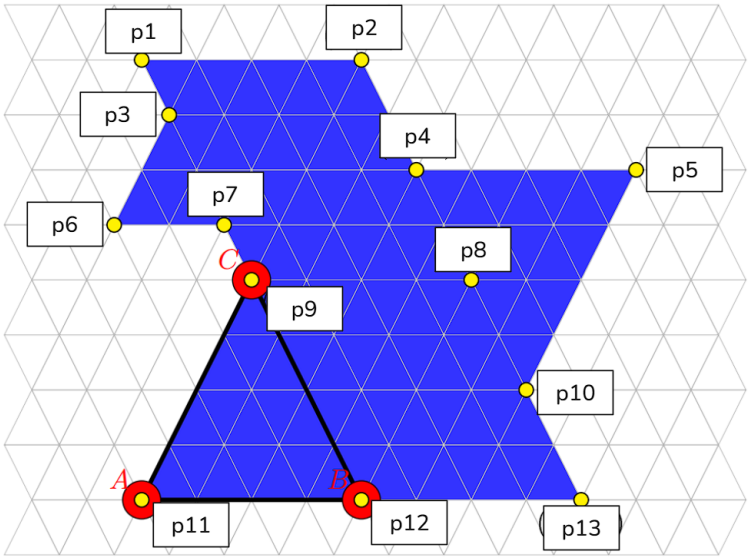
Um sicherzustellen, dass mit diesen baryzentrischen Koordinaten der Punkt p repräsentiert werden kann, wird folgende Proberechnung durchgefüht:

$$p = \lambda_0 * p_0 + \lambda_1 * p_1 + \lambda_2 * p_2 + \lambda_3 * p_3 + \lambda_4 * p_4$$
$$\begin{pmatrix} 3.5 \\ 2 \end{pmatrix} = \frac{1}{6} * \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \frac{8}{33} * \begin{pmatrix} 4 \\ 0 \end{pmatrix} + \frac{4}{11} * \begin{pmatrix} 6 \\ 3 \end{pmatrix} + \frac{2}{33} * \begin{pmatrix} 3 \\ 4 \end{pmatrix} + \frac{1}{6} * \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

2.3 Baryzentrische Koordinaten in 2D

Tragen Sie in den folgenden Zeichnungen für alle gelb markierten Punkte die baryzentrischen Koordinaten bezogen auf die jeweiligen Punkte A, B und $C \in \mathbb{R}^2$ ein. Für eiunen Punkt p mit den baryzentrischen Koordinaten $(\lambda_0, \lambda_1, \lambda_2)$ soll also $p = \lambda_0 * A + \lambda_1 * B + \lambda * C$ gelten. Als Hilfestellung ist das Ergebnis für einen gelb markierten Punkt bereits eingetragen. Ersetzen Sie bei allen gelb markierten Punkte die Tripel (?, ?, ?) mit den baryzentrischen Koordinaten $(\lambda_0, \lambda_1, \lambda_2)$





$$
\begin{aligned}
p_1 &= (0, -1, 2) \\
p_2 &= (-1, 0, 2) \\
p_3 &= (0, -0.75, 1.75) \\
p_4 &= (-1, 0.5, 1.5) \\
p_5 &= (-2, 1.5, 1.5) \\
p_6 &= (0.5, -0.75, 1.25) \\
p_7 &= (0, -0.25, 1.25) \\
p_8 &= (-1, 1, 1) \\
p_9 &= (0, 0, 1) \\
p_{10} &= (-1, 1.5, 0.5) \\
p_{11} &= (1, 0, 0) \\
p_{12} &= (0, 1, 0) \\
p_{13} &= (-1, 2, 0)
\end{aligned}
$$

3 Räumliche Datenstrukturen

3.1 Datenstruktur Updates

Für Hüllkörperhierarchien, reguläre Gitter und Octrees seien folgende naiven Ansätze vorgegeben, welche beschreiben, welche Anpassungen vorgenommen werden, sobald ein Objekt im Raum bewegt wird (d.h. seine Position verändert).

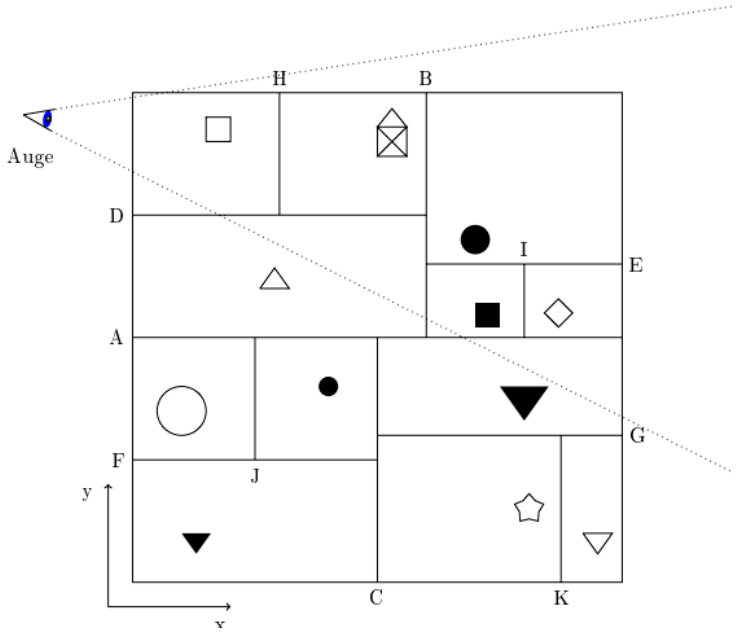
- **Hüllkörperhierarchien: Das Bounding Volume (BV) des Objekts selbst wird für die neue Position neu berechnet und ersetzt das alte BV in der Hierarchie. Die in der Hierarchie übergeordneten BVs werden entsprechend vergrößert bzw. verkleinert, sodass das neue BV des verschobenen Objekts vollständig enthalten ist.**
- **Reguläre Gitter: Das Objekt wird aus allen Zellen entfernt, in denen es nicht mehr enthalten ist, und wird zu allen Zellen hinzugefügt, in denen es neu enthalten ist.**
- **Octrees: Genauso wie bei Regulären Gittern**

Erklären Sie für alle drei Datenstrukturen ob und warum diese Änderungen für dynamische Szenen ausreichend sind. Überlegen Sie sich insbesondere inwiefern der Zweck der Datenstruktur (die beschleunigte Suche von Objekten im Raum) weiter erfüllt wird.

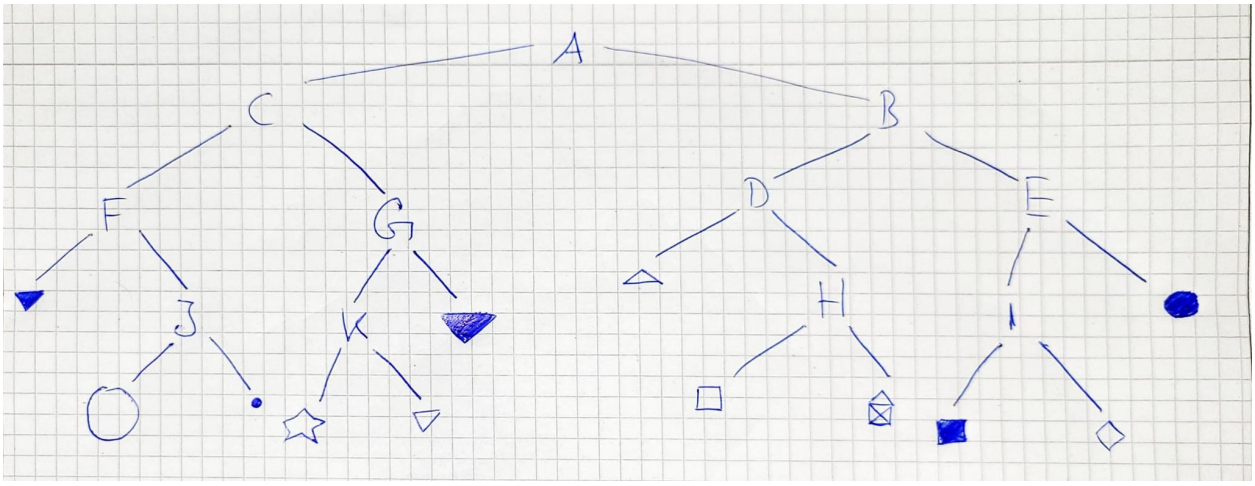
- Hüllkörperhierarchien: Die Neuberechnung der Bounding Volumes kann dafür sorgen, dass diese an verschiedenen Positionen im Raum liegen als vor der Objektbewegung. Hierbei werden Bounding Volumes die als "Parent-BV" fungierten unterschiedlich größer/kleiner. Die Aufteilung des Raums durch die Hüllkörperhierarchie wird verändert und es wird nicht die selbe Hierarchie widergespiegelt wie zuvor, also muss diese definitiv angepasst werden. Aufgrund der dauerhaften neu-berechnung stellt sich diese Methode in diesem Kontext sehr ineffizient dar.
- Reguläre Gitter: Die regulären Gitter können weiterhin verwendet werden. Die Datenstruktur die sich hieraus ergibt ist nach den Veränderungen ebenfalls repräsentativ der Szene, denn das entfernen und neu-hinzufügen der Objekte entspricht dem Resultat, wenn Objekte verschoben werden, nur etwas umständlicher.
- Octrees: Das Verändern der Objektpositionen im Octree kann dafür sorgen, dass eine Zelle die im Octree zuvor nur ein Objekt enthalten hatte nun mehrere Objekte hat. Die Zellen werden bei der Initialisierung des Octrees aufgeteilt, und mehrere Objekte in Zellen zu haben, wo zuvor nur ein Objekt war, kann die Suchgeschwindigkeit sehr verlangsamen. Aufgrunddessen muss die Raumaufteilung angepasst werden.

3.2 KD-Tree

Gegeben ist eine 2D-Szene mit Objekten, wobei jedes Symbol ein Objekt repräsentiert. Weiterhin gegeben ist eine KD-Tree-Zerlegung dieser Szene. Die Buchstaben kennzeichnen die Geraden, die die Halbebenen voneinander abtrennen.



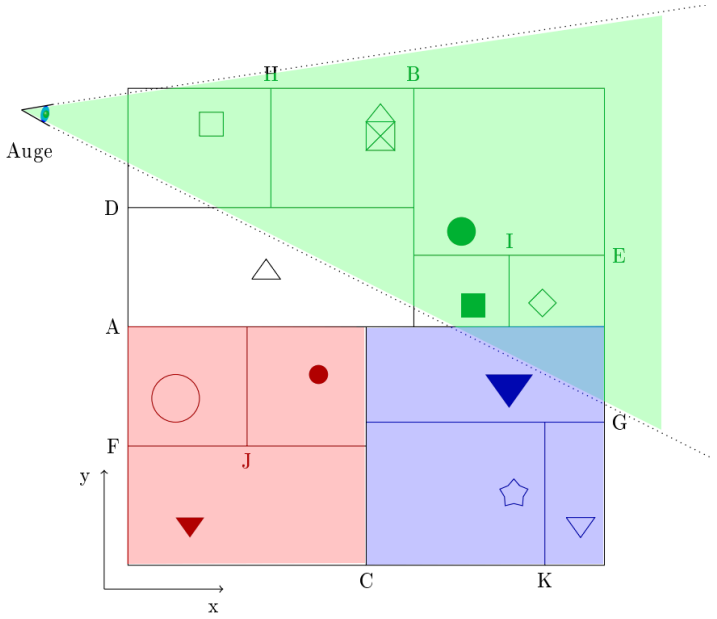
Zeichnen sie den dazugehörigen Baum dieser Szene. Geheh sie dabei so vor, dass der linke Teilbaum den Raum mit kleineren Koordinaten enthält, der recht Teilbaum den Raum mit größeren Koordinaten. In den Blättern soll jeweils ein Objekt enthalten sein.



3.3 Culling

welche Teilbäume des KD-Trees können für das in b) eingezeichnete View Frustum verworfen werden?

Das View Frustum des Teilbaums mit Wurzel F (roter Bereich in Abbildung) und des Teilbaums mit Wurzel K (unterer Teil des Blauen Bereichs in Abbildung) können verworfen werden.



Es kann jedoch nicht der Gesamte Teilbaum mit Wurzel G entfernt werden, aufgrunddessen dass das View Frustum durch das Bounding Volume vom großen umgedrehten schwarzen Dreieck geht. Aber der Rest sollte verworfen werden können.

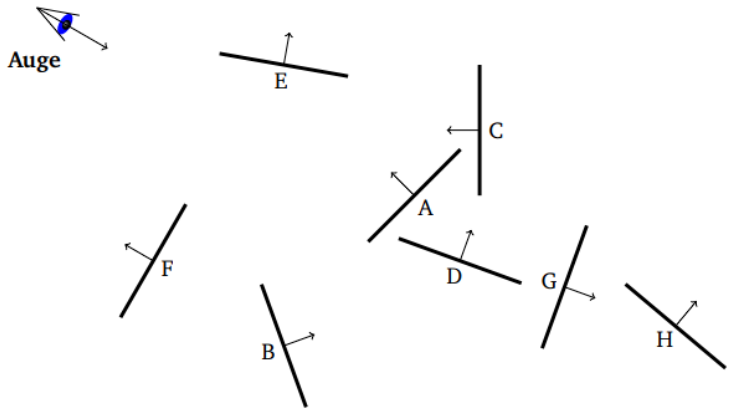
Theorieübung 3

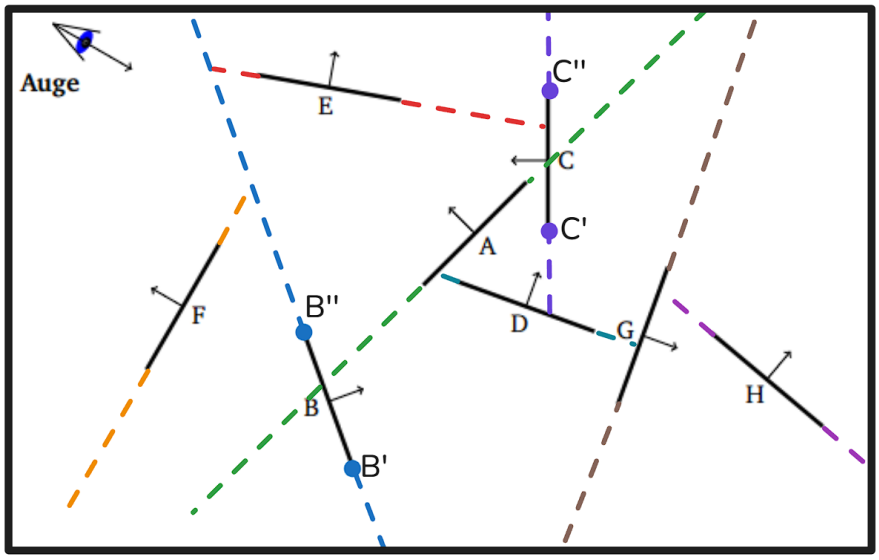
Graphische Datenverarbeitung I
WiSe 2024/25
Team 01

1 Räumliche Datenstrukturen

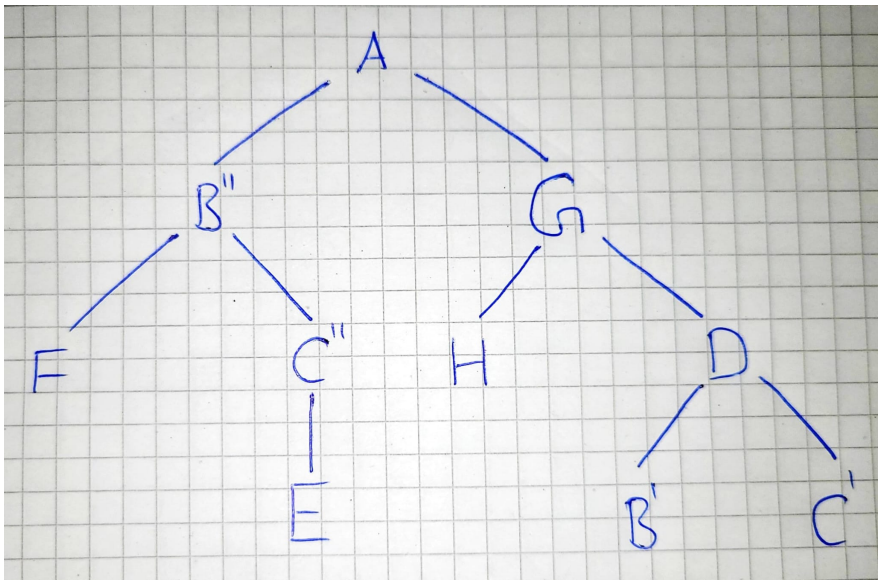
Im folgenden ist eine Szene dargestellt, in der die Liniensegmente Objekte darstellen, die den Raum jeweils in Halbräume unterteilen. Die Pfeile auf den Liniensegmenten stellen die Normalen dar. Sie unterteilen den Raum in Vorne (in Normalenrichtung) und Hinten. Erstellen Sie den BSP-Baum dieser Szene. Beachten sie dabei folgendes:

- **Wählen Sie Liniensegment A als Wurzel des Baumes**
- **Wählen Sie eine sinnvolle Reihenfolge für die weiteren Unterteilungen. Konkret soll folgendes erfüllt werden:**
  - **Außer der anfänglichen Unterteilung von B und C durch A dürfen keine weiteren Unterteilung von Liniensegmenten stattfinden.**
  - **Die Raumunterteilungen durch die Liniensegmente lassen jeweils zwei neue Halbräume entstehen. Die Größe der zwei Halbräume sollte bei jeder Unterteilung so ähnlich wie möglich sein und die Anzahl der verbleibenden Liniensegmente in der beiden entstehenden Halbräumen sollte ebenfalls so ähnlich wie möglich sein.**





Traversieren Sie den BSP-Baum aus Aufgabenteil a) gemäß des eingezeichneten Augpunktes und geben Sie die resultierende Zeichenreihenfolge an. Die Traversierung erfolgt hier entsprechend der Zeichenreihenfolge in back-to-front Reihenfolge.

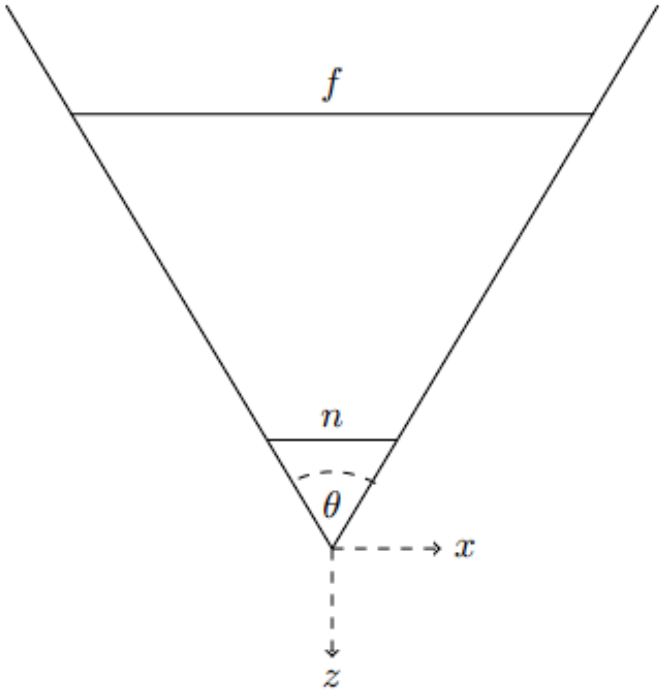


2 Projektionen

2.1 View Frustum

Bei der Projektion wird das 3D View Frustum immer in das kanonische Sichtvolumen transformiert. In dieser Aufgabe wollen wir diese Transformation nachvollziehen. Wir

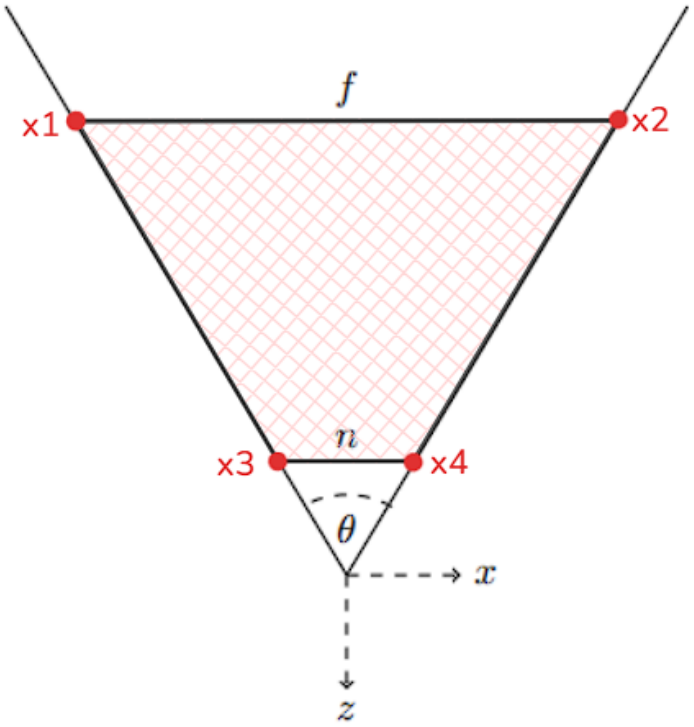
betrachten hier den 2D-Fall, da dieser auf 3D übertragbar ist.
Wir betrachten also ein 2D View Frustum in der $x, z-$Ebene. Der $z-$Wert der near-plane sei n, der $z-$Wert der far-plane sei f. Der Öffnungswinkel des Frustums sei Θ.



1. **Berechnen Sie die 4 Eckpunkte des Vierecks, welches den sichtbaren Bereich darstellt.**
2. **Führen Sie anschließend die perspektivische Transformation dieser 4 Punkte durch. Das bedeutet, Sie müssen die 4 Eckpunkte mit der perspektivischen Transformationsmatrix R multiplizieren.**
3. **Vollziehen Sie schließlich die Transformation des Sichtbarkeitsbereiches in den Einheitswürfel. Das bedeutet, Sie müssen die 4 perspektivisch transformierten Eckpunkte des sichtbaren Rechtecks mit der Projektionsmatrix P_0 multiplizieren.**

---

1. Das Viereck, welches gemeint ist, ist der rot-gekennzeichnete Bereich in der folgenden Abbildungen mit den Eckpunkten x_1, x_2, x_3 und x_4.



Die $z-$Koordinate ist für die Punkte x_3 und x_4 (near-plane) n. Für x_1 und x_2 (far-plane) ist sie f. Sei $\overrightarrow{x_3x_4}$ die Strecke zwischen den zwei Punkten x_3 und x_4, somit ist die Hälfte dieser Strecke die x-Koordinate für x_4 und als negative Zahl die x-Koordinate für x_3.

$$\tan(\frac{\Theta}{2}) = \frac{\overrightarrow{x_3x_4} * 0.5}{n}| * n$$

$$\tan(\frac{\Theta}{2}) * n = \overrightarrow{x_3x_4} * 0.5$$

Somit ist die x-Koordinate von x_4 $\tan(\frac{\Theta}{2}) * n$. Dementsprechend werden die Werte für x_1, x_2 und x_3 bestimmt.

$$x_1 = \begin{pmatrix} -f * \tan(\frac{\theta}{2}) \\ f \end{pmatrix}$$
$$x_2 = \begin{pmatrix} f * \tan(\frac{\theta}{2}) \\ f \end{pmatrix}$$
$$x_3 = \begin{pmatrix} -n * \tan(\frac{\theta}{2}) \\ n \end{pmatrix}$$
$$x_4 = \begin{pmatrix} n * \tan(\frac{\theta}{2}) \\ n \end{pmatrix}$$

2. Perspektivische Projektionsmatrix R

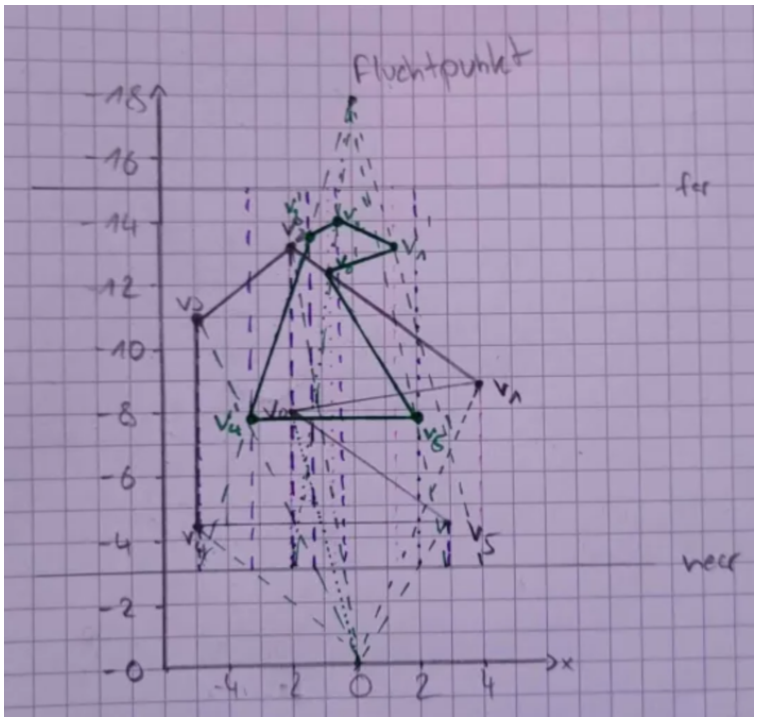
$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+\frac{f}{n} & -f \\ 0 & \frac{1}{n} & 0 \end{pmatrix}$$

2.2 Geometrische Projektion

Gegeben seien die Eckpunktkoordinaten eines Objektes:

$$v_0 = \begin{pmatrix} -2 \\ -8 \end{pmatrix}, v_1 = \begin{pmatrix} 4 \\ -8.73 \end{pmatrix}, v_2 = \begin{pmatrix} -2 \\ -13.2 \end{pmatrix} v_3 = \begin{pmatrix} -5 \\ -11 \end{pmatrix}, v_4 = \begin{pmatrix} -5 \\ -4.54 \end{pmatrix}, v_5 = \begin{pmatrix} 3 \\ -4.54 \end{pmatrix}$$

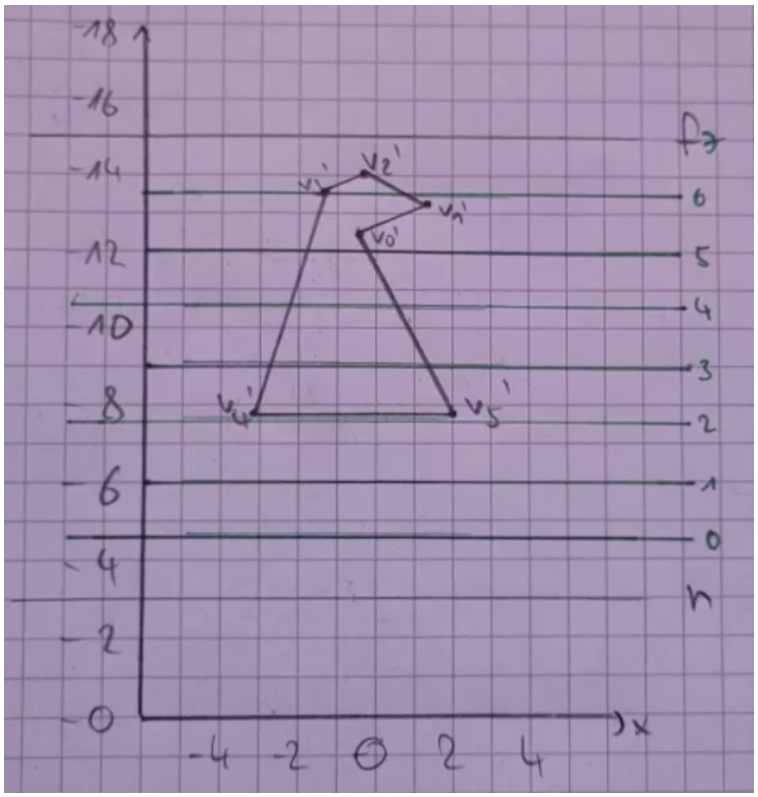
Der Augpunkt liegt im Ursprung und die Blickrichtung ist entlang der negativen z-Achse. Die near-plane liegt bei $z = -3$ und die far-plane bei $z = -15$. Fertigen Sie eine 2D-Skizze dieser Szene an und zeichnen Sie den Fluchtpunkt ein. Ermitteln Sie zeichnerisch das perspektivisch transformierte Polygon



2.3 Tiefenpuffer 1

Gegeben sei ein Tiefenpuffer mit einer 3-Bit-Auflösung (8 Tiefenwerte 0,...,7). Die Tiefenwerte werden gleichmäßig über das perspektivisch transformierte Sichtvolumen verteilt. Entnehmen Sie aus der Zeichnung von Aufgabenteil b) die Tiefenwerte der

Eckpunkte des transformierten Objektes. Fertigen Sie eine neue Skizze mit dem transformierten Objekt an und zeichnen Sie die Grenzen der einzelnen Tiefenwerte im Sichtvolumen ein.



$$Tiefenwert(v_0') = 6, Tiefenwert(v_1') = 6, Tiefenwert(v_2') = 7$$
$$Tiefenwert(v_3') = 7, Tiefenwert(v_4') = 3, Tiefenwert(v_5') = 3$$

2.4 Tiefenpuffer 2

Bei der perspektivischen Transformation von Aufgabenteil c) werden Punkte mit einem z-Wert $\in [n, f]$ auf Punkte mit einem z'-Wert $\in [n, f]$ abgebildet. Allerdings tendenziell näher zur far-plane. Stellen Sie die Gleichung $z' = f(z)$ auf und lösen Sie anschließend die Gleichung nach z auf, also bestimmen Sie $z = g(z')$.

Jeder Tiefenwert 0, ..., 7 umfasst einen gewissen Teilraum. Bestimmen Sie den Teilraum $z \in [a, b]$, der nach der perspektivischen Transformation den Tiefenwert 1 zugewiesen bekommt. Bestimmen Sie ebenfalls den Teilraum, der nach der perspektivischen Transformation den Tiefenwert 6 bekommt.

Bemerkung: Um alle z-Werte in [n, f] eindeutig einem Tieferwert zuordnen zu können, müssten die Intervalle eigentlich teilweise halboffen [a, b) oder (a, b] sein. Hier müssen die Intervalle nur als abgeschlossene Intervalle [a, b] angegeben werden.

Near und Far plane Werte in die Projektionsmatrix einsetzen:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+\frac{f}{n} & -f \\ 0 & \frac{1}{n} & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+\frac{-15}{-3} & 15 \\ 0 & \frac{1}{-3} & 0 \end{pmatrix}$$

Gleichung erhält man durch z-Komponente der Matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1+\frac{-15}{-3} & 15 \\ 0 & \frac{1}{-3} & 0 \end{pmatrix} * \begin{pmatrix} x \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ \frac{-18}{-3}z + 15 \\ -\frac{1}{3}z \end{pmatrix}$$

Komponenten multiplizieren mit -3

$$\begin{pmatrix} -3x \\ -18z - 45 \\ z \end{pmatrix}$$

Komponenten durch z teilen

$$\begin{pmatrix} \frac{-3x}{z} \\ -18 - \frac{45}{z} \\ 1 \end{pmatrix}$$

$$z' = f(z) = -18 - \frac{45}{z}$$

$$z = g(z') = -\frac{45}{z' + 18}$$

Tiefenwert 1 ist von -4.5 bis -6:

$$g(-4.5) = -\frac{45}{-4.5 + 18} = -3.33$$

$$g(-6) = -\frac{45}{-6 + 18} = -3.75$$

$$z \in [-3.33, -3.75]$$

Tiefenwert 6 ist von -12 bis -13.5:

$$g(-12) = -\frac{45}{-12 + 18} = -7.5$$

$$g(-13.5) = -\frac{45}{-13.5 + 18} = -10$$

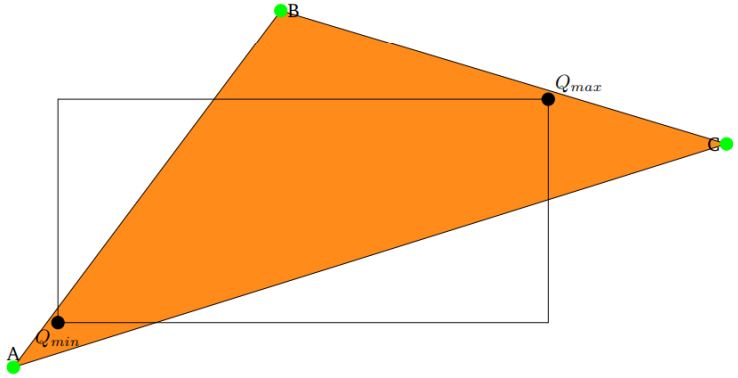
$$z \in [-7.5, -10]$$

3 Clipping

Clippen Sie das Dreieck $A = (0,0)^T, B = (6,8)^T, C = (16,5)^T$ am Fenster $Q_{\min} = (1,1)^T$ und $Q_{\max=(12,6)^T}$ mit SHA. Hierfür wird in jedem Clipping-Schritt CSA angewendet, wodurch neue Schnittpunkte mit dem Liang-Barsky-Algorithmus berechnet werden.

Halten Sie sich dabei an folgende Konventionen:

- **Die 4 bit der CSA-Codierung sind in der Reihenfolge Oben—Unten—Rechts—Links angeordnet.**
- **Die Reihenfolge der Clipping-Schritte des SHA soll wie folgt durchgeführt werden: Rechts, Oben, Unten, Links.**



Theorieübung 4

Graphische Datenverarbeitung I
WiSe 2024/25
Team 01

1 Raytracing

1.1 Basic

At least one ray per pixel is cast from the "camera" into the 3D scene to determine the color of that pixel based on the objects and light sources in the scene. Ray tracing is used to simulate the way the light interacts with different kinds of surfaces realistically. The first basic step is ***ray casting*** in which for each pixel one ray is generated originating from the camera and passing through the corresponding pixel on the image plane. For this ray we then ***test for intersection*** points with objects in the scene. We evaluate further based on the object that is hit first by each ray. The ray extracts color information from the surface it hits as well as sometimes lighting-information stemming from ***global illumination*** which simulates indirect, diffuse light stemming from light bouncing between surfaces (radiosity).

Furthermore, when the ray intersects a surface, additional rays are generated recursively (the depth of this recursion determines the quality of the ray-tracing). This is done to determine the color value of the pixel corresponding to the ray by factoring in light from various sources using a shading model (like the ***Phong*** model). One ray simulates lighting information corresponding to a ***reflective*** surface. One ray per light source is cast from the intersection point, to determine wether a point is in shadow (also called shadow ray sometimes). Additionally the Halfway-vector between the light and the viewing-direction and the surface normal are calculated to enable simulating ***dispersed*** light (diffuse and specular light). Additionally, if the surface should be ***transparent***, another ray would be generated to simulate light passing through the surface with some set refractive index.

All this accumulated data (color information of the surface itself, lighting by reflection, dispersion, refraction and global illumination, etc.) is then used to compute the final color value of the displayed pixel.

Implementation

For calculationg the rays mathematically we use the following formula:

$$I_p = I_a k_a + \sum_{i=1}^{n} S_{i,p} I_i (k_d f_d + k_s f_s) + k_r I_r$$

Since there is only one light-source is in this scene, we can shorten it to:

$$I_p = I_a k_a + S_p I (k_d f_d + k_s f_s) + k_r I_r$$

Where I_r is the intensity of the point where the reflected ray hits another surface. Thereby I_r is calculated just like I_p just using the respective parameters for the reflected ray on the surface the

reflected ray hits. $I_{p \text{ without reflection}}$ can be calculated by subtracting $k_r I_r$ from I_p (or by just not adding it in the first place. Since we use a recursive depth of 2, we calculate one initial ray and the first reflection of this ray. Furthermore we know: $f_d = L_i \cdot N_p$ and $f_s = (H_i \cdot N_p)^{k_e}$. Let's first compute the necessary terms for the first camera ray. After normalizing the initial vectors from the visualization on the task-sheet, we get: $V = (\frac{-1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})^T$, $N = L = (-1, 0)^T$, $H = \frac{L+V}{||L+V||_2} = (-0.92388, -0.38268)^T$ Since the lightsource is not covered we get $S_p = 1$. Furthermore by inserting the vectors we get: $f_d = 1$ and $f_s = (0.92388)^{15} = 0.305$. Since the recursion depth is 2, we cast one more ray simulating the reflection of the viewing ray V, which we get using the following formula: $R(V) = -V + 2N(V \cdot N)$. After inserting we get $R(V) = (\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^T$. We also need V, N, L and H for the reflection, let's call them V^r, N^r, L^r, H^r. Since the viewing-vector is always from the surface that is viewed to the observer we get it by negating the incoming reflection-ray: $V^r = -R(V) = (\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}})^T$. Computing the normal is not possible without knowing the exact radius of the circle C1 (as is computing whether the ray intersects with the circle), so we used the visualization and drew a line from the center of the circle to the center of the light source. since this line crosses the point where the incoming reflection-ray hits the surface of C1, the light-vector also is the normal-vector for this point, namely (after normalizing): $L^r = N^r = (\frac{-1}{\sqrt{17}}, \frac{-4}{\sqrt{17}})^T$. Using the same formula as before we get $H^r = \frac{(0.46457, -1.67725)^T}{1.740399} = (0.2669, -0.9637)^T$. With those vectors we can finally compute $f_d^r = L^r \cdot N^R = 1$ and $f_s^r = (H^r \cdot N^r)^{k_e^r} = 0.062$. Using the first equation with $I_r = I_a^r \cdot k_a^r + S_p^r \cdot I^r(k_d^r f_d^r + k_s^r f_s^r)$ we get

$$I_p = I_a k_a + S_p I(k_d f_d + k_s f_s) + k_r(I_a^r \cdot k_a^r + S_p^r \cdot I^r(k_d^r f_d^r + k_s^r f_s^r))$$

For the regular k_a, k_d, k_s, k_r we consult the table for R1 and for the reflected k_a^r, k_d^r, k_s^r we consult the table for C1.

Finally we need to make a distinction for the intensity of the light source, since the light-intensity normally falls of with the square of the distance d: $I_{actual} = \frac{I}{d^2}$. For I the distance is $d = 4$ making $I_{actual} = \frac{0.75}{4^2} = \frac{3}{64}$. Similarily for I^r the distance is $d = \sqrt{1^2 + 4^2} = \sqrt{17}$ making $I^r_{actual} = \frac{0.75}{\sqrt{17}^2} = \frac{3}{68}$. For $I_a = 0.75$ the intensity of the light source as said in the moodle-forum. Inserting all values into the formula we get:

$$I_r = (0.75 \cdot 0.9 + 1 \cdot \frac{3}{68}(0.1 \cdot 1 + 0.1 \cdot 0.062)) = 0.6797$$

$$I_{p \text{ without reflection}} = 0.75 \cdot 0.2 + 1 \cdot \frac{3}{64}(0.6 \cdot 1 + 0.3 \cdot 0.305) = 0.1824$$

$$I_p = I_{p \text{ without reflection}} + I_r k_r = 0.041789 + 0.3 \cdot 0.6797 = 0.3863$$

For computing the final color value we get:

$$C_p = C_{\text{base}} \cdot I_{p \text{ without reflection}} + C_{\text{reflected}} \cdot I_r k_r$$

$$C_p = (38, 235, 235)^T \cdot 0.1824 + (235, 182, 38)^T \cdot 0.2039 = (54.85, 79.97, 50.61)^T$$

Since this color is kinda dark, I suspect we shouldn't account for light-intensity falloff, so... without accounting for the light-intensity fall-off.

$$I_r = (0.75 \cdot 0.9 + 1 \cdot 0.75(0.1 \cdot 1 + 0.1 \cdot 0.062)) = 0.7547$$

$$I_{p \text{ without reflection}} = 0.75 \cdot 0.2 + 1 \cdot 0.75(0.6 \cdot 1 + 0.3 \cdot 0.305) = 0.6686$$

$$I_p = 0.6686 + 0.7547 \cdot 0.3 = 0.8950$$

Resulting in the color values:

$$C_p = (38, 235, 235)^T \cdot 0.6686 + (235, 182, 38)^T \cdot 0.2264 = (78.61, 198.33, 165.73)^T$$

— Now we do the same for Camera 2: After normalizing the initial vectors from the visualization on the task-sheet, we get: $V = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})^T$ The rest is a bit more complicated this time. The normal of R2 is a unit vector $(1, 0)^T$ rotated by 10 $N = Rot(10)(1, 0)^T = (0.9848, 0.1736)$. Since the side of R2 that is hit by the ray is clearly occluded from the light source by itself, we can set $S_p = 0$, which lets us disregard the entire term $I_i(k_d f_d + k_s f_s)$. We again compute the reflecting ray using $R(V) = -V + 2N(V \cdot N)$. After inserting we get $R(V) = (0.9063, -0.4226)^T$. Now we just need V, N, L and H for the reflection, again let's call them V^r, N^r, L^r, H^r. Since the viewing-vector is always from the surface that is viewed to the observer we get it by negating the incoming reflection-ray: $V^r = -R(V) = (-0.9063, 0.4226)^T$. We already have the normal and the last camera-ray: $N^r = (-1, 0)^T$. For the light-ray we calculate the point where the reflection-ray V^r hits R1, by solving the following formula for y: $(4, 6)^T + r \cdot (0.9063, -0.4226)^T = (7, y)^T$ we get: $y = 4.601$ so the point where the reflection-ray hits R1 is $(7, 4.601)^T$ which by calculating the difference between it and the center of the light-source and normalizing once more gives us the light-vector $L^r = (-0.9951, 0.0993)^T$. Furthermore, we compute $H^r = \frac{L^r + V^r}{||L^r + V^r||_2} = (-0.9644, 0.2647)^T$. Inserting those vectors in the formulas for f_d^r and f_s^r we get: $f_d^r = 0.9951$ and $f_s^r = (0.9644)^{15} = 0.5802$. Now we can finally compute the intensity of the reflection using the same respective formula as for camera1:

$$I_r = 0.75 \cdot 0.2 + 1 \cdot \frac{3}{64}(0.6 \cdot 0.9951 + 0.3 \cdot 0.5802) = 0.1861$$

$$I_{p \text{ without reflection}} = 0.75 \cdot 0.1 = 0.075$$

$$I_p = 0.075 + 0.1861 \cdot 0.7 = 0.2053$$

For computing the final color value we get:

$$C_p = (237, 66, 66)^T \cdot 0.075 + (38, 235, 235)^T \cdot 0.1303 = (22.73, 35.57, 35.57)^T$$

Again, without accounting for the light-intensity fall-off we would get the following values instead:

$$I_r = 0.75 \cdot 0.2 + 1 \cdot 0.75(0.6 \cdot 0.9951 + 0.3 \cdot 0.5802) = 0.7283$$

$$I_{p \text{ without reflection}} = 0.75 \cdot 0.1 = 0.075$$

$$I_p = 0.075 + 0.7283 \cdot 0.7 = 0.5848$$

Resulting in the color values:

$$C_p = (237, 66, 66)^T \cdot 0.075 + (38, 235, 235)^T \cdot 0.5098 = (37.15, 124.75, 124.75)^T$$

2 Textures

2.1 Footprint-Assembly

Vectors for MipMap calculation: $r_1 = (\frac{\delta u}{\delta x}, \frac{\delta v}{\delta x})^T$,$r_2 = (\frac{\delta u}{\delta y}, \frac{\delta v}{\delta y})^T$

$$r_1 = (\frac{4}{1}, \frac{2}{1})^T$$

$$r_2 = (\frac{4}{1}, \frac{8}{1})^T$$

$$\text{MipMap-Step } d = \log_2(\min(|r_1|, |r_2|))$$

$$|r_1| = \sqrt{20}, |r_2| = \sqrt{80}$$

$$\log_2(\sqrt{20}) \approx 2.16 \Rightarrow \text{next power of two: } 2^1 = 2; d = 2$$

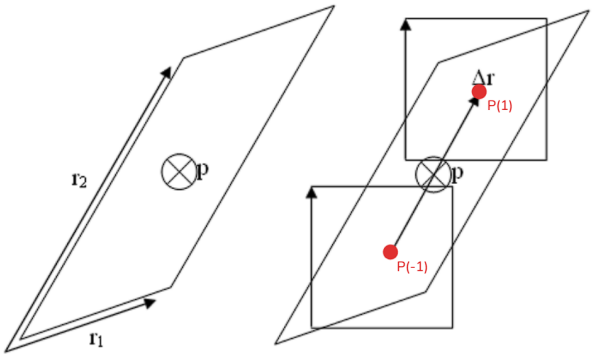
Calculation of the number of texture sections $N = \frac{\max(|r_1|, |r_2|)}{\min(|r_1|, |r_2|)} = \frac{\sqrt{80}}{\sqrt{20}} = 2$

$$\text{Scale bye } \frac{1}{N} : \triangle r = \frac{1}{2} * (4, 8)^T = (2, 4)^T$$

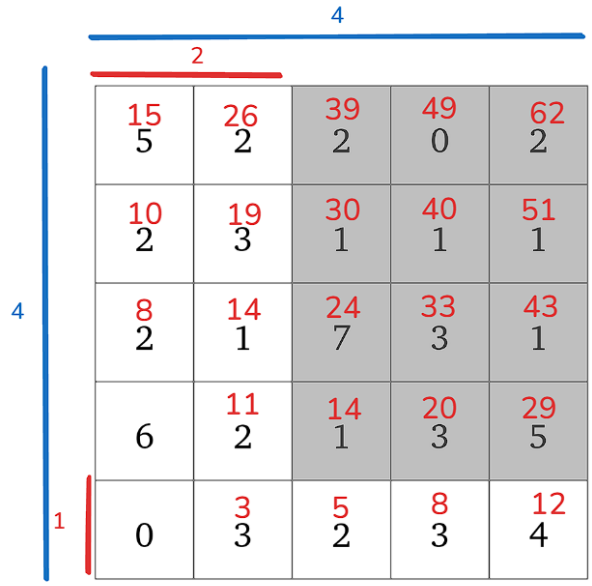
$$\text{Calculate sampling points: } P_n = P + \frac{n}{2} * (2, 4)^T; n \in \{\pm 1\}$$

$$P_{-1} = P + \frac{-1}{2} * (2, 4)^T = P * (-1, -2)^T$$

$$P_1 = P + \frac{1}{2} * (2, 4)^T = P * (1, 2)^T$$



2.2 Summed Area Table



The according values of C_{SAT} are red in upper image.

$$i_0 = 2, i_1 = 4; j_0 = 1, j_1 = 4$$

$$C_{avg}(i_0, j_0, i_1, j_1) = \frac{C_{sat}[i_1, j_1] - C_{sat}[i_0 - 1, j_1] - C_{sat}[i_1, j_0 - 1] + C_{sat}[i_0 - 1, j_0 - 1]}{(i_1 - i_0 + 1)(j_1 - j_0 + 1)}$$

$$C_{avg}(2, 1, 4, 4) = \frac{62 - 26 - 12 + 3}{12} = \frac{27}{12} = 2.25$$

How many texture accesses do we save in the calculation of C_{avg} in the example by using the SAT?
Complexity: $O((4-2) * (4-1)) = 12$
Complexity SAT: 4
Saved texture accesses: 8

3 Radiosity

Given Data

Patches and Reflection Coefficients

- $P_0 = (1, 5), \quad P_1 = (5, 2), \quad P_2 = (7, 2), \quad P_3 = (3, 1)$
- $\rho_0 = 0.9, \quad \rho_1 = 0.3, \quad \rho_2 = 0.3, \quad \rho_3 = 0.5$

Form Factors F_{ij}

$F_{00} = 0, \qquad F_{11} = 0, \qquad F_{22} = 0, \qquad F_{33} = 0$
$F_{01} = 0.2, \qquad F_{12} = 0, \qquad F_{23} = 0.35$
$F_{02} = 0.15, \qquad F_{13} = 0.6,$
$F_{03} = 0.02$

Initial Emission and Change in Energy

$B_{e,0} = 0.7, \qquad \Delta B_0 = 0.7$
$B_{e,1} = 0.1, \qquad \Delta B_1 = 0.1$
$B_{e,2} = 0.1, \qquad \Delta B_2 = 0.1$
$B_{e,3} = 0.1, \qquad \Delta B_3 = 0.1$

3.1 Iteration 1

Step 1: Identify the emitting patch (highest ΔB

- Patch P_0 has the highest $\Delta B_{e,0} = 0.7$, so it is the emitting patch.

Step 2: Compute the new values using

$B_i^k = B_i^{k-1} + \rho_i \cdot F_{ij} \cdot B_j^{k-1}$

- $B_1^1 = B_1^0 + 0.3 \times 0.2 \times 0.7 = 0.1 + 0.042 = 0.142$
- $B_2^1 = B_2^0 + 0.3 \times 0.15 \times 0.7 = 0.1 + 0.0315 = 0.1315$
- $B_3^1 = B_3^0 + 0.5 \times 0.02 \times 0.7 = 0.1 + 0.007 = 0.107$

Step 3: Compute the new energy changes

$\Delta B_1^1 = 0.3 \times 0.2 \times 0.7 = 0.042$
$\Delta B_2^1 = 0.3 \times 0.15 \times 0.7 = 0.0315$
$\Delta B_3^1 = 0.5 \times 0.02 \times 0.7 = 0.007$

3.2 Iteration 1

Step 1: Identify the emitting patch (highest ΔB

- Patch P_1 has the highest $\Delta B_1^1 = 0.042$.

Step 2: Compute the new values

- $B_2^2 = B_2^1 + 0.3 \times 0 \times 0.142 = 0.1315$ (no change)
- $B_3^2 = B_3^1 + 0.5 \times 0.6 \times 0.142 = 0.107 + 0.0426 = 0.1496$

Step 3: Compute the new energy changes

$\Delta B_2^2 = 0.3 \times 0 \times 0.142 = 0$
$\Delta B_3^2 = 0.5 \times 0.6 \times 0.142 = 0.0426$

3.3 Iteration 3

Step 1: Identify the emitting patch (highest ΔB

- Patch P_3 has the highest $\Delta B_3^2 = 0.0426$

Step 2: Compute the new values

- $B_2^3 = B_2^2 + 0.3 \times 0.35 \times 0.1496 = 0.1315 + 0.0157 = 0.1472$

Step 3: Compute the new energy change

$\Delta B_2^3 = 0.3 \times 0.35 \times 0.1496 = 0.0157$

Final Table

Iteration	B_0^k	ΔB_0^k	B_1^k	ΔB_1^k	B_2^k	ΔB_2^k	B_3^k	ΔB_3^k	Emitting Patch
0	0.7	0.7	0.1	0.1	0.1	0.1	0.1	0.1	-
1	0.7	0	0.142	0.042	0.1315	0.0315	0.107	0.007	P_0
2	0.7	0	0.142	0	0.1315	0	0.1496	0.0426	P_1
3	0.7	0	0.142	0	0.1472	0.0157	0.1496	0	P_3

Final Answer

After three iterations, the energy distribution is:

$B_0^3 = 0.7, \quad B_1^3 = 0.142, \quad B_2^3 = 0.1472, \quad B_3^3 = 0.1496$

And the last emitting patch was P_3.