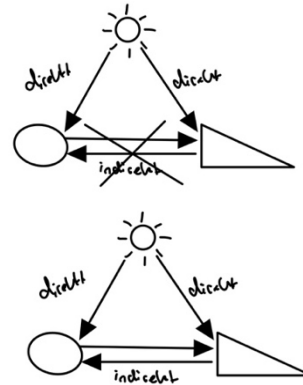


Teilthema 1: Lokale Beleuchtungsmodelle

a) Geben Sie die Definition eines lokalen Beleuchtungsmodells an. (RP, 2 P)

Definition 6.1 (Lokales Beleuchtungsmodell)

- Man berechnet die Intensität (Farbe) eines Punktes oder Pixels in Abhängigkeit von direktem Lichteinfall einer oder mehrerer Lichtquellen.
- Indirektes Licht wird nicht reflektiert, es gibt nur direkte Beleuchtung.



Definition 6.2 (Globales Beleuchtungsmodell)

- Man berechnet die Intensität (Farbe) eines Punktes oder Pixels in Abhängigkeit von direktem und indirektem Lichteinfall einer oder mehrerer Lichtquellen
- Licht, das über die Reflexion an anderen Objekten in der Szene eintrifft wird beachtet.

Erläutern Sie den Unterschied zwischen einem globalen und einem lokalen Beleuchtungsmodell

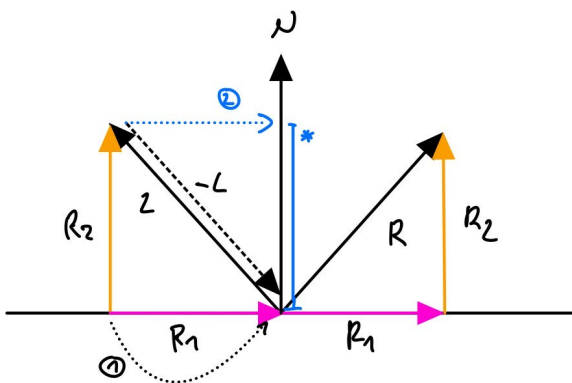
b) Geben Sie die beide prinzipielle möglichen Arbeitsweisen eines Schattierungsverfahrens an (in Bezug auf den Zusammenhang zum Beleuchtungsmodell) (RP, 2 P)

- Auswertung eines Beleuchtungsmodells **für die Vertices**,
Farben der Zwischenwerte für die (feiner aufgelösten) Pixel werden per Interpolation bestimmt
- Auswertung eines Beleuchtungsmodells **für jedes Pixel**
(aufwändiger, da es wesentlich mehr Pixel als sichtbare Vertices gibt)

c) Leiten Sie anhand einer Skizze die Formel für die perfekt spiegelnde Reflexion her. Welche vereinfachende Annahme wird hierzu gemacht? (RP, 3 P)

Annahme:

- L und R liegen in einer Ebene und sind normiert \rightarrow Einfallswinkel = Ausfallswinkel
- Die Größen in der Formel sind Vektoren
- L = Lichtvektor, R = Reflektionsvektor, N = Punktnormale



$$\begin{aligned}
 R &= R_1 + R_2 \\
 &\stackrel{①}{\Rightarrow} (R_2 - L) + R_2 \\
 &\Rightarrow 2 \cdot R_2 - L \\
 &\stackrel{②}{\Rightarrow} 2 \cdot (L \cdot N) \cdot N - L \\
 &\quad (N, L \text{ normiert})
 \end{aligned}$$

$$① R_1 = R_2 - L$$

$$② R_2 = (L \cdot N) \cdot N$$

Skalarprodukt
* \Rightarrow Faktor

d) Nennen Sie die 3 Komponenten, die beim Phong-Beleuchtungsmodell modelliert werden und geben Sie die vereinfachenden Annahmen an, die dem Phong-Modell zugrunde liegen. (RP, 3 P)

- Diffuses Licht, spiegelndes Licht und ambientes Licht

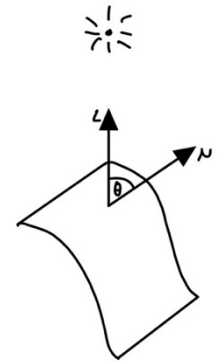
Vereinfachte Annahmen:

- Reflexionen werden nur lokal (also unter direktem Lichteinfall) betrachtet
- Lichtquellen sind punktförmig
- die Geometrie der Oberflächen, außer den Oberflächennormalen, wird ignoriert

e) Erläutern Sie anhand einer Skizze die Berechnung der perfekten diffusen Reflexion beim Phong-Beleuchtungsmodell. Beachten Sie dabei folgende Teilaspekte: (RP, 3 P)

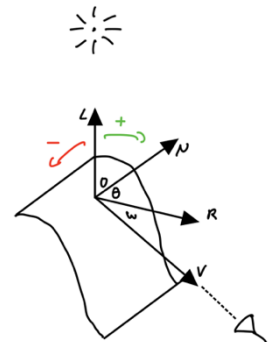
Perfekte diffuse Reflexion

- Beschriften Sie Ihre Skizze!
- Erläutern Sie die verwendeten Bezeichner
 - I_d = Intensität des reflektierten Lichts (skalärer Wert)
 - I_i = Intensität des einfallenden Lichts
 - L = Lichtvektor, N = Punktnormale θ = Winkel zwischen N und L
- Geben Sie die Formel zur Berechnung an.
 - $I_d = I_i * \cos \theta = I_i * (L * N)$
- Welche mathematische Eigenschaft müssen die Vektoren in der Formel besitzen? (Kategorie VZ)
 - die Vektoren müssen normiert sein



Unvollkommene spiegelnde Reflexion

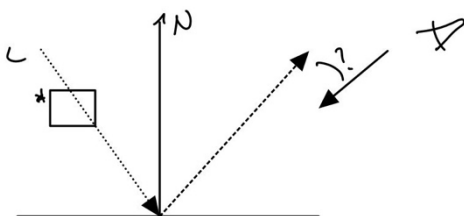
- Der Lichtstrahl wird bei der Reflexion „aufgespalten“, es entsteht eine **Reflexionskonus** um die ausgezeichnete Reflexionsrichtung. Das ist **abhängig** von der Betrachtungsrichtung.
- $I_s = I_i * \cos^S \omega = I_i * (R * V)^S$
- I_s = Die zu berechnende Intensität des reflektierten Lichts (skalärer Wert)
- I_i = Intensität des einfallenden Lichts
- ω = Winkel zwischen N und L
- R = Reflektionsrichtung, V = Betrachtungsrichtung, L = Lichtvektor, N = Punktnormale
- S = Perfektionsgrad, $S \rightarrow \infty$ (in Richtung R)
- die Vektoren müssen normiert sein



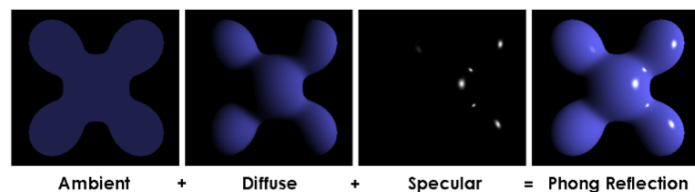
Gesamtmodell - Linearkombination des reflektierten Lichts

$$I = k_d I_d + k_s I_s + k_a I_a = I_i (k_d (L * N) + k_s (R * V)^n) + k_a I_a \quad (k_d + k_s + k_a = 1)$$

- I_a = Ambientes Licht, konstante Grundhelligkeit



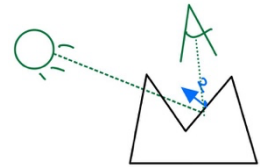
* kein direktes Licht
=> Schatten
Lösung => I_a



f) Erläutern Sie anhand einer Skizze die beiden Probleme „Schatten“ und „Lichtabgewandte Objekt-Teile“ bei lokalen Beleuchtungsmodellen (VZ, 3 P)

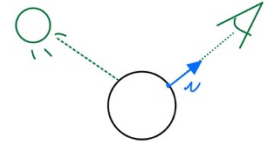
Schatten:

- Punkt liegt im Schatten, aber wird trotzdem beleuchtet, da das Modell nur auf die Oberflächenpunkte achtet und den Weg des einfallenden Lichts nicht berücksichtigt



Lichtabgewandte Objekt-Teile:

- Punkt ist dem Licht abgewandt und die Intensität des einfallenden Lichts ist 0. Das würde bedeuten Punkt ist schwarz, deswegen gibt es das ambiente Licht, welches unabhängig vom einfallenden Licht ist.



g) Was ist der Half-Way Vektor und warum wird er verwendet? (RP, 2 P)

- Das Problem bei dem spiegelnden Anteil ist das wir für jeden Punkt den reflektierten Strahl neu ausrechnen müssen → Performance Verlust.
Mit dem Half-Way-Vektor ist dies billiger zu berechnen und man erkennt kaum einen Unterschied.
- Der Half-Way Vektor liegt zwischen L und V.

h) Welcher Fehler kann bei der Benutzung des Half-Way Vektors passieren und wie wird er vermieden? (RP, 2 P)

- Das das Skalarprodukt $L \cdot V$ zwischen der Lichtquelle L und der Betrachtungsrichtung V positiv ist, obwohl das Licht bereits die Rückseite der Oberfläche bestrahlt.
- Es kann ein Fehler für größere Winkel entstehen ($H \cdot N > 0$), dies kann man umgehen, indem man zuerst den diffusen Anteil berechnet und ihn davon abhängig macht.
- Falls dieser 0 ist muss der spiegelnde auch 0 sein.

Teilthema 2: Interpolative Schattierungstechniken

a) Erläutern Sie den Zusammenhang zwischen Beleuchtung und Schattierung. Welche beiden grundlegenden Varianten bei der Schattierung gibt es? (RP, 2 P)

- Beleuchtung beschreibt, wie man die Intensität bzw. Farbe berechnet und das Schattierungsverfahren beschreibt, wo und wie häufig es berechnet wird.
- Entweder pro Polygon flat-shading, pro Eckpunkt gouraud-shading oder für jeden Pixel phong-shading.

b) Erläutern Sie die grundlegende Funktionsweise von Gouraud-Shading. Welche visuellen Effekte werden erzeugt und welche visuellen Probleme ergeben sich? (RP, 4 P)

- Die Beleuchtung erfolgt an den Polygoneckpunkten und wird mit der `vertex_normal` ausgerechnet und dann interpoliert.

Nachteil:

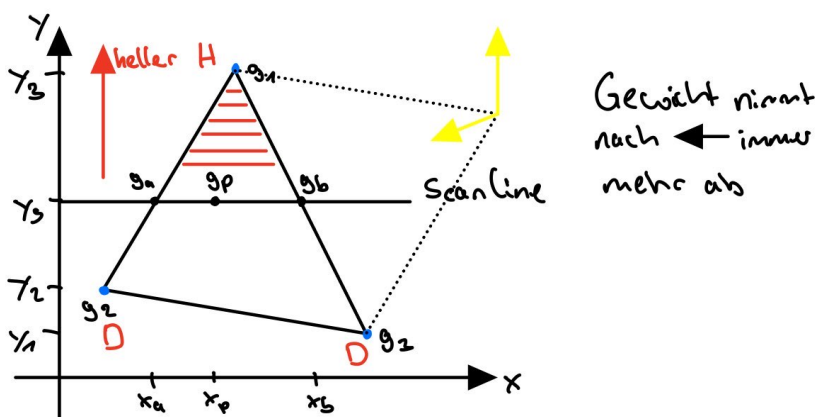
- Wenn das Licht nah an dem Polygon ist, dann flackert es
=> Highlights springt, deswegen nie bei bewegter Kamera
- Kleine Highlights innerhalb eines Polygons werden nicht berücksichtigt, da Ecken diese nicht abbekommen
- Silhouette bleibt eckig – Geometrie wird nicht verändert

c) Erläutern Sie an welchen Stellen bzw. unter welchen Bedingungen die polygonale Struktur eines Netzes beim Gouraud-Shading sichtbar werden kann. Begründen Sie, warum die Struktur sichtbar wird. (Achtung es gibt zwei unterschiedliche Effekte, die dazu führen) (VZ, 4P)

- Beim Highlight. Zum einen wegen der Interpolation werden die Highlights nach den Polygonen verschieden stark intensiv und nicht stetig. Zum anderen bei Bewegung, weil die Highlights dann springen und plötzlich verschwinden oder sichtbar werden
- An der Silhouette, weil die Geometrie nicht verändert wird

d) Erläutern Sie anhand einer Skizze das Prinzip der bi-linearen Interpolation für Punkte, Normalen oder Intensitäten innerhalb eines Dreiecks (RP, 3 P)

- Es wird ein Helligkeitsverlauf im Inneren des Polygons (Scan Line) erzeugt und auf dem Rand/Kanten wird auch einer ermittelt.
- Die Intensitätswerte werden entlang der Polygonkanten im Bildraum linear interpoliert, und danach zwischen den Kanten entlang der so genannten Scan-Lines



e) Erläutern Sie das Problem von glatten bzw. scharfen Kanten bei der Konstruktion von Objekten in Bezug auf die Beleuchtung. (VZ, 3 P)

- Schattierungen wie Gouraud oder Phong wollen Kanten glätten. Wenn man bewusst scharfe Kanten haben möchte, dann muss man doppelte Kanten festlegen, um verschiedene Normalen für einen Punkt zu haben.

Probleme bei je einem Netz für jede Teilfläche:

- Man verliert Konnektivitätsinformationen → Deformationen nicht möglich
- Doppelte Punkte mit der selben geometrischen Position
- Netzauflösung muss in beiden Meshes gleich sein!

Flat Shading

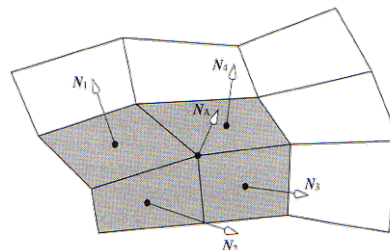
Die Beleuchtung wird pro Polygon genau einmal in einem ausgewählten Oberflächenpunkt (face_normal) ausgewertet und alle Punkte des Polygons übernehmen die Farbe.

Vorteile:

- Einfaches und effizientes Verfahren
- Interpolation findet nicht statt.
- Wird meist als Voransicht verwendet

Nachteile:

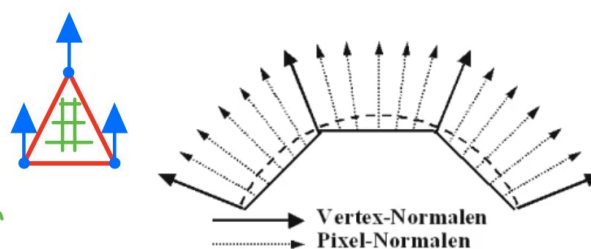
- Die meisten Kanten bleiben sichtbar
- Runde Objekte brauchen extrem viele Polygone



Phong-Shading

- Die Auswertung des Beleuchtungsmodells erfolgt für jeden Pixelmittelpunkt der Polygonoberfläche.
- Die notwendigen Oberflächennormalen in den Polygonpunkten werden mittels Interpolation aus den Eckpunktnormalen ermittelt.
 - o innere Kanten in Polygonnetzen werden geglättet
 - o Highlight sind immer vorhanden und haben die erwartete Form.
 - o **NT**: Da die Geometrie nicht verändert wird, bleibt auch hier die Silhouette „eckig“
- Ziel: Helligkeit pro Pixel berechnen und erfordert Normale und Pos. von Pixel, werden mit Interpolation bestimmt

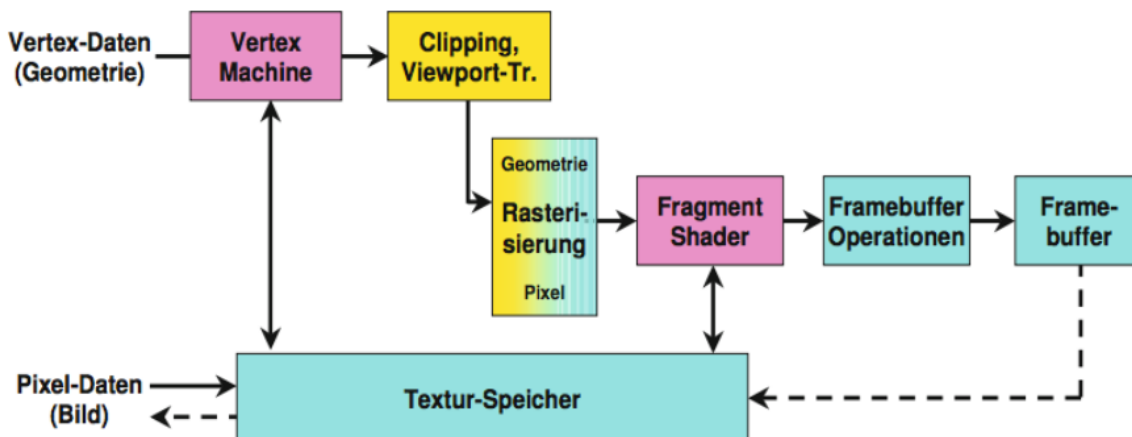
Helligkeit
Helligkeitsinterpolation
Ziel: Helligkeit pro pixel
erfordert normale + position pro pixel



Flat-Shading	Gouraud-Shading	Phong-Shading
Ein Beleuchtungswert für das gesamte Polygon	Interpolation der Beleuchtungswerte an den Eckpunkten	Interpolation der Normalen an den Eckpunkten und Berechnen der Beleuchtungswerte

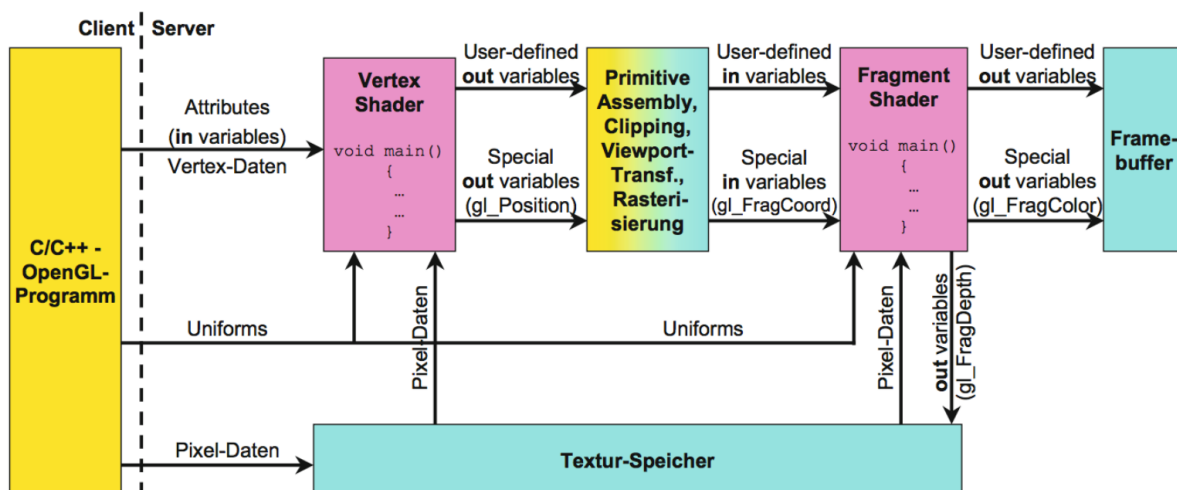
Teilthema 3: OpenGL Pipeline

- a) Geben Sie den Prinzipiellen Ablauf der OpenGL-Pipeline als Skizze an und erläutern Sie kurz die Funktion jeder Komponente (1 Satz pro Komponente, Texturspeicher muss nicht erläutert werden) (RP, 5 P)



- Berechnungen in gelb: kontinuierlich, und in blau: pixelbasiert
- Vertex-Machine: besteht aus mind. einem Vertex-Shader, ein Programm, dass durch alle Vertex einmal ausgeführt wird
- Clipping, Viewport-Tr.: Objekte, die nicht gesehen werden, werden abgeschnitten und das Bild wird in die richtige Größe transformiert. Verdeckungsproblem wird gelöst
- Rasterisierung: projizierte Punkte (stetig) werden in Pixel (diskret) umgewandelt
- Fragment Shader: ähnlich wie Vertex-Shader, aber hier für alle Pixel
- Framebuffer Operationen: zum Beispiel Anti-Aliasing
- Framebuffer: enthält Pixel als Array

- b) Erläutern Sie anhand einer Skizze die Funktionsweise der verschiedenen Variablentypen in der OpenGL Pipeline (Datentransfer) (RP, 4 P)



- Es gibt 3 Daten-Arten
- In-Variablen, die pro Vertex oder Fragment variieren
- Uniform-Variablen, die pro Grafik-Primitiv variieren
- Uniform sampler2D Variablen, häufig Pixeldaten
- Zusätzlich gibt es out-Variablen, die können benutzerdefiniert sein oder in OpenGL fest eingebaut sein

c) Wird die Auswertung des Beleuchtungsmodells im Fragment-Shader für alle Pixel ausgeführt?

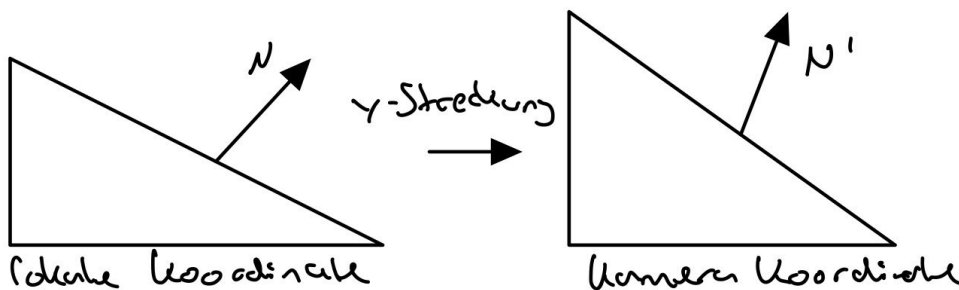
Begründen Sie Ihre Antwort. (TR, 2 P)

- Nicht für alle Pixel ausgeführt, weil man es auch in der Vertex-Machine gemacht werden kann. Und Überprüfung davor, ob es verdeckt ist, dann kann man es verwerfen.
- 1. Überprüfung ob davor ob überhaupt beleuchtet
- Die Formel wird nur dann ausgewertet, wenn der Oberflächenpunkt „von vorne“ beleuchtet wird, also wenn Licht drauf fällt. Wird bei der Beleuchtungsberechnung getestet unabhängig davon ob ich pro Pixel oder pro Vertex rechne, der Verweis auf Fragment-Shader war also nur eine Nebelkerze.

a.) Wieso wird eine Normal-Matrix benötigt? Geben Sie eine Skizze an und erläutern Sie die Problematik. (RP, 3 P)

Erläutern Sie, warum die Normalen nicht mit derselben kombinierten Transformationsmatrix (Welt-LookAt-Projektion) transformiert werden können wie die Punkte.

- Da eine nicht gleichförmige Skalierung in x, y, und z die Normale manipuliert



d) Leiten Sie die Berechnung der Normal-Matrix incl. Formeln her und erläutern Sie die Vorgehensweise. (RP+VZ, 5 P)

- Normale N und N' transformiert
- Tangente T und T' transformiert. T ist orthogonal zu N
- G ist die zu suchende Normal-Matrix
- M_3 ist die model_view_matrix ohne homogene Koordinate

$$N' \cdot T' = (G \cdot N) \cdot (M_3 \cdot T) = 0$$

$$\Leftrightarrow (G \cdot N)^T \cdot (M_3 \cdot T) = N^T \cdot G^T \cdot M \cdot T = 0$$

$$G^T \cdot M = I \quad \Leftrightarrow \quad G = (M^{-1})^T = G$$

- Matrix multiplikation
- Skalar produkt

Da N und T nach der Transformation orthogonal sein müssen gilt: $G = M$

e) Erläutern Sie, warum die Beleuchtungsberechnung in Raster- Koordinaten mathematisch nicht korrekt ist. (RP, 2 P)

- Weil die Rasterkoordinaten nicht mehr stetig sind, sondern diskret. Deswegen sind die Teilverhältnissen falsch bei der Projektion von Pixeln in Raster

- b.) Vertex-Shader beschreiben. Was kann er, Was kann er nicht. Was er bekommt und was er weitergibt
- Vertex Shader erhält Vertex / Normale in lokalen Koordinaten als Eingabe und alle benötigten Transformationsmatrizen
 - Berechnet die Position des Vertex in NDC und ggf. dessen Helligkeit/Farbe
 - gibt diese Ergebnisse weiter in der CG-Pipeline zur weiteren Verarbeitung.
- c.) Was sind die Aufgaben, Vor- und Nachteile der Vertex Machine?
- besteht aus mind. einem Vertex-Shader, ein Programm, dass durch alle Vertex einmal ausgeführt wird (parallel)
 - **Vorteil:** skaliert gut, d.h. schnellere GK sind um den Faktor der parallel vorhandenen Pipelines schneller
 - **Nachteil:** Geometrie-Konnektivität nicht verfügbar
- d.) Vertex-Shader Code zeigt Multiplikation aller Matrizen immer wieder pro Punkt. Sagen ob und weshalb dies ineffizient ist.
- Die Matrixmultiplikation muss nur 1x auf der CPU ausgerechnet werden. Da kann man die Assoziativität ausnutzen