

Aufgabe 1

- a) Die subjektiv empfundene Stärke von Sinneseindrücken ist proportional zum Logarithmus der Intensität des physikalischen Reizes.

$$1\%-2\% \approx k = \frac{\Delta L_{JND}}{L}$$

Addition required for the change to be perceived (Just Noticeable Difference)
 ← original Intensity (Helligkeit)

- b) Die darstellbaren Farben eines Ausgabegeräts (bzw. die darstellbaren eines Eingabegeräts)

	RGB	CMY	HSV	CIE x,y
Additiv:	X			
Subtraktiv:		X		
Multiplikativ:			(X)	
trennt Luminanz: Chrominanz:			X	X
enthält alle sichtbaren Farben:				X
nativ auf peripherie	X	(X)		

→ Was heißt nativ? und ist Zeitung peripherie
oder ein schwarz/weiß monitor z z z z z

Aufgabe 2a)

Vorteile:

- kompakte Beschreibung statt Speicherung
- Erzeugung von großen Szenen / Viel Content

Nachteile:

- schwer zu kontrollieren
- Approximation
- rechenintensiv / langsam

Beispiele:

- Vegetation
- Landschaften (Gelände mit Height maps)

b)

Noise-Funktionen nach Ken Perlin (1985)

- Grundlage für stochastische Modellierung / procedurale Texturen
- Basis bilden Pseudo Zufallszahlen aus denen Bilder mit gewissen Eigenschaften erzeugt werden

Eigenschaften / Anforderungen

- reproduzierbar: $n(x)$ liefert für jede Auswertung von x dasselbe Resultat.
- keine Periodizität (zumindest nicht sichtbar)
- begrenzter Wertebereich (ASS auf Farben / druckfarben...)
- definierte Frequenzverteilung, bandlimitiert (Aliasing reduzieren)
- Stetigkeit, räumliche Korrelation: $n(x) \approx n(x + \epsilon)$

Erzeugung: z.B. durch Lattice Vektordotprodukte

- Zufallszahlen auf einem Gitter

$$Z(x) = \text{random}(Lx], Ly], Lz])$$

- räumliche Korrelation und Bandbreitenbegrenzung durch Interpolation

Bsp:

$$n(x) = Z(x) \cdot (1 - f_x) + Z(x+1) \cdot f_x$$

mit $f_x = x - Lx]$ (f_x nach Komponenten von x)

Aufgabe 3

Rekursiv?

a)

- View Ray
(Primärstrahl)
- Reflection Ray
(Reflexionsstrahl)
- Shadow Ray
(Schattenstrahl)
- Refraction Ray
(Gebrochener Strahl
(Transmissionsstrahl))

Ja

Ja

b)

• Rekursions tiefe Vorteil: Einfach

- Nach ein Anzahl von Reflexion ist die Intensität des Strahls so gering, dass abgebrochen werden kann da sich die Farbe bzw Ergebnis kaum noch verändert.

Vorteil: Sehr genau, verschwendet wenig Zeit in manchen Fällen (z.B. Strahl wird durch mehrere ~~sehr~~ wenig Transparenze Objekte gebrochen)

c)

(Wenn ein Abbruchkriterium erfüllt ist) xD

Ansonsten wenn das Material weder Reflektiert noch Transmittiert (durchlässig ist.)

Aufgabe 4

a) Magnification

- Abbildung weniger Texel auf viele Pixel
- Interpolation mehrerer Texelwerte z.B. bilinear, bikubisch, ...
bikubische Interpolation und Interpolation höherer Ordnung sind Texer

Minification

- Abbildung mehrerer Texel auf einen Pixel
- Mögliche Lösungen:
 - Supersampling (Texer)
 - Wird jedes Texel auslesen und dann interpolieren
 - Vorfilterung
 - z.B. durch Mip Mapping

b) Latitude/Longitude - Maps

- Parametrisierung über Polwinkel zur z-Achse
- Vergleichsweise "naure" Berechnung der Texturkoordinaten
- sehr ungleichmäßige Abtastung an den Polen:
viele Texel für wenige Richtungen.
- + Einfach?

Sphere Mapping

- Fotografiere kleine Spiegelkugel, das Foto wird zur Sphäre auslesen der Map mittels Abtastung Richtung (v).
- ungleichmäßige Abtastung
- Nur für Blickrichtung ähnlich so geeignet.

Cube Mapping

- Abbilden des Umgebungs auf einen Würfel um den Betrachter
- Benötigen 6 x 2 D-Texturen + Mip Mapping möglich

Aufgabe 5

- a) Die Idee ist potentiell geschnittene Objekte schneller zu finden.
Weiterhin keine Objekte zu testen, die nicht in Frage kommen.

Dadurch wird die Performance des Raytracings drastisch erhöht.

b)

	BVH	Octree	Gitter	BSP
Aufbau adaptiv	X	X		X
Datenstruktur Binärbaum	X			X
Mehrfachschnitt		X	X	X
leerer Raum überspringen	(X)	X	X	(X)
Achsenparallele Unterteilung	X	X	X	
Schnelle Area	X			X

Aufgabe 6

- a) ~~P_1 Octcode~~ $y_{min}, x_{min}, x_{max}$

P_1 Octcode enthält y_{min} und x_{min}

P_2 Octcode enthält x_{max}

Octcodes entstehen in dem jedes Kante eine Bit-Verkettung gesehen wird,

Ob y_{min} oder x_{min} geschnitten hängt von den Winkel

zwischen $\overline{P_1 P_2}$ und y_{min} bzw. x_{min} ab. Also

davon ob x_{min} vor y_{min} geschnitten wird bzw. umgekehrt.

- b) wenn Octcode (A) \wedge Octcode (B) $\neq \emptyset$ sind.

Eine Veränderung wird ausgeführt.

Aufgabe 6

a)

Normale interpoliert

Mach-Bund-Effekt

Vertex Position Interp.

Flächennormale

ASbildung

OpenGL

Flat

Gouraud

Phong

~~X~~

X

X

X

X

X

X

X

X

b)

Berechnung durch lineare Komponentenweise Interpolation der Vertex-Normalen anhand der baryzentrischen Koordinaten

geg: n_1, n_2, n_3 eines Dreiecks $\Delta(P_1, P_2, P_3)$

ges: n_Q am Punkt Q

Lösung: berechne $\lambda_1, \lambda_2, \lambda_3$, dann ist $n_Q = \lambda_1 n_1 + \lambda_2 n_2 + \lambda_3 n_3$

$$\frac{n_Q}{|n_Q|}$$

Denn lineare Komponentenweise Interpolation ist nicht Längen erhaltend.

c).

Zu Blinn Phong benötigt Vektor zur Kamera, Vektor zur Lichtquelle und Reflexionsvektor, aus denen dann noch der Half-way Vektor berechnet wird. All diese Vektoren existieren auch wenn man Flat Shading verwendet.

(Phong-Shading != Phong Beleuchtungsmodell.)

Aufgabe 8

Sichtbarkeit vor Rasterisierung

Wahr

Falsch
X

Opake Primitive

X

Tiefenwert aus bsp. zentrisch

X

Abhängigkeit Near/Far-Plane

(X)

Transparente Oberflächen

X

Open GL Tiefentest vor Alpha

✗

X

Aufgabe 9

- a) Sei N die Anzahl der ~~3~~ Dreiecke die gezeichnet werden sollen.
Normalerweise werden $3N$ Vertices benötigt.
Beim Triangle Strip wird nach dem ersten Dreieck für jedes weitere Dreieck nur noch ein zusätzlicher Vertex benötigt.
Insgesamt also nur $n+2$ Vertices, dass spart Speicher.

b)

Ein Array mit Vertices

+ Ein Array mit Reihenfolge (Indices)

Grund: Vertex meint Bestandteil mehrerer Dreiecke

Verbesserung: Weniger Berechnungen und Vertex Caching

c)

Kombination der Frontside im Framebuffer mit Füllen von Fragmenten

Anwendung semi-transparente Objekte

d) 1) GL_DST_COLOR, GL_ZERO

2) GL_ONE, GL_ONE

e)

frei-programmiersbare Transformation einzelner Vertices und Verarbeitung deren Attribute

- keine Vertex Erzeugung und Löschung

- Eingabe ist:

- Vertex und seine Attribute
- globale Zustände, Konstanten

- Ausgabe: Normale für Phong, (Allgemein Attribute für frag. Shader)

Aufgabe 10

a) de Casteljau-Algorithmus

b) 1. $F(u)$ liegt in der abgeschlossenen konvexen Hülle des Kontrollpolygons

2. Endpunktinterpolation $F(0) = b_0$ und $F(1) = b_n$

3. Tangentenbedingung: $F'(0) = n(b_1 - b_0)$ und $F'(1) = n(b_n - b_{n-1})$

4. affine Invarianz: sei $\varphi(x) = Ax + t$ eine affine Abbildung, dann gilt:

$$\varphi(F(u)) = \sum_{i=0}^n B_i^n(u) \varphi(b_i)$$

d.h. um die Kurve zu transformieren genügt es, die b_i zu transformieren

5. Variationsreduzierung („variation diminishing Property“)

„Eine Bezielskurve F wackelt nicht stärker als ihr Kontrollpolygon B “

$$\binom{0}{k} = 0 \quad \binom{n}{0} = 1$$

$$c) B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}, \text{ mit } i=0, \dots, n \quad \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$B_0^3 = 1 \cdot 1 \cdot (1-u)^3 = (1-u)^3$$

$$B_1^3 = \frac{3!}{1!(2)!} \cdot u \cdot (1-u)^2 = 3u(1-u)^2$$

$$B_2^3 = \frac{3!}{2!(1)!} \cdot u^2 \cdot (1-u) = 3u^2(1-u)$$

$$B_3^3 = \frac{3!}{3!(0)!} u^3 = u^3$$

$$b_0^0 = (1, 1) \quad b_1^0 = (1, 10) \quad b_2^0 = (10, 10) \quad b_3^0 = (1, 1)$$

$$b_0^1 = (1-u) \cdot b_0^0 + u \cdot b_1^0$$

$$b_1^1 = (1-u) \cdot b_1^0 + u \cdot b_2^0$$

$$b_2^1 = (1-u) \cdot b_2^0 + u \cdot b_3^0$$

$$b_3^1 = (1-u) \cdot b_3^0 + u \cdot b_4^0$$

$$b_0^2 = (1-u) \cdot b_0^1 + u \cdot b_1^1 = \begin{pmatrix} 2 \\ 6 \end{pmatrix} \quad b_1^2 = (1-u) \cdot b_1^1 + u \cdot b_2^1 = \begin{pmatrix} 3 \\ 7 \end{pmatrix}$$

$$b_2^2 = (1-u) \cdot b_2^1 + u \cdot b_3^1 = \begin{pmatrix} 5 \\ 9 \end{pmatrix}$$

$$b_0^2 = \frac{2}{3} b_0^1 + \frac{1}{3} b_1^1 = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

$$= \frac{2}{3} b_1^0 + \frac{1}{3} b_2^0 = \begin{pmatrix} 4 \\ 10 \end{pmatrix}$$

$$= \frac{2}{3} b_2^0 + \frac{1}{3} b_3^0 = \begin{pmatrix} 7 \\ 2 \end{pmatrix}$$