

Computer-Graphik II

Tone Mapping / Tone Reproduction

G. Zachmann

Clausthal University, Germany

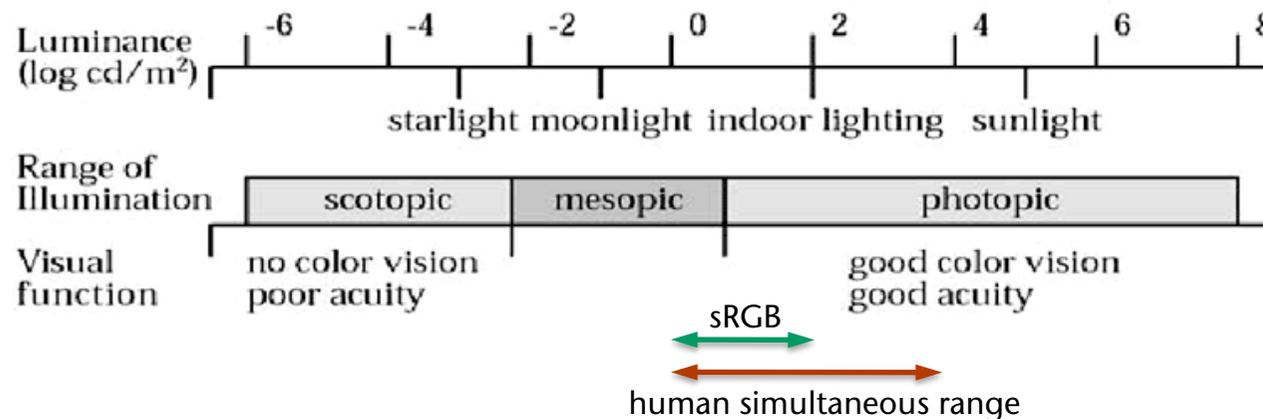
cg.in.tu-clausthal.de



Motivation



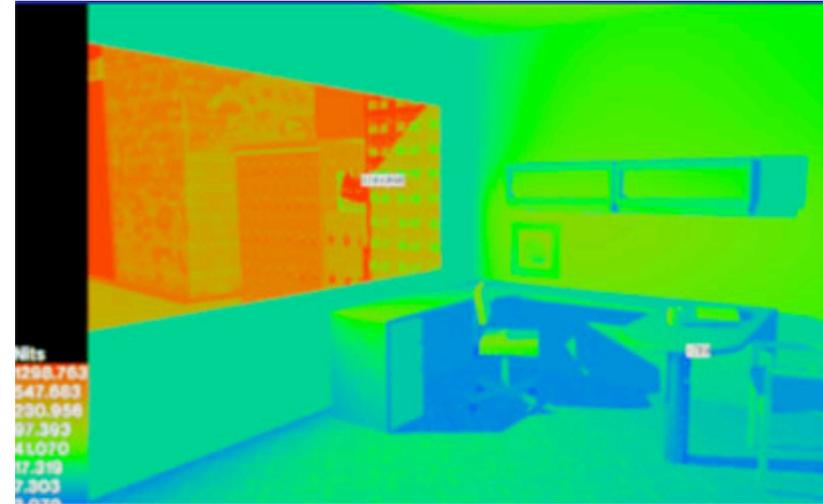
- Definition:
 - Der **Dynamikbereich** (*dynamic range*) eines Bildes ist das Kontrastverhältnis zwischen dem hellsten und dunkelsten Teil
 - Der **Dynamikbereich eines Displays bzw. optischen Sensors** ist das Verhältnis der hellsten darstellbaren bzw. wahrnehmbaren Leuchtdichte zur dunkelsten
- Der Dynamikbereich des menschlichen Sehsystems (Human Visual System, HVS):





Quellen von High Dynamic Range Images (HDRI)

- Ray-Tracing: physikalisch korrekte synthetische Bilder
- Photographie:
 - mehrere Aufnahmen mit verschiedenen Belichtungszeiten
 - Ineinander "blenden" (benötigt kalibrierte Antwortkurve der Kamera)





Darstellung von HDR-Bildern

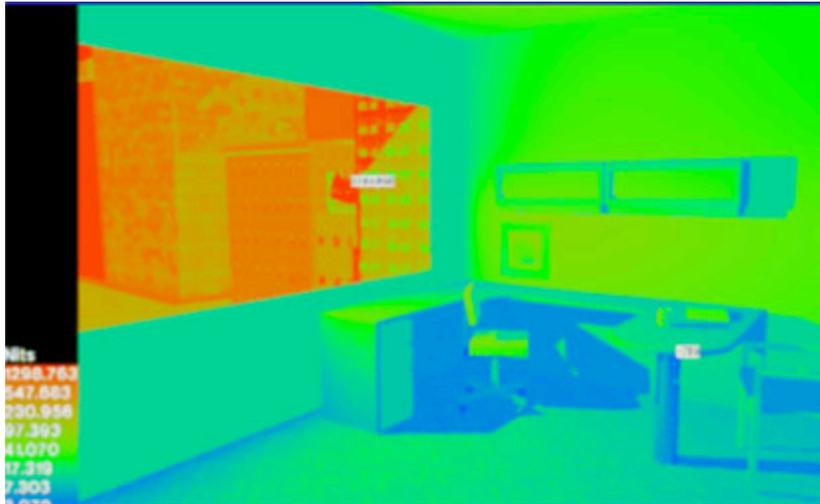
- Verwende entweder HDR-Displays ...
- ... oder LDR-Displays; dann benötigt man:



- **Tone mapping/ tone reproduction** = Abbildung des potentiellen "high dynamic range" (HDR) von realen Leuchtdichten auf eine "low dynamic range" (LDR) eines Displays mit begrenzter Leuchtdichtenbandbreite.

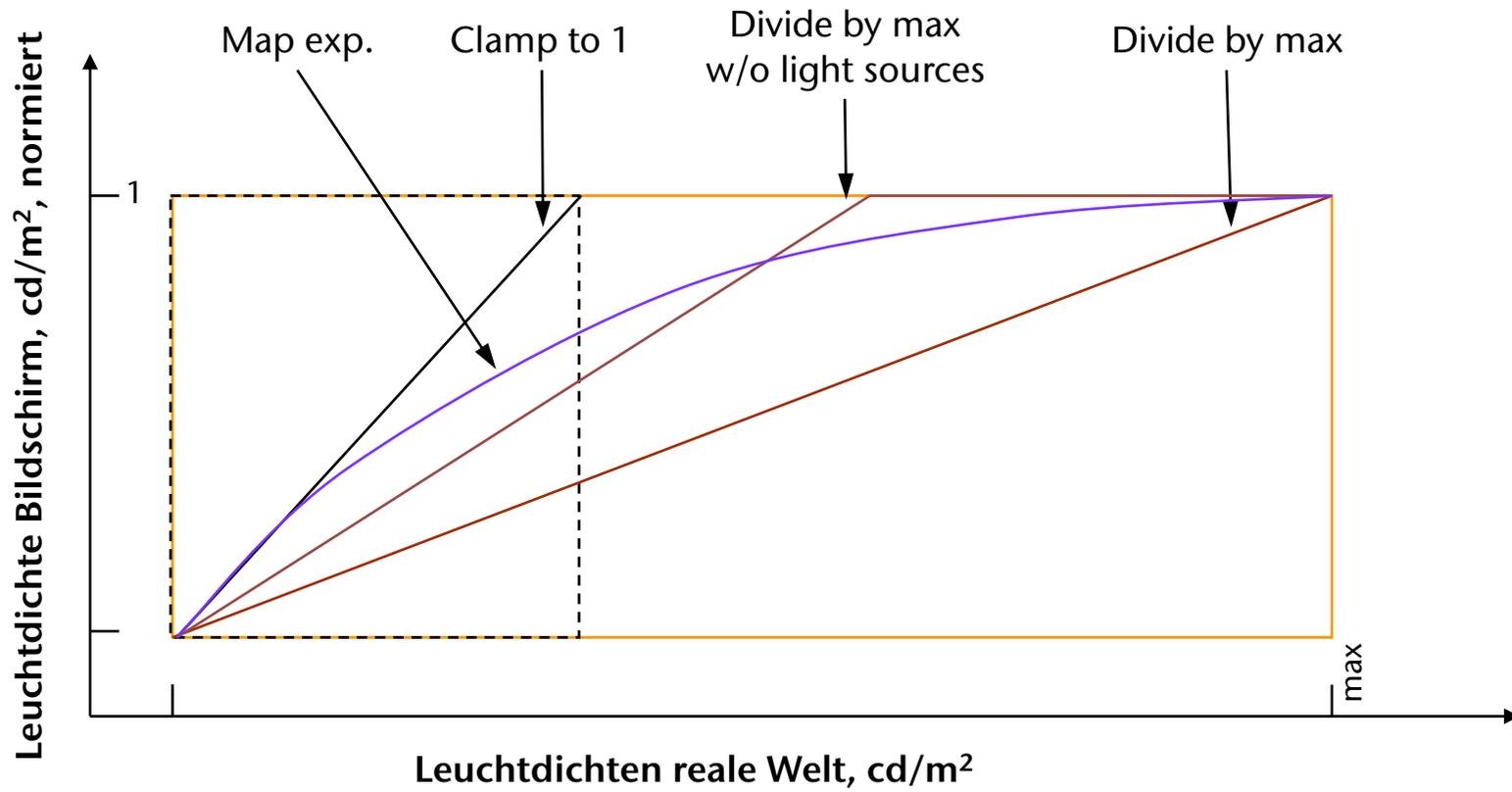


Informelle Darstellung des Problems





Naives Mapping





Resultat des naiven Mappings



Divide by Max



Clamp to 1



Exp. mapping



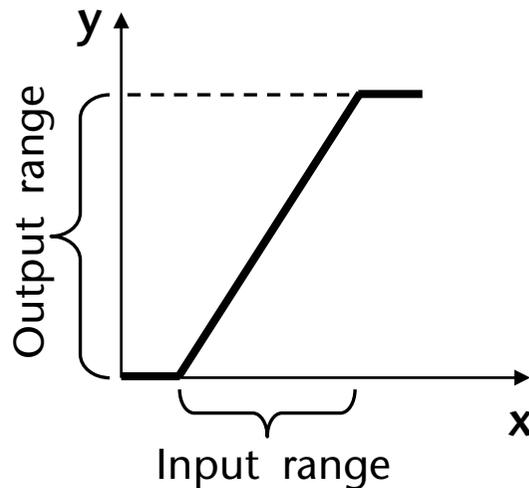


Graustufentransformationen

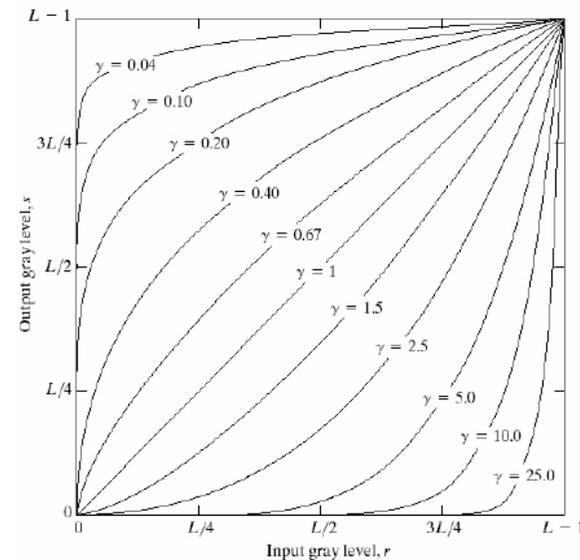


- Betrachte zunächst reine "Punktoperationen":
 - $y = T(x)$
 - T heißt **Transferfunktion**
 - Völlig unabhängig von Nachbarpixeln
- Beispiele:

Lineare Skalierung



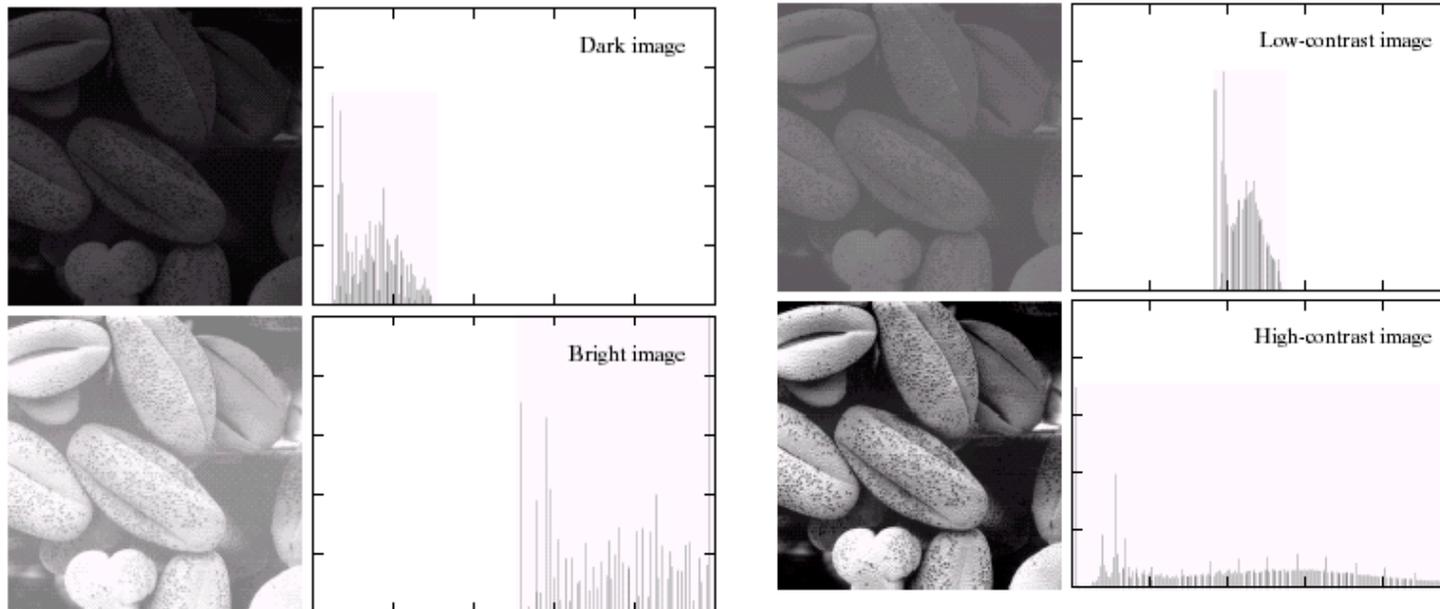
Gammakorrektur





Das Luminanz-Histogramm

- Unbalanciertes Histogramm nutzt nicht den vollen dynamischen Wertebereich aus
- Balanciertes Histogramm ergibt ein angenehmeres Bild und gibt den Inhalt wesentlich besser wieder





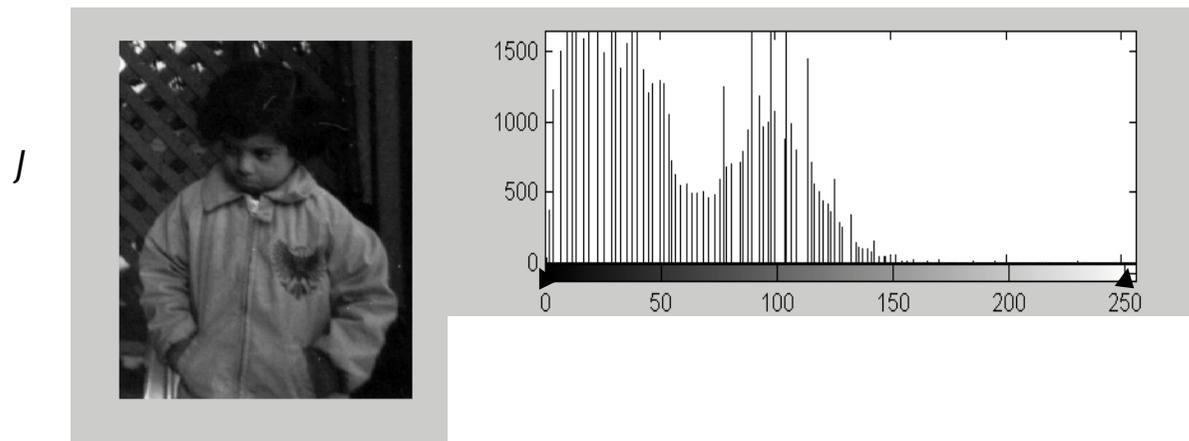
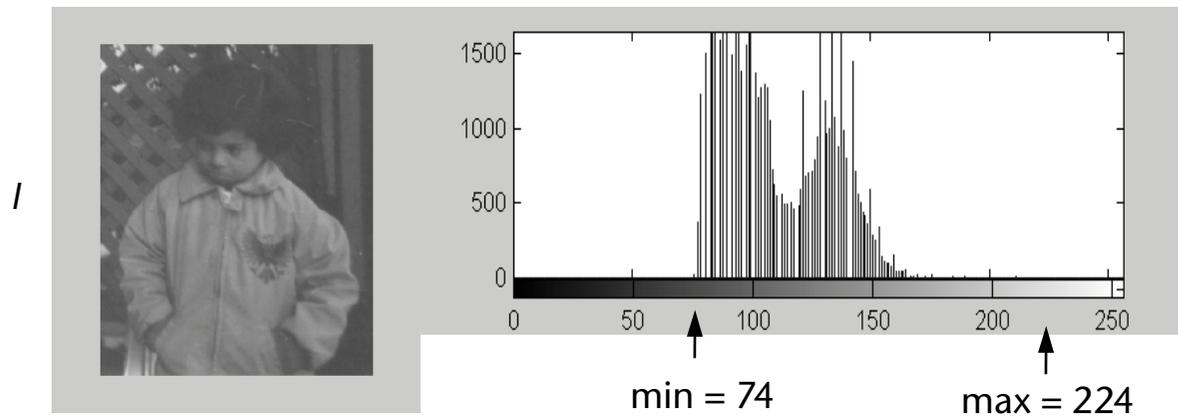
- Das Histogramm eines Bildes enthält wertvolle Informationen über die Graustufen
- Es enthält **keine räumlichen** Informationen
- Alle folgenden Bilder haben exakt das gleiche Histogramm!





Histogram stretching

- Lineare Skalierung = "*histogram stretching*":
$$J = \frac{I - I_{\min}}{I_{\max} - I_{\min}} \cdot J_{\max}$$





Interpretation des Histogramms

- Handle Pixel als **unabhängige, gleichverteilte Zufallsvariablen** (*i.i.d. random variables = independent, identically distributed RVs*)
- Histogramm = diskrete Approximation der Wahrscheinlichkeitsdichtefunktion (*probability density function, PDF*)





Diskrete vs. kontinuierliche Formulierung

$L =$ Anzahl Levels

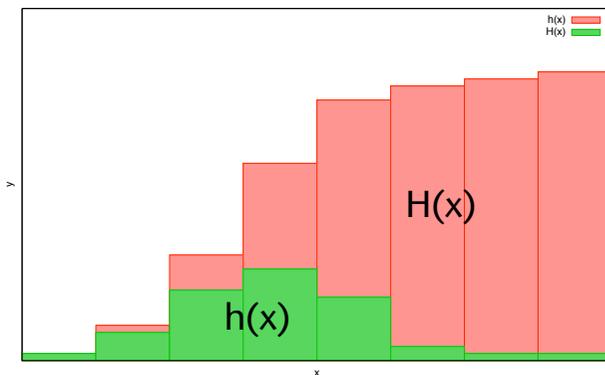
$x \in 0, \dots, L - 1$

Histogramm:

$h(x) =$ Anz. Pixel mit Level x

Kumulatives Histogramm:

$$H(x) = \sum_{u=0}^x h(u)$$



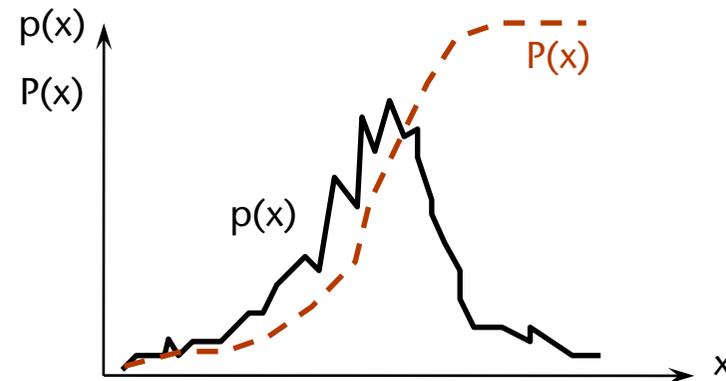
$x \in [0, 1]$

PDF:

$p(x) =$ "Dichte" am Level x

Cumul. distrib. function (CDF):

$$P(x) = \int_0^x p(u) du$$





- Klar ist:

$$H(L - 1) = \sum_{u=0}^{L-1} h(u) = \text{Anzahl Pixel } N$$

- Deswegen wird oft $h(x)$ bzw. $H(x)$ mit $\frac{1}{N}$ normiert
- Die Wahrscheinlichkeit, daß das Ereignis " $X \leq x$ " eintritt, ist

$$P[X \leq x] = P(x) = \int_0^x p(u) du$$

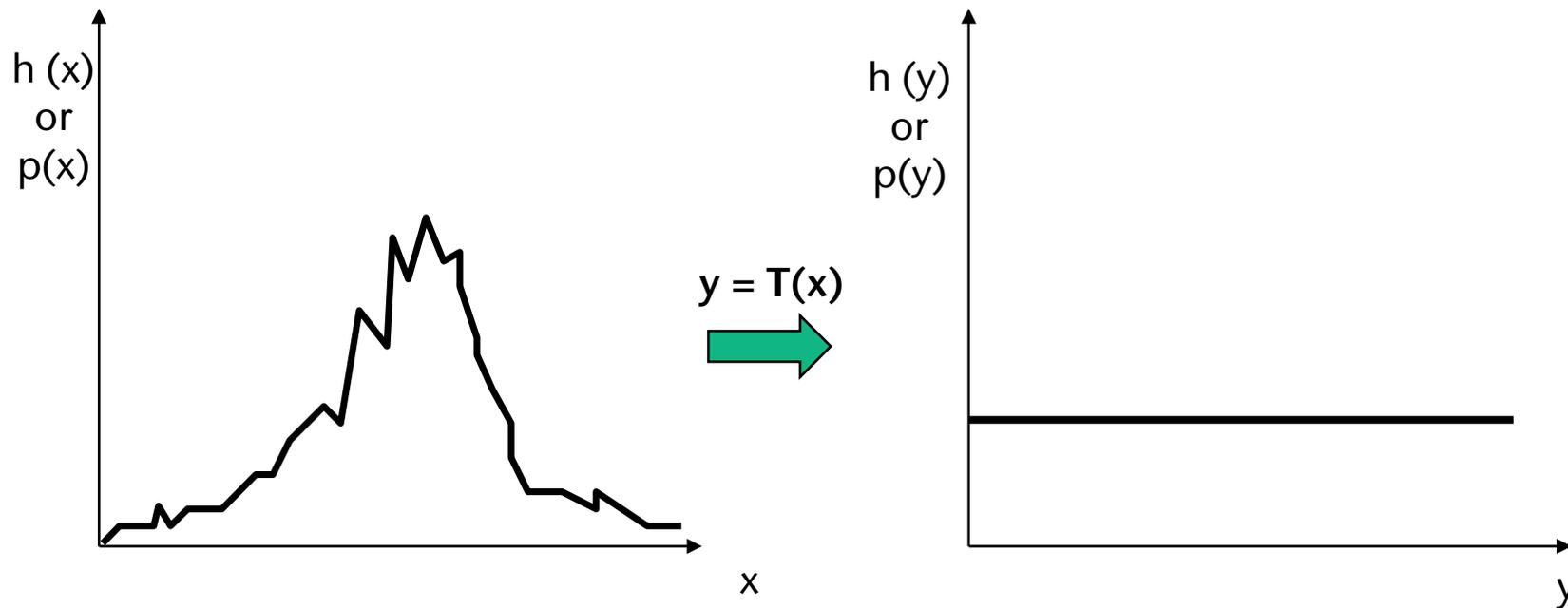
bzw.

$$P[X \leq x] = P(x) = \frac{1}{N} \sum_0^x h(u)$$



Histogram Equalization

- Gegeben: eine Zufallsvariable X mit bestimmter PDF p_X
- Gesucht: Funktion T , so daß die Zufallsvariable $y = T(x)$ eine gleichverteilte PDF $p_Y \equiv \text{const}$ hat
- Diese Transformation heißt **Histogram Equalization**

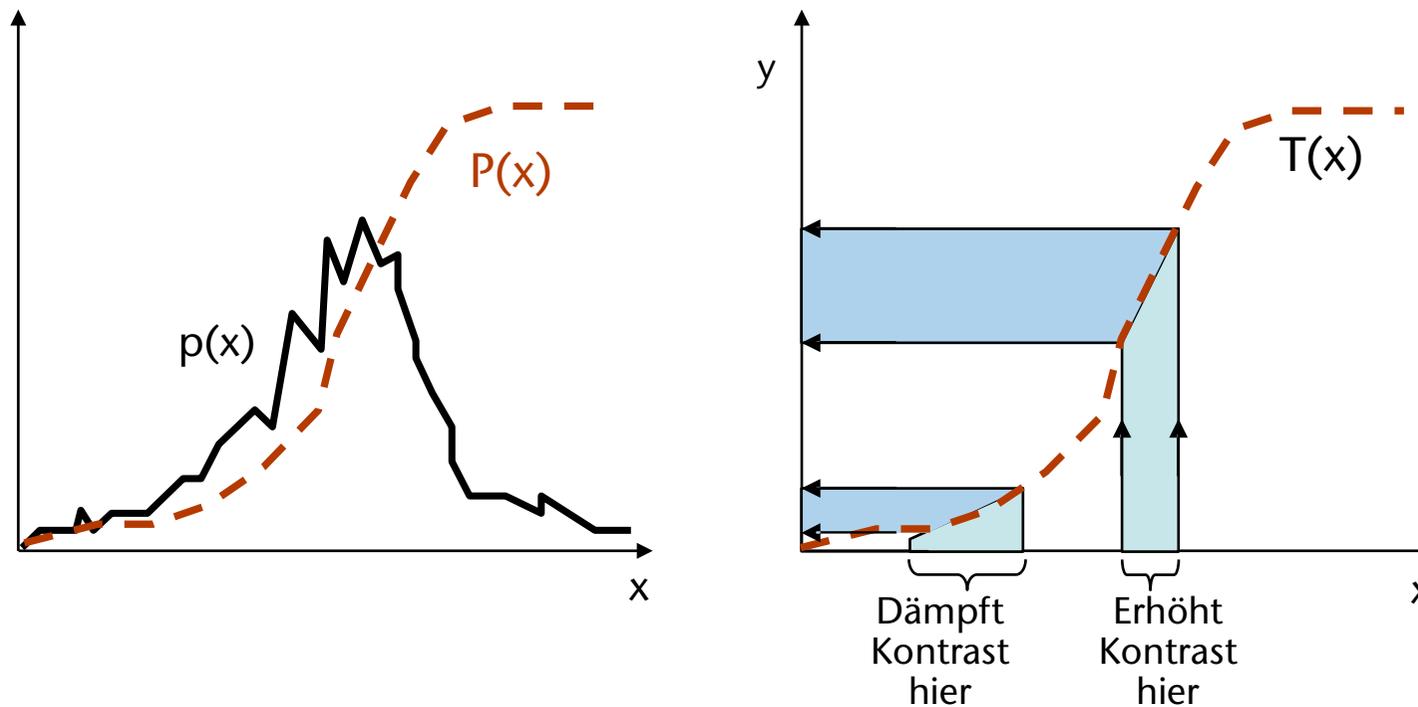




- Behauptung: die Transferfunktion

$$y = P(x) = \int_0^x p(u) du$$

leistet genau diese Histogramm-Equalization





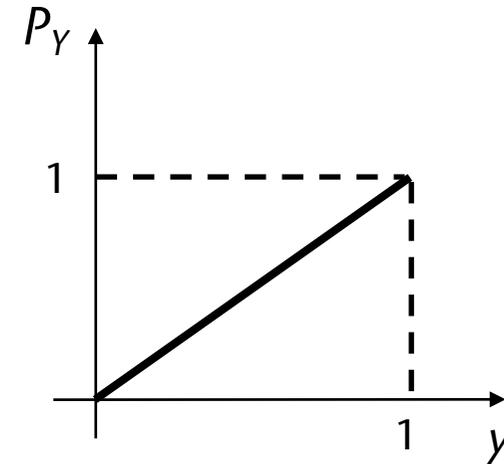
1. Beweisvariante

- Zu zeigen: die CDF

$$P_Y(y) = y$$

- Beweis durch Einsetzen:

$$\begin{aligned} P_Y(y) &= P[Y \leq y] \\ &= P[T(X) \leq y] \\ &= P[P_X(x) \leq y] \\ &= P[x \leq P_X^{-1}(y)] \\ &= P_X(P_X^{-1}(y)) \\ &= y \end{aligned}$$





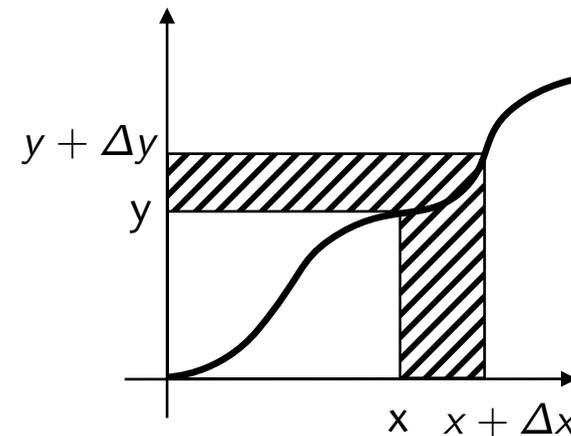
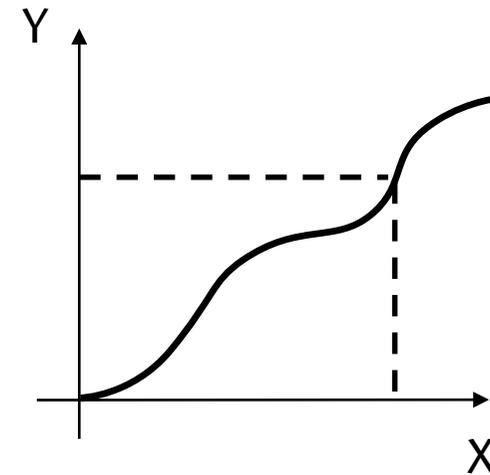
2. Beweisvariante

- Sei X eine stetige Zufallsvariable
- Sei $Y = T(X)$ (Y ist also auch stetige ZV)
- Sei T diff'bar und monoton wachsend
- Damit existiert T' und T^{-1}
- Da T alle $x \leq s \leq x + \Delta x$ auf $y \leq t \leq y + \Delta y$ abbildet, gilt

$$\int_x^{x+\Delta x} p_X(s) ds = \int_y^{y+\Delta y} p_Y(t) dt$$

- Für kleine Δx gilt also

$$p_Y(y) \Delta y \approx p_X(x) \Delta x \quad p_Y(y) \approx p_X(x) \frac{\Delta x}{\Delta y}$$





- Wenn $\Delta x \rightarrow 0$, dann wird die Gleichung exakt:

$$p_Y(y) = \lim_{\Delta x \rightarrow 0} p_X(x) \frac{\Delta x}{\Delta y} = p_X(x) \lim_{\Delta x \rightarrow 0} \frac{1}{\Delta y / \Delta x}$$

$$\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{T(x + \Delta x) - T(x)}{\Delta x} = T'(x)$$

- Zusammen:

$$p_Y(y) = \frac{T_X(x)}{T'(x)}$$



- Jetzt noch $x = T^{-1}(y)$ einsetzen ergibt

$$p_Y(y) = \frac{p_X(T^{-1}(y))}{T'(T^{-1}(y))}$$

- Nebenresultat: wir wissen jetzt, wie man Verteilungsfunktionen umrechnen muß, wenn eine Zufallsvariable eine Funktion einer anderen Zufallsvariable ist.
- Zurück zur Histogramm-Equalization ...



- Gesucht ist T , so daß

$$p_Y(y) \equiv 1$$

- Einsetzen liefert

$$\frac{p_X(T^{-1}(y))}{T'(T^{-1}(y))} = 1$$

$$T'(T^{-1}(y)) = p_X(T^{-1}(y))$$

- Einsetzen von $x = T^{-1}(y)$ liefert $T'(x) = p_X(x)$

- Gesucht war T , also noch integrieren :

$$T(x) = \int_0^x T'(u) du = P_X(x)$$



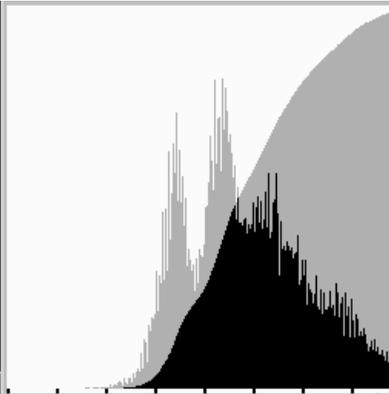
Beispiele



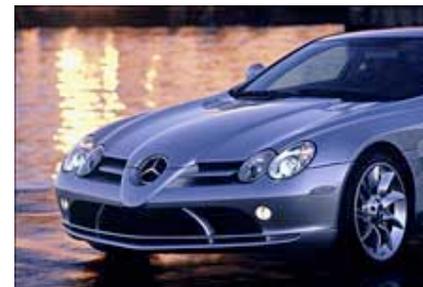
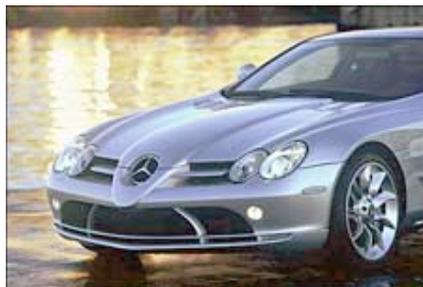
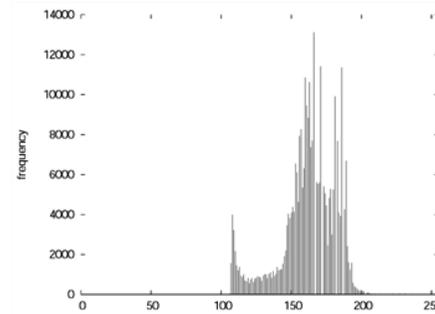
Orig. Bild



Histogramm



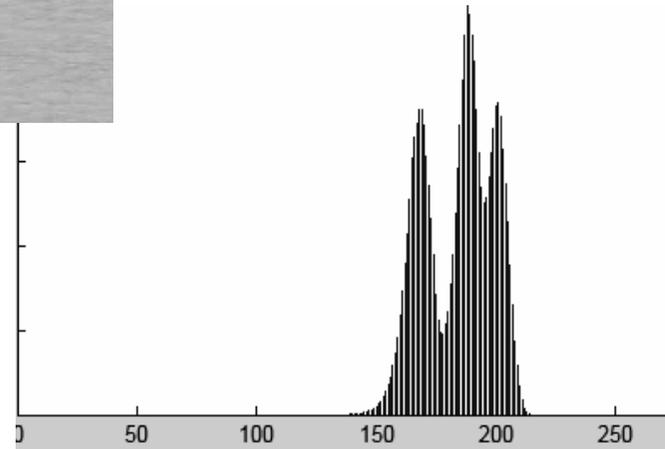
Resultat





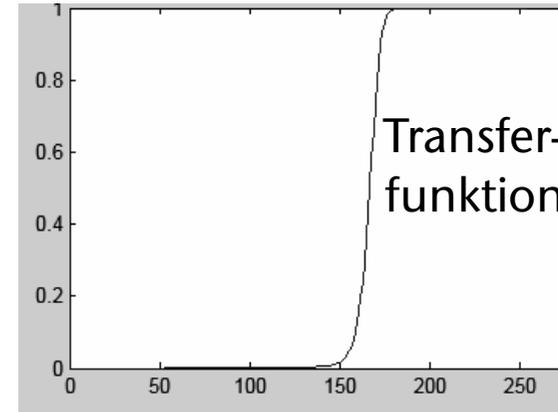
Problem der Histogramm-Equalization

- Problem: ein schmales Histogramm

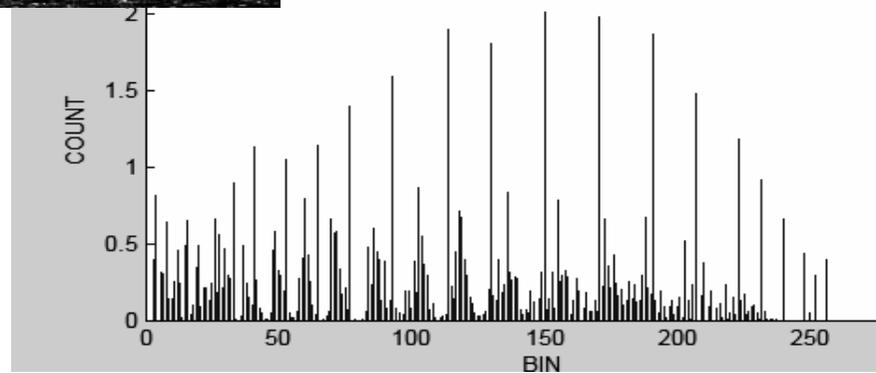




- Ergebnis: unerwünschter Kontrast



Resultierendes
Histogramm



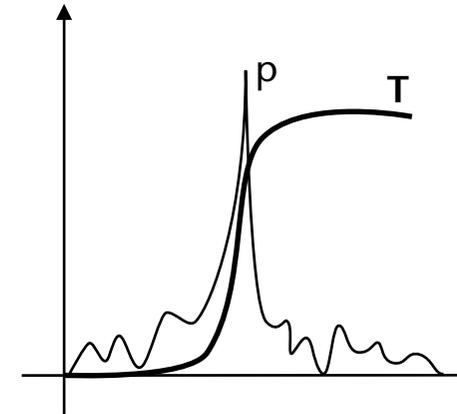


Tone Reproduction nach Ward et al.

[1997]



- Problem der Histogramm-Equalization:
 - Sehr steile Abschnitte der Transferfunktion T können sichtbares Rauschen hervorrufen
- Idee: beschränke die Steigung von T
- Algo:
 1. Bestimme das Histogramm h
 - Erinnerung $h \approx p = T'$
 2. Clampe zu große Bins auf einen Wert $\alpha \cdot \frac{N}{B}$, wobei $\alpha \approx 0.5 \dots 1.5$,
 $N = \text{Anzahl Pixel}$, $B = \text{Anzahl Bins}$
 3. Setze $N' = \sum_{i=0}^{L-1} h(x_i)$.
 4. Wiederhole ein paar Mal.





Exkurs: Das Weber-Fechner-Gesetz

[~1850]



- Experimenteller Befund:
 - Die "just noticeable difference" (JND) eines Stimulus (z.B. Gewicht) hängt von der Stärke des Stimulus ab.
 - Das Verhältnis von JND zu Stärke des Stimulus ist eine (stimulusabhängige) Konstante.
- Mathematische Formulierung:
 - Sei S der Stimulus, ΔS die JND
 - Weber's Gesetz:

$$\frac{\Delta S}{S} = k$$



- Das Weber-Fechner-Gesetz:

Sei E die Stärke des wahrgenommenen Sinneseindrucks von S ("gefühltes Gewicht").

Dann ist

$$\Delta E = k \frac{\Delta S}{S} \qquad \frac{dE}{dS} = k \frac{1}{S}$$

- Integrieren liefert:

$$E = k \cdot \ln S + c$$

- Dabei ist c eine Konstante, die den minimaler Stimulus S_0 beschreibt, bei dem gerade noch ein Sinneseindruck $E \approx 0$ entsteht:

$$c = -k \cdot \ln S_0$$

- Zusammen:

$$E = k \cdot \ln \frac{S}{S_0}$$



- Im Fall des Sehsinnes kann man es noch genauer angeben:

$$\Delta E = \begin{cases} -2.8 & , \log L < -3.9 \\ (0.4 \log L + 1.6)^{2.2} - 2.8 & , -3.9 \leq \log L < -1.4 \\ \log L - 0.4 & , -1.4 \leq \log L < -0.02 \\ (0.3 \log L + 0.7)^{2.7} - 0.7 & , -0.02 \leq \log L < 1.9 \\ \log L - 1.3 & , \log L \geq 1.9 \end{cases}$$



Anwendung im Tone Mapping

- Annahme: zwei benachbarte Pixel im Originalbild haben gerade einen Intensitätsunterschied von der JND, also

$$\Delta L = L_1 - L_2 = J(L_1)$$

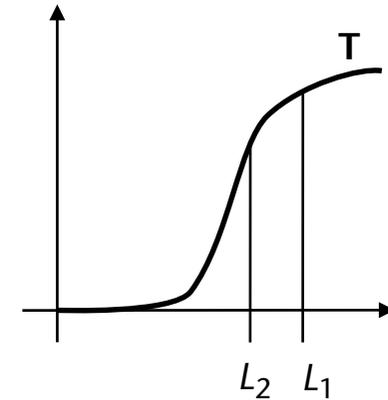
(oBdA ist $L_1 > L_2$)

- Gesucht ist eine Transferfunktion T , so daß diese Bedingung eine Invariante ist, also

$$T(L_1) - T(L_2) \leq T(J(L_1))$$

- Umformen:

$$p(L_1) = T'(L_1) \approx \frac{T(L_1) - T(L_2)}{L_1 - L_2} \leq \frac{J(T(L_1))}{L_1 - L_2} = \frac{J(T(L_1))}{J(L_1)}$$





■ Algorithmus:

1. Berechne das Histogramm
2. Berechne das kumulative Histogramm, d.h., T
3. Clampe alle Bins, so daß

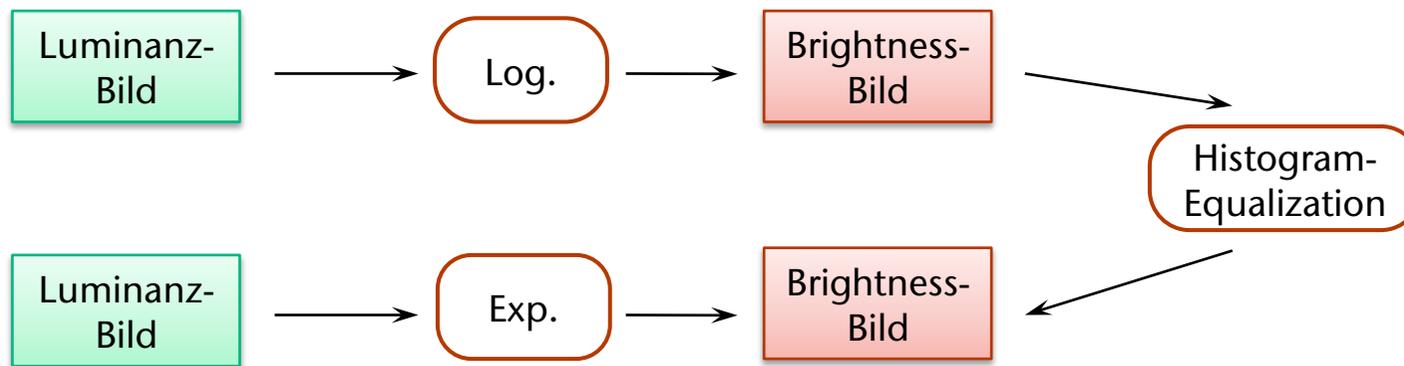
$$h(i) \leq \frac{J(T(L_i))}{J(L_i)}$$

wobei L_i der Intensitätslevel von Bin i ist

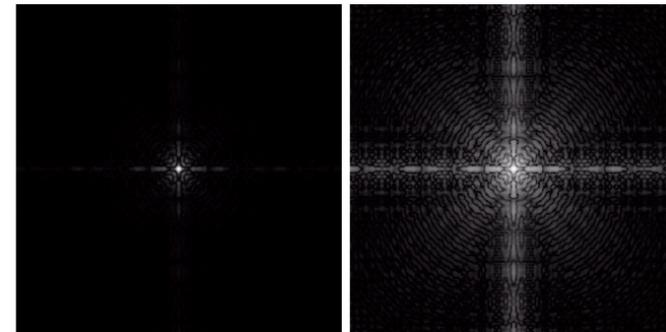
4. Wiederhole ein paar Mal



- Nebenbemerkung: das Weber-Fechner-Gesetz ist auch der Grund, warum die Histogramm-Equalization bzw. das Tone-Mapping sehr oft im sog. "Log-Space" durchgeführt wird



Fourier-Spektrum
eines Bildes
vor und nach
dem Log.





Beispiel





Weitergehende Ideen



- Problem: diese Methode verhindert $\Delta L > J(L)$ auch zwischen Pixeln, die **nicht** benachbart sind
 - Idee: clamp nicht das Histogramm, sondern jedes Pixel individuell unter Berücksichtigung der Nachbar-Pixel
 - Lokaler Tone-Mapping-Operator (TMO)
 - Führt wieder zu anderen Problemen (z.B. sog. "Halos")
- Weitere Beschränkungen des Human Visual Systems (HVS):
 - Blendung (glare): starke Lichtquellen in der Peripherie reduzieren Kontrastempfindlichkeit des Auges
 - Skotopisches / mesopisches Sehen: bei niedriger Luminanz nimmt die Farbempfindlichkeit stark ab
 - Ebenso nimmt räumliches Auflösungsvermögen ab
 - Könnte man im TMO ausnutzen



Erzeugen eines Grauwertbild-Histogramms



- Gegeben: Grauwertbild (= Textur)
- Ziel: Histogramm als 1D-Textur
 - Jedes Texel = ein Bin
- Problem: "Verteilen" auf Bins
 - Ziel-Adresse eines Fragment-Shaders ist ja fest
- Erste Idee:
 - Pro Pixel im Originalbild einen Punkt (GL_POINT) "rendern",
 - im Vertex-Shader das entsprechende Bin ausrechnen (statt Transf. mit MVP-Matrix),
 - die "Koordinate" dieses Bins als Koordinate des Punktes setzen
- Problem:
 - Hohes Datenübertragungsvolumen CPU → GPU
 - Z.B.: $1024^2 \times 2 \times 4$ Bytes = 8 MB zusätzlich zum 1024^2 -Bild

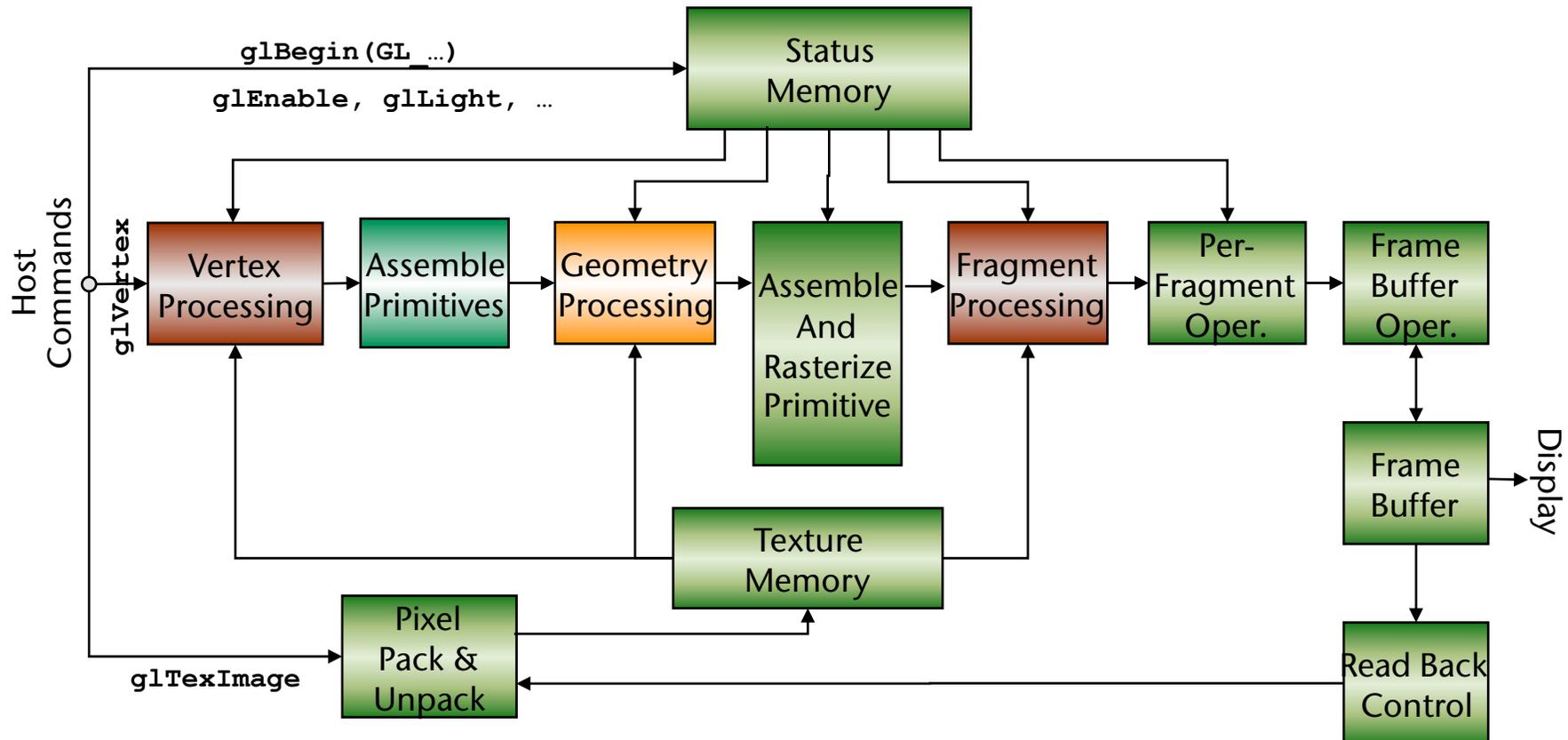


Enter ... the Geometry Shader

[Nov 2006]



- Eine dritte Shader-Art:



- Doku: http://www.opengl.org/registry/specs/NV/geometry_shader4.txt



Funktionsweise des Geometry Shaders



- Applikation generiert irgendwelche Primitive (Points, Lines, Triangle-Fans, etc.)
- Vertex-Shader transformiert diese (immer ein Vertex auf einmal)
- Geometry-Shader bekommt von der Assembly-Stufe Primitive
 - Nur GL_POINTS, GL_LINES, GL_TRIANGLES (+ 2 weitere)
- Geometry-Shader gibt neue Primitive aus
 - Nur GL_POINT, GL_LINE_STRIP, GL_TRIANGLE_STRIP
 - Muß nichts mit der eingegebenen Geometrie zu tun haben
 - Anzahl kann (fast) beliebig sein, ist unabhängig von der Anzahl der eingegebenen Primitive
- Typ der Input-/Output-Geometrie muß vorab festgelegt werden
- Zugriff auf OpenGL-State und Texturen wie üblich

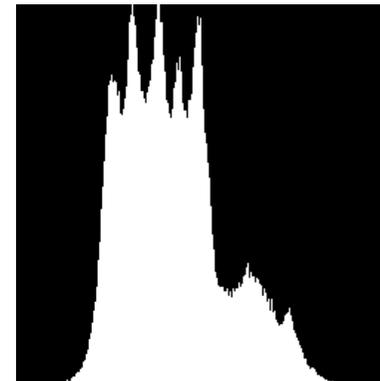


- Output des Vertex-Shaders:
`gl_Position` →
`gl_Normal` →
`gl_TexCoord` →
...
- Input des Geometry-Shaders:
`gl_PositionIn[]` →
`gl_NormalIn[]` →
`gl_TexCoordIn[][]` →
- Output des Geometry-Shaders:
`gl_Position`
`gl_Normal`
`gl_TexCoord[]`



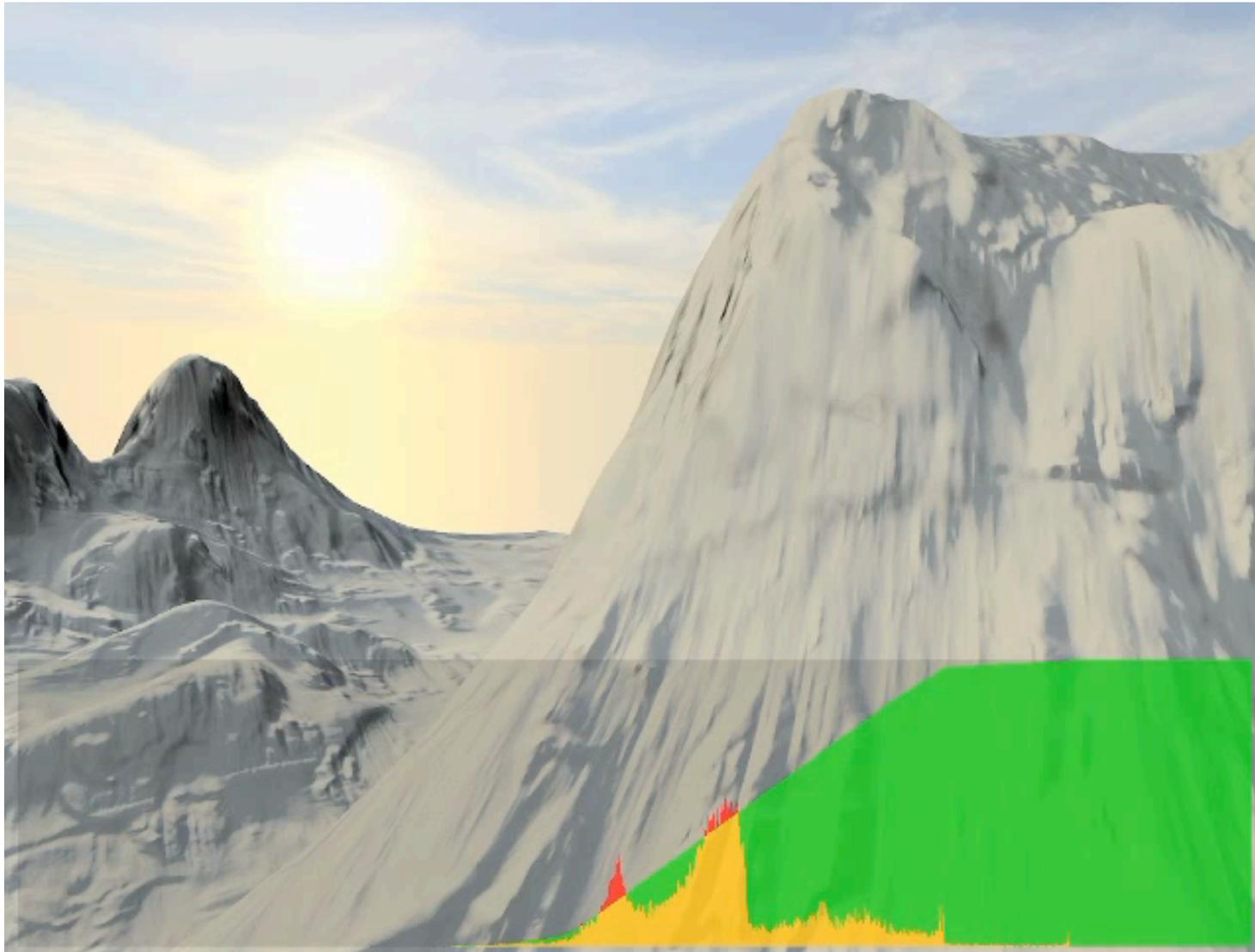
Histogramme mit Geometry-Shader

- Ein Quad in der Applikation rendern
- Vertex-Shader ist (fast) leer
- Der Geometry-Shader ...
 - läuft durch das Bild,
 - erzeugt für jedes Pixel ein Point-Primitiv mit der x-Koordinate = Bin , $y=0$
- Fragment-Shader ...
 - nimmt die Points,
 - gibt Farbe (1,0,0,0) aus,
 - an der Position (x,0)
- Fragment-Operation ...
 - ist auf Blending eingestellt mit `glBlendFunc (GL_ONE , GL_ONE) =` Akkumulation (aktuelle Karten können das auch mit FP-FBOs)





Video



Thorsten Scheuermann, Justin Hensley; 2007.
Graphics Product Group, Advanced Micro Devices Inc.





Berechnung der Transferfunktion auf der GPU



- Erinnerung: *parallel prefix sums* bzw. *summed area tables*



High-Dynamic Range Imaging in der Photographie



- Waren sogar zuerst da [Charles Wyckoff, 1930-40]
- Inzwischen alles in Photoshop & Co. integriert



Beispiele





