

LEHRSTUHL FÜR RECHNERARCHITEKTUR UND PARALLELE SYSTEME

Grundlagenpraktikum: Rechnerarchitektur

Gruppe 143 – Vortrag zur Aufgabe A208

Image Processing – Gamma Korrektur

Ar Pazari

Edera Ndoj

Demil Omerovic

Wintersemester 2023 – 2024

12 März 2024

Vortragsinhalte

- Einleitung
- Ansätze
- Ergebnisse
- Performanzanalyse
- Zusammenfassung

Einleitung: Gamma Korrektur

- Grundlegende Algorithmus des Image Processing Gebiets
- Ziel: die Helligkeit und Kontraste in digitalen Bildern anpassen

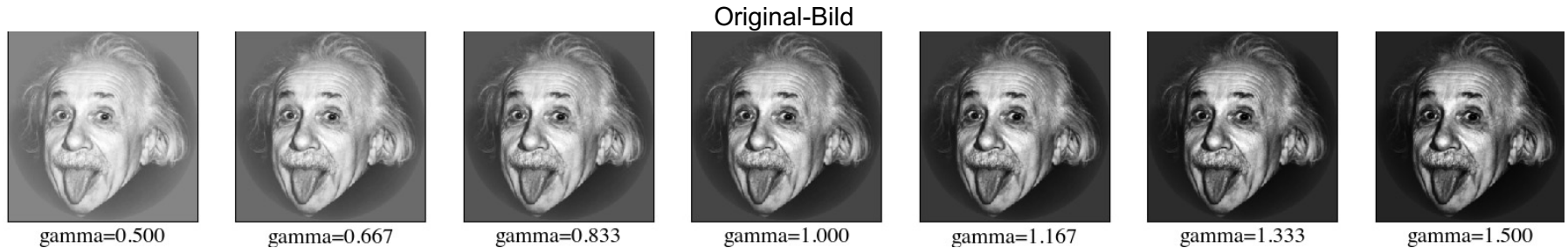
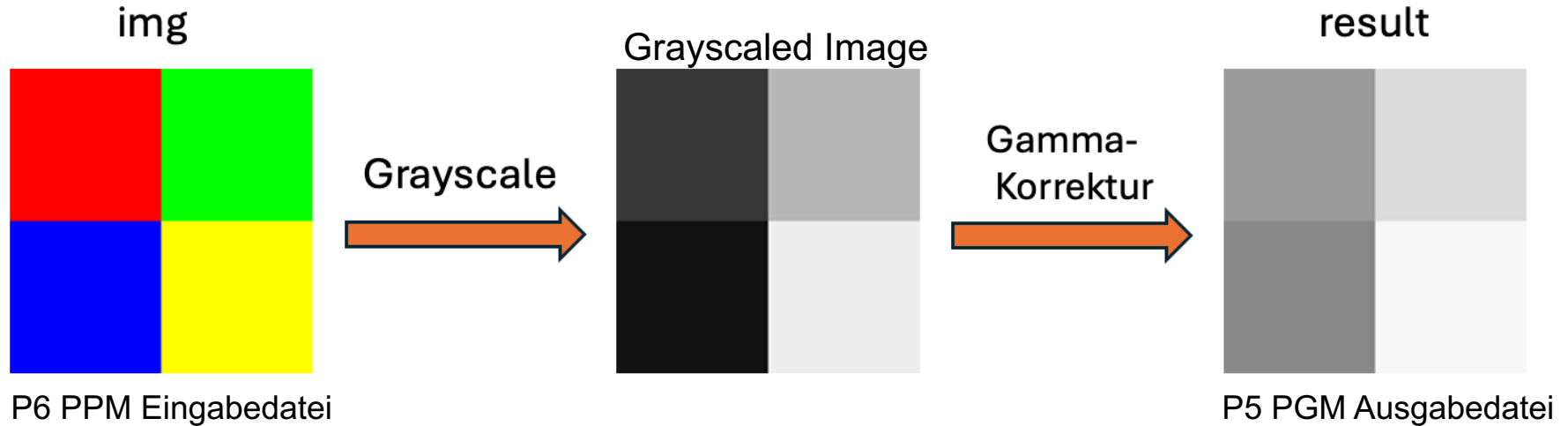
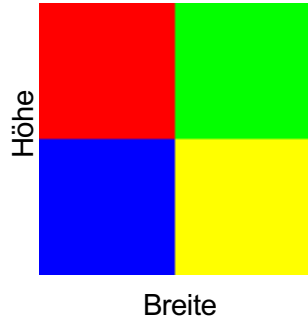


Photo von: <https://laurentperrinet.github.io/sciblog/posts/2019-11-28-role-of-gamma-correction-in-sparse-coding.html>

Funktionsweise des Algorithmus



24bpp PPM (P6) Eingabedatei



input.ppm:

```
87654321 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
00000000: 5036 0a32 2032 0a32 3535 0aff 0000 00ff P6.2 2.255.1.1...
00000010: 0000 00ff ffff 00
.....
```

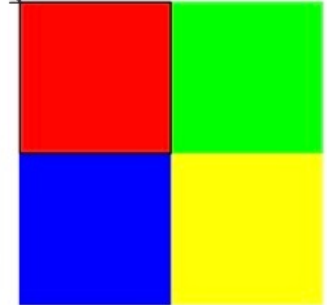
Magischer Wert - "P6"

Breite des Bilds

Höhe des Bilds

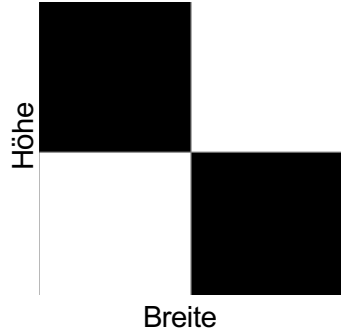
Maximalwert des Farbwertes

New line



Rot Value des Pixels	Grün Value des Pixels	Blau Value des Pixels
FF	00	00
255	0	0

PGM (P5) Ausgabedatei



output.pgm:

```
87654321 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
00000000: 5035 0a32 2032 0a32 3535 0a00 ffff 00 P5.2 2.255.1...
```

Magischer Wert - "P5"

Breite des Bilds

Höhe des Bilds

Maximalwert des Farbwertes

Wert des Pixels

New line



Andere wichtige Komponente

- **Grayscale Koeffizienten:**

Gemäß HVS wissenschaftlicher Literatur:

Genutzte Standard: ITU-R BT.601-7 (SDTV)

a = 0.299

b = 0.587

c = 0.114

Formel der Graustufekonveriterung:

$$D = \frac{a \cdot R + b \cdot G + c \cdot B}{a + b + c}$$

- **Gamma:** $\gamma \in [0, \infty)$
 - Für $\gamma < 1$: Die Ausgabebild wird heller.
 - Für $\gamma > 1$: Die ausgabebild wird dunkler.
 - Für $\gamma = 1$: Die Abbildung ist linear.
 - Standartwert: $\gamma = 2.2$

Formel der Gamma Korrektur:

$$Q_{(x,y)} = D$$

$$Q'_{(x,y)} = \left(\frac{Q_{(x,y)}}{255} \right)^{\gamma} \cdot 255$$



$\gamma = 0.5$



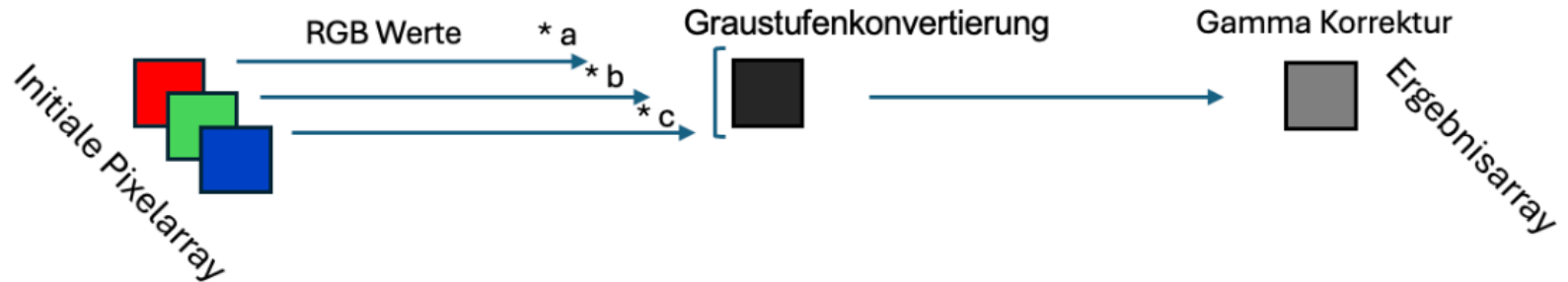
$\gamma = 1.7$



$\gamma = 7.7$

Ansätze

- **gamma_V0.c**
- Berechnung der $Q'_{(x,y)}$ mithilfe von **math.h-Bibliothek**



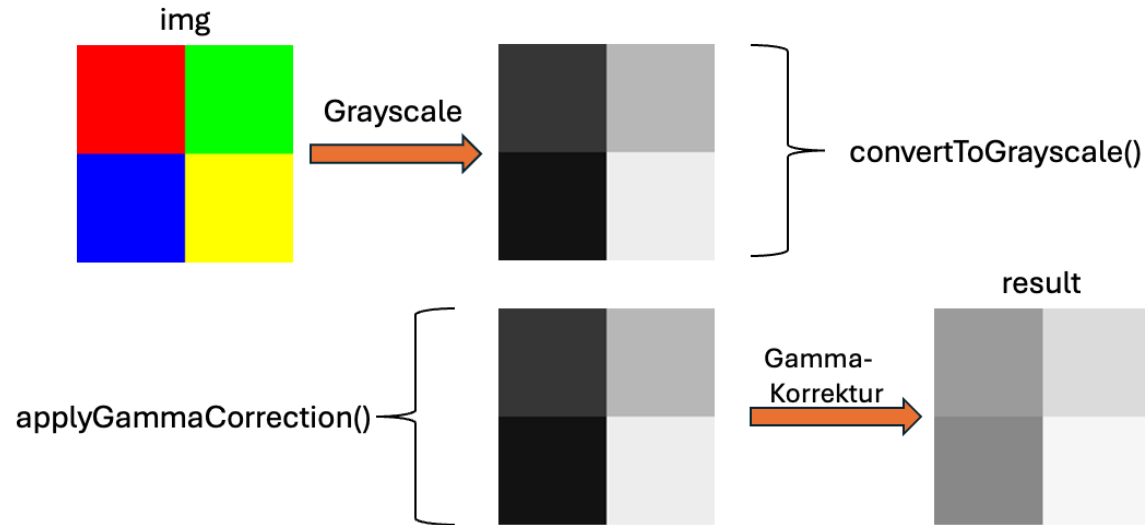
- **gamma_V1.c**
- Berechnung der $Q'_{(x,y)}$ mithilfe von **grundlegende mathematische Operationen**
- Trennung der Exponent in Vorkommastellen (ganzzahliges Teil) und Nachkommastellen (fraktionales Teil)

Für das ganzzahliges Teil: $result \ast = base$

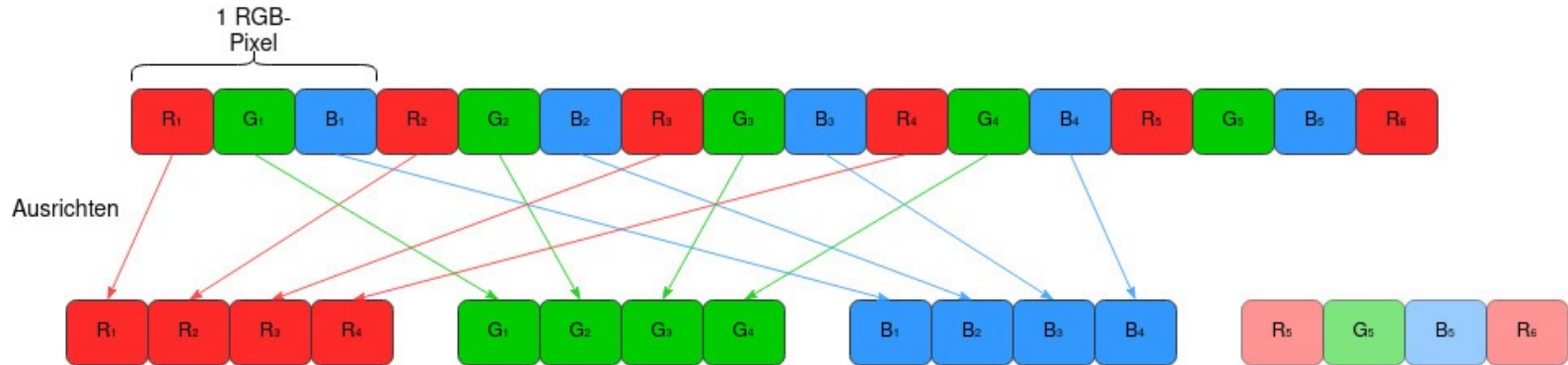
Für das fraktionales Teil:

fractionalPart < 0.5	fractionalPart >= 0.5
$1 + fractionalPart \ast (base - 1)$	$x_{n+1} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right), \quad n \in \mathbb{N},$

- **gamma_V2.c**
- Getrennte Funktionen für Graustufekonvertierung und Gamma Korrektur
- Berechnung der $Q'_{(x,y)}$ mithilfe von $\log()$ und $\exp()$: $a^b = \exp(b \cdot \log(a))$.



- **gamma_V3.c**
- Ziel: Optimierung der RGB-Bildverarbeitung
- Hybrider Ansatz: Kombination von SIMD und skalaren Operationen



- **gamma_V4.c**
- Weiterentwicklung vorheriger Implementierung
- Optimierung: Vollständige SIMD-Berechnungen und Taylor-Reihen-Approximation
- Maximierte parallele Datenverarbeitung
- Minimierte Overheads
- **Tradeoff: Performance vs Korrektheit**

$$\ln(1 + x) \approx x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$$

$$e^x \approx 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \dots$$

Ergebnisse – 1

Die Vergleichung der Ausgabedateien generiert von verschiedenen Implementierungen mit $\gamma = 0.5$



Eingabedatei 1280x816



gamma_V1.c



gamma_V3.c



gamma_V0.c



gamma_V2.c



gamma_V4.c

Ergebnisse - 2

Was passiert wenn die Eingabedatei sehr groß ist und auch hohe Resolution hat?



Eingabedatei 2300x1300

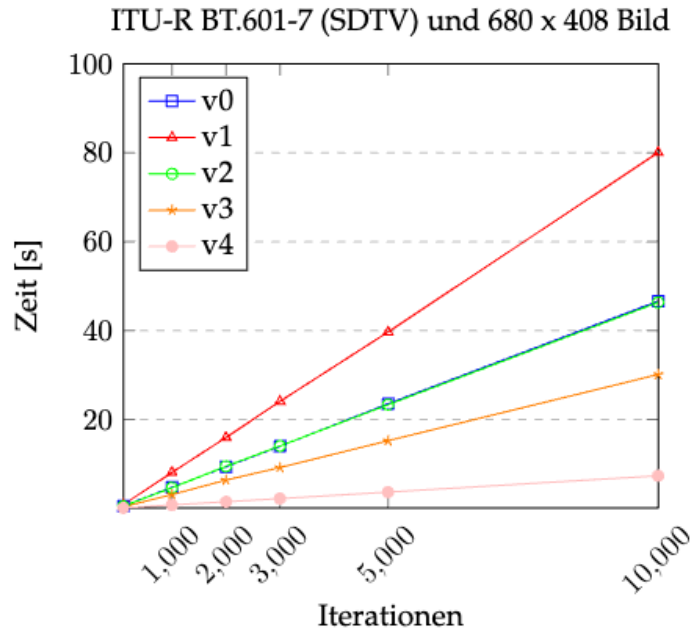


Realisiert mit gamma_V0.c und $\gamma = 0.5$

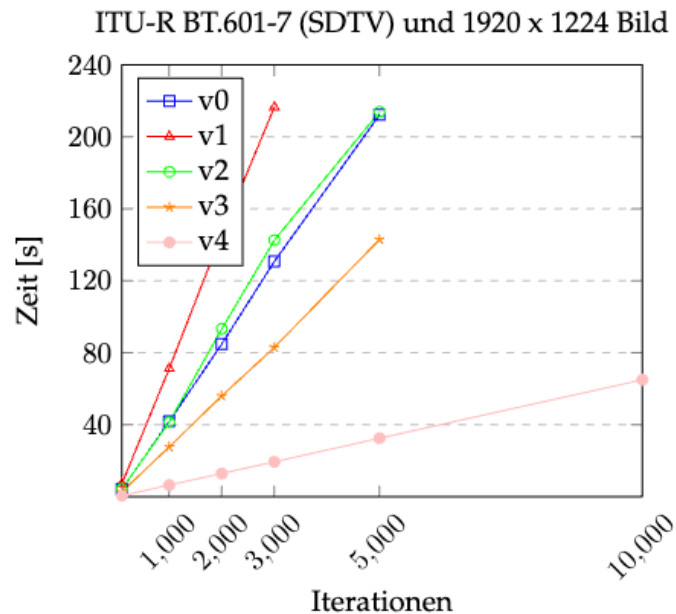
<https://www.welt.de/kmpkt/article176582222/Kein-Zufall-Zu-diesem-bestimmten-Zeitpunkt-finden-wir-Hundewelpen-am-suessesten.html>

Performanzanalyse - 1

Experiment 1:

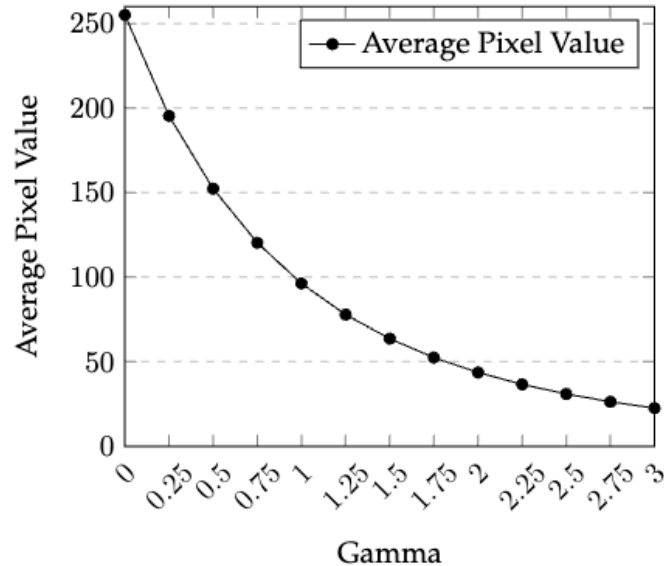


Experiment 2:



Performanzanalyse - 2

Average Pixel Value vs. Gamma 2beeM Bild und ITU-R BT.601-7 (SDTV)



- Das verwendete System: Intel Core i7 mit 3.6 GHz, 16 GB DDR4 und Ubuntu 22.04.3 LTS
- Die Softwareumgebung: C17 und GCC 11.4.0.

Zusammenfassung

- Bild anhand der Graustufenkonvertierung und Gamma Korrektur verändern
- Wechselung zwischen PPM und PGM während der Transformationen
- Fünf verschiedene algorithmische Ansätze mit unterschiedlicher Performanz
- Performanzüberlegenheit von SIMD gegenüber den Basisimplementierungen
- Leistungssteigerung durch SIMD + Parallel Computing