

Projektpräsentation: Bildentrauschung

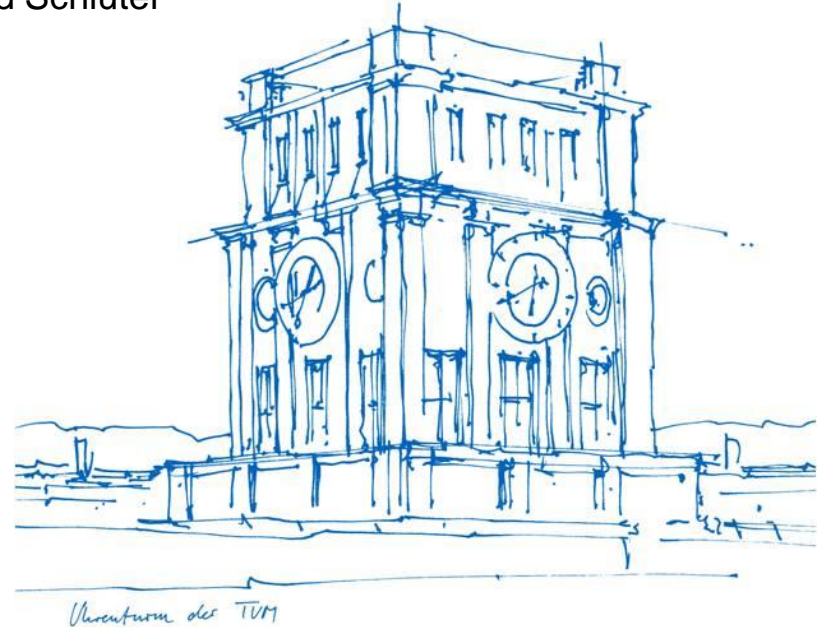
Gruppe 128: Ashkan Hassani, Aosimanjiang Aihaiti, Konrad Schlüter

Grundlagenpraktikum: Rechnerarchitektur

Technische Universität München

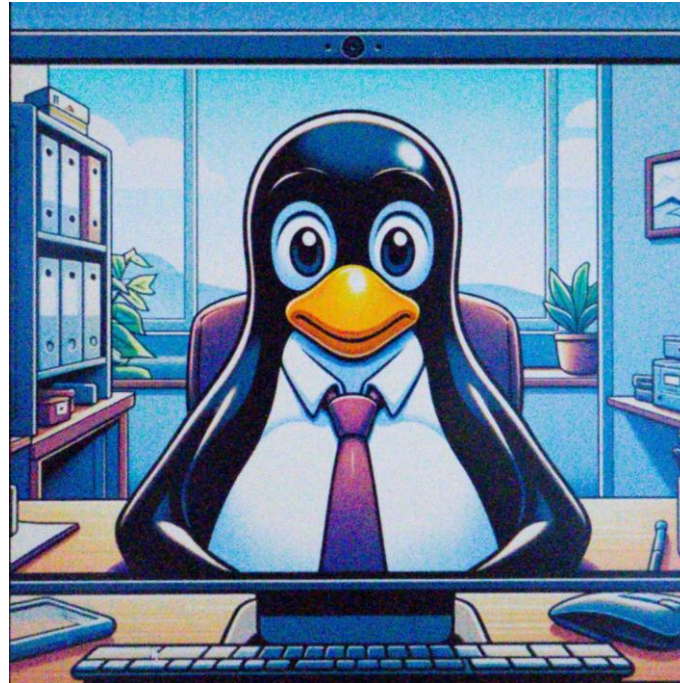
Lehrstuhl für Rechnerarchitektur und Parallele Systeme

Garching, 11. März 2024



Einführung

Das Problem: Bildentrauschung



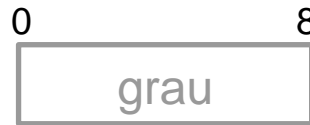
PPM Bilder mit RGB Farbkanälen

- 24bpp ppm Files
- 3 Farben pro Pixel, 8 Bits für jede Farbe (Rot , Grün , Blau)



PGM Bilder mit Graustufenpixeln

- 8bpp pgm Files
- Gewichteten Durchschnitt der drei Farben



Inhaltsübersicht

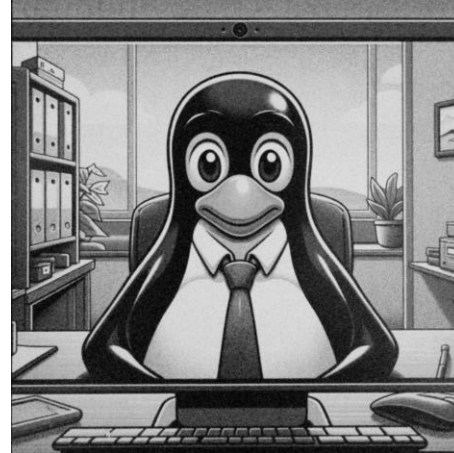
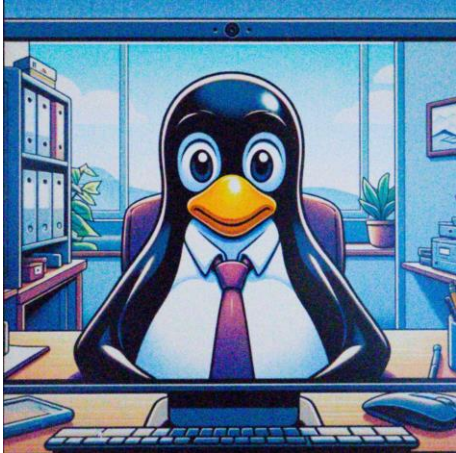
- Einführung
- Lösungsansatz
- Implementierung
- Genauigkeit
- Performanz
- Ausblick

Lösungsansatz

Graustufenkonvertierung

- Mittels Luminanzmethode durch gewichteten Durchschnitt

$$D = \frac{a \cdot R + b \cdot G + c \cdot B}{a + b + c}, \quad a=0.2126 \quad b=0.7152 \quad c=0.0722$$

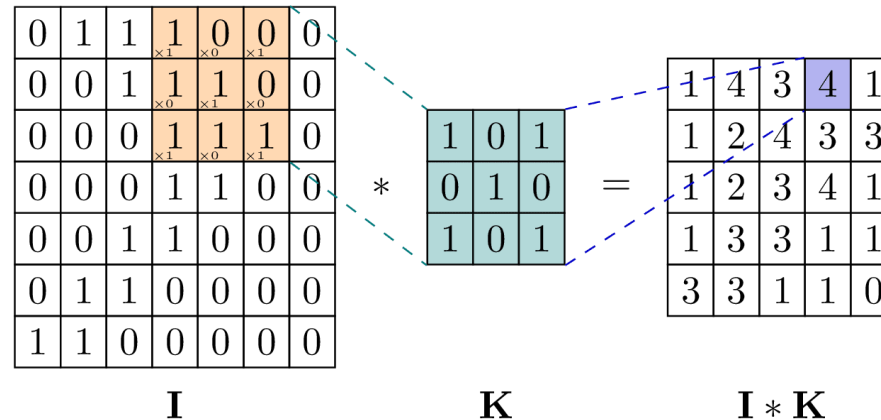


Faltung

Formel für diskrete 2D Faltung:

$$I^*(x, y) = \sum_{i=1}^n \sum_{j=1}^n I(x-i+a, y-j+a) K(i, j)$$

K := Filter I := Bild

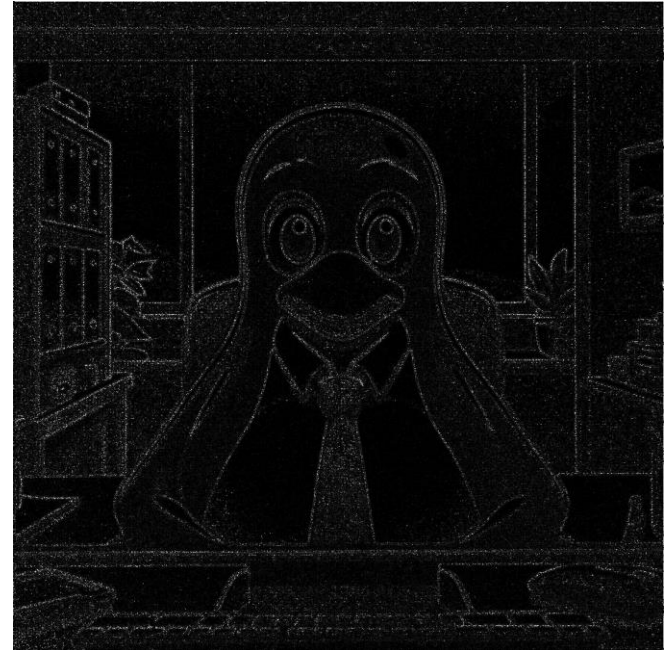


Kantendetektion

- Diskrete Laplace Filter:

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

- Identifiziert Stellen mit schnellen Änderungen in der Intensität eines Bildes
- Betont Kanten durch Hervorhebung von Intensitätssprüngen
- Eine Gewichtung, um Kanten von Hintergrund zu trennen

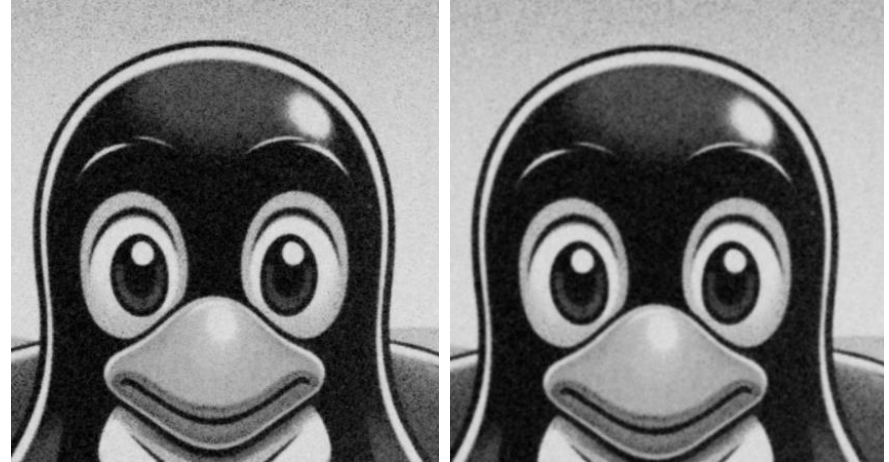


Weichzeichnung

- 2D Gauß'scher Filter:

$$K = \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Berechnet den Durchschnitt von Pixelwerten in der Umgebung
- Reduziert Rauschen im Bild



Zusammensetzung

- Zusammensetzen des Graustufebildes I mit Ergebnissen von Kantenerkennung I^L und Weichzeichnung I^W :

$$I' = \frac{|I^L|}{1020} \cdot I + \left(1 - \frac{|I^L|}{1020}\right) \cdot I^W$$

- An Kanten: stärkere Gewichtung des schärferen Ursprungsbildes
- In glatten Bereichen: Weichzeichnung stärker gewichtet
- Resultat: Ausgewogene Rauschreduzierung, Erhaltung wichtiger Details wie Kanten

Implementierung

Drei Implementierungen:

Drei Implementierungen:

„Accurate“

- Naive Implementierung
- Höchste Genauigkeit
- SISD

Drei Implementierungen:

„Accurate“

- Naive Implementierung
- Höchste Genauigkeit
- SISD

„Integer“

- Erste Optimierungen, vor allem durch **Ganzzahlarithmetik**
- Verringerte Genauigkeit
- Zwei Faltungen in einem Durchlauf
- SISD

Drei Implementierungen:

„Accurate“

- Naive Implementierung
- Höchste Genauigkeit
- SISD

„Integer“

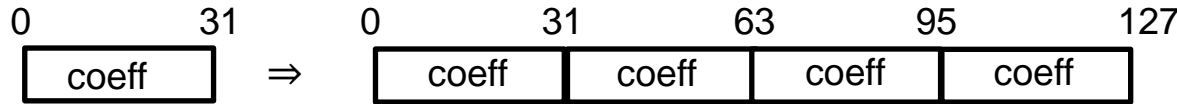
- Erste Optimierungen, vor allem durch **Ganzzahlarithmetik**
- Verringerte Genauigkeit
- Zwei Faltungen in einem Durchlauf
- SISD

„SIMD“

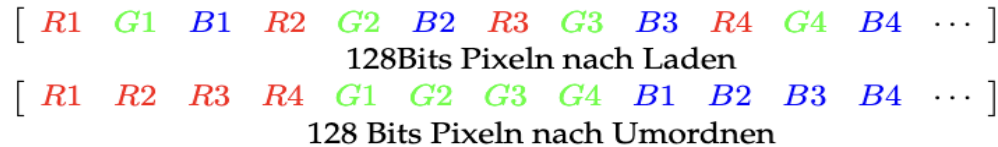
- Stärkste Optimierung
- Verringerte Genauigkeit
- SIMD

Graustufenkonvertierung in SIMD

- Laden und Replizieren der Koeffizienten a, b und c auf jeweils eigene 128 bit Registers



- Laden 128Bits Pixel
- Umordnung der geladenen Pixel



- Konvertierung jeder Farbe von 8Bits zu 32Bits und Laden Werte einer Farbe in ein eigenes Register
- Multiplikation jedes Pixelregisters mit zugehörigem Koeffizienten-Register
- Speichern des Ergebnisses

Faltung in SIMD

- Problem:
 - Pixel in 1D Array => Pixel am Anfang der nächsten Zeile könnten mitgeladen werden
 - Zugriffe außerhalb des Bildes müssen korrekt behandelt werden
- Lösung: Padding

10	20	30	40	50
50	100	200	250	255
0	30	60	90	120
150	180	210	240	255

0	0	0	0	0	0	0	0
0	10	20	30	40	50	0	0
0	50	100	200	250	255	0	0
0	0	30	60	90	120	0	0
0	150	180	210	240	255	0	0
0	0	0	0	0	0	0	0

x0y0	0	0	0	0	0	0	0
x0y1	0	10	20	30	40	50	0
x0y2	0	50	100	200	250	255	0

x1y0	0	0	0	0	0	0	0
x1y1	10	20	30	40	50	0	0
x1y2	50	100	200	250	255	0	0

x2y0	0	0	0	0	0	0	10
x2y1	20	30	40	50	0	0	50
x2y2	100	200	250	255	0	0	0

x0y0	0	0	0	0	0	0	0
x0y1	0	20	40	60	80	100	0
x0y2	0	50	100	200	250	255	0

x1y0	0	0	0	0	0	0	20
x1y1	40	80	120	160	200	0	200
x1y2	100	200	400	500	510	0	0

x2y0	0	0	0	0	0	0	10
x2y1	40	60	80	100	0	0	100
x2y2	100	200	250	255	0	0	30
Sum	280	610	990	1275	1040	355	110

280	610	990	1275	1040	355	110	470
-----	-----	-----	------	------	-----	-----	-----

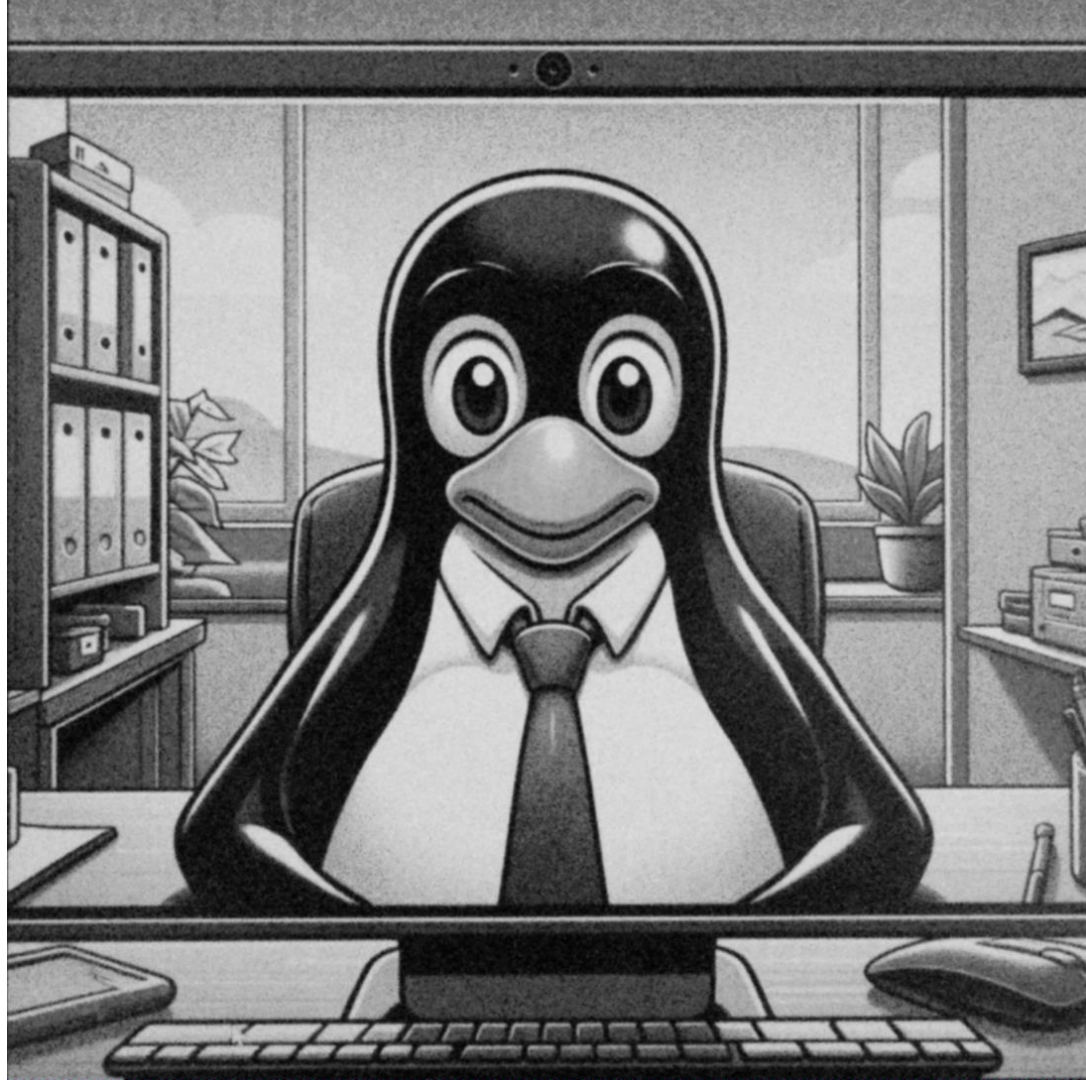
17	38	61	79	65	22	6	29
----	----	----	----	----	----	---	----

0	0	0	0	0	0	0
0	17	38	61	79	65	22
6	29	68	116	151	124	...
...	46	88	129	163	135	...
...	61	97	120	140	114	0
0	0	0	0	0	0	0

17	38	61	79	65
29	68	116	151	124
46	88	129	163	135
61	97	120	140	114

Genauigkeit

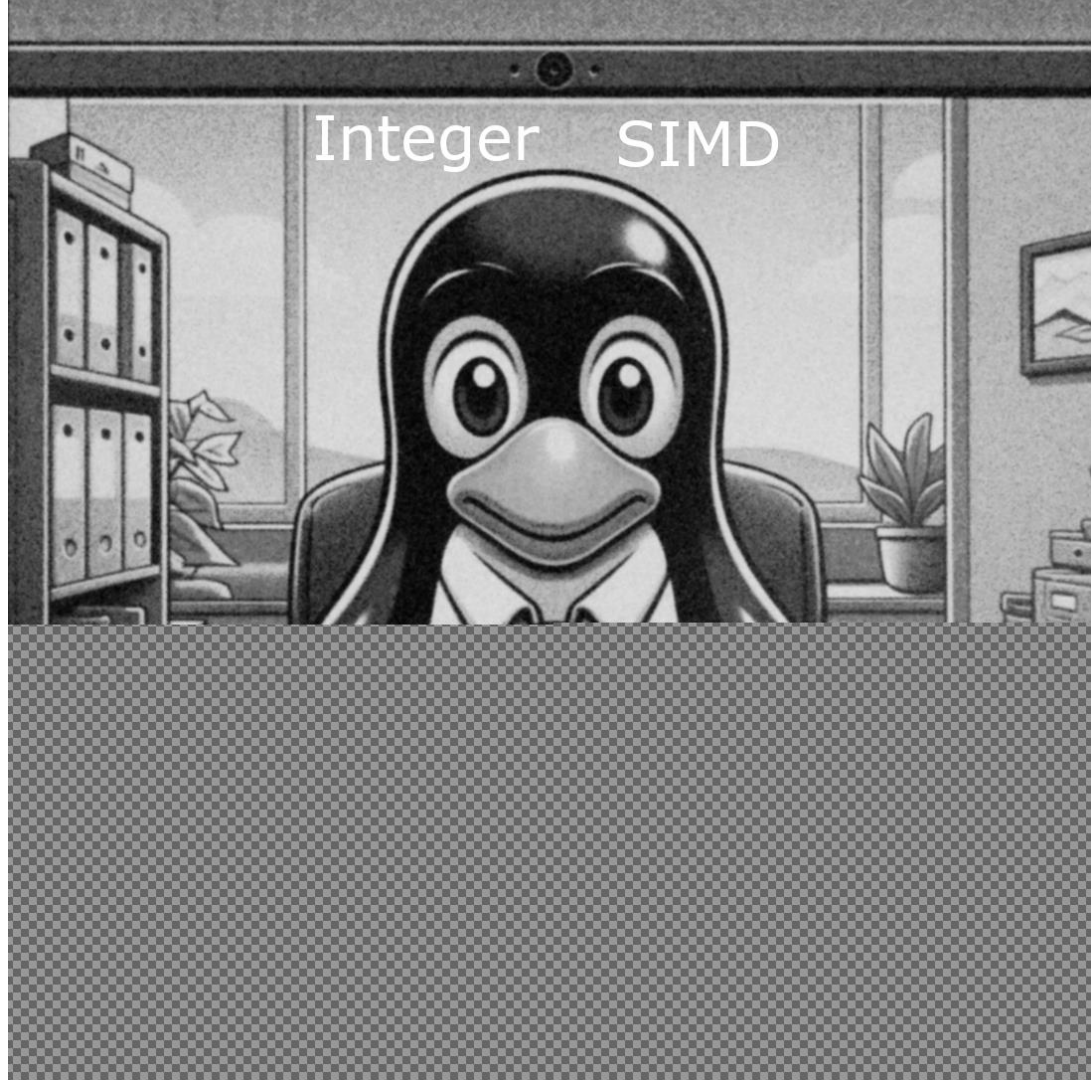
Auswirkungen der Ungenauigkeiten in Integer und SIMD



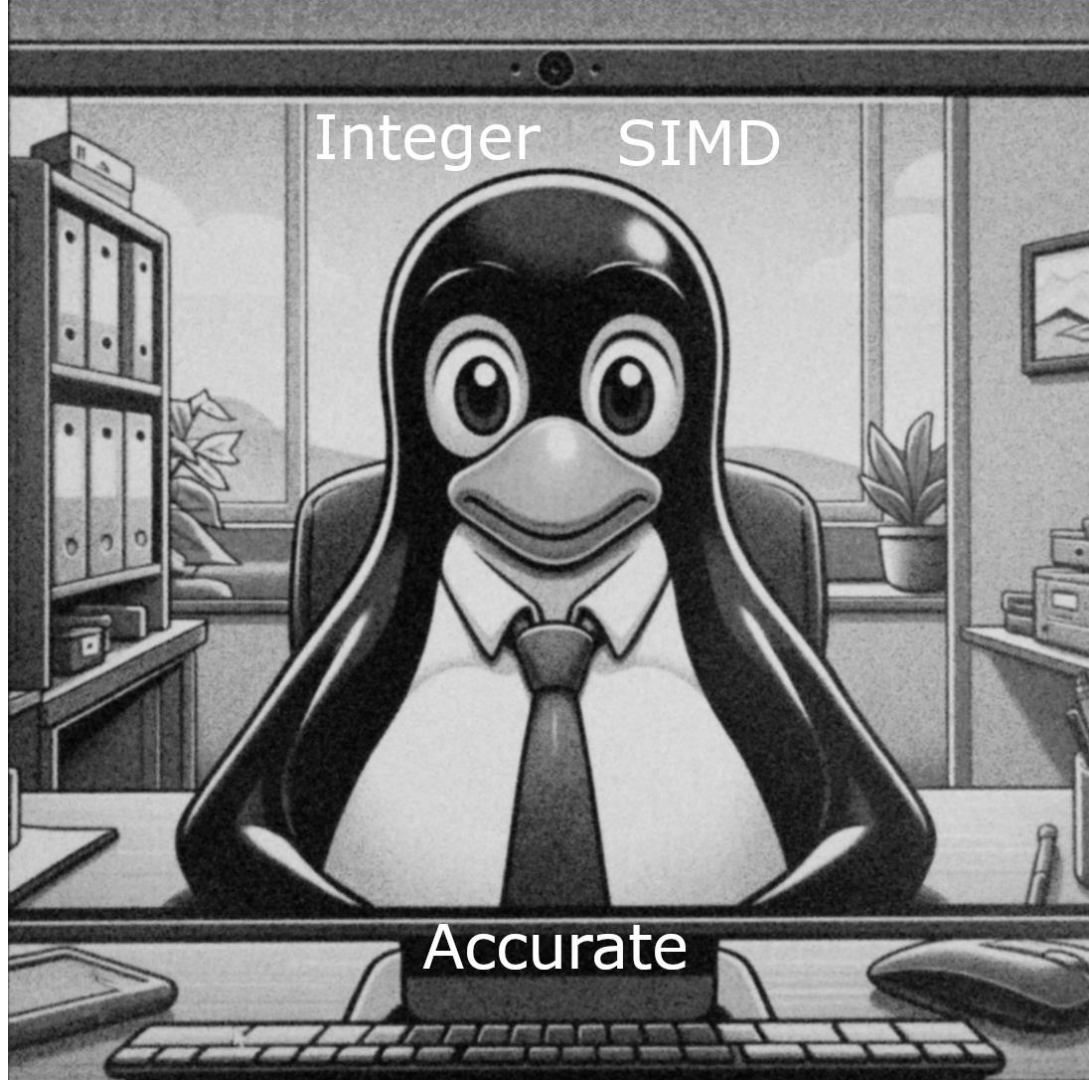
Auswirkungen der Ungenauigkeiten in Integer und SIMD



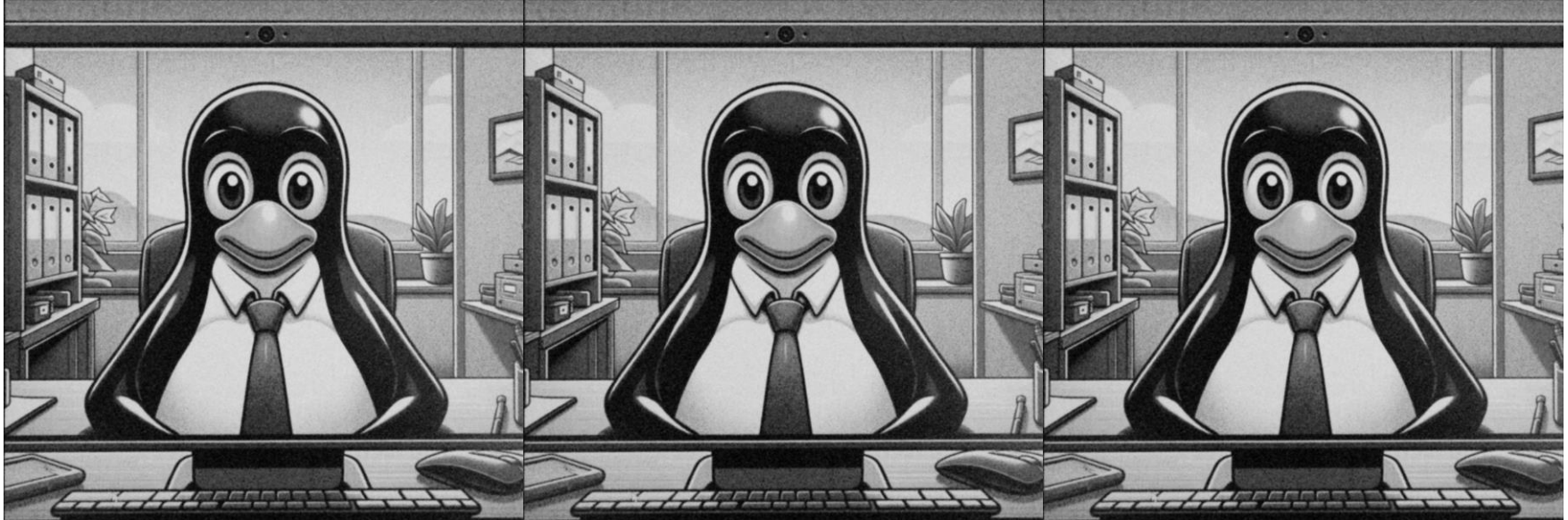
Auswirkungen der Ungenauigkeiten in Integer und SIMD



Auswirkungen der Ungenauigkeiten in Integer und SIMD



Auswirkungen der Ungenauigkeiten in Integer und SIMD

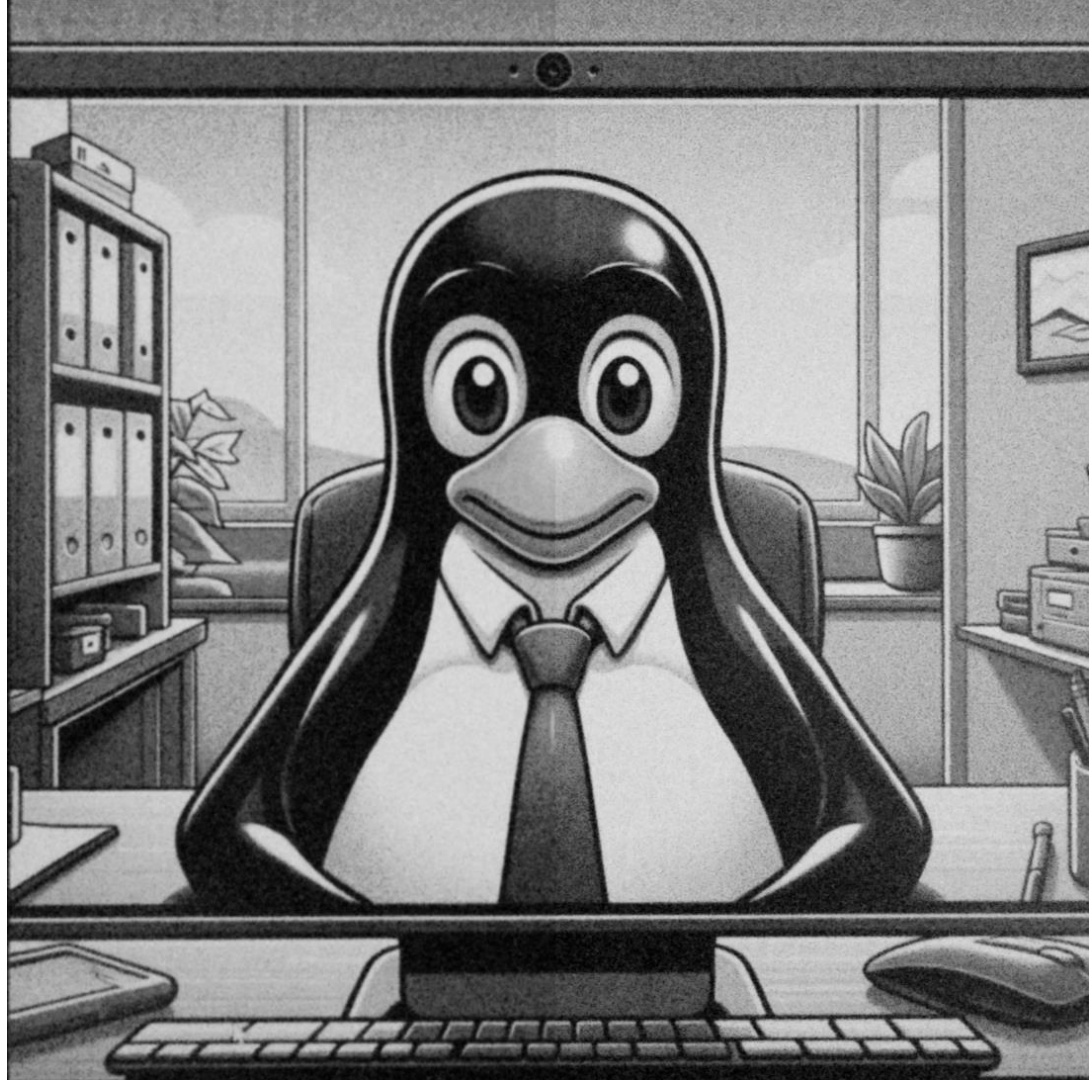


Accurate

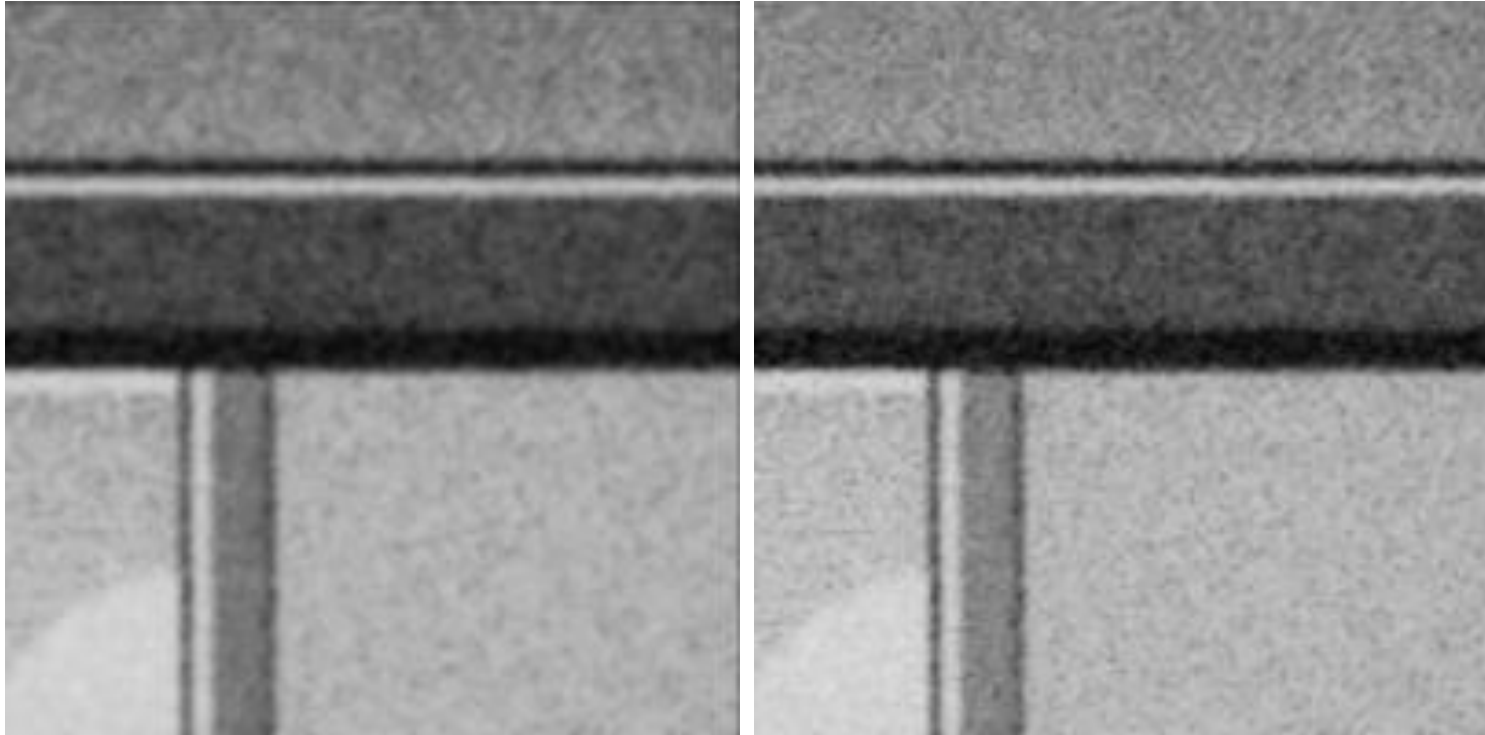
Integer

SIMD

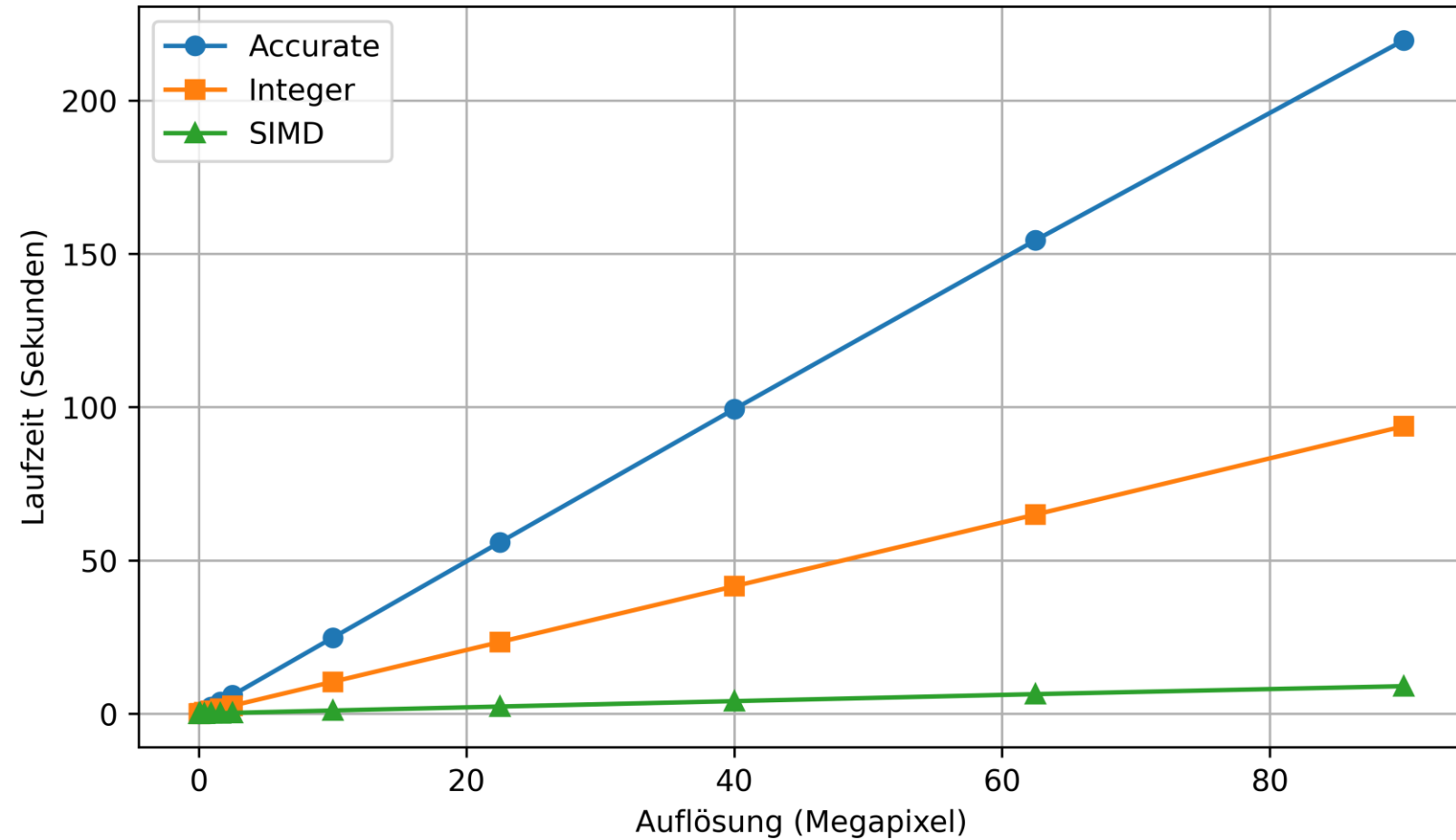
Ergebnis

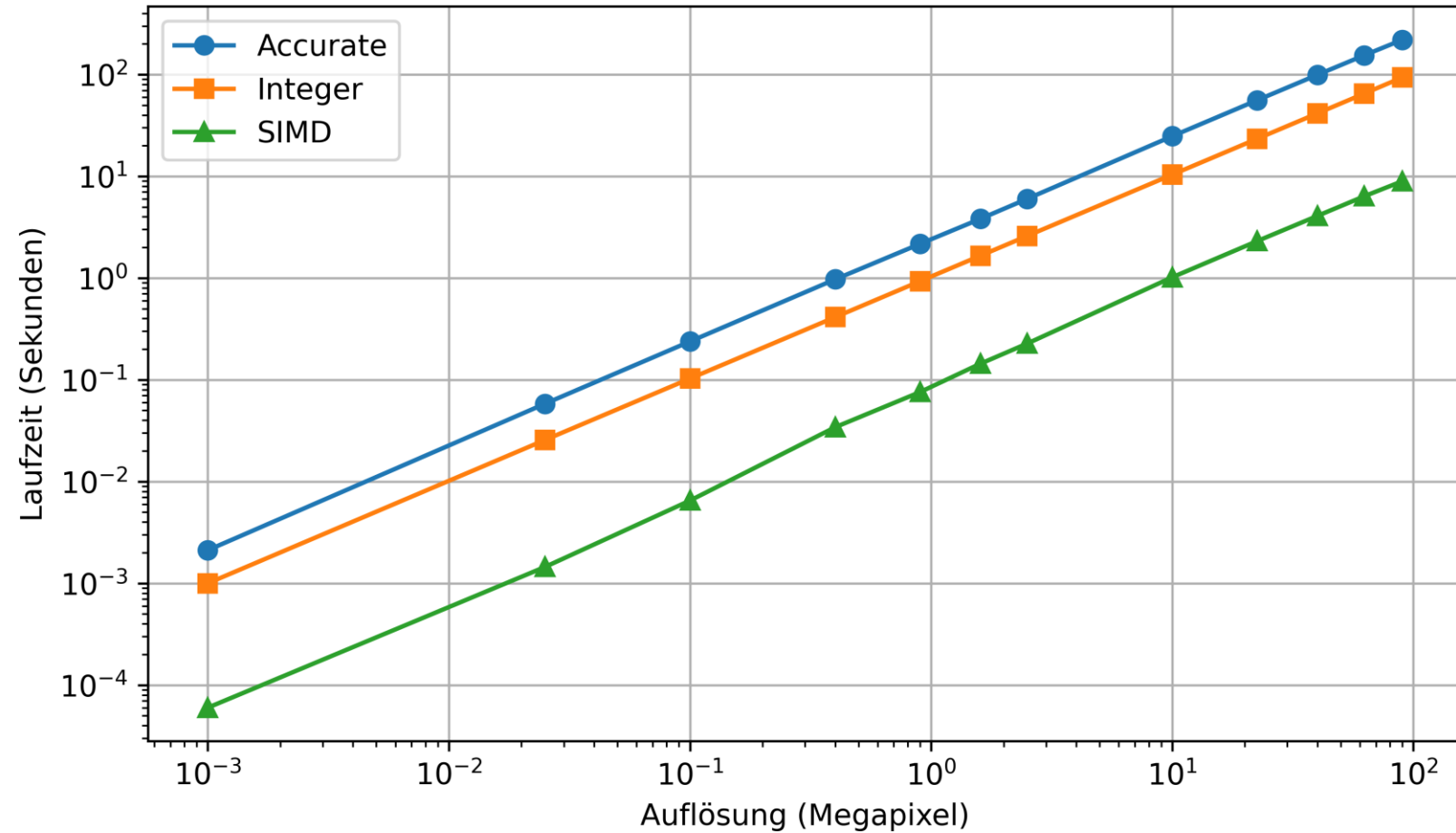


Ergebnis



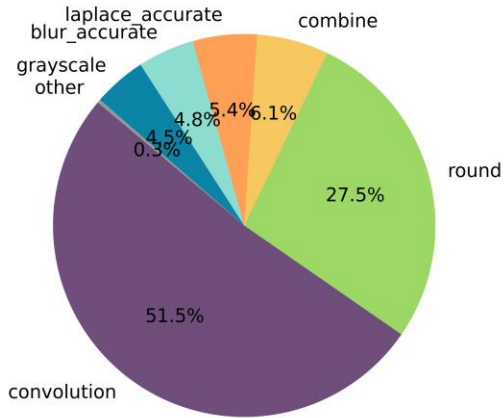
Performanz



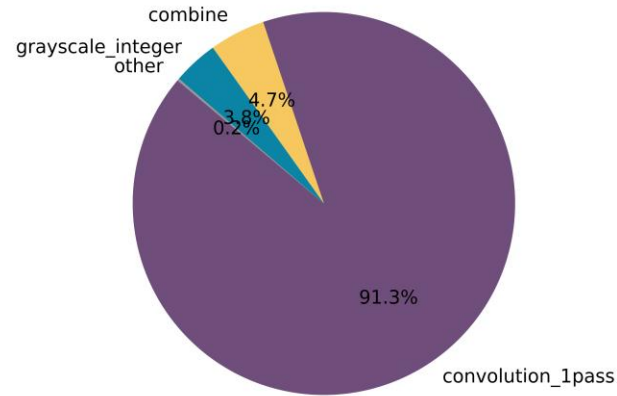


Profiling

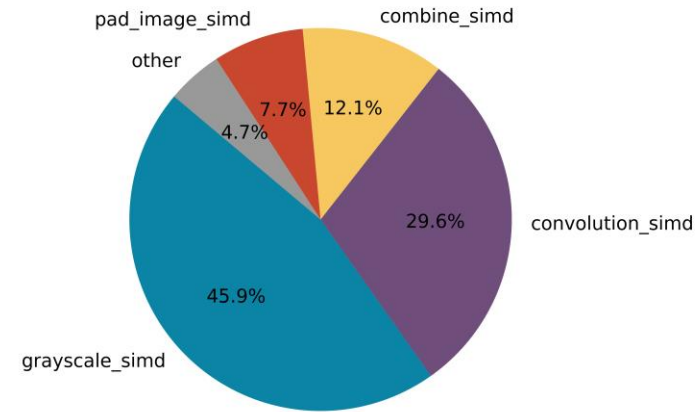
Accurate



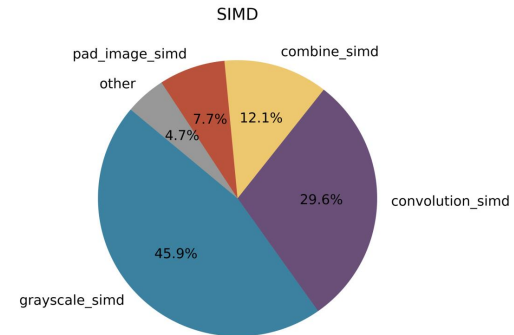
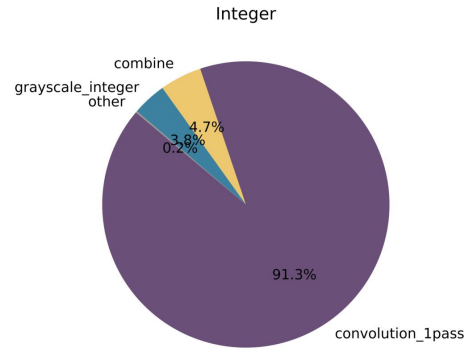
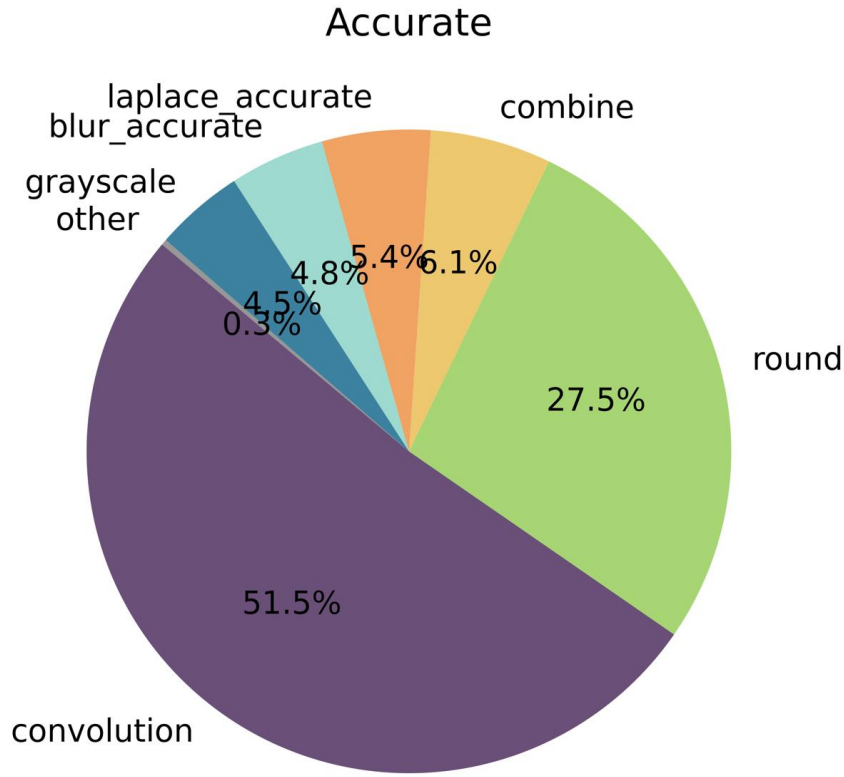
Integer



SIMD

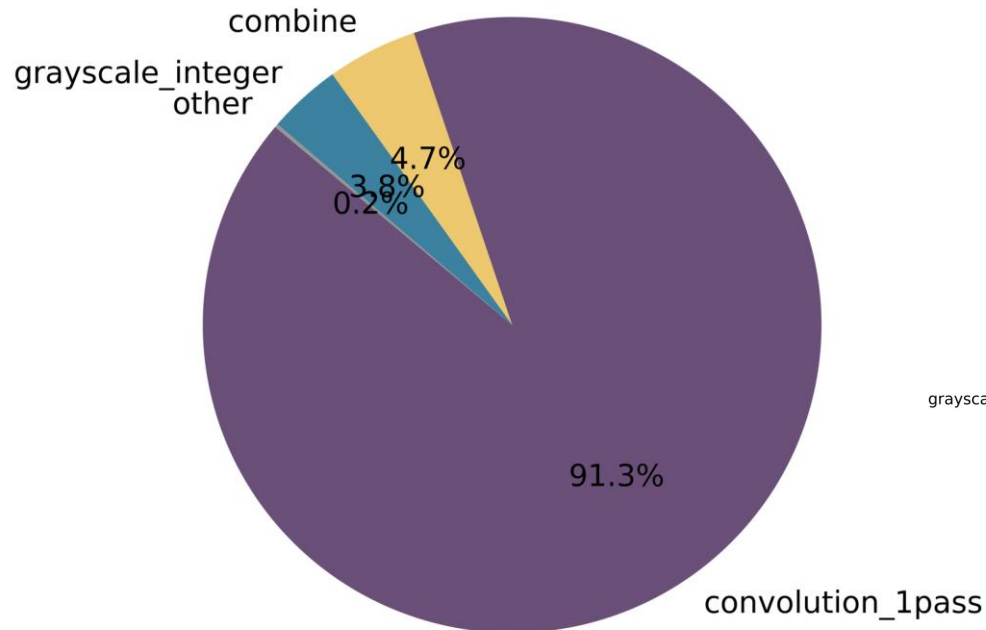


Profiling

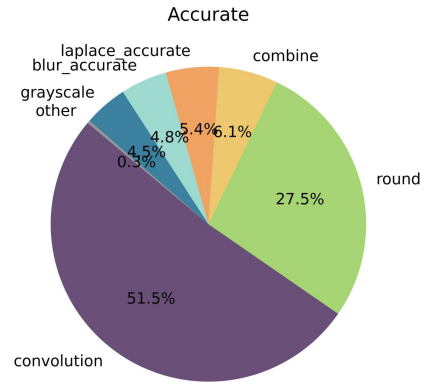
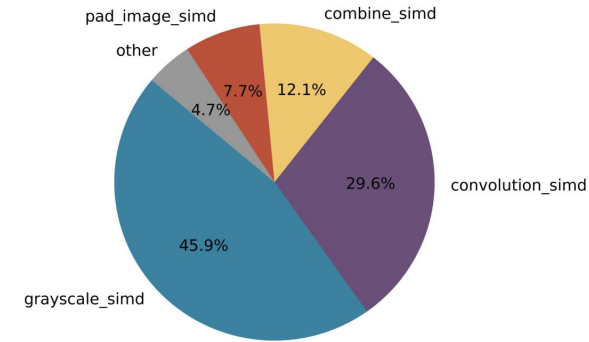


Profiling

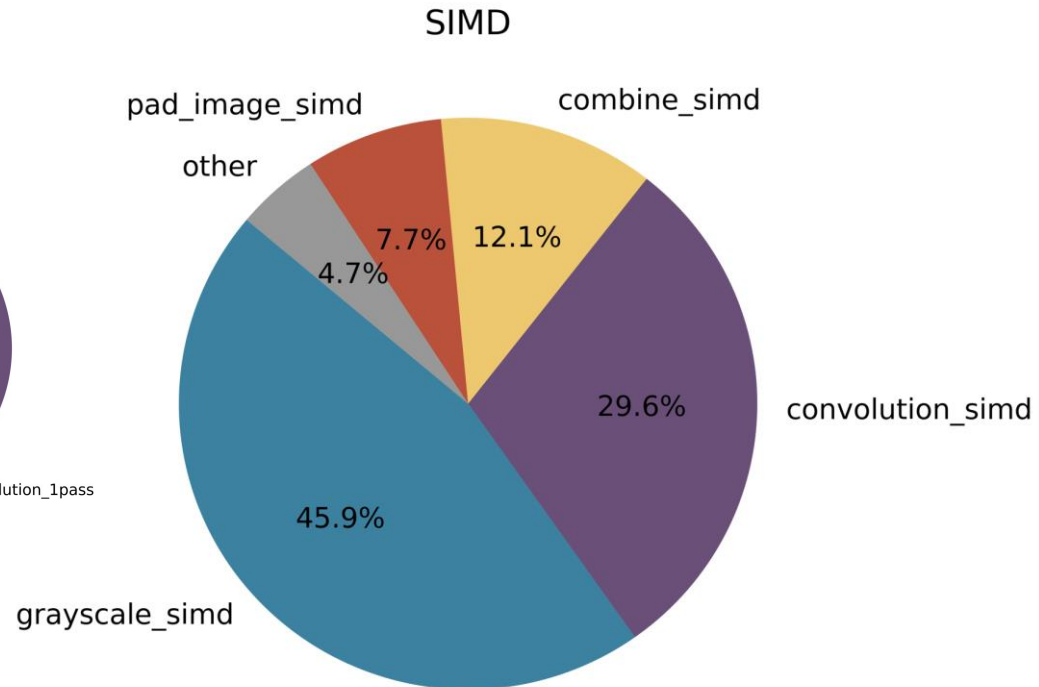
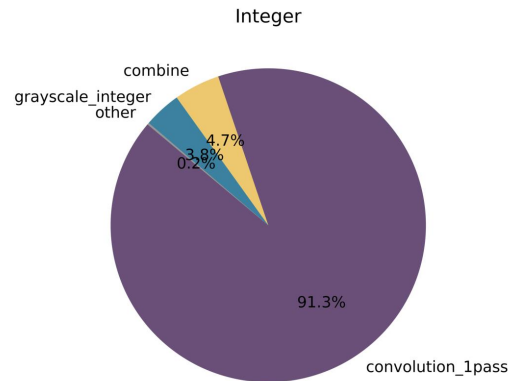
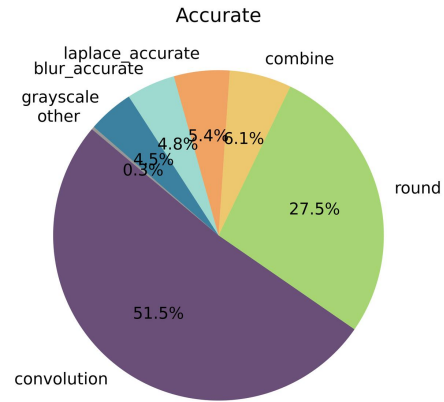
Integer



SIMD

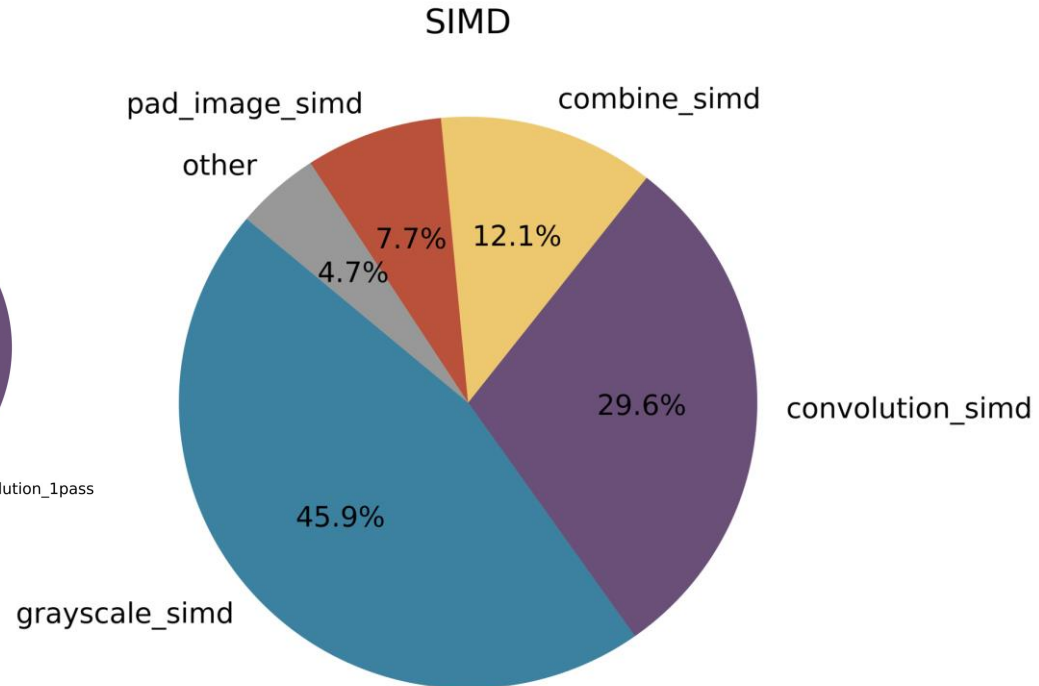
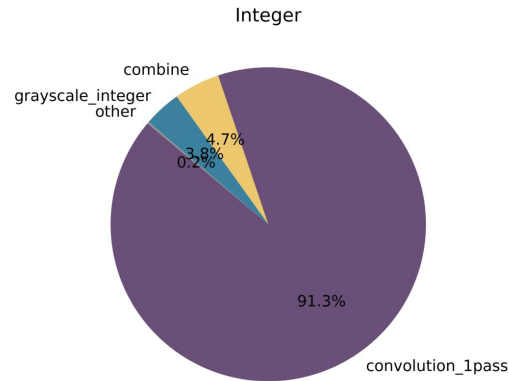
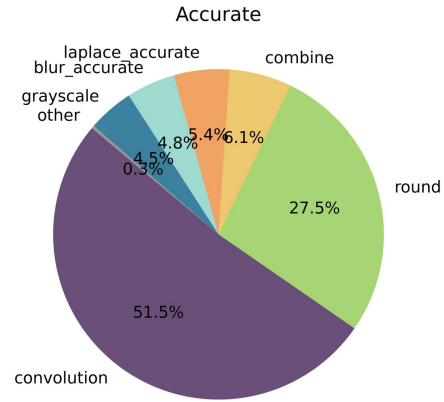


Profiling



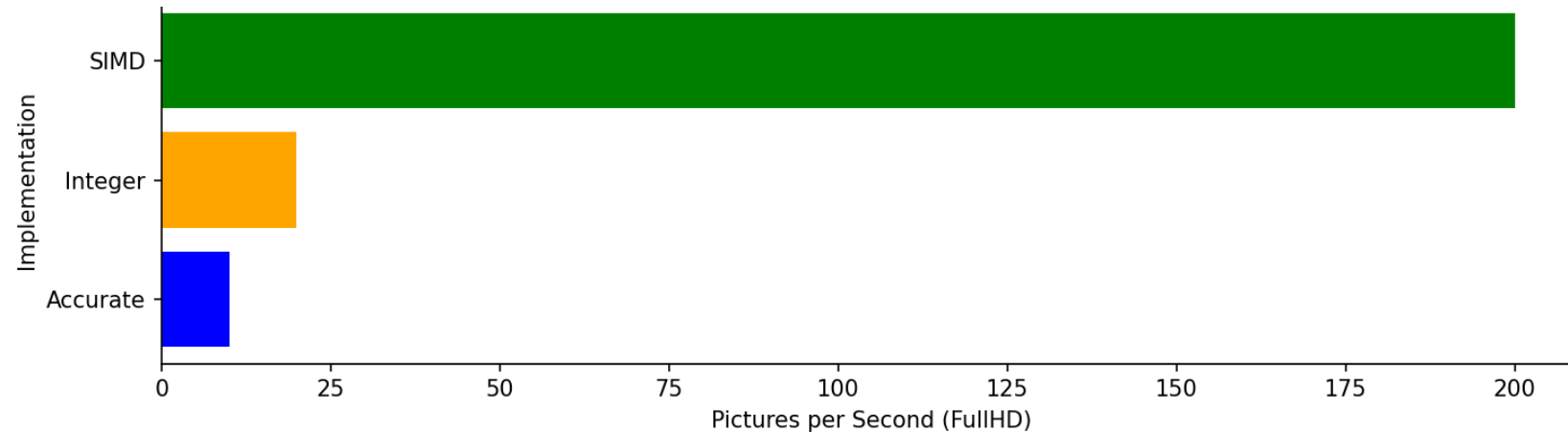
Profiling

$$\begin{bmatrix} R1 & G1 & B1 & R2 & G2 & B2 & R3 & G3 & B3 & R4 & G4 & B4 & \dots \end{bmatrix}$$

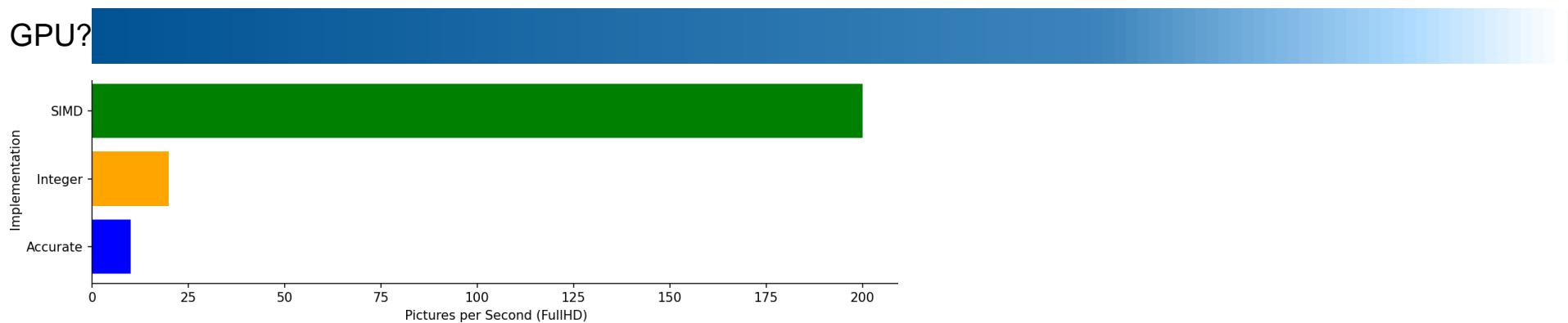
$$\begin{bmatrix} R1 & R2 & R3 & R4 & G1 & G2 & G3 & G4 & B1 & B2 & B3 & B4 & \dots \end{bmatrix}$$


Ausblick

Wie sinnvoll ist die Optimierung?



Wie sinnvoll ist die Optimierung?



Der Pinguin dankt für ihre Aufmerksamkeit

