

Grundlagenpraktikum: Rechnerarchitektur

Simon Bußmann, Nico Lintner, Manuel Walter Mußbacher

Technische Universität München

Gliederung

Problemstellung

- ▶ Kantenerkennung
- ▶ Kantenerkennung mit Sobel

Performanzanalyse

Ausblick

Quellen

Problemstellung

Wie erkennt man Kanten in einem Bild?

- ▶ Was ist eine Kante?
- ▶ Laplace-Filter, Scharr-Operator, Prewitt-Operator, Sobel-Operator, ...

Repräsentation von BMP-Bildern

Farbbilder

- ▶ Speichern die Intensität der 3 Grundfarben in einem Pixel

Graustufenbilder

- ▶ Speichern nur die Helligkeit des Pixels

Kantenerkennung mit dem Sobel-Operator

Faltungsmatrizen

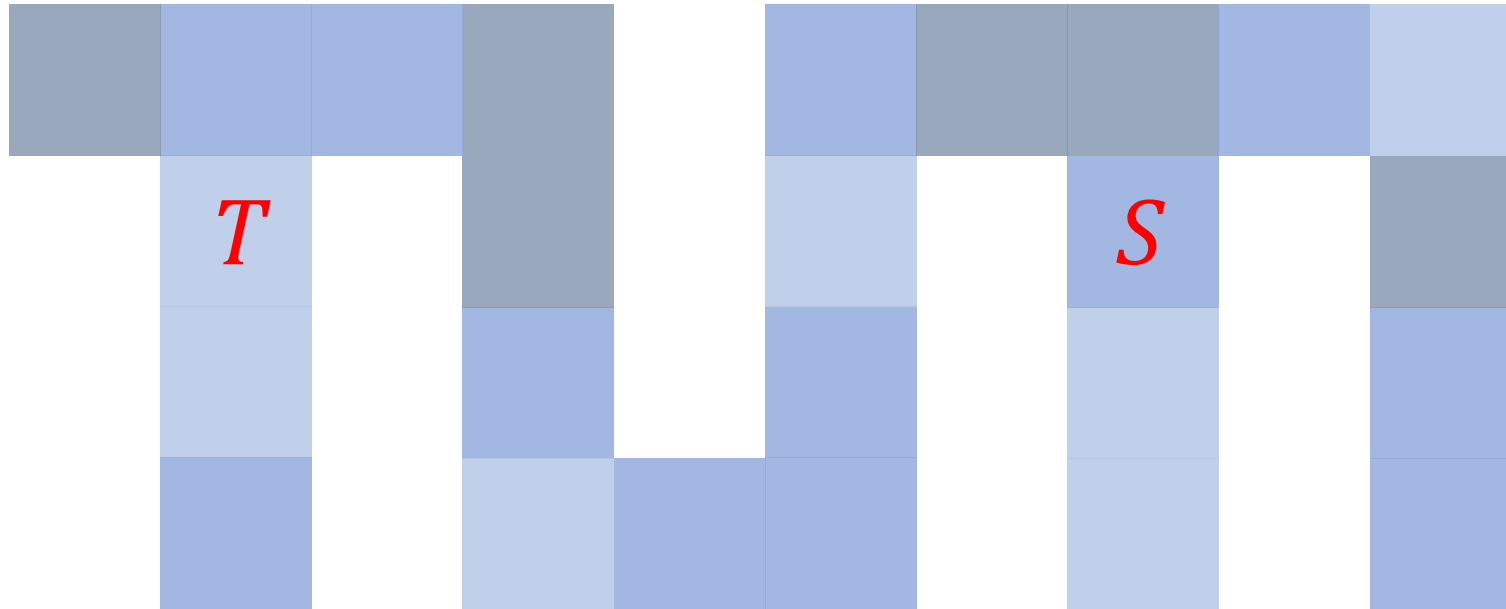
Sobel basiert auf den beiden Faltungsmatrizen M^v und M^h :

$$M^v : \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad M^h : \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

- Werden zur Berechnung der vertikalen und horizontalen Kanten verwendet

Kantenerkennung mit dem Sobel-Operator

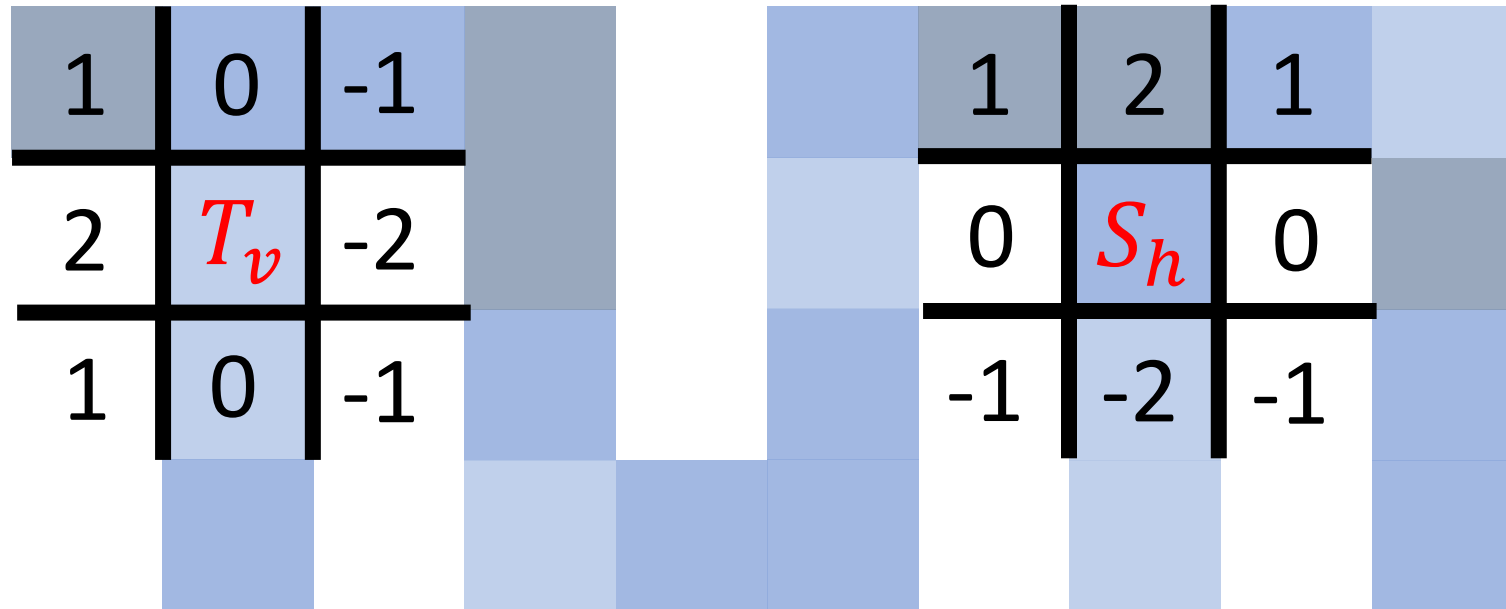
Anwendung am Beispiel



- Die Werte von *T* und *S* sind gesucht

Kantenerkennung mit dem Sobel-Operator

Anwendung am Beispiel



- Veranschaulichung:
Berechnung mit M^v links und M^h rechts

Kantenerkennung mit dem Sobel-Operator

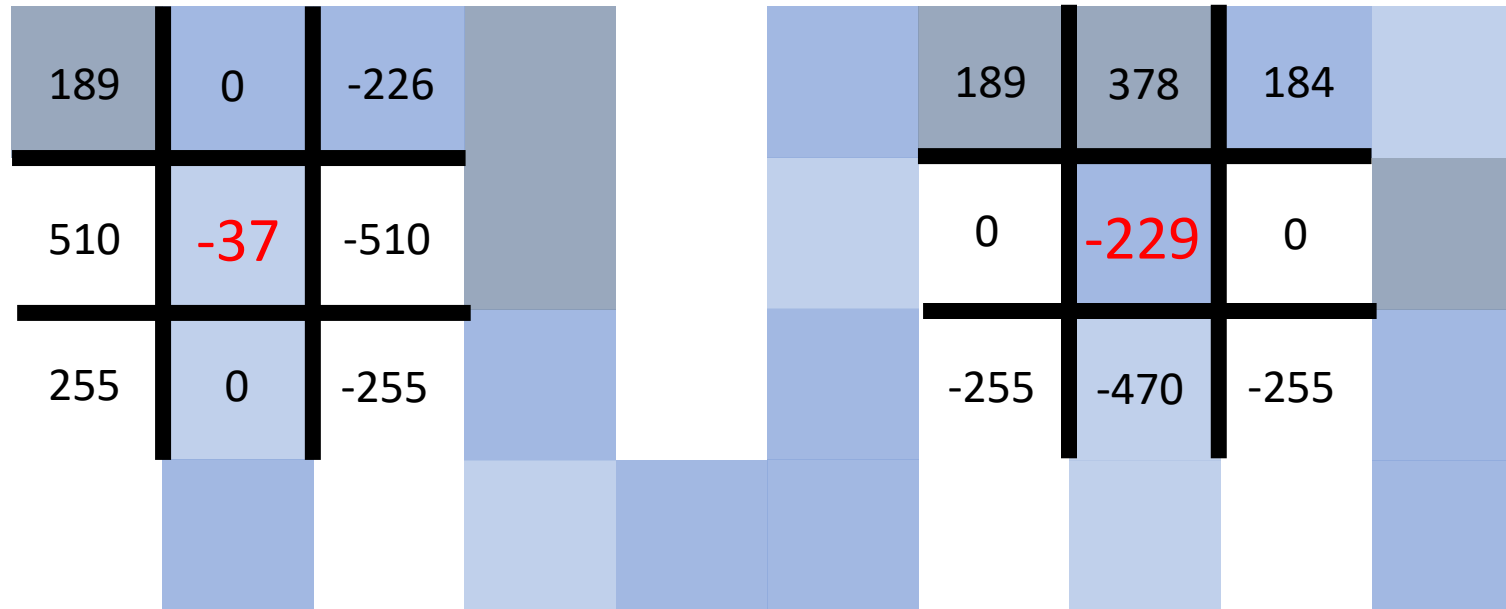
Anwendung am Beispiel

189	0	-226				189	378	184	
510	T_v	-510				0	S_h	0	
255	0	-255				-255	-470	-255	

- Wert umliegender Pixel mit darüberliegender Matrix multiplizieren

Kantenerkennung mit dem Sobel-Operator

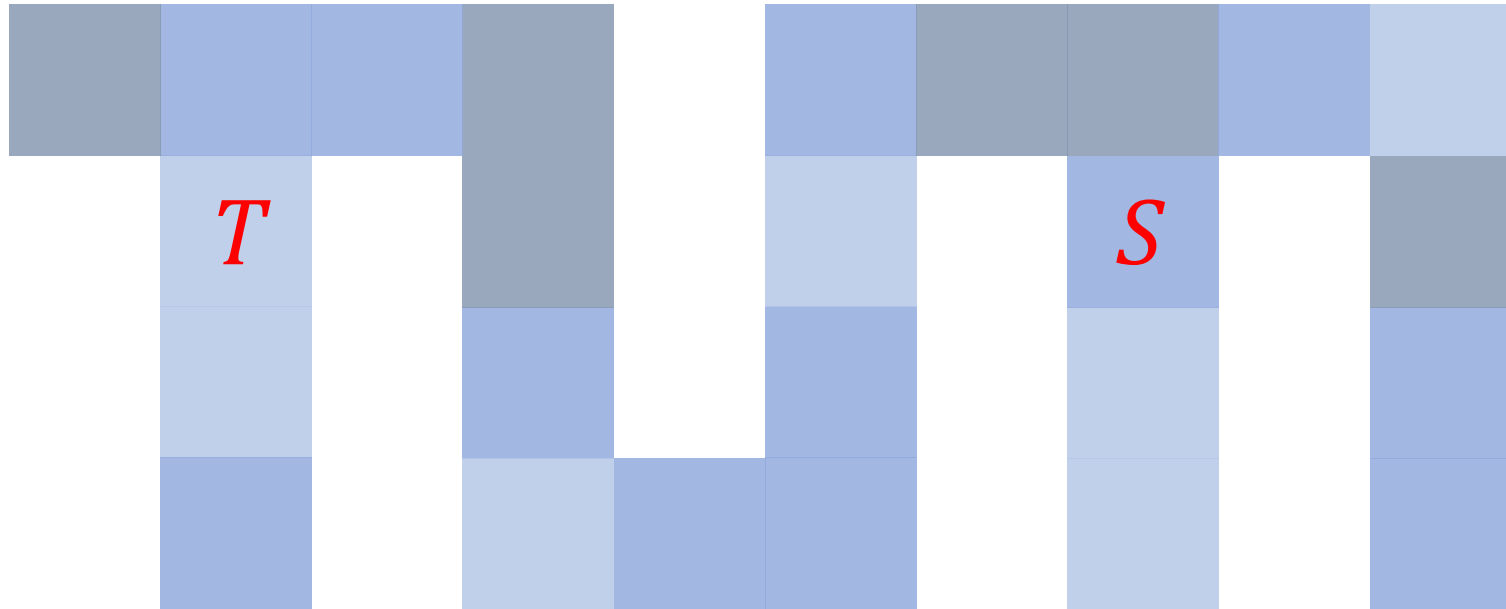
Anwendung am Beispiel



- Resultierende Produkte aufsummieren

Kantenerkennung mit dem Sobel-Operator

Anwendung am Beispiel

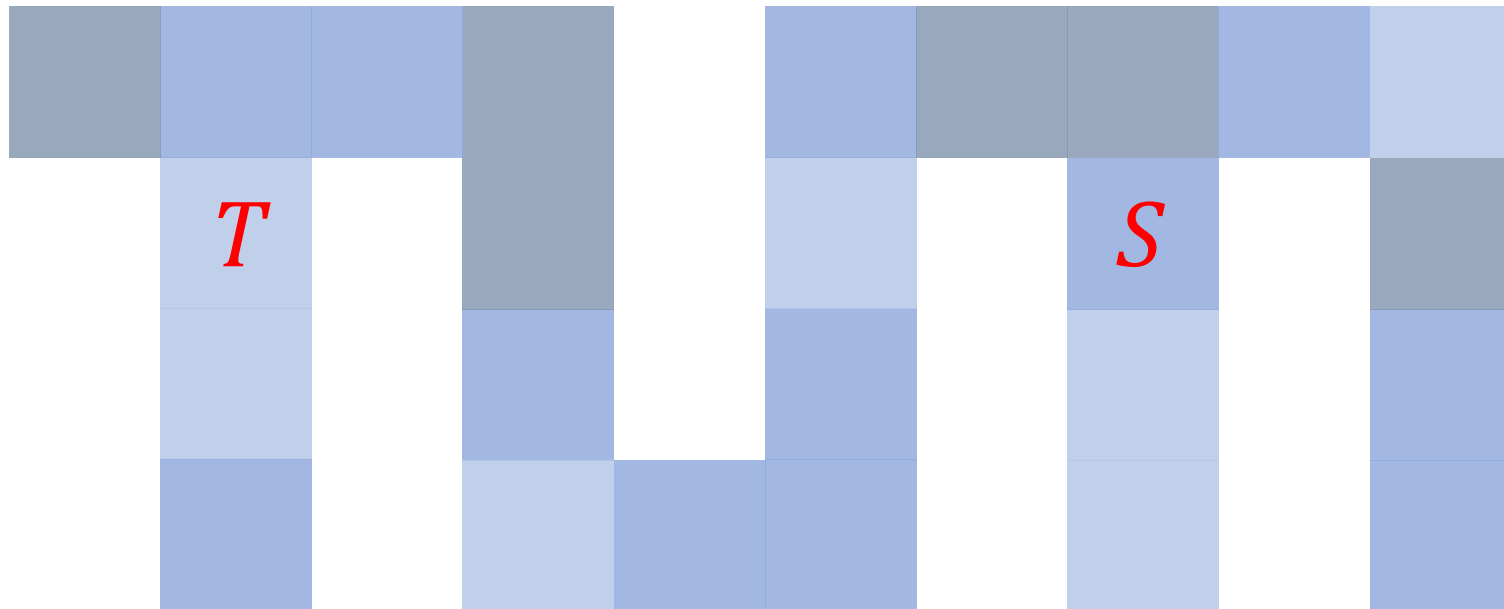


- Wert von T mit $T_v = -37$ und $T_h = -113$

$$T = \sqrt{T_v^2 + T_h^2} = \sqrt{(-37)^2 + (-113)^2} = 118$$

Kantenerkennung mit dem Sobel-Operator

Anwendung am Beispiel

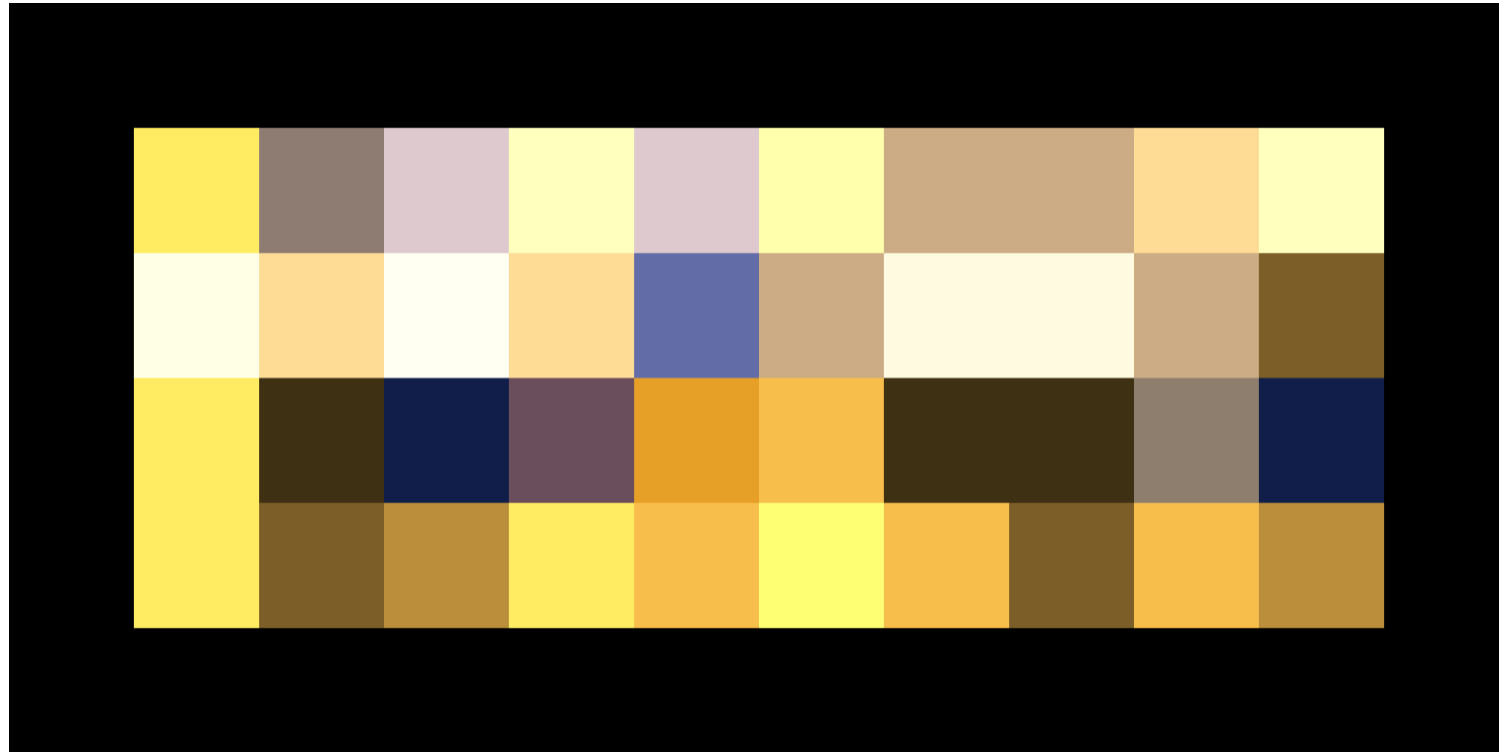


- Wert von S mit $S_v = -37$ und $S_h = -229$

$$S = \sqrt{S_v^2 + S_h^2} = \sqrt{(-37)^2 + (-229)^2} = 231$$

Kantenerkennung mit dem Sobel-Operator

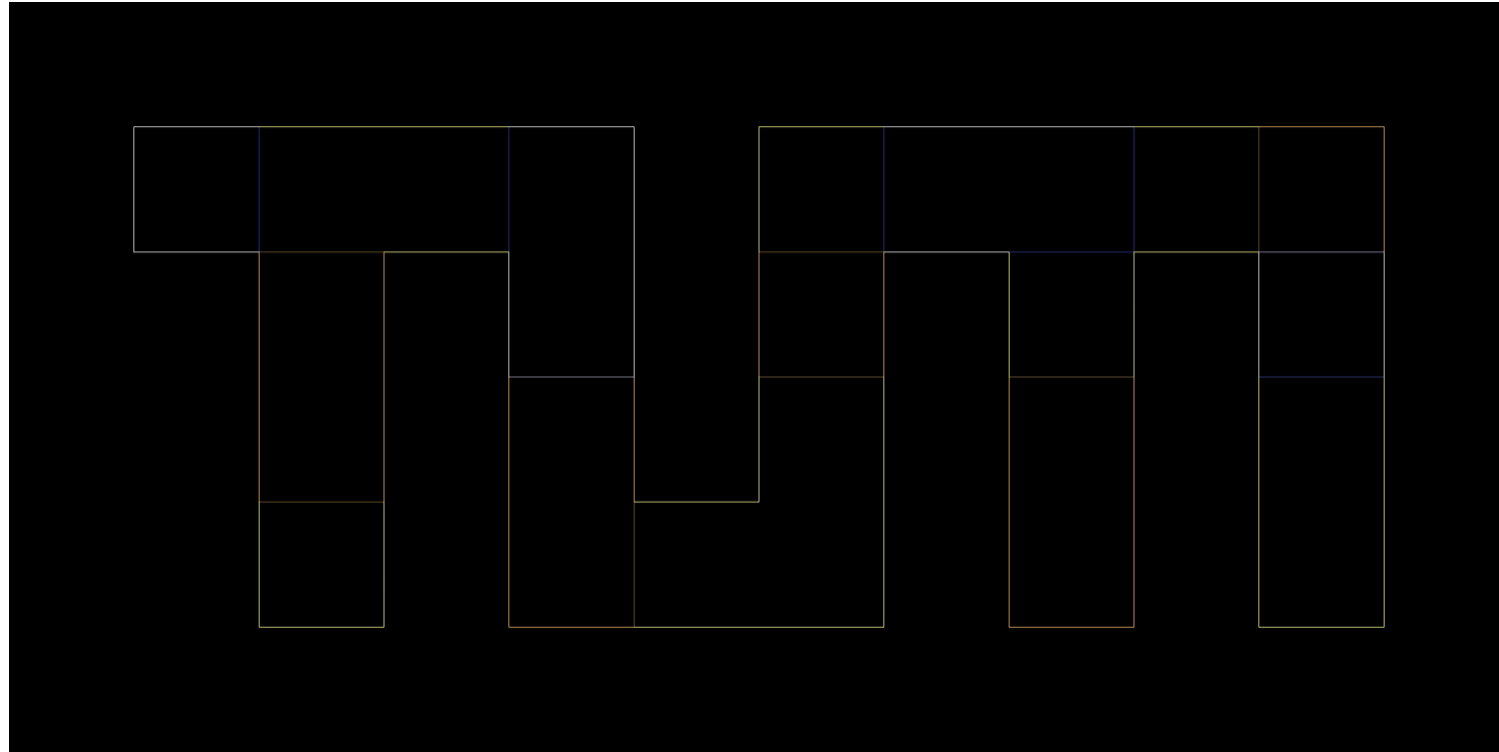
Anwendung am Beispiel



- Anwendung auf alle Farbkanäle

Kantenerkennung mit dem Sobel-Operator

Anwendung am Beispiel



- Anwendung auf alle Farbkanäle

Performanzanalyse

Spezifikationen

6 Versionen (je 24 & 8 Bit Pixel)

- ▶ 0 & 3: Naive Implementierung
- ▶ 1 & 4: Implementierung mit SIMD
- ▶ 2 & 5: Implementierung mit Multithreading + SIMD

Technische Daten Messsystem

- ▶ AMD Ryzen-5 1600 (6 Kerne, 12 Threads), 16GB DDR4 RAM
- ▶ Betriebssystem: Manjaro 22.1.3 (Kernelversion 6.1)

Testdaten

- ▶ Gängige Bildgrößen und Formate

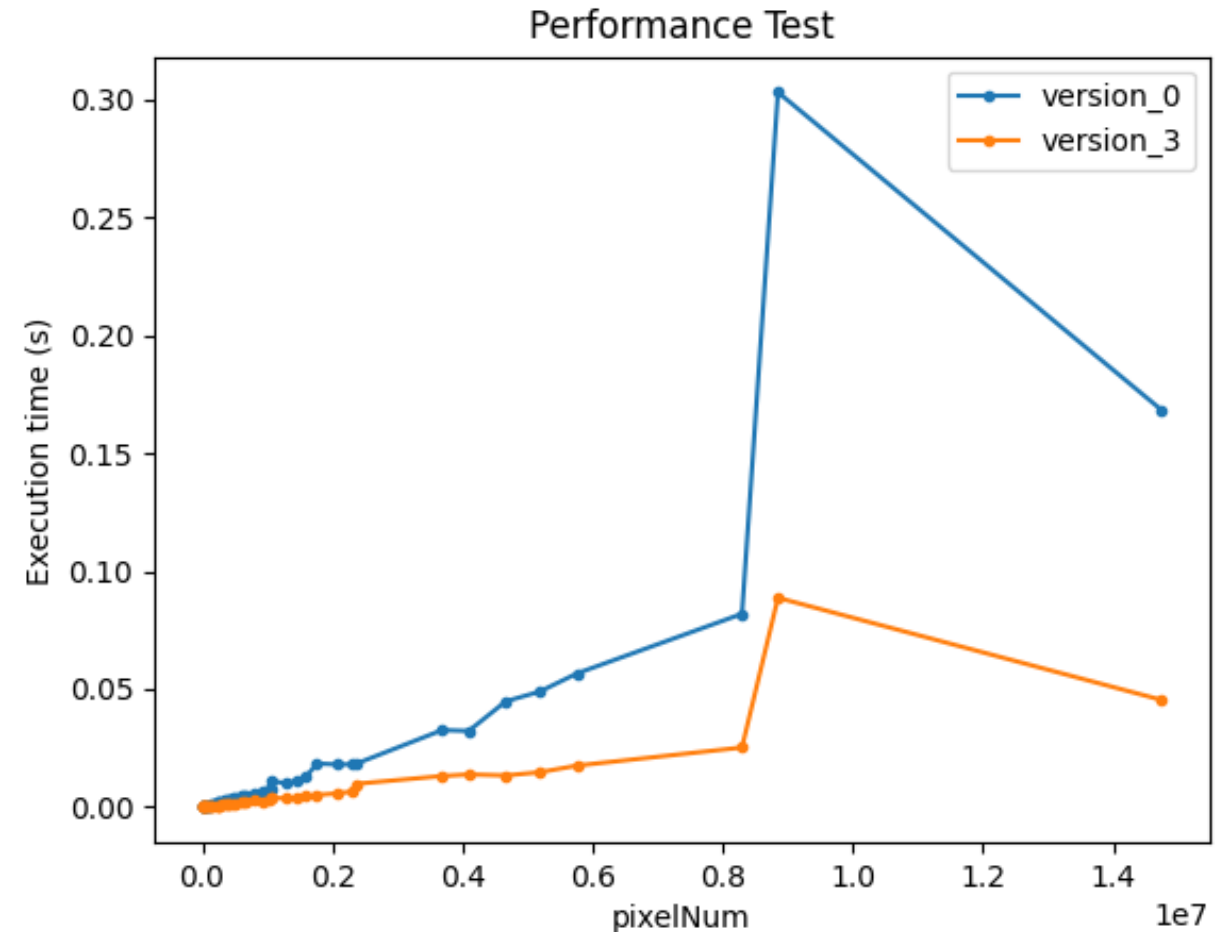
Compiler

- ▶ GCC 13.1.1, -O3

Performanzanalyse

Auffälligkeit bei naiver Implementierung (V0 bzw. 3)

- Ausreißer erzeugt durch Stall-Zyklen

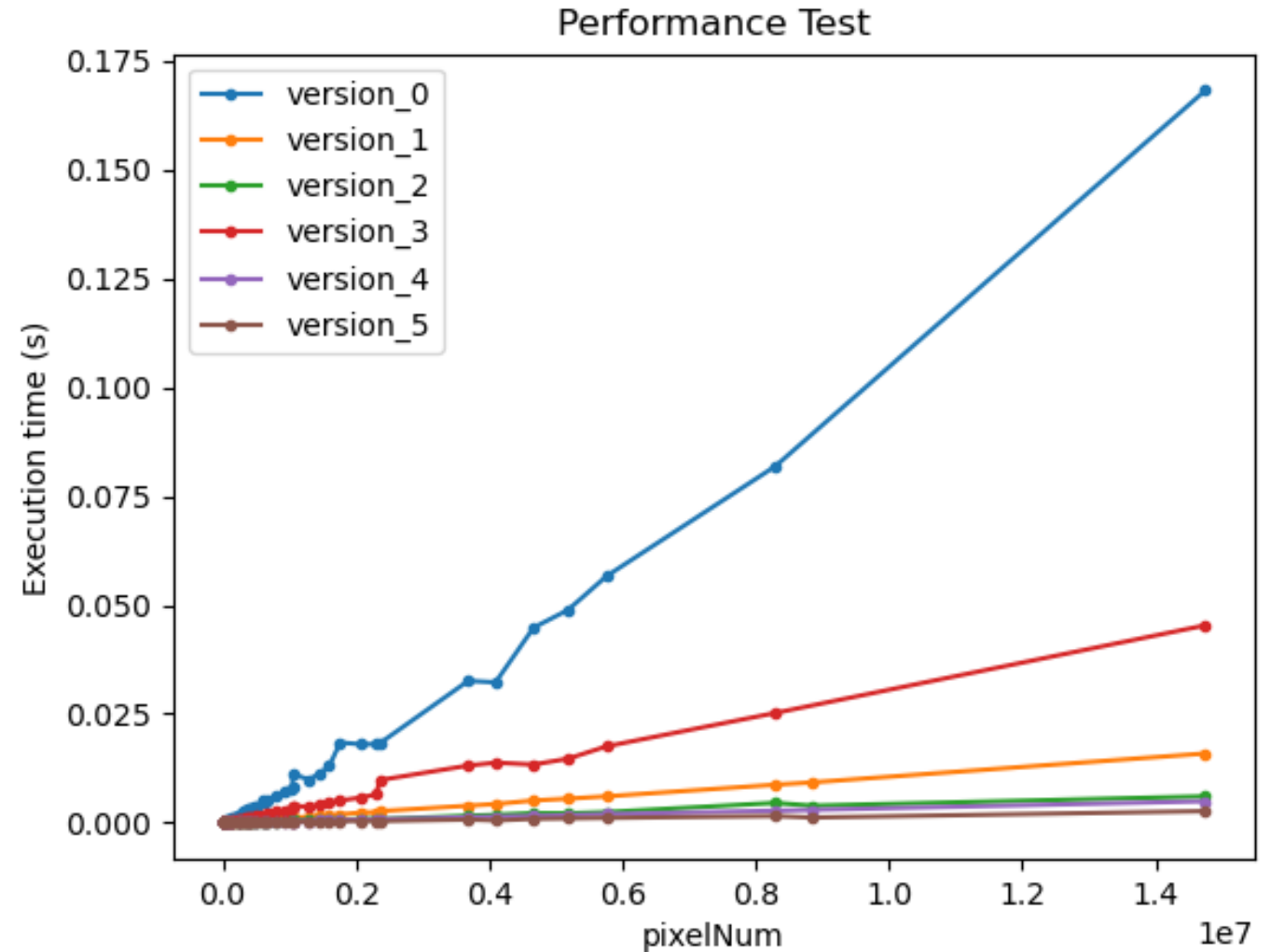


Performanzanalyse

Gegenüberstellung aller Versionen

Wenig überraschender
Sieger: Version 5

► Höchste Parallelität,
niedrigste Pixelbreite

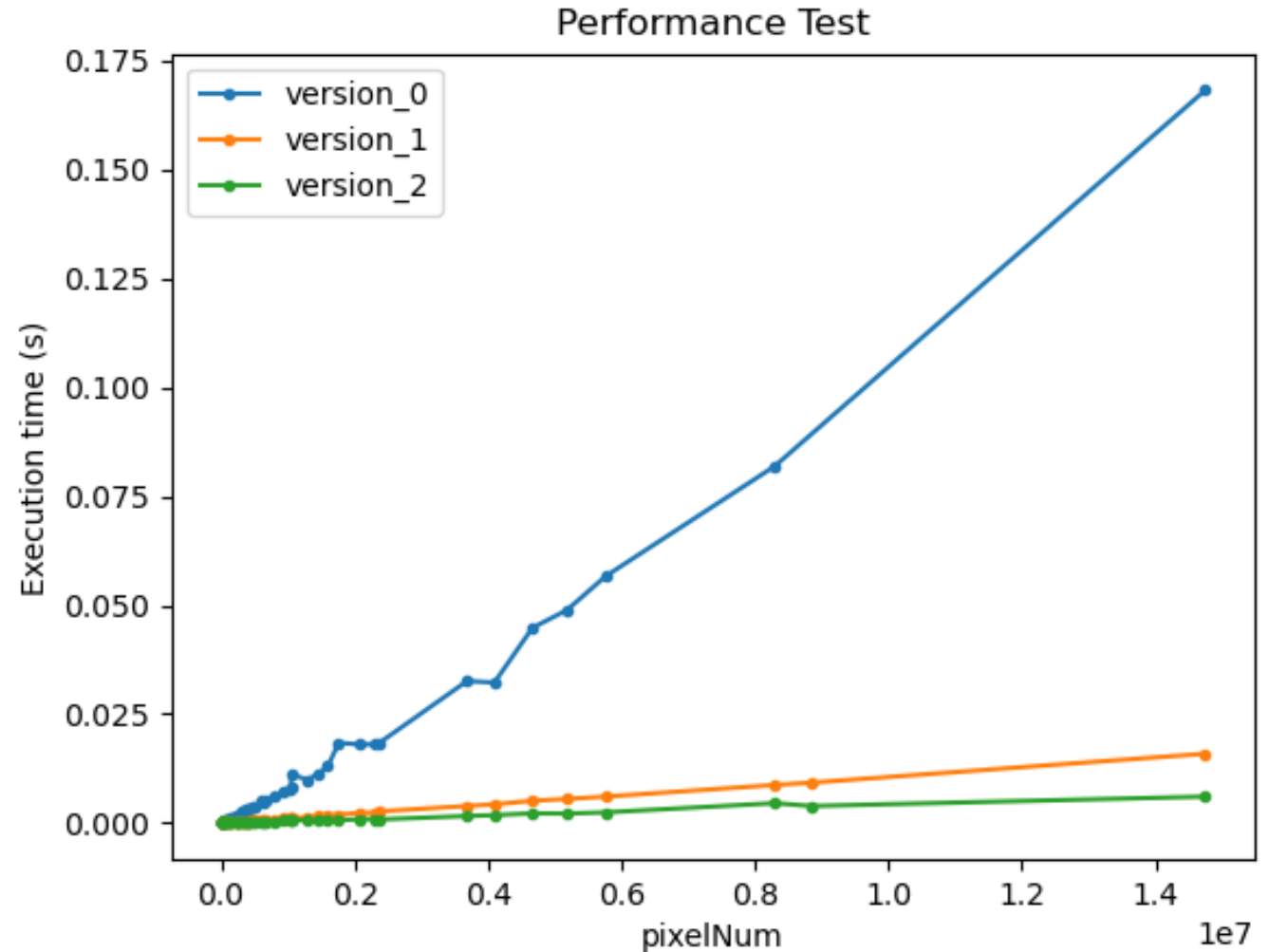


Performanzanalyse

Basisversion vs. SIMD vs. Threading

SIMD ist hier signifikant
schneller als SISD

► Sehr parallelisierbares
Problem

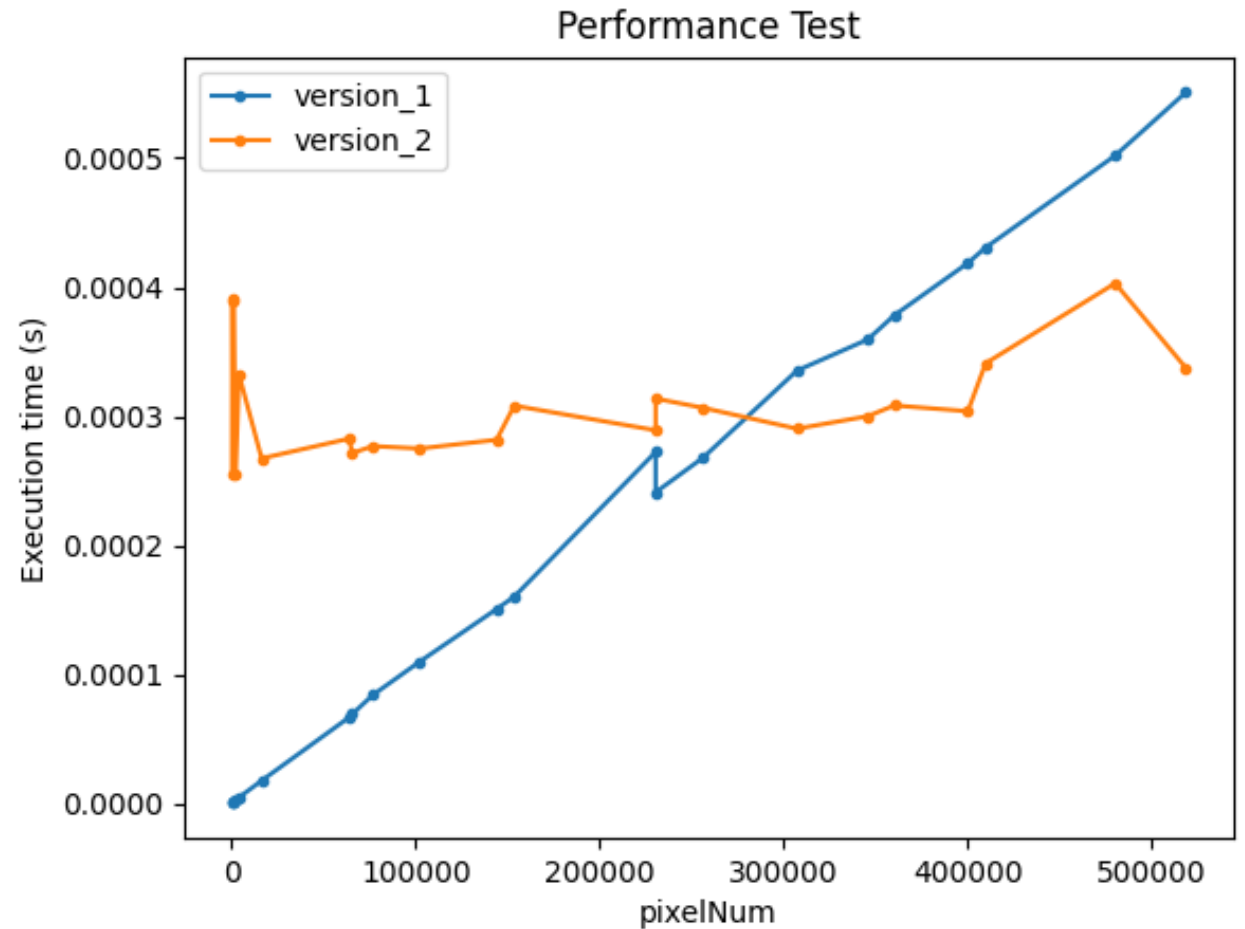


Performanzanalyse

SIMD vs. SIMD + Threading auf kleinen Bildern

Auffällig: SIMD für kleine Bilder schneller

- Threadverwaltungsoverhead

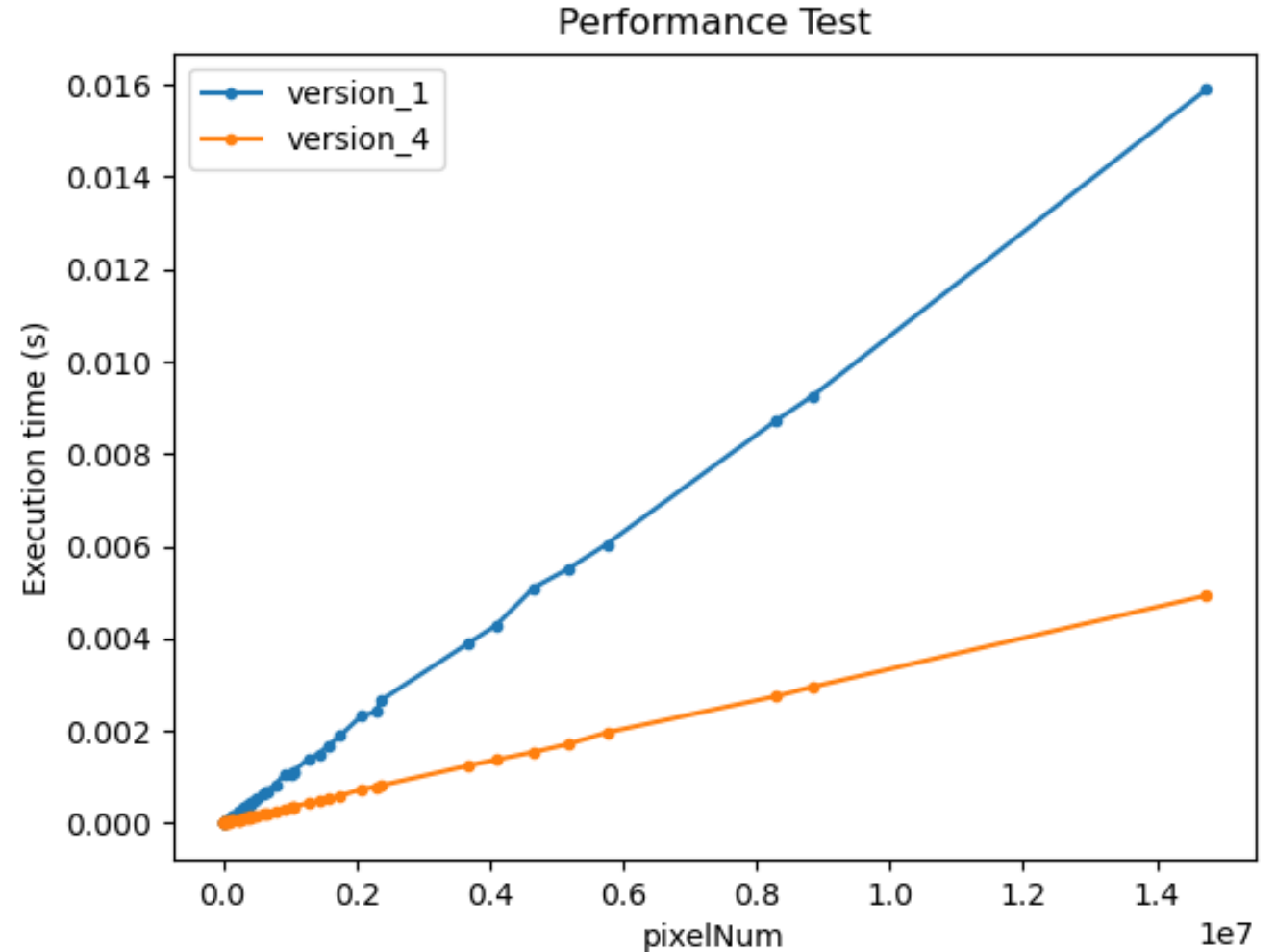


Performanzanalyse

24 vs. 8 Bit Pixelbreite

Speedup durch
Graustufenbilder

- ▶ Nur noch ein Drittel der Daten
- ▶ Dreifache Geschwindigkeit

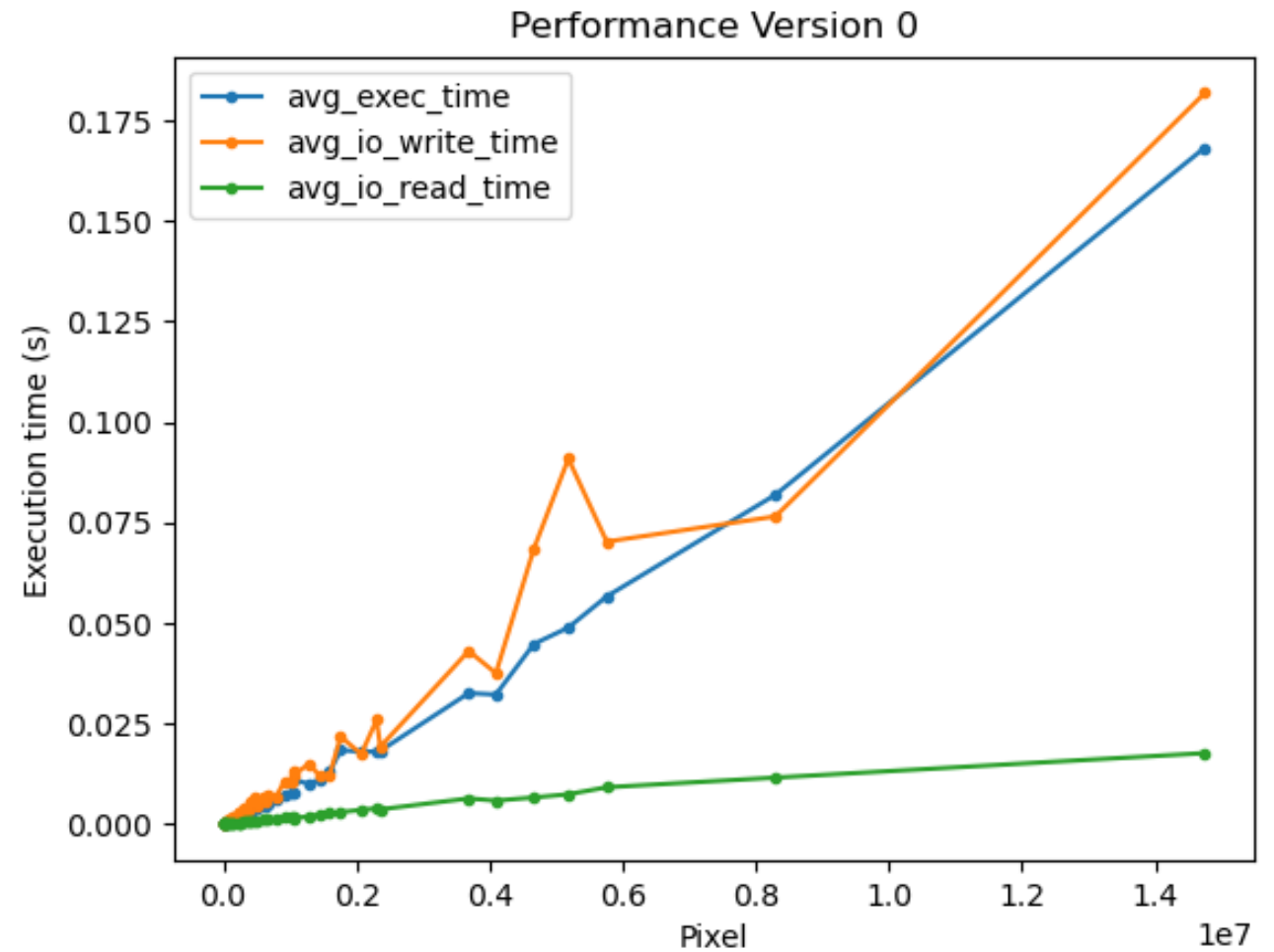


Performanzanalyse

Filesystem vs. Basisversion

Interessant: Read ist sehr schnell

► Ausblick: Nachforschen woran das liegt



Ausblick

- Bildbearbeitung? → Grafikkarten
 - V.a. für Echtzeitanwendungen bzw. [Videos](#) interessant