



Sobel-Filter

Grundlagenpraktikum: Rechnerarchitektur SS23

Simon Bußmann

Nico Lintner

Manuel Walter Mußbacher

Übersicht

- Funktionsweise von Sobel
- Implementierungen
 - Version 0
 - Version 1
 - Overflows
 - Version 2

Funktionsweise

- Um den Sobel-Wert des Pixels in der Mitte zu berechnen, werden zwei Matrizen

- Mv:

1	0	-1
2	0	-2
1	0	-1

und Mh:

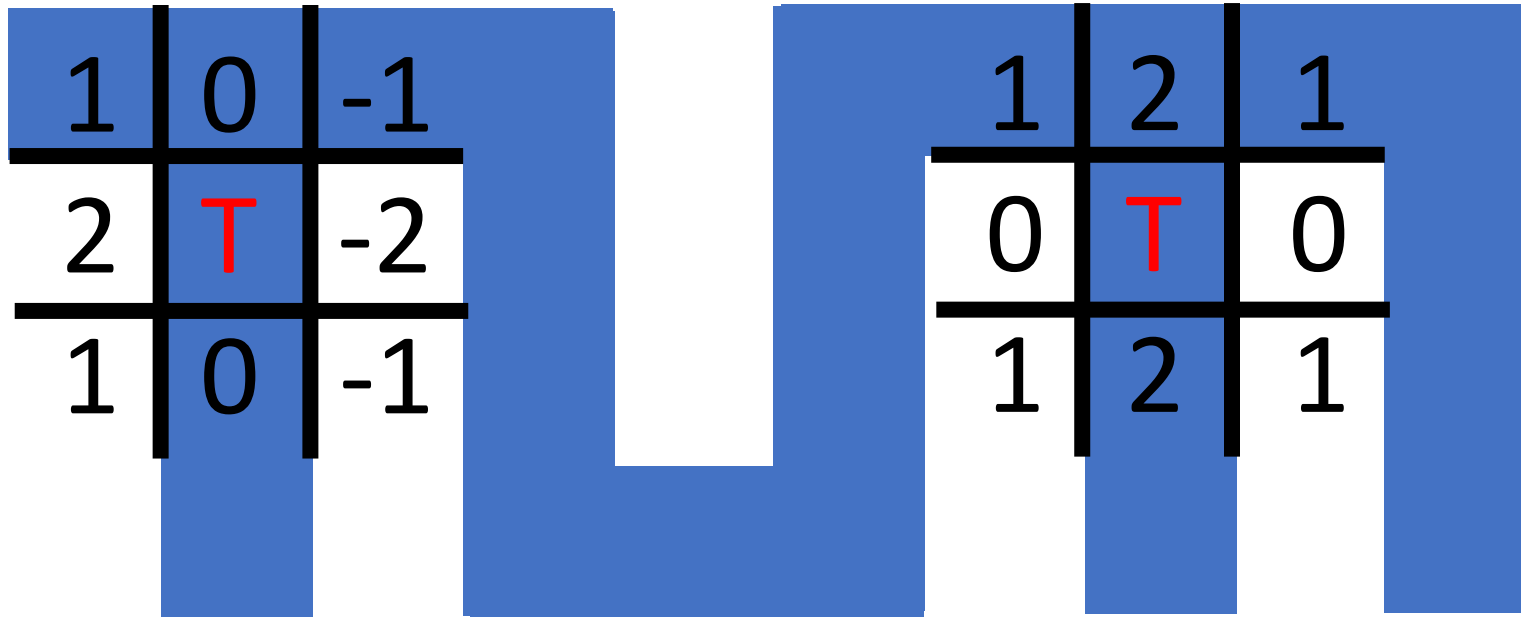
1	0	-1
2	0	-2
1	0	-1

auf die umliegenden Pixel angewandt

- Berechnen die Änderung der Farbwerte in diesem 3x3 Bereich

Funktionsweise

Berechnung der vertikalen bzw. horizontalen Kanten in einem Bild:



Wert des roten Ts ergibt sich durch die Summe der Produkte der Pixel mit dem Wert in der Matrix

Funktionsweise

- Nach der Berechnung dieser beiden Werte:
 - „Länge“ der beiden „Vektoren“ berechnen: $\sqrt{(\textcolor{red}{T} \text{ von } Mv)^2 + (\textcolor{red}{T} \text{ von } Mh)^2}$
 - Alternativ lässt sich dieser Wert auch effizienter mit $|\textcolor{red}{T} \text{ von } Mv| + |\textcolor{red}{T} \text{ von } Mh|$ annähern
- Alle Pixel in neuem Bild abspeichern
 - Insbesondere entstehen dadurch keine Datenabhängigkeiten zwischen Input und Output-Bild

Version 0 – Übersetzung in C

- Ineffizient, aber auch nicht darauf ausgelegt



Version 1 - SIMD

- SIMD-Instruktionen wenden dieselbe Operation auf mehreren Daten gleichzeitig an
 - Mächtige Speedups möglich!
- Kleine Datenbereiche, Großes Problem
 - Um Overflows zu verhindern, werden 16 Byte auf 2 16-Byte Vektoren aufgeteilt

SIMD: Overflows vermeiden

Speicherbereich mit Bytes 0-15:

B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	b15	b16
----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----

Vektor 1 mit allen ungeraden Bytes

00	B1	00	B3	00	B5	00	B7	00	B9	00	B11	00	B13	00	b15
----	----	----	----	----	----	----	----	----	----	----	-----	----	-----	----	-----

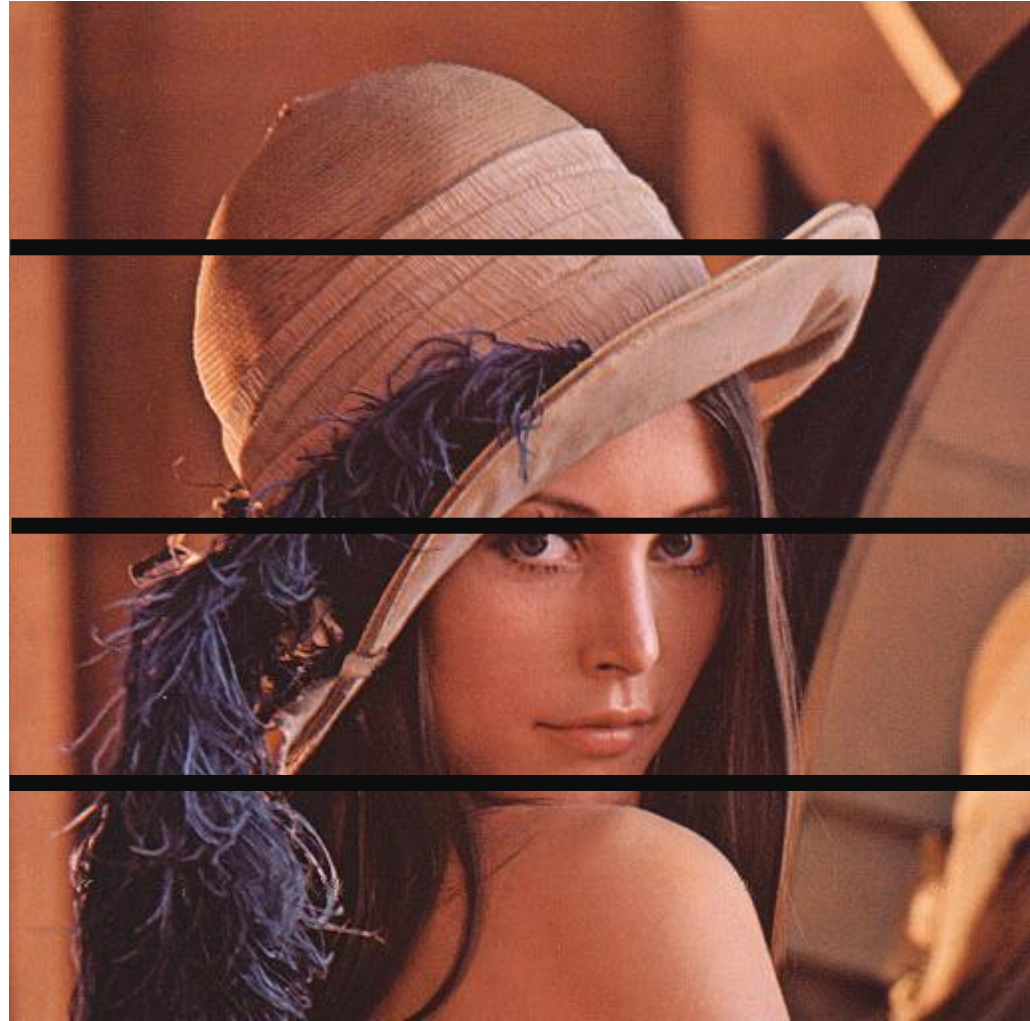
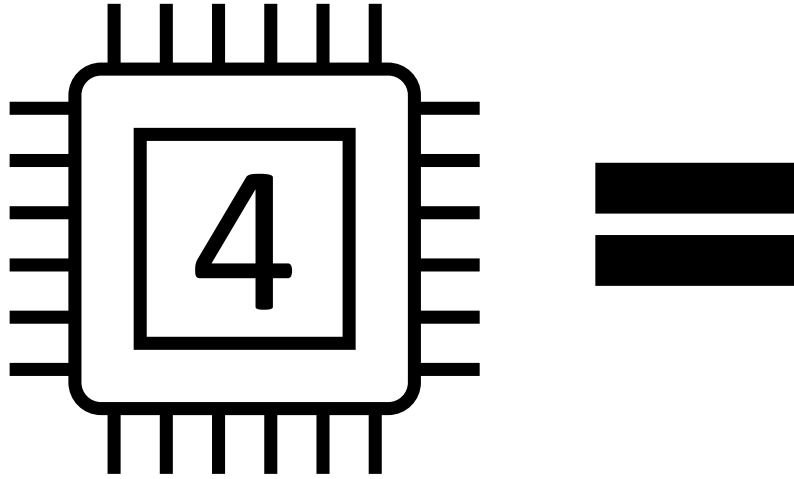
Vektor 1 mit allen geraden Bytes

00	B2	00	B4	00	B6	00	B8	00	B10	00	B12	00	B14	00	b16
----	----	----	----	----	----	----	----	----	-----	----	-----	----	-----	----	-----

→ Können jetzt als 16-Bit Integer (-32768 bis +32767 - leicht ausreichend) interpretiert werden

Version 2 – Threading

- Version 1 als Armee



Versionen 3-5

