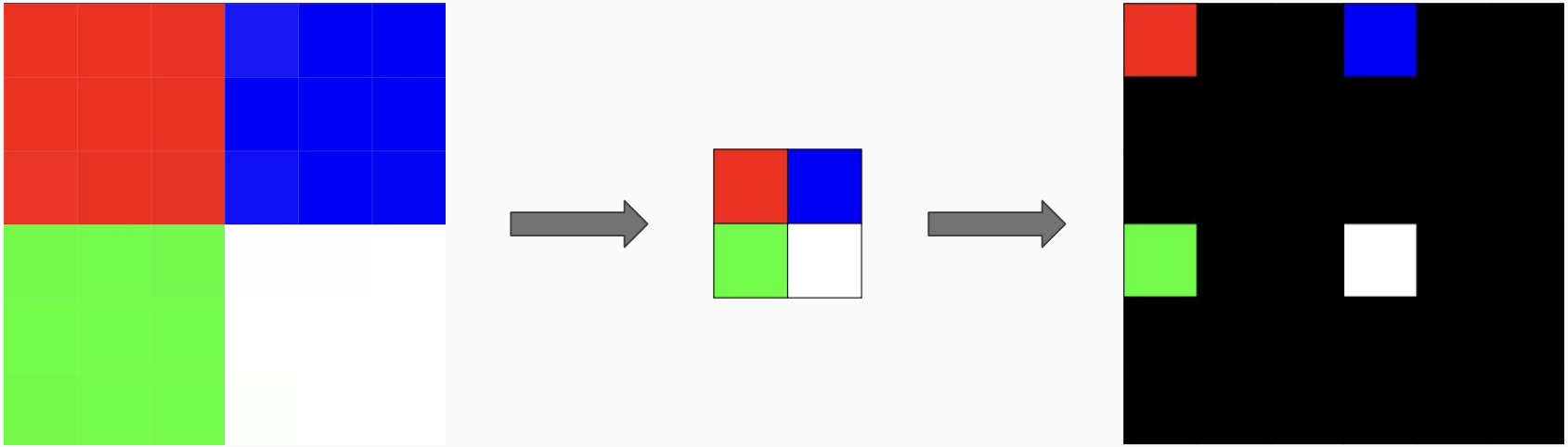
Pixelwiederholung

Team 158: Arda Mat, Barış Özgün, Yiğit Odakır

Problemstellung

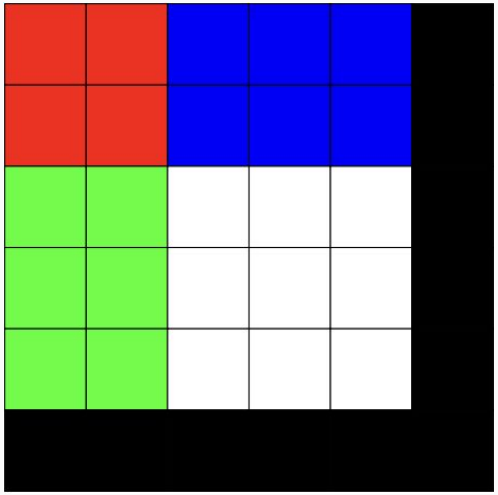
- Lesen der Informationen aus der gegebenen BMP-Datei
- Einen Fenster aus dem Bitmap ausschneiden
- Das ausgeschnittene Bild vergrößern
- Eine neue BMP-Datei erstellen

Beispiel

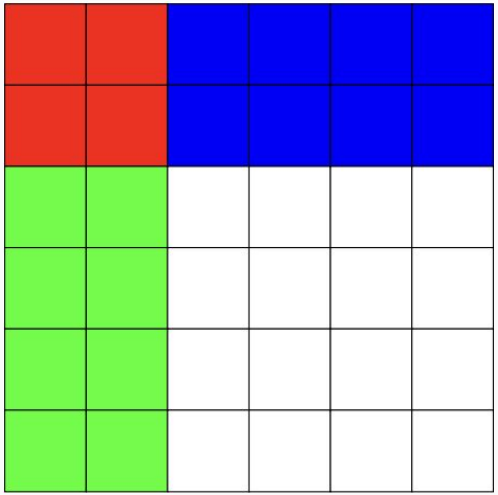


Beispiel

Direkten Nachbarn gefüllt



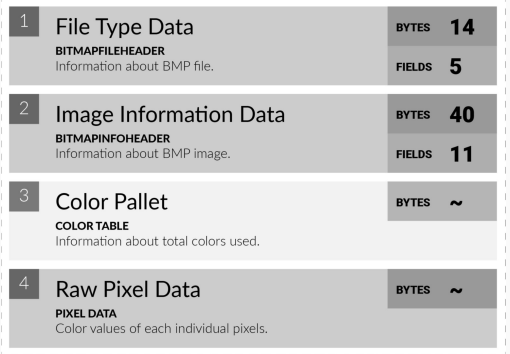
alternativen Nachbarn gefüllt



BMP File Format

BMP-Datei enthält 3 Teile:

- Der Dateikopf: 14 Byte, gibt die Größe der Datei und an welchem Byte die Pixeln anfangen, an
- Der Informationsblock: 40 Bytes, enthält Informationen wie die Breite und die Höhe
- Die Bilddaten: alle Pixeln



Lösungsansatz

- Extrahieren der Informationen aus der BMP-Datei

  in Form eines Arrays speichern

- window()-Methode:
  - Lesen der Höhe/Breite des Bildes aus der BMP-Datei
  - Berechnen der Anfang-Offsets
    - Top-Down-Ansatz: Erste Pixel liegt oben links im Array
    - Bottom-Up-Ansatz : Erste Pixel unten links im Array
  - Speichern der Pixelwerte in den Pufferspeicher

Lösungsansatz

- zoom()-Methode:
  - Setzen von originale Pixeln auf skaliertes Array
  - Direkten Nachbarn der originale Pixeln füllen
  - Restliche Lücken füllen

- Erstellen der neuen BMP-Datei:
  - Setzen der nötige Größen und Informationen
  - Pixeln aus der zoom()-Methode aufschreiben.

SIMD Optimierung

window_v1()-Methode:

- Ladung und Speicherung der Pixeldaten durch SSE-Instruktionen
- Gleichzeitiges Lesen und Schreiben mehrerer Daten

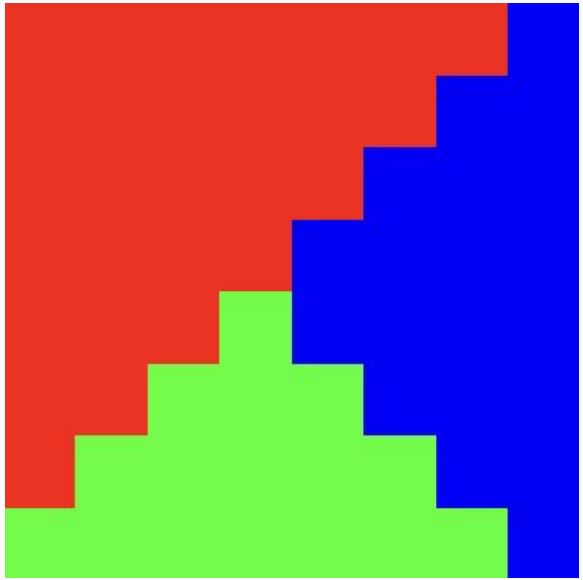
Vorteil:

- Optimierte Laufzeit niedriger bei großen Bilder

Nachteil:

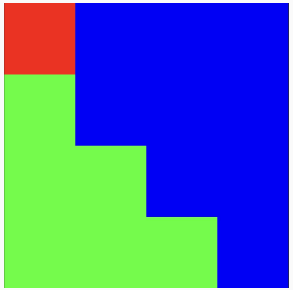
- Laufzeit ändert sich nicht bei kleineren Bilder

Beispiel

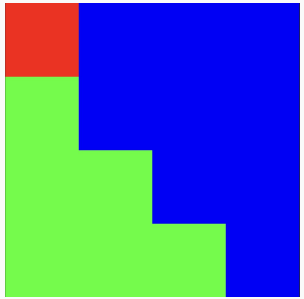


Startindex: (3,3)

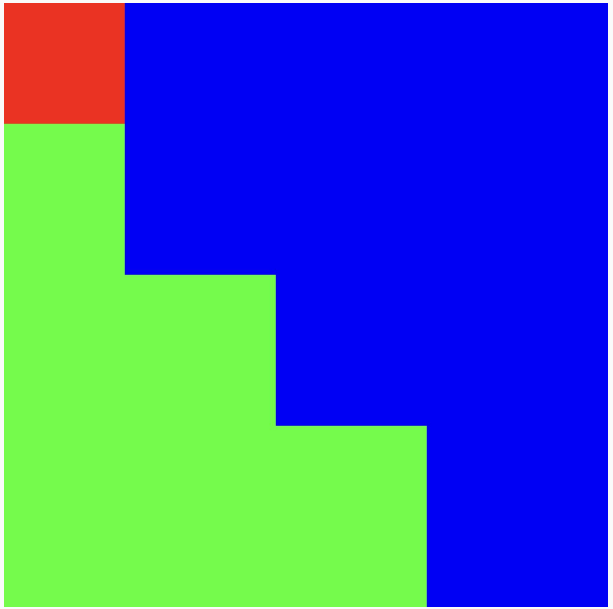
Höhe = 4, Breite = 4



Beispiel

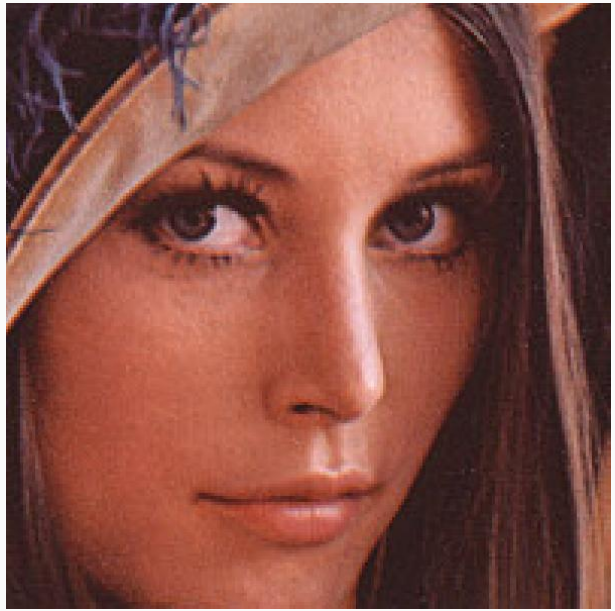


mit Faktor 5 skalieren



Beispiel

-Startindex: (200,200)
-Geschnittenes Fenster:200x200
-Skalierungsfaktor: 3
-Resultat: 600x600



Beispiel

-Startindex: (200,10)
-Geschnittenes Fenster:250x250
-Skalierungsfaktor: 4
-Resultat: 1000x1000

Performanz: Analyse

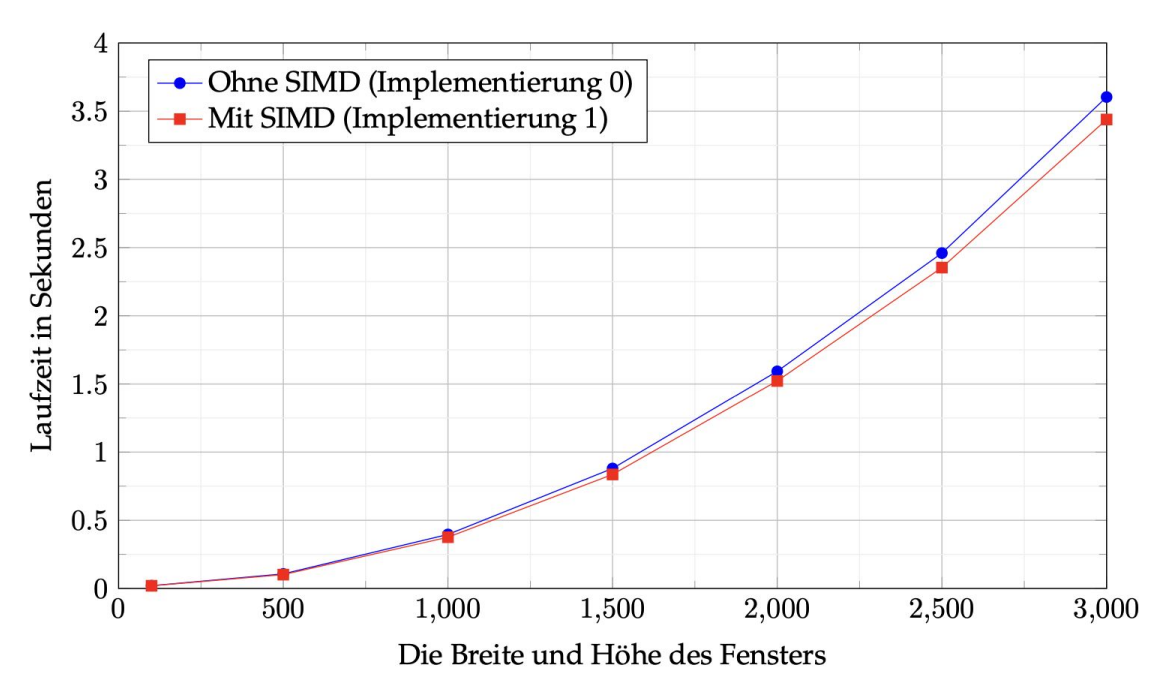
Laufzeit hängt von: die Breite-/ Höhe des Bildes und der Skalierungsfaktor ab

window(): O(width * height)

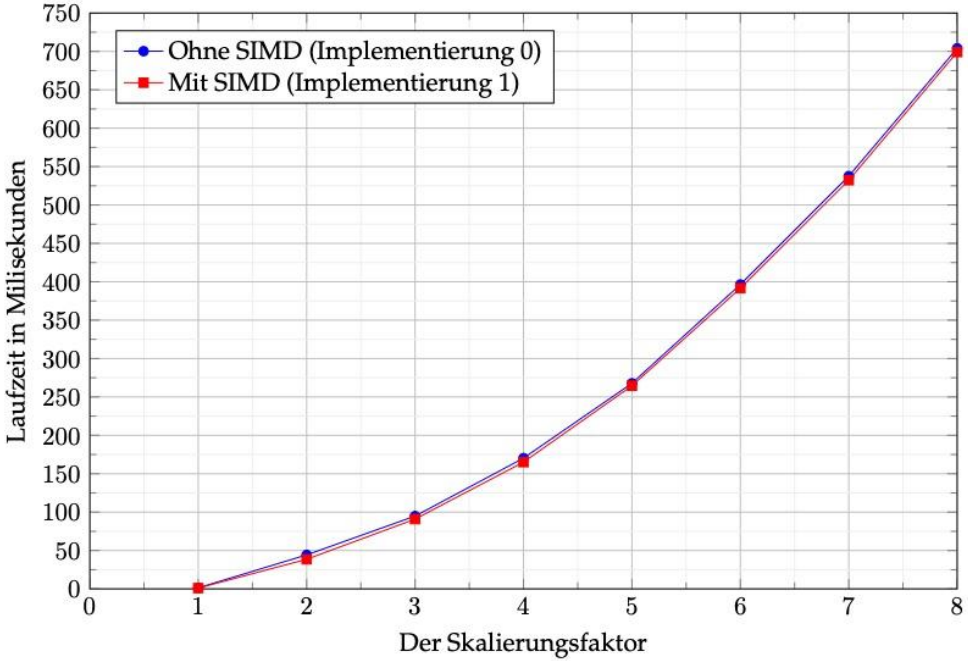
zoom(): O(width * height * scale_factor^2)

Laufzeit des Algorithmus: O(n^2)

Performanz: Bildgröße-Laufzeit Beziehung



Performanz: Skalierungsfaktor-Laufzeit Beziehung



Zusammenfassung

- BMP-Datei lesen (I/O-Operationen)
- Ein Fenster ausschneiden: Top-Down-/Bottom-Up Ansätze
- Fenster mit dem Skalierungsfaktor vergrößern
- Eine neue BMP Datei erstellen
- Optimierung: SIMD
- Performanz: Laufzeit des Algorithmus = O(n^2)