

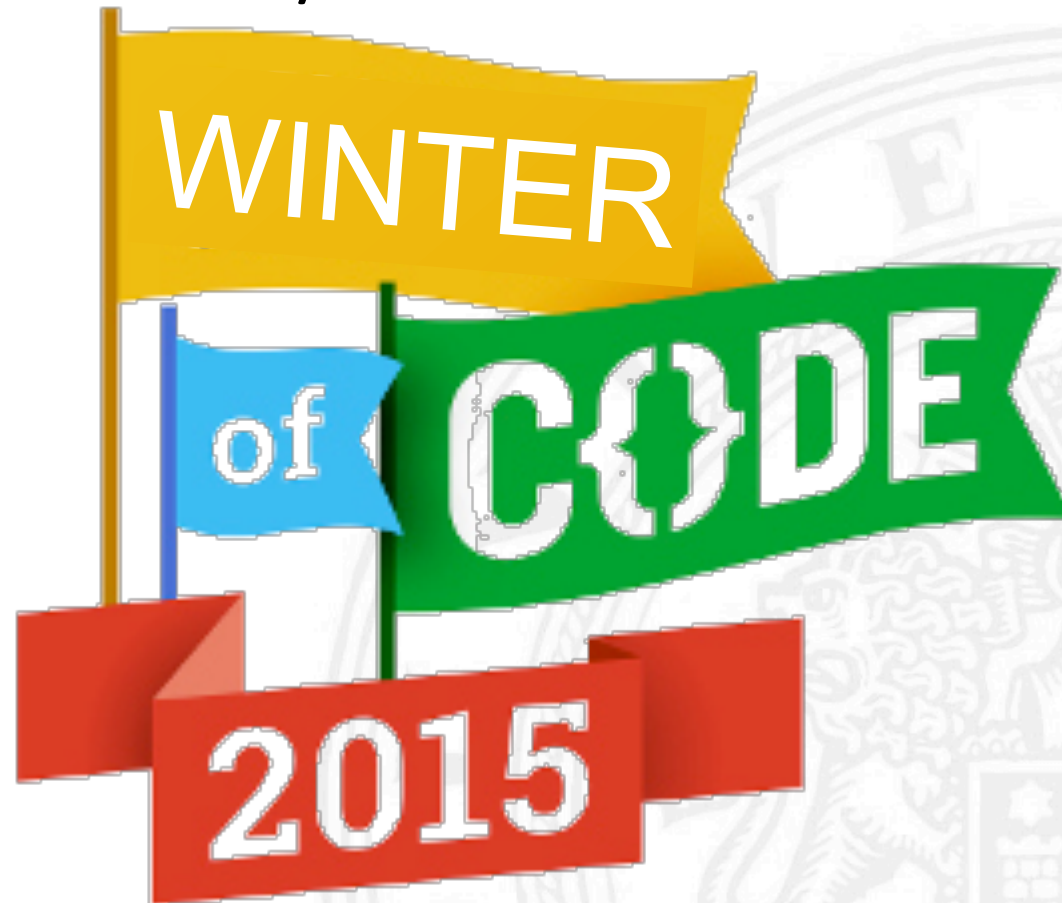


Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

MIN-Fakultät  
Fachbereich Informatik  
Arbeitsbereich SAV/BV (KOGS)

SAV/BV



---

# Die Aufgabe

- Implementierung elementarer, häufig benötigter Bildverarbeitungsfunktionen (siehe folgende Folien)
  - Zeitmessung der Laufzeit
  - Kompletter Code muss C/C++ kompilierbar sein
  - C/C++, ASM, OpenCL, CUDA, Multi-Threading und Multi-Processing sind erlaubt!
  - Tipp: Legt ein Experimentier-Tagebuch an und dokumentiert eure Versuche von Anfang an!
-

---

# Das Ziel

- Vergleich der Möglichkeiten zur effizienten Bildverarbeitung auf aktuellen Architekturen
  - Erstellung von Empfehlungen aufgrund der Statistiken
  - Untersuchung von Laufzeit, Arbeitsspeicher und Auslastung!
  - Referenz: Python 2.8 + Numpy
-

---

# Setting (1)

- Alle folgenden Bildverarbeitungsoperationen sollen auf folgenden Bilddaten ausgeführt werden:
    - Grauwertbilder (2D-Arrays)
    - Farbbilder (3D-Arrays, 3. Dimension: RGB)
    - Bildgrößen: 2048x2048 Pixel, 4096x4096 Pixel
    - Pixeldatentypen: UInt8, float(32)
  - Pro Test: 8 Variationen!
  - Bei insgesamt 10 Operationen: 80 Tests pro Team
  - Bei 5 Teams (+1 Referenz): 480 Ergebnisse
-

---

# Setting (II)

- Es zählt nur die Laufzeit der Operation, Laden und Speichern des Bildes werden nicht mitgezählt!
  - Aber:
    - Alle Tests müssen je 100x laufen!
    - Maß am Ende: Gesamtlaufzeit!
  - 3 Typen von Bildverarbeitungsverfahren:
    - Map/Reduce
    - Filter
    - Transformationen
-

---

# Map/Reduce

- Algorithmus 1 (Image Blending):

$$Img\_dest = MINMAX_{DT} \text{ oder } (Img\_src1 + Img\_src2)/2$$

- Algorithmus 2 (Schwellenwert):

$$Img\_dest = Img\_src > Scalar/RGB$$

- Algorithmus 3 (Histogramm, Ann.:  $-1 < I(x,y) < 256$ )

$$Hist(i) = card(\{(x,y) \mid I(x,y) = i\}) \text{ f\"ur } -1 < i < 256$$

\*  $MINMAX_{DT}$  stellt sicher, dass die Grenzen des Datentyps nicht über/unterschritten werden!

---

---

# Filter (Faltung)

- Alle Filter als 5x5 und 11x11 implementieren!
- Im Folgenden werden beispielhaft Faltungen der Größe (5x5) gezeigt:

```
For (int y=2; y!=height-2; ++y)
  For (int x=2; x!=width-2; ++x)
    For (int j=-2; j!=3; ++j)
      For (int i=-2; i!=3; ++i)
        Img_dest(x,y)
          = Img_src(x,y)*mask(i+2,j+2)
```

---

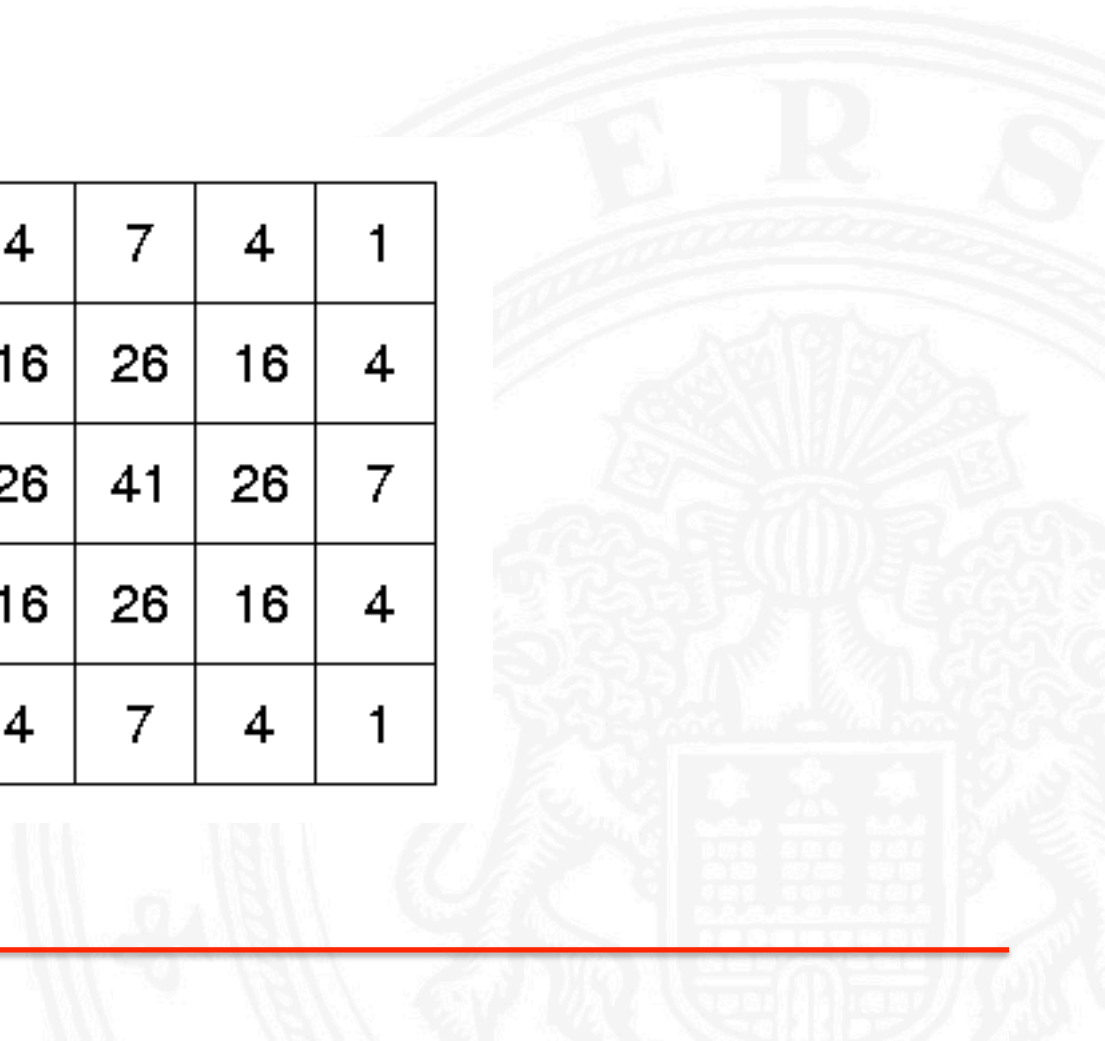
---

# Gauß'sche Glättung ( $\sigma=1$ , näherungsweise)

Mask(i,j):

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



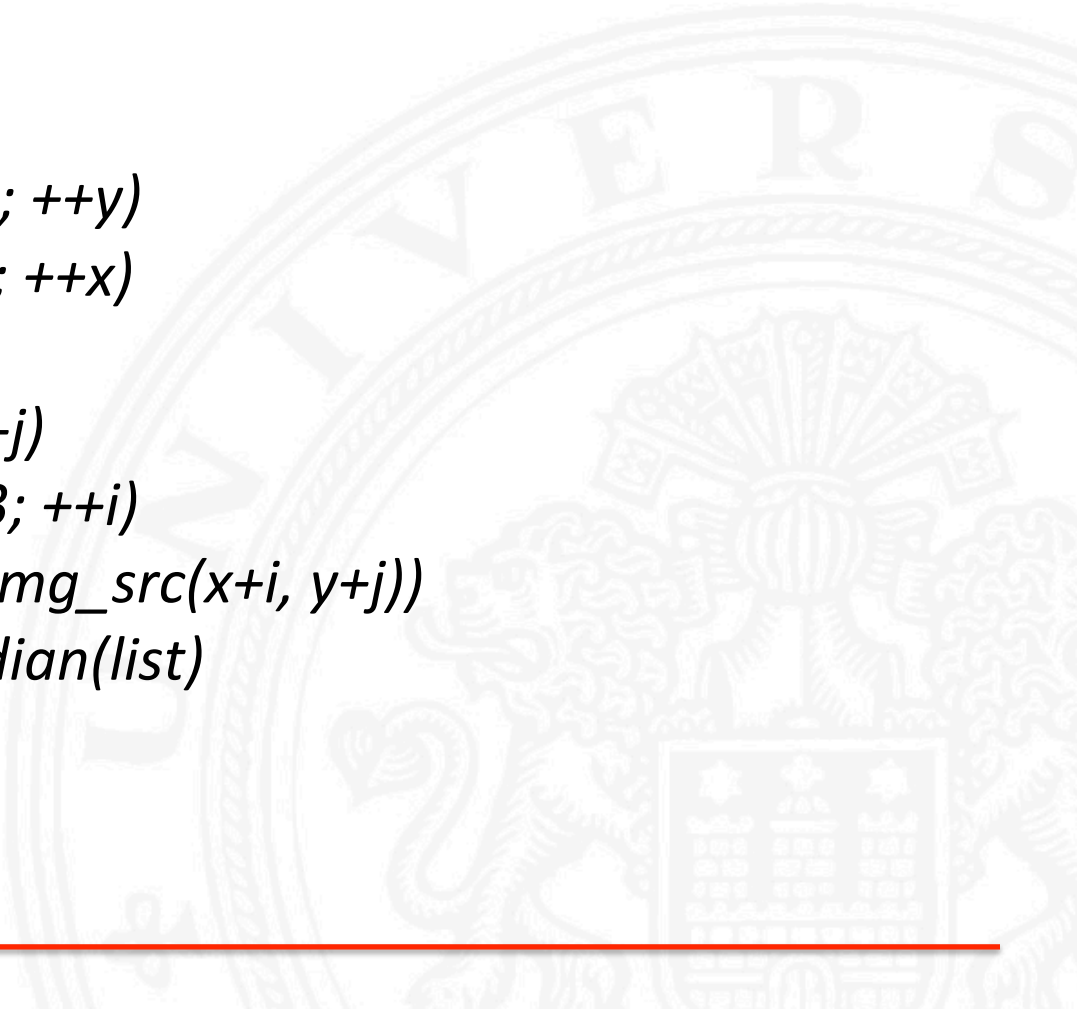


---

# Median-Filter

- Beispielhaft für 5x5:

```
For (int y=2; y!=height-2; ++y)
  For (int x=2; x!=width-2; ++x)
    list = []
    For (int j=-2; j!=3; ++j)
      For (int i=-2; i!=3; ++i)
        list.append(lmg_src(x+i, y+j))
    lmg_dest(x,y) = median(list)
```



---

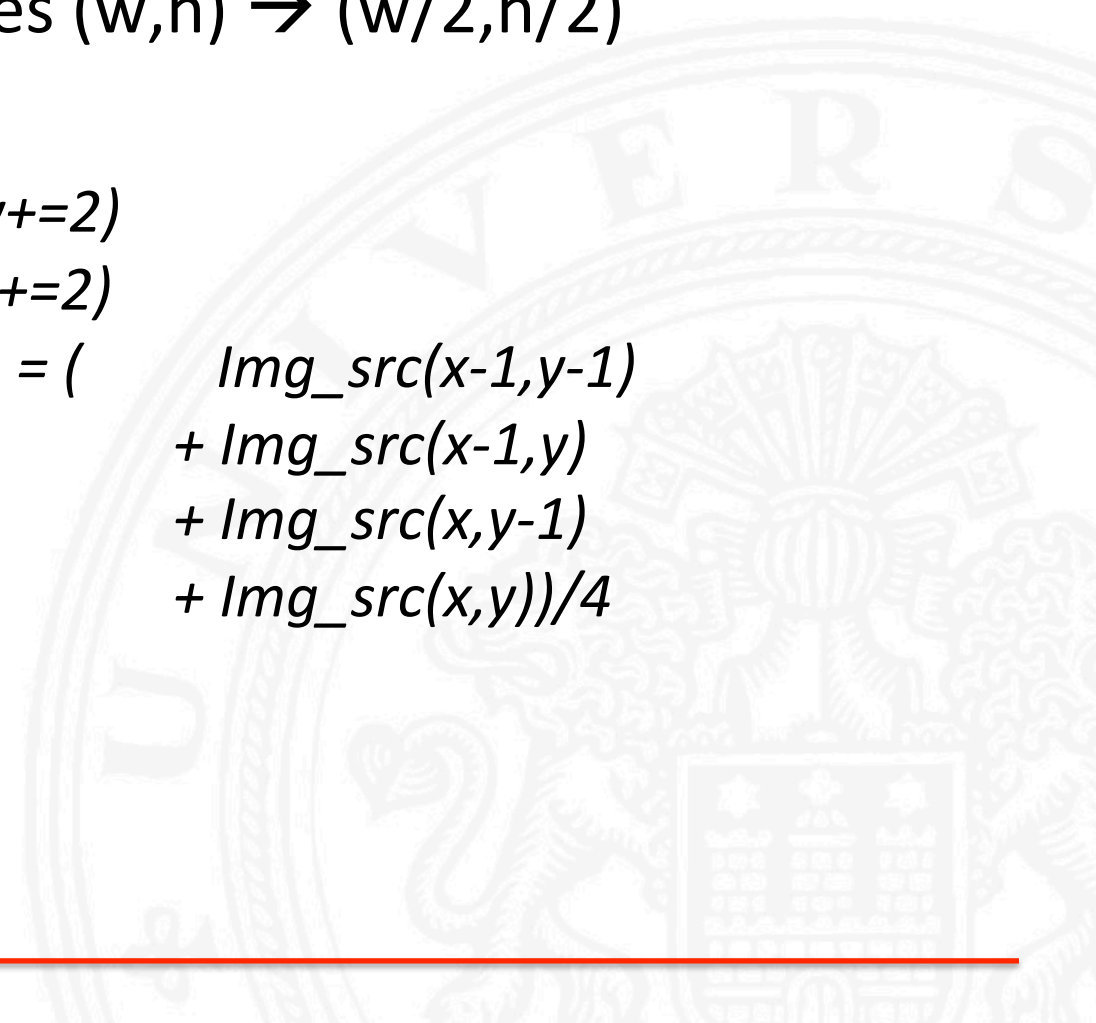
# Subsampling (Binning)

- Verkleinern des Bildes  $(w,h) \rightarrow (w/2,h/2)$

*For (int y=1; y<height; y+=2)*

*For (int x=1; x<width; x+=2)*

**+   
*+   
*+****



---

# Bildtransformationen (Rotation)

- Transformation der Bildkoordinaten, nicht der Bildintensitäten!

- Beispiel, für einen Winkel  $\alpha$ :

$$img\_dest(r\_x, r\_y) = img\_src(x, y)$$

$$\text{mit: } (r\_x, r\_y)^T = R_\alpha * (x, y)^T$$

$$\text{und: } R_\alpha = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}.$$

- Varianten:
    - Nächste Nachbarn (NN-) Interpolation
    - Bi-linear interpolieren!
-

---

# Zusammenfassung

1. Bild-Blending
  2. Schwellenwert
  3. Histogrammberechnung
  4. Gauß'sche Glättung (5x5)
  5. Gauß'sche Glättung (11x11)
  6. Medianfilter (5x5)
  7. Medianfilter (11x11)
  8. Binning (2x2->1x1)
  9. Rotation (NN-Interpolation)
  10. Rotation (Bi-lineare Interpolation)
-

---

# Abgabe

- Abgabe bis zum 31.3.2015
  - Preise für die schnellsten Implementationen!
  - Format: Archiv mit allen Dateien und Hinweise auf benötigte Bibliotheken
  - Gut dokumentierter Code!
  - Definition der einzelnen Funktionen und mehrfacher Aufruf!
-

# Viel Erfolg!

Times to beat [s] für Grauwertbilder

(2x2,66GHz Xeon

Dual Core,5 GB RAM)

Quelltext verwendet numpy/scipy

(Bald) verfügbar auf der Homepage!

	small, uint8	small, float	large, uint8	large, float
Blending	0,52	3,96	2,52	22,84
Schwellenwert	0,58	1,09	2,75	4,81
Histogramm	19,76	25,46	79,04	101,75
2D-Faltung (5x5)	37,26	19,88	149,15	90,00
Median-Filter (5x5)	213,82	210,31	856,19	851,17
Subsampling	1,61	4,95	5,90	19,52
Rotation (bi-lin)	71,83	74,86	286,27	324,76

