

# Evaluation einer modernen Zynq-Plattform am Beispiel der Implementierung einer Hough Transformation

## Zwischenpräsentation der Bachelorarbeit

Dominik Weinrich

[dominik.weinrich@tu-dresden.de](mailto:dominik.weinrich@tu-dresden.de)

Dresden, 31.05.2018

# Gliederung

- Aufgabenstellung
- Motivation
- Fortschritt
- Implementierung
- Herausforderungen
- Ergebnisse
- Geplantes HW/SW Codesign

# Aufgabenstellung

- Softwareimplementierung einer Hough Transformation
  - Grayscaler
  - Gauß-Filter
  - Canny Edge Detection
  - Circle Hough Transformation
- Iterative Auslagerung einzelner Komponenten auf den FPGA
- Evaluation

# Motivation

# Fortschritt

- Softwareimplementierung einer Hough Transformation
  - Grayscaler
  - Gauß-Filter
  - Canny Edge Detection
  - Circle Hough Transformation
- Iterative Auslagerung einzelner Komponenten auf den FPGA
- Evaluation



# Implementierung - Grayscaler

```
1  /*
2  ** Converts 32-Bit colored image data to 8-Bit grayscale image data
3  ** The function needs and returns only pixel data without any kind of compression.
4  */
5
6 #include "grayscale.h"
7
8 uint8_t* grayscale(uint32_t* input, int width, int height) {
9     uint8_t *output = malloc(width * height * sizeof(uint8_t));
10
11 #ifdef _OPENMP
12 #pragma omp parallel for
13#endif
14 // Convert image to grayscale
15 for(int y = 0; y < height; y++) {
16     for(int x = 0; x < width; x++) {
17         uint8_t r, g, b;
18         int index = y * width + x;
19
20         r = input[index] >> 16 & 0xFF;
21         g = input[index] >> 8 & 0xFF;
22         b = input[index] & 0xFF;
23         output[index] = 0.3*r + 0.59*g + 0.11*b;
24     }
25 }
26
27 return output;
28}
29
```

# Herausforderungen

- Objekte werden zum Teil mehrfach erkannt
  - Lösungsansatz: Bias compensation
- Es werden viele Parameter benötigt
  - Möglicher Ansatz: Automatische Parameterbestimmung über neuronale Netze (nicht Teil der Arbeit)
-