

Projektgruppe FastSense

# Meilenstein 2

## Hardware Accelerated TSDF SLAM

28. Oktober 2020

## Ziele und Anforderungen

- Ziele für MS2

- Funktionale Anforderungen

- Nicht-Funktionale Anforderungen

## Hauptspeise

- Recap: Prototyping Demo

- Algorithmus

- Hardware Implementierung

- FastSense Prototyp

- Kommunikation

## Evaluation

- Strom

- Zeit

## Fazit

- Bisherige Verbesserungen

- Verbesserungspotenzial

- Projektmanagement

## Ausblick / MS3

# Ziele und Anforderungen

- Implementation von inkrementellem SLAM mit TSDF in "autarker" Box
- Vorimplementation in Software
- Implementation von Bottleneck-Komponenten in Hardware
- Speicherung von Pose-Graph und TSDF-Karte zur Rekonstruktion des kompletten explorierten Bereichs
- Evaluation durch Zeit- und Strommessung

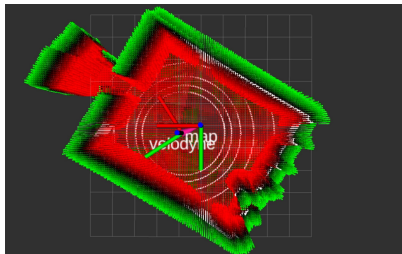
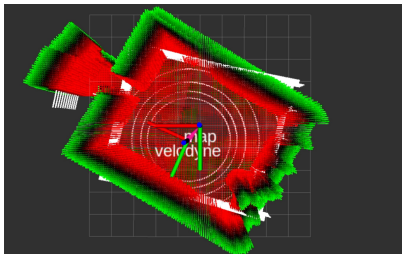
- Lokale TSDF-Map ausgeben
- Aktuelle 6D-Pose ausgeben
- Map auf Basis der IMU und Velodyne-Daten
- Trajektorie und TDSF-Map für jede Pose speichern
- Parameter zur Laufzeit anpassbar

- HW-Plattform: Trenz-Board, SW-Plattform: Vitis
- FPGA-Beschleunigung der Algorithmen
- Sensoren direkt am Board
- Unit-Tests
- Testbench
  - Integration, Strommessung, Zeitmessung, Visualisierung
- Logging

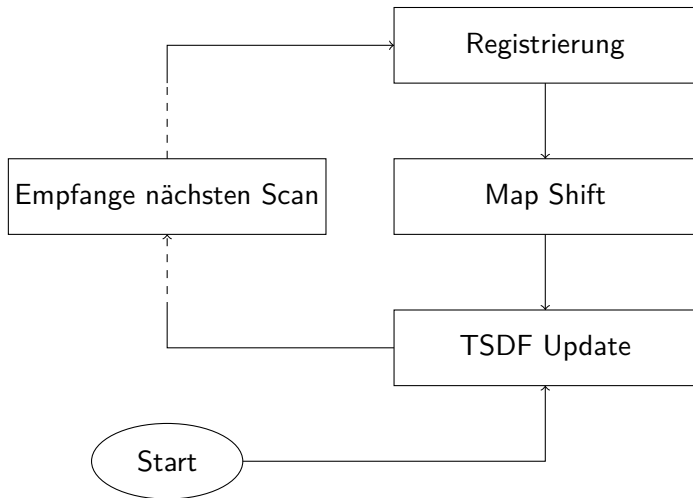
Hauptspeise

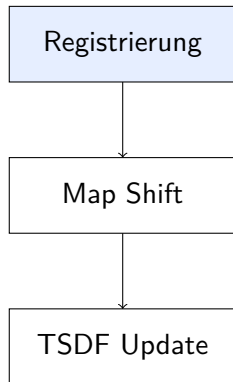
# Recap: Prototyping Demo

- Registrierung mithilfe von IMU
- Gute Parameterkombination herausgefunden
- Geplante Funktionalität war vorhanden und in RViz darstellbar
- Erkannte Probleme / Bottlenecks:
  - Laufzeit stark abhängig von der Auflösung der Karte
  - Probleme mit Orientierung (kurz nach Demo gefixt)
  - Insgesamt noch recht langsam ( $\sim 0.5s/\text{Scan}$  auf Glumanda,  $2-5s/\text{Scan}$  in Testwelt)
  - Bottlenecks Registrierung und TSDF-Update (dachten wir)









$T$  = Initial Transform;

Transform point cloud with  $T$ ;

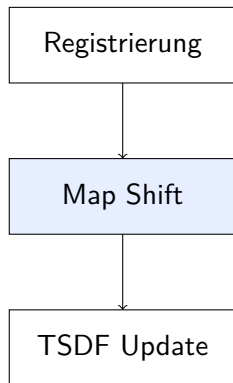
**do**

$E$  = Minimization of the sum of  
signed distances squared;

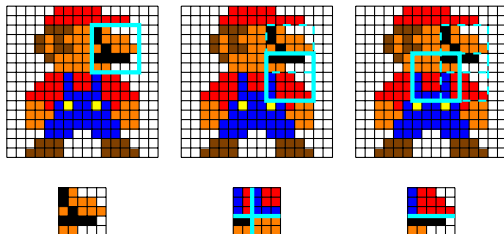
Transform point cloud with  
 $T(E)$ ;

**while** *under max iterations and  $E$   
not converged*;

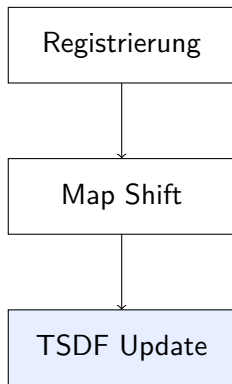
**return**  $T$ ;



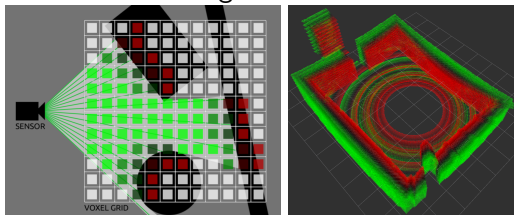
- LocalMap (aka RingBuffer)
  - Geteilt zwischen SW und HW



- Werte außerhalb in GlobalMap
  - Chunk-basiert
- Speicherung in HDF5



## 1. TSDF Generierung

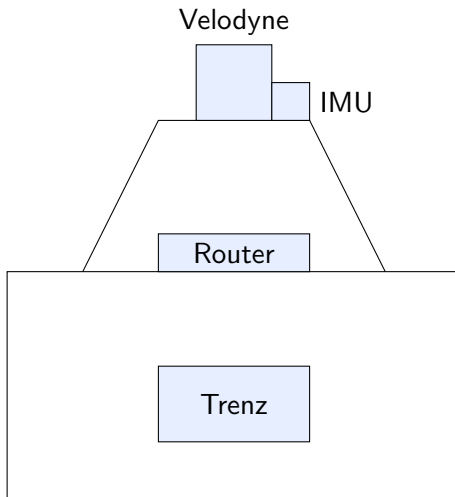


## 2. Update mit gewichtetem Mittelwert

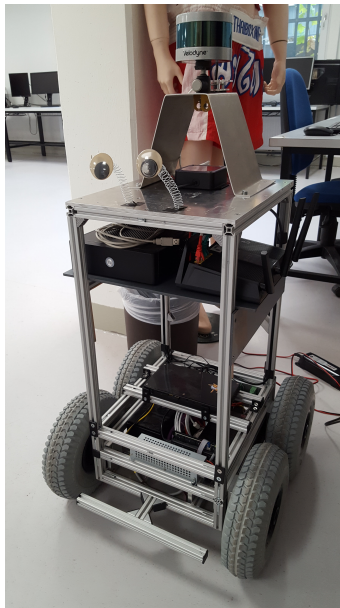
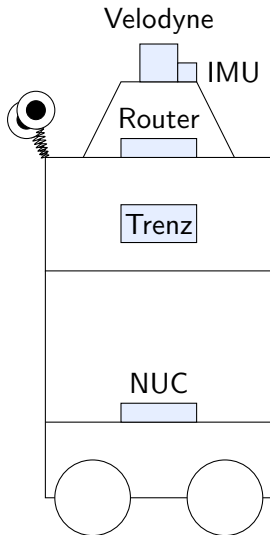
$$M_V = \frac{M_V \cdot M_W + v \cdot w}{M_W + w}$$

$$M_W = \min(M_W + w, W_{\max})$$

TODO: infos zu reg und tsdf, wie visuell darstellen?



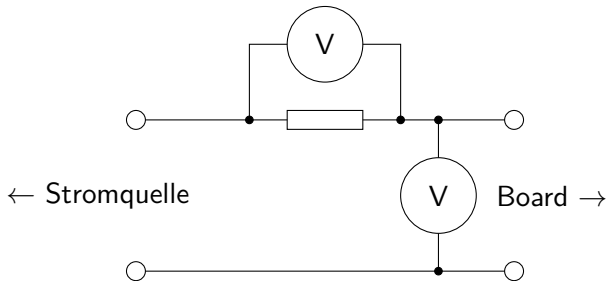
# FastSense Prototyp



TODO: Kommunikation von Julian



# Evaluation



TODO: Ergebnisse

Zeitmessung (ms)			
Abschnitt	Durchschnitt	Min	Max
Preprocessing	???	???	???
Registrierung	???	???	???
TSDf Update	???	???	???
Map Shift	???	???	???

Vergleich Vitis – Realität		
Abschnitt	Vitis	Gemessen
Registrierung	$0.9\text{ms} \cdot 100 = 90\text{ms}$	???
TSDF Update	477ms	???

Vergleich (Durchschnitt, ms)			
Programm	FastSense	Prototyp	Prototyp
System	Board	Board	NUC
Preprocessing	???	???	???
Registrierung	???	???	???
TSDF Update	???	???	???
Map Shift	???	???	???

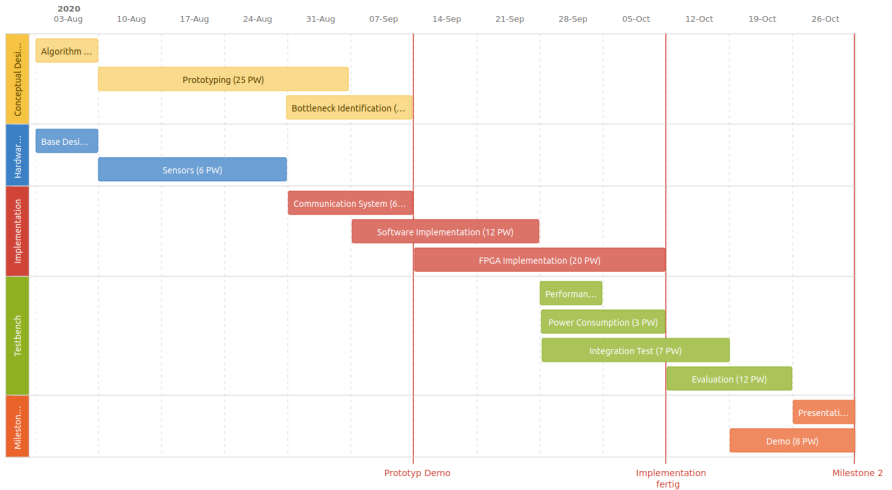
# Fazit

- Registrierung
  - Auslagerung von Point to TSDF auf Hardware
  - Auslagerung von Pointcloud Transformation auf Hardware
- TSDF

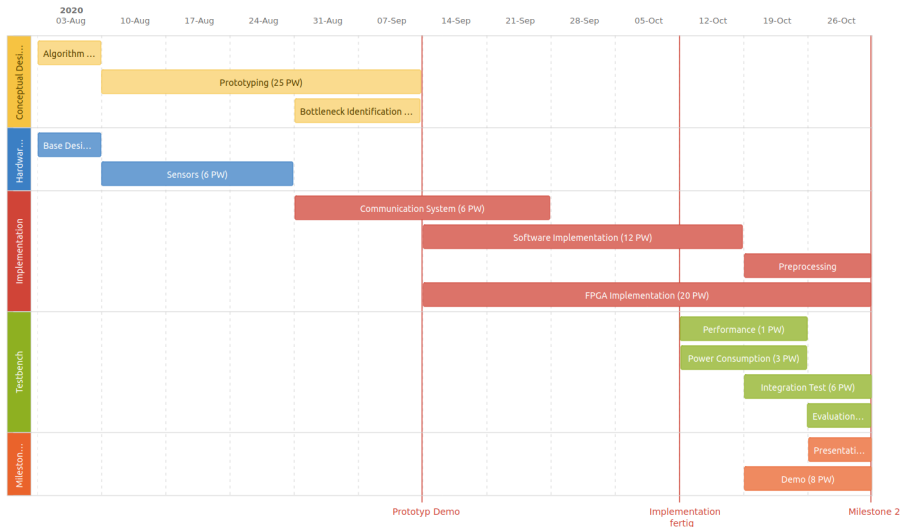


- Registrierung
  - Drift entfernen (aktuell noch leichter Drift (1cm/s) in alle 3 Richtungen)
  - Komplet in Hardware (Overhead ist fast dreimal so hoch wie der eigentliche Aufruf)

# Projektmanagement



# Projektmanagement



- Aufbau einer SLAM-Box mittels CAD
  - Nutzung als Sensor
  - Einfache Portierung zwischen Drohne, Roboter, Rucksack etc.
  - Festes Interface, einfache Bedienung, Kapselung
- Verbesserung und Optimierung des Algorithmus
- Mesh-Generierung auf Basis der TSDF Werte
- Loop Closing
- ???

TODO: Mehr Ideen für MS3?