Projektgruppe FastSense

Meilenstein 2

Hardware Accelerated TSDF SLAM

28. Oktober 2020

Inhalt

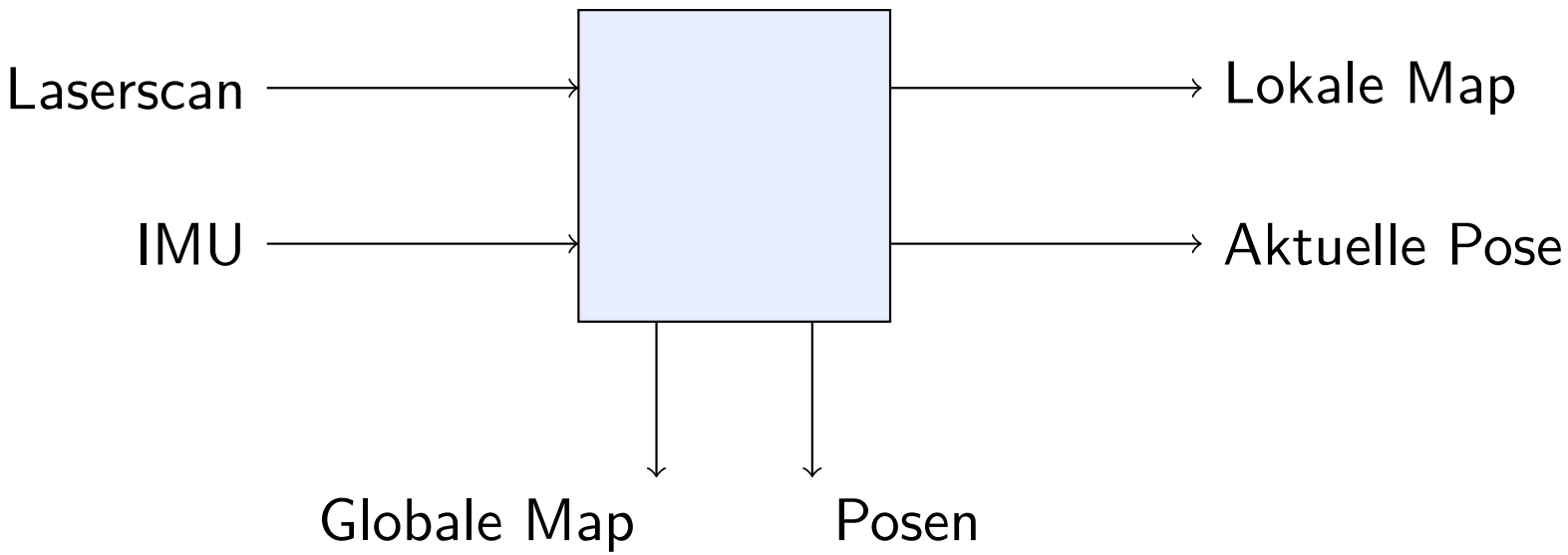
Ziele und Anforderungen

Ziele für MS2

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

Funktionale Anforderungen

- 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



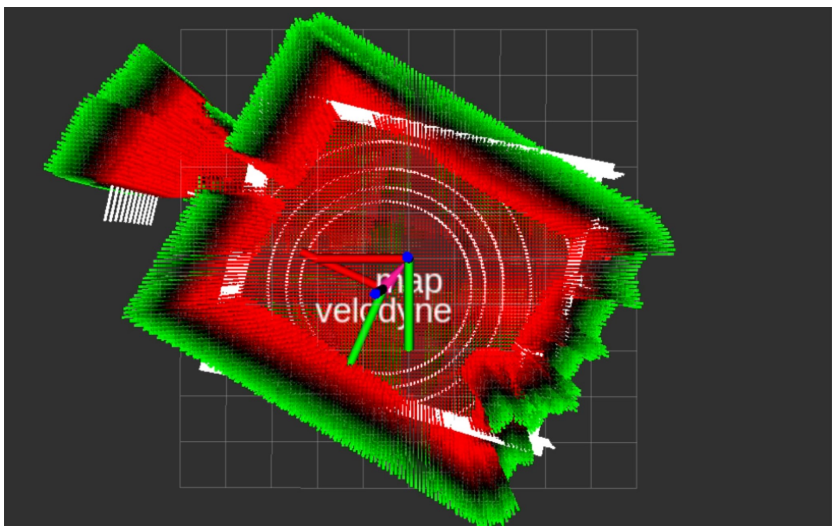
Nicht-Funktionale Anforderungen

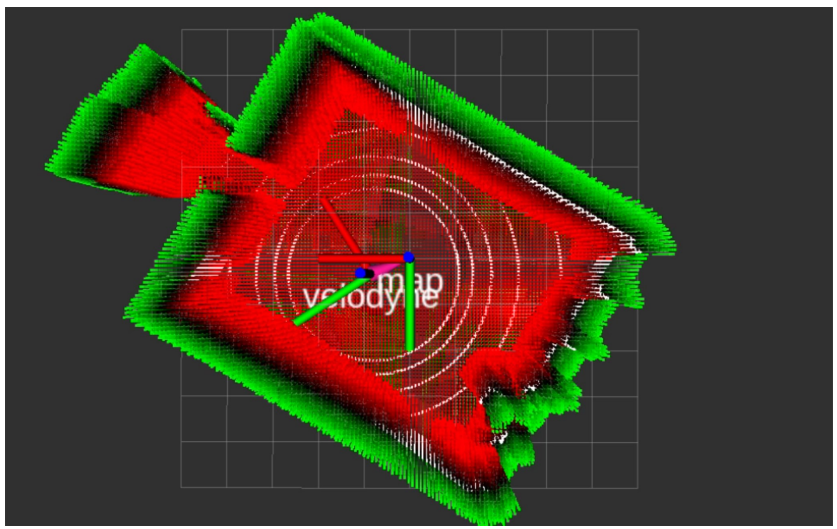
- HW-Plattform: Trenz-Board
- Entwicklungsplattform: Vitis
- Beschleunigung der Algorithmen durch FPGA
- Sensoren direkt am Board
- Unit-Tests
- Testbench
  - Integration, Strommessung, Zeitmessung, Visualisierung
- Logging

Implementierung

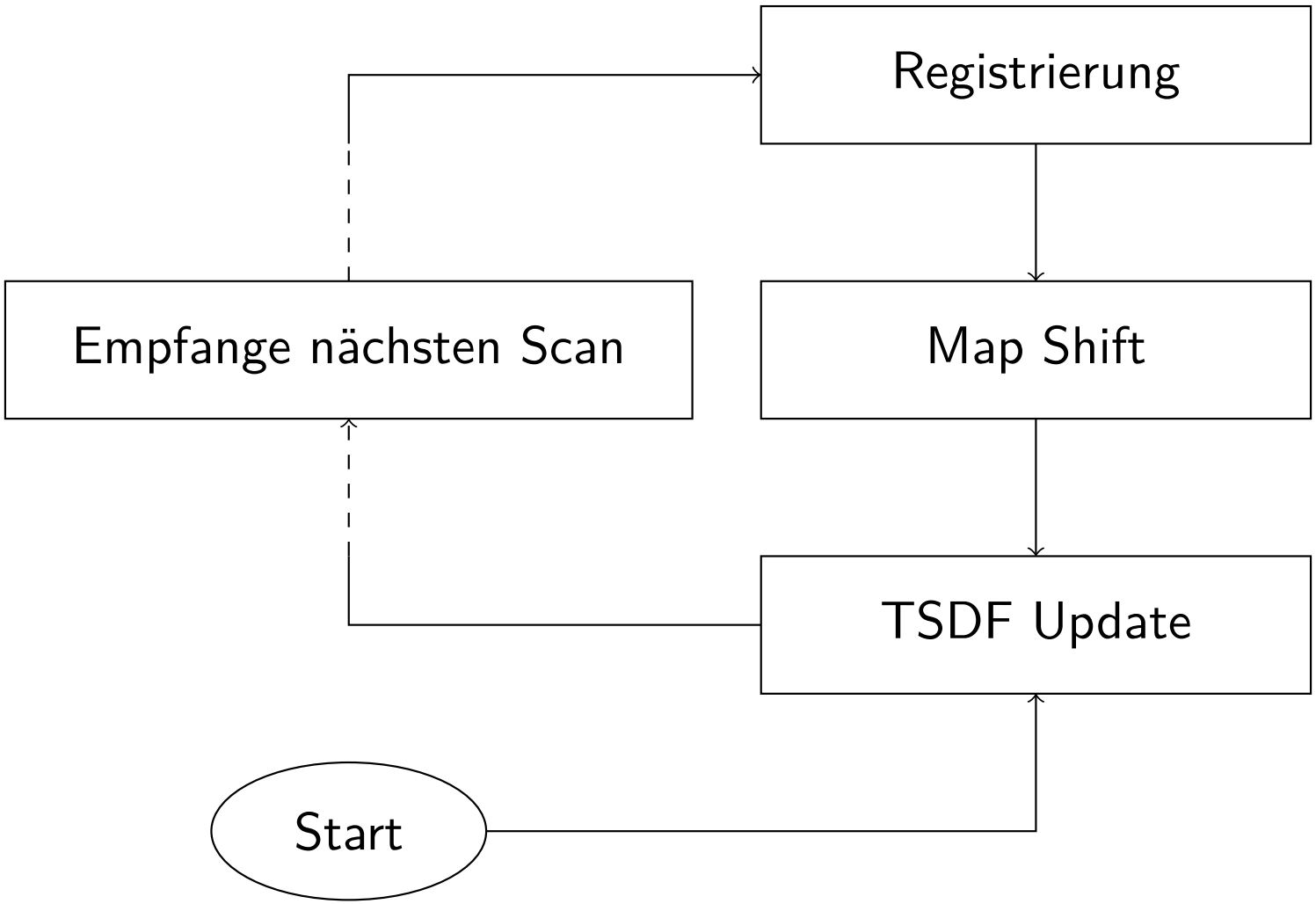
Recap: Prototyping Demo

- 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 (∼0.5s/Scan auf Glumanda, 2-5s/Scan in Testwelt)
  - Registrierung und TSDF-Update als parallelisierbare Bottlenecks

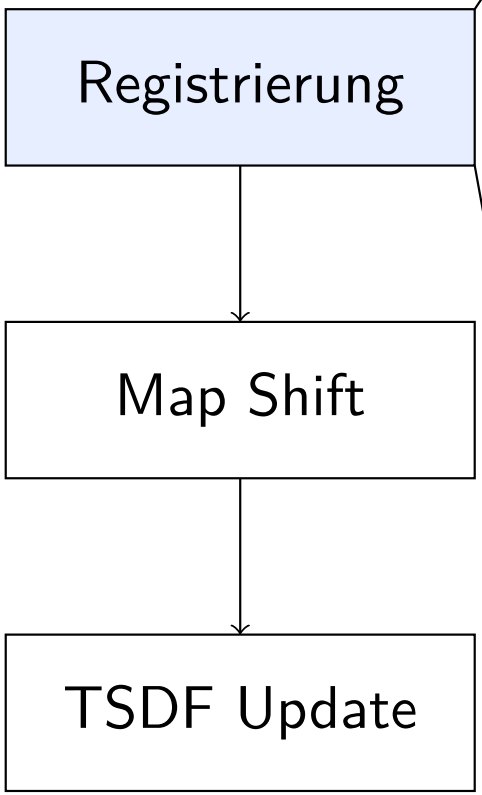




Algorithmus



Algorithmus



T = Initial Transform;

Transform point cloud with T;

do

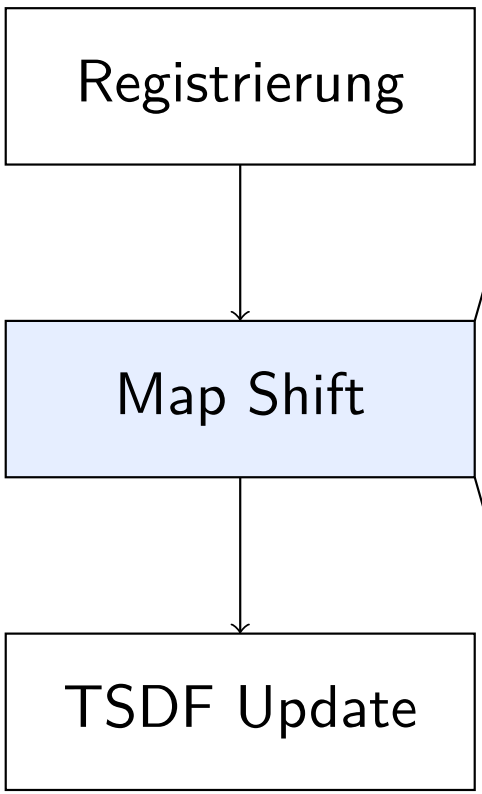
  E = Minimization of the sum of signed distances squared;

  Transform point cloud with $T(E)$;

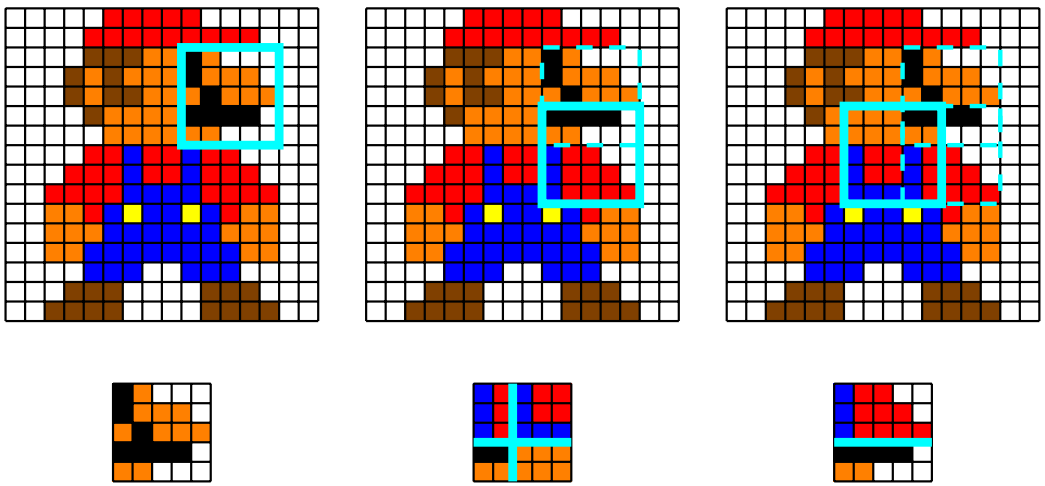
while *under max iterations and E not converged*;

return T;

Algorithmus

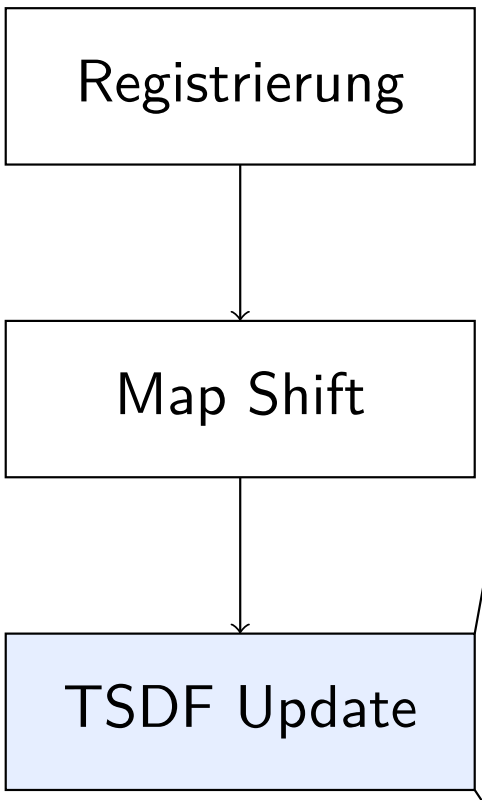


- LocalMap (aka RingBuffer)
  - Geteilt zwischen SW und HW

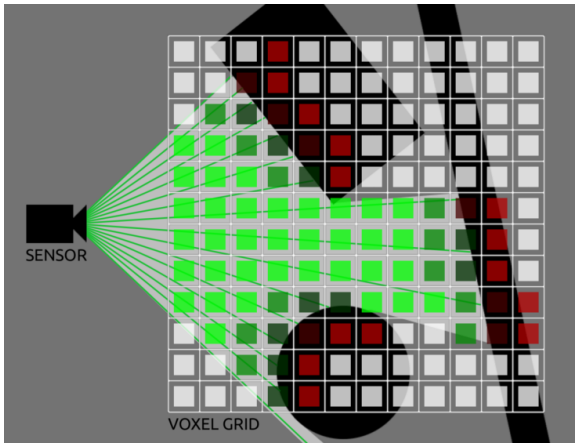


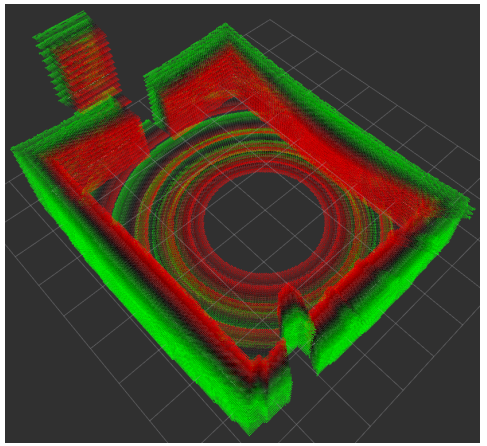
- Werte außerhalb in GlobalMap
  - Chunk-basiert
- Speicherung in HDF5

Algorithmus



1. TSDF Generierung





2. Update mit gewichtetem Mittelwert

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

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

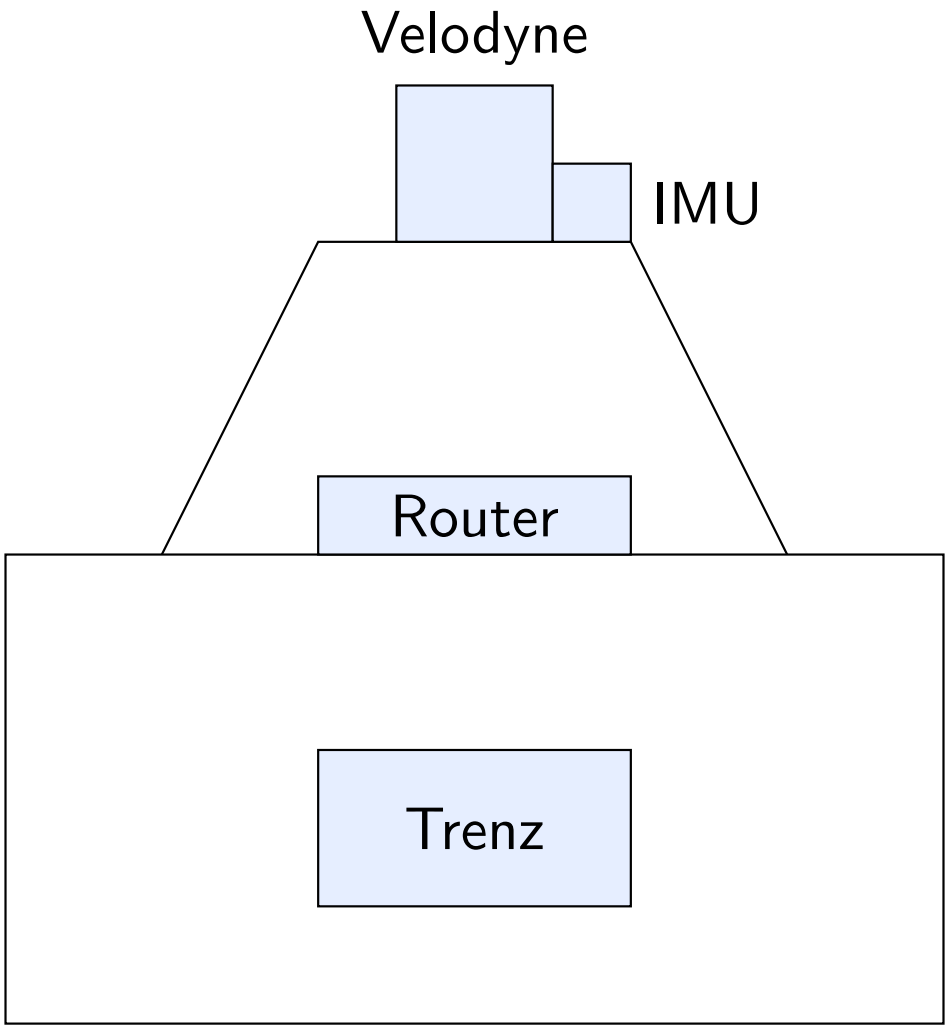
Registrierung

- Parallelisierbare Schritte der Schleife in HW (Berechnungen für alle Punkte)
- Matrixinvertierung in der Schleife bleibt in SW

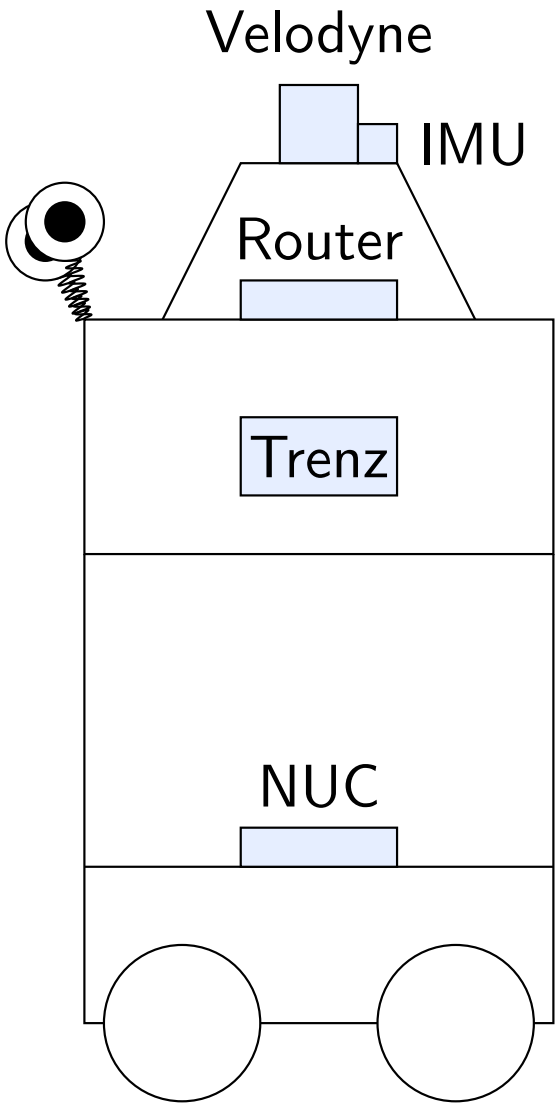
TSDF Update

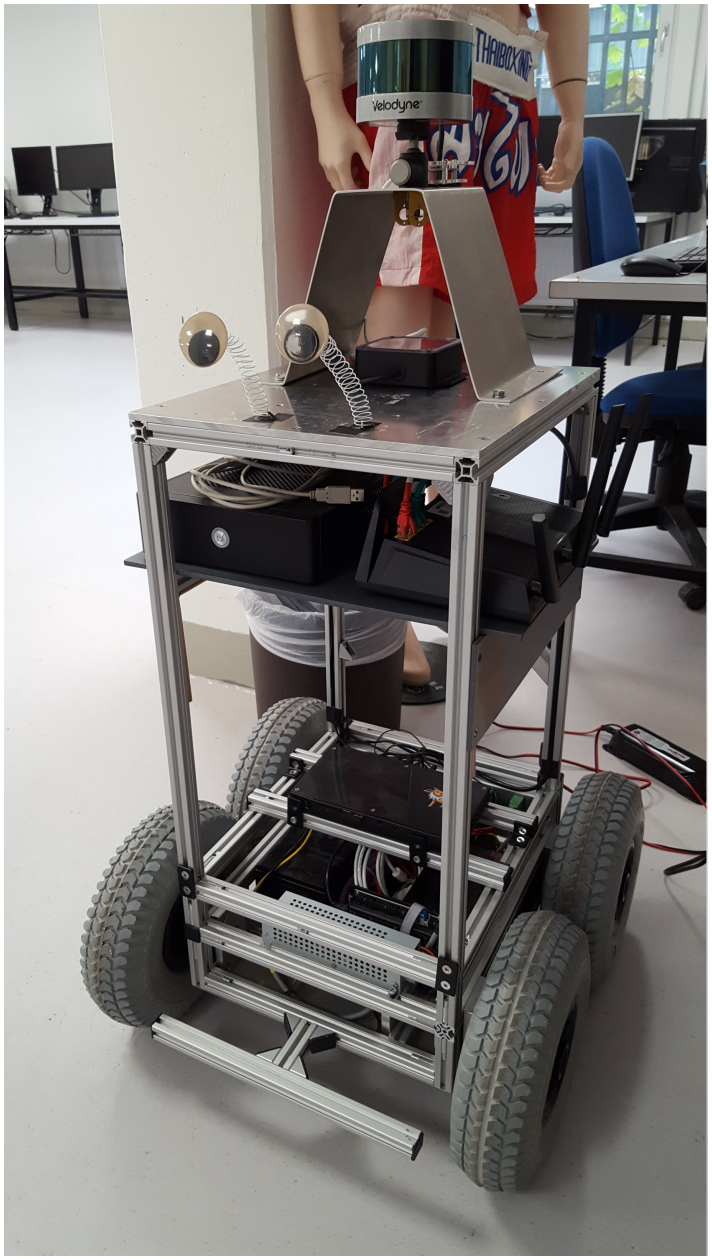
- Vollständig in HW
- Bresenham Algorithmus für die Iteration über Punkte entlang eines Strahls
- Synchronisation angepasst

FastSense Prototyp



FastSense Prototyp



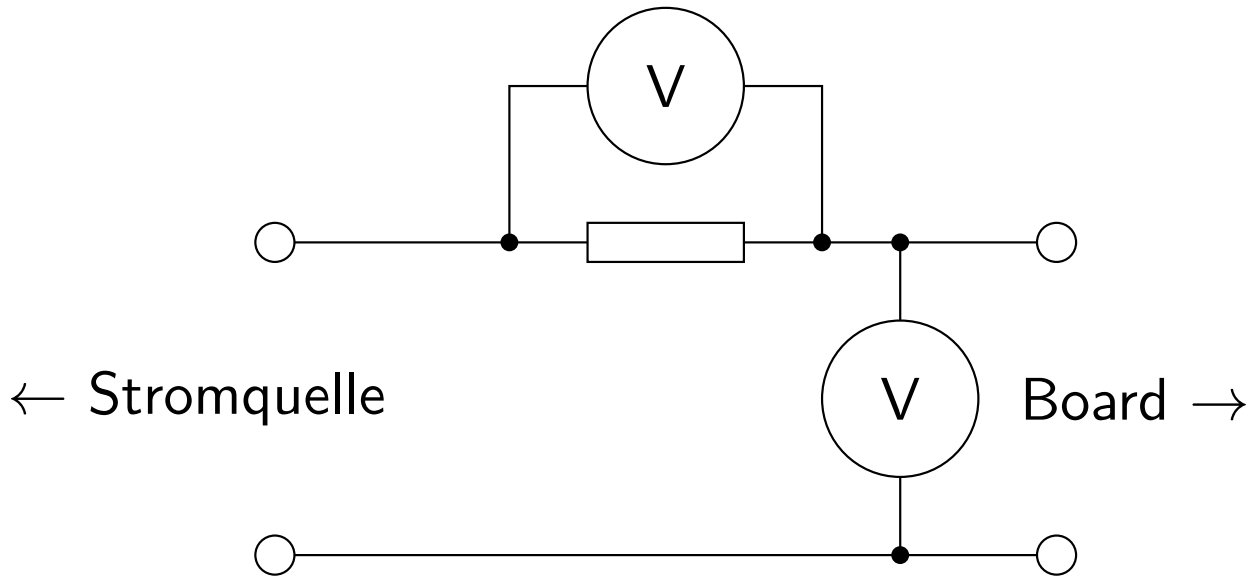


Kommunikation

TODO: Kommunikation von Julian

Evaluation

Strom



Strom

TODO: Ergebnisse

Zeit

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

Zeit

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

Zeit

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

Fazit

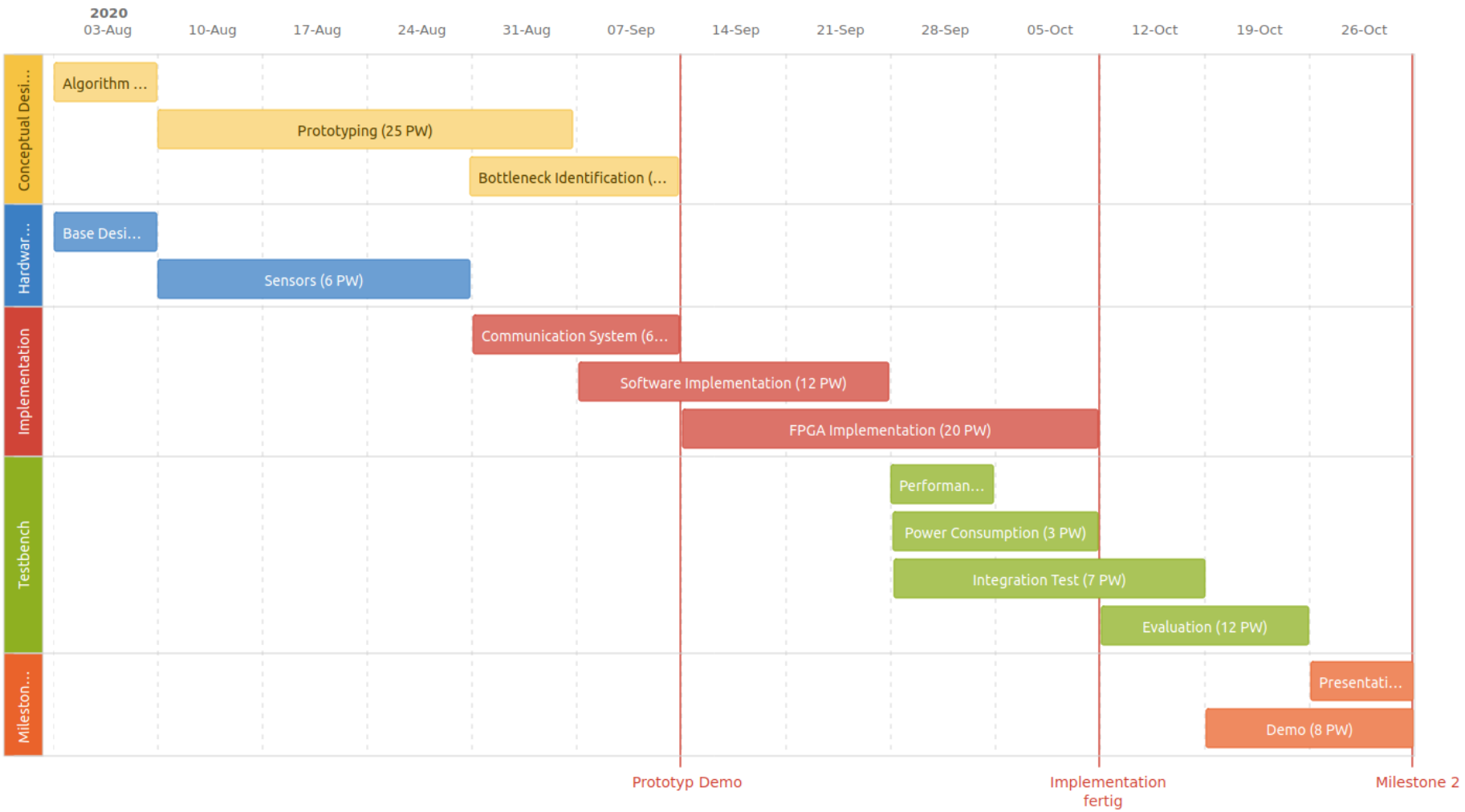
Bisherige Verbesserungen

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

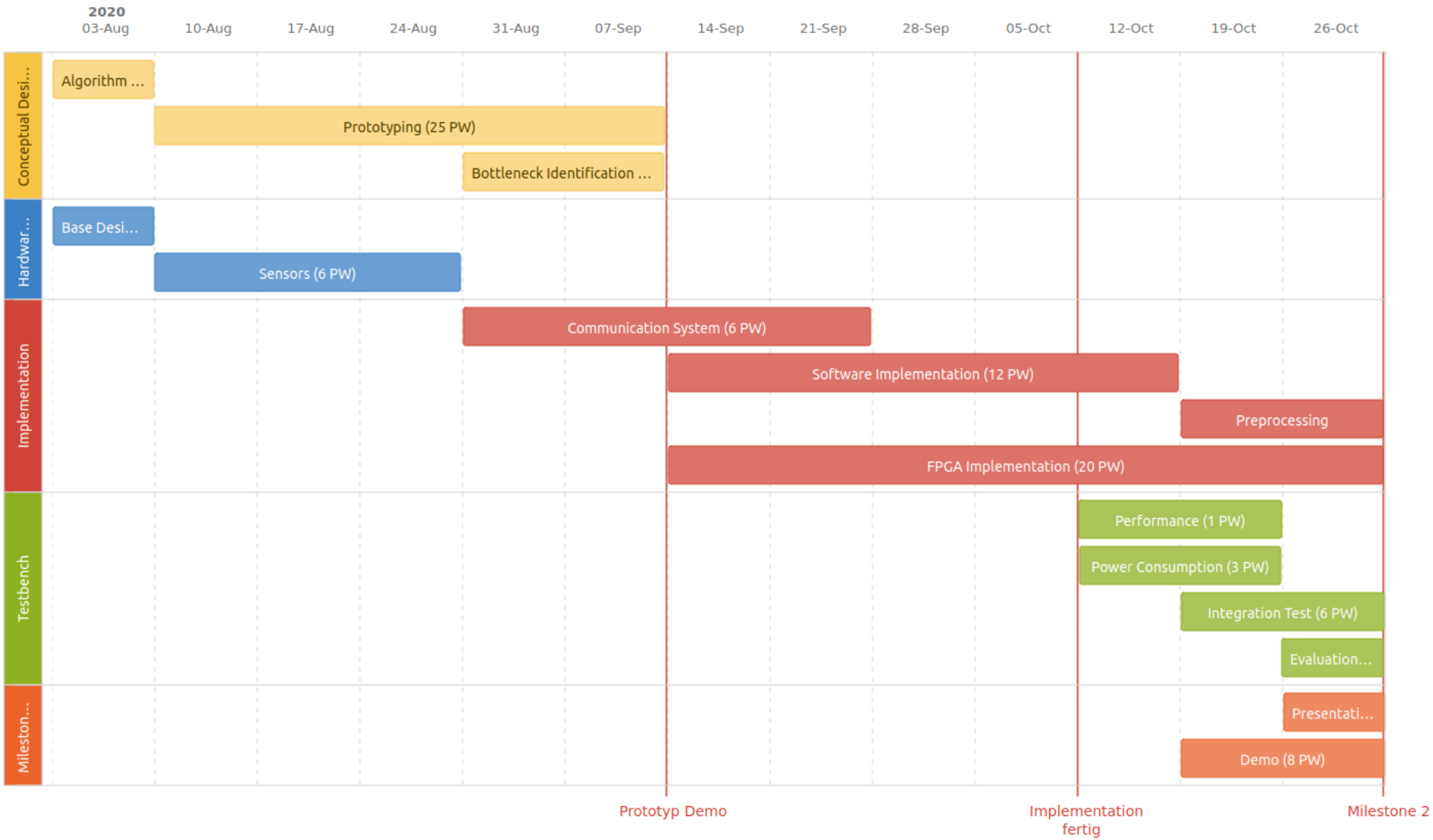
Verbesserungspotenzial

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

Projektmanagement



Projektmanagement



Ausblick / MS3

- 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?