

Projektgruppe FastSense

Meilenstein 2

Hardware Accelerated TSDF SLAM



28. Oktober 2020

Inhalt

Ziele und Anforderungen

Funktionale Anforderungen

Nicht-Funktionale Anforderungen

Implementierung

Recap: Prototyping Demo

Algorithmus

Aufbau Demo

Kommunikation

Live Demonstration

Evaluation

Strom

Zeit

Fazit

Verbesserungspotenzial

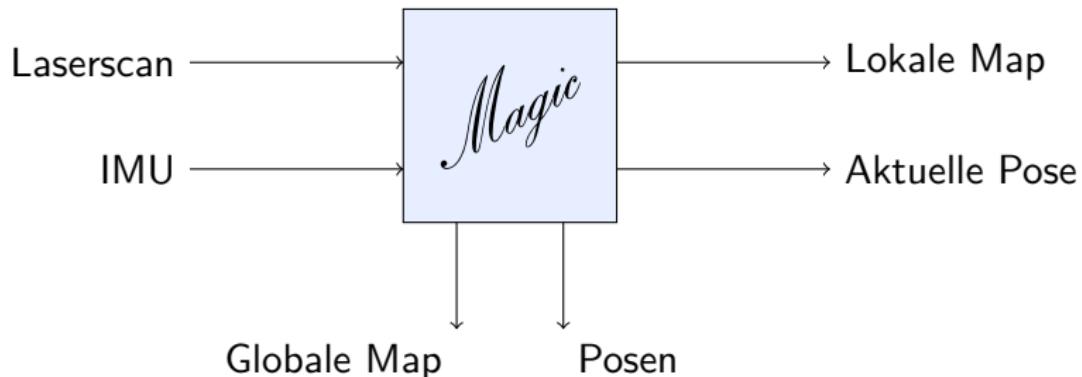
Projektmanagement

Ausblick / MS3

Ziele und Anforderungen

Funktionale Anforderungen

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



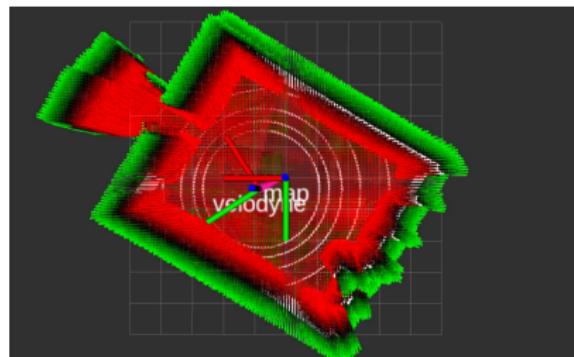
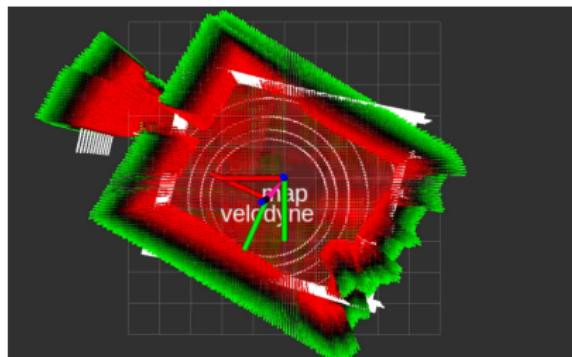
Nicht-Funktionale Anforderungen

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

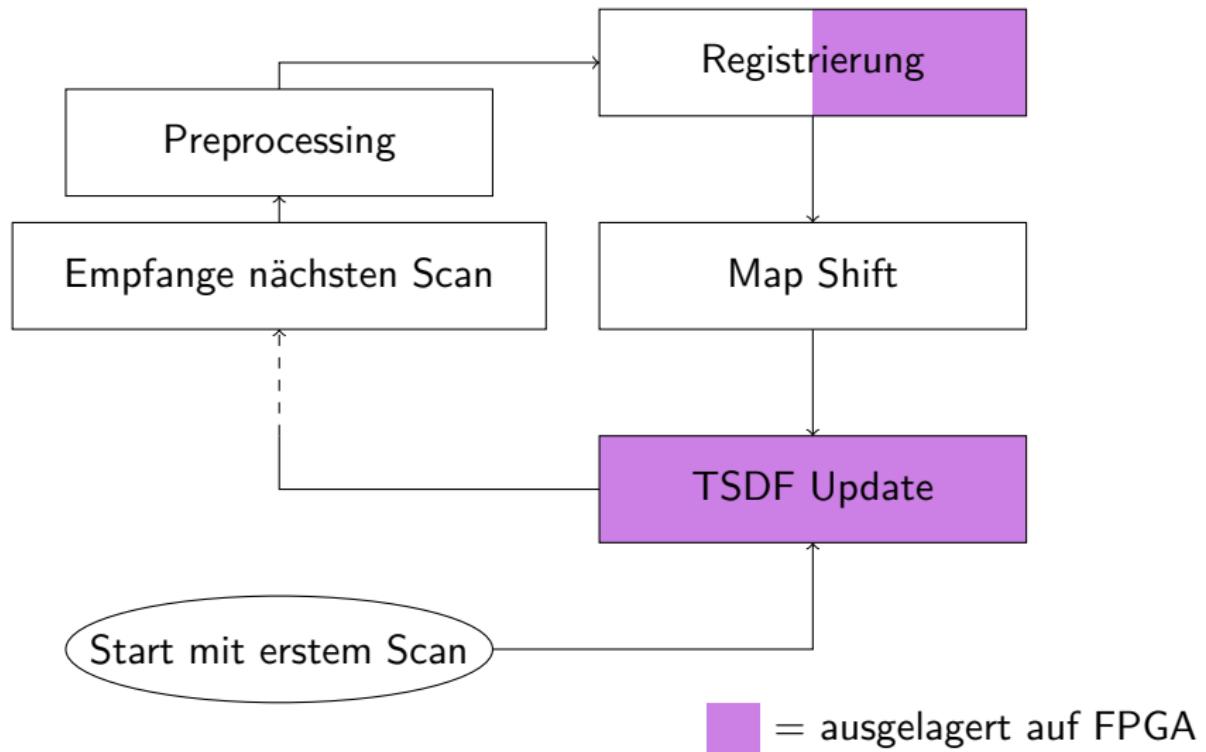
Implementierung

Recap: Prototyping Demo

- Gute Parameterkombination herausgefunden
- Geplante Funktionalität war vorhanden und in RViz darstellbar
- Probleme / Bottlenecks erkannt

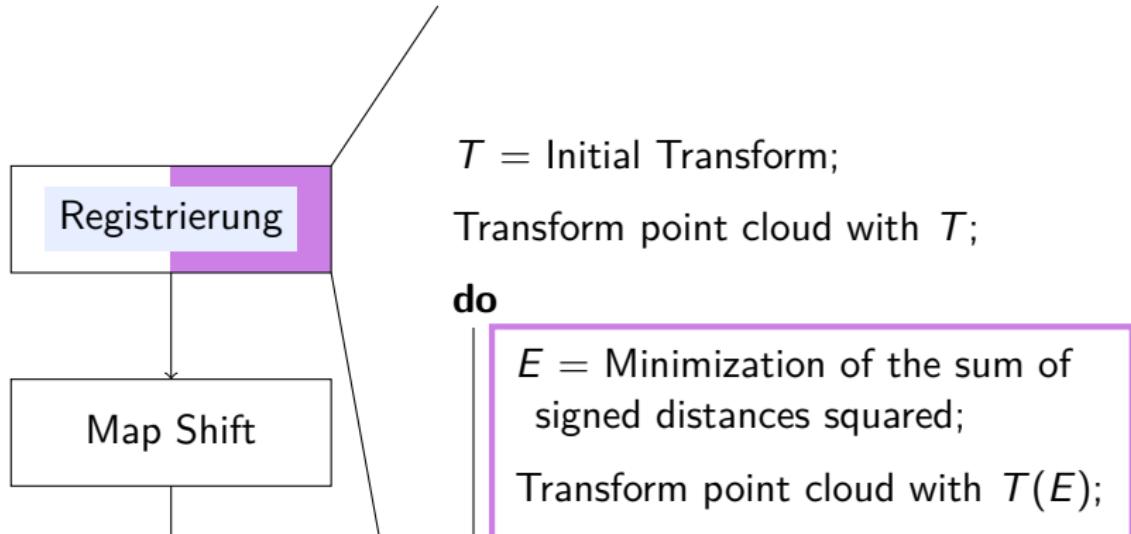


Algorithmus



= ausgelagert auf FPGA

Algorithmus



$T = \text{Initial Transform};$

Transform point cloud with T ;

do

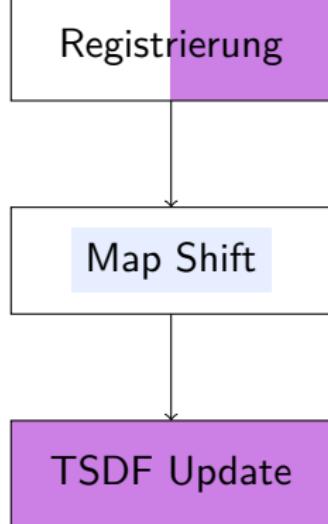
$E = \text{Minimization of the sum of signed distances squared};$

Transform point cloud with $T(E)$;

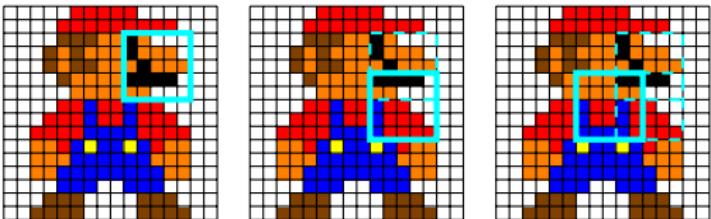
while *under max iterations and E not converged*;

return T ;

Algorithmus

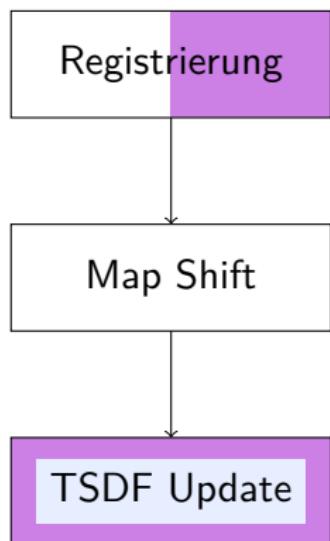


- LocalMap (aka RingBuffer)
 - Geteilt zwischen SW und HW

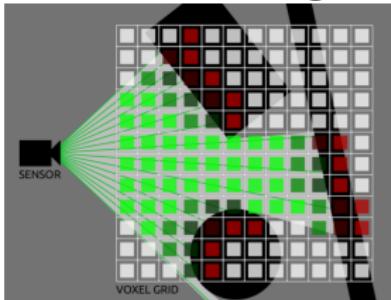


- Werte außerhalb in GlobalMap
 - Chunk-basiert
- Speicherung in HDF5

Algorithmus



1. TSDF Generierung

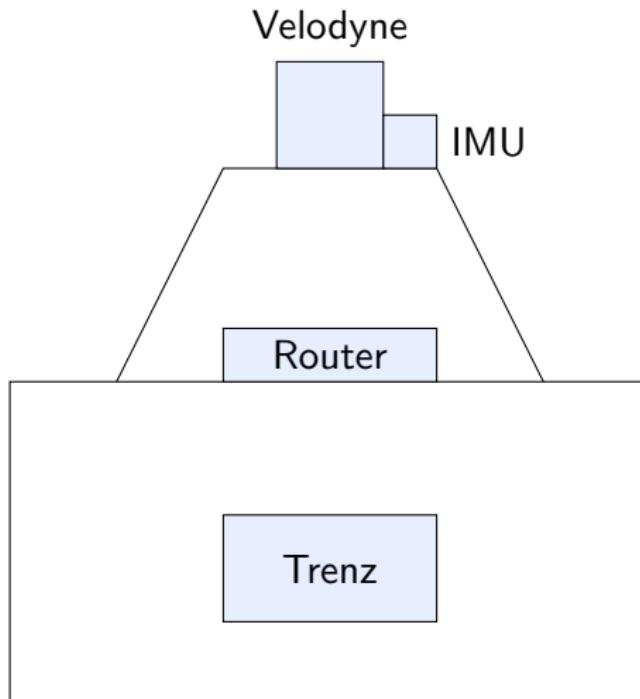


2. Update mit gewichtetem Mittelwert

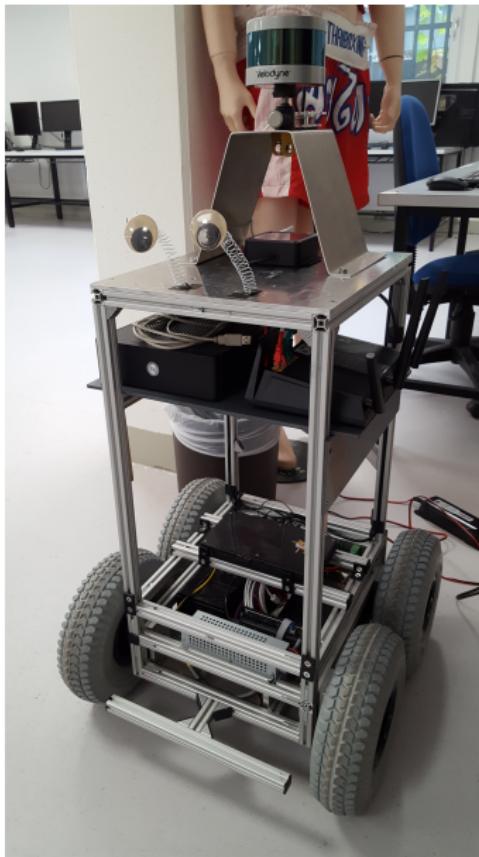
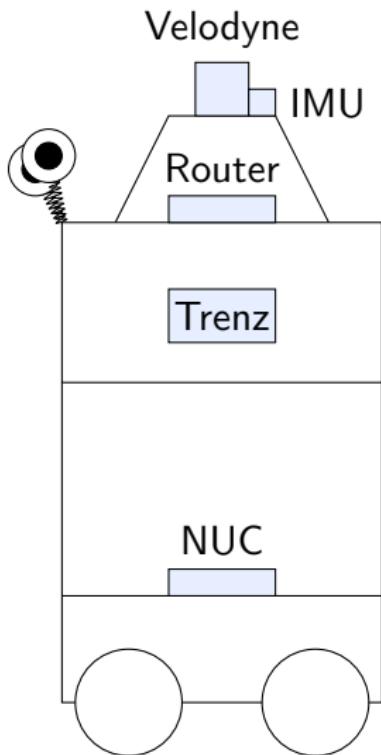
$$M_V = \frac{M_V \cdot M_W + v \cdot w}{M_W + w}$$

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

Aufbau Demo

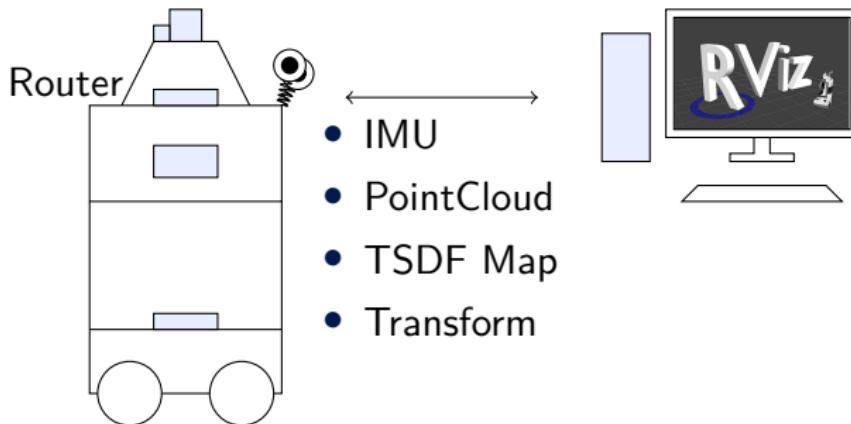


Aufbau Demo



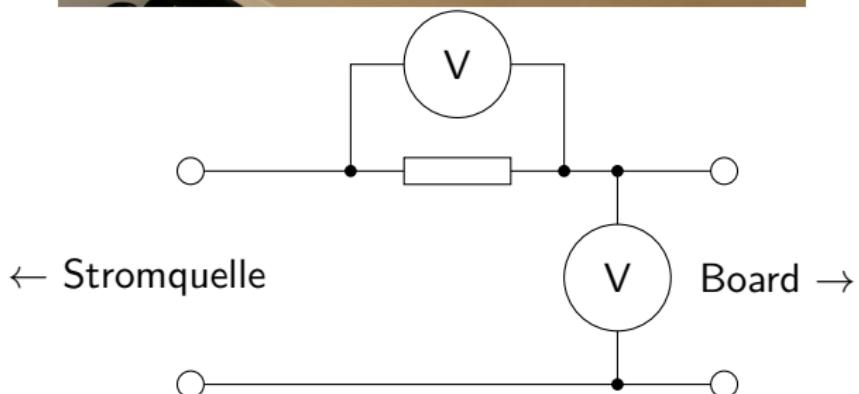
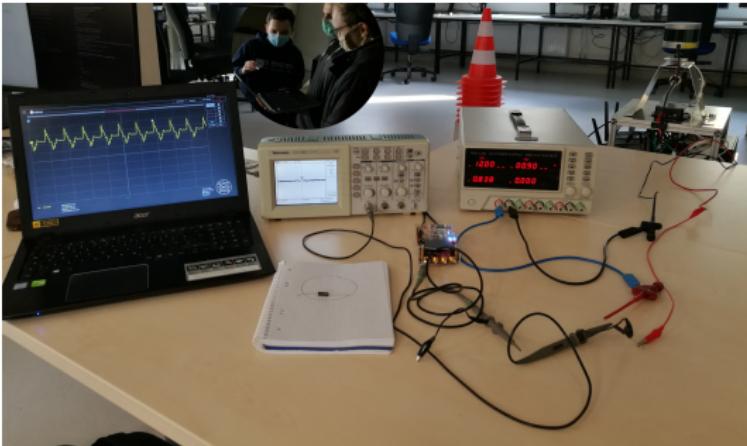
Kommunikation

- Kommunikation via „Bridge“
- Datenaustausch über TCP, basierend auf ZeroMQ

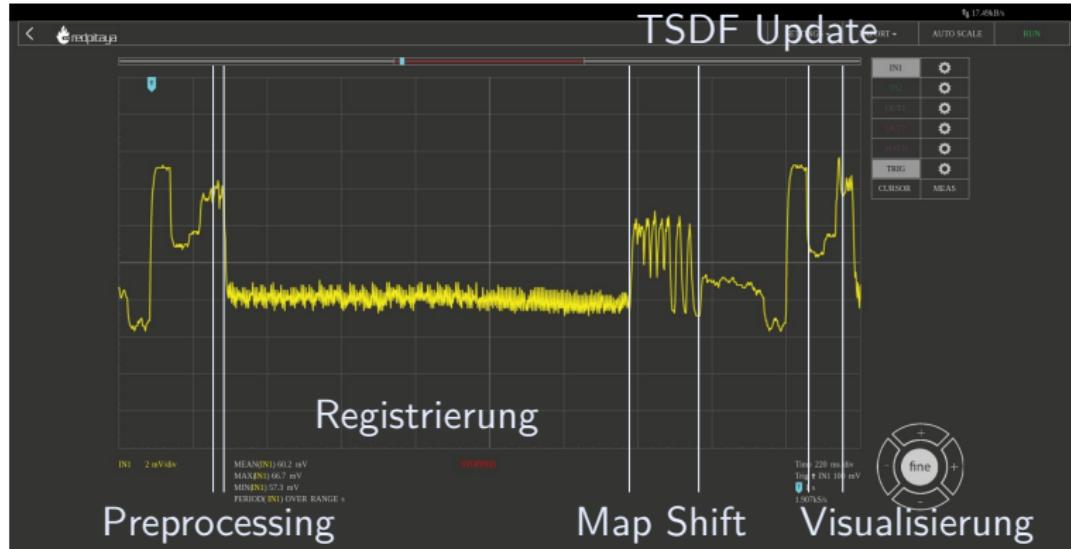


Evaluation

Strom



Strom



	Durchschnitt	Min	Max
U [mV]	60.2	57.3	66.7
I [A]	0.86	0.82	0.95
P [W]	10.32	9.84	11.4

Zeit

Zeitmessung [ms]			
Abschnitt	Durchschnitt	Min	Max
Preprocessing	35	24	51
Registrierung	676	142	2894
Map Shift	105	0	1681
TSDF Update	337	319	346

Zeit

Vergleich Vitis – Realität [ms]		
Abschnitt	Vitis	Gemessen
Registrierung	0.905	3.667
TSDF Update	124.483	552.924

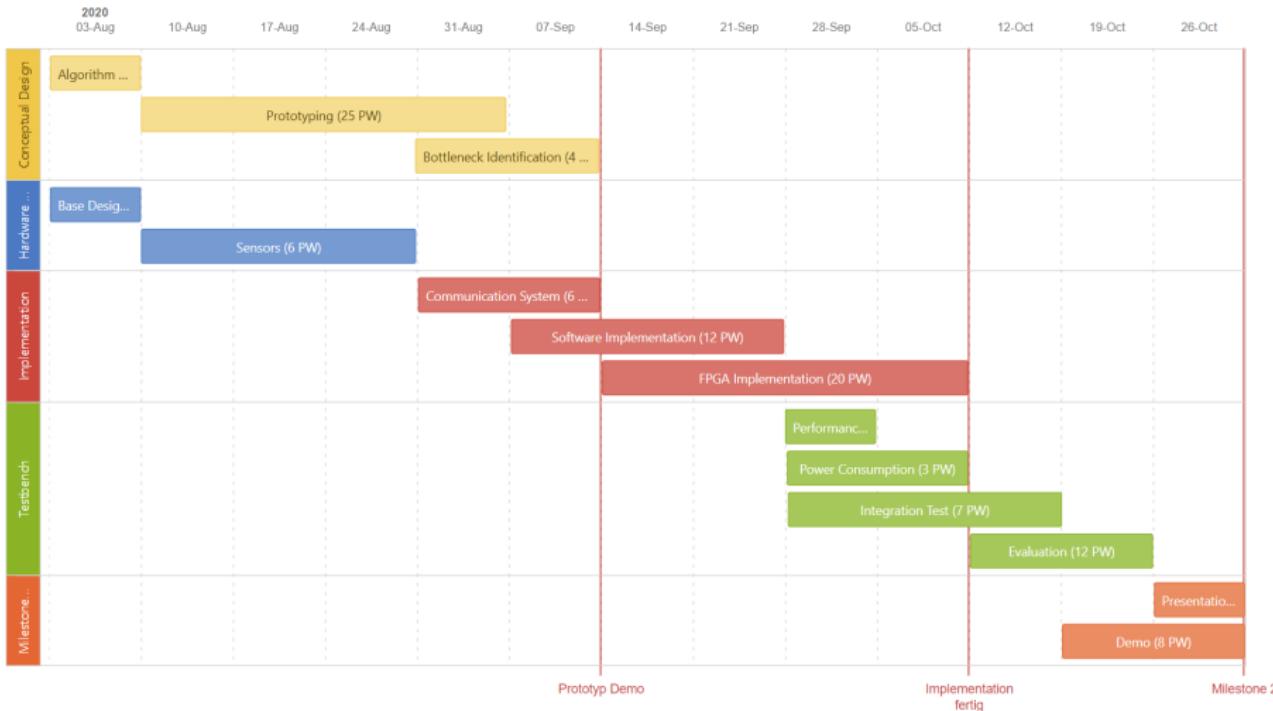
- Bei der Registrierung wird die meiste Zeit auf Speicher gewartet (95%)

Fazit

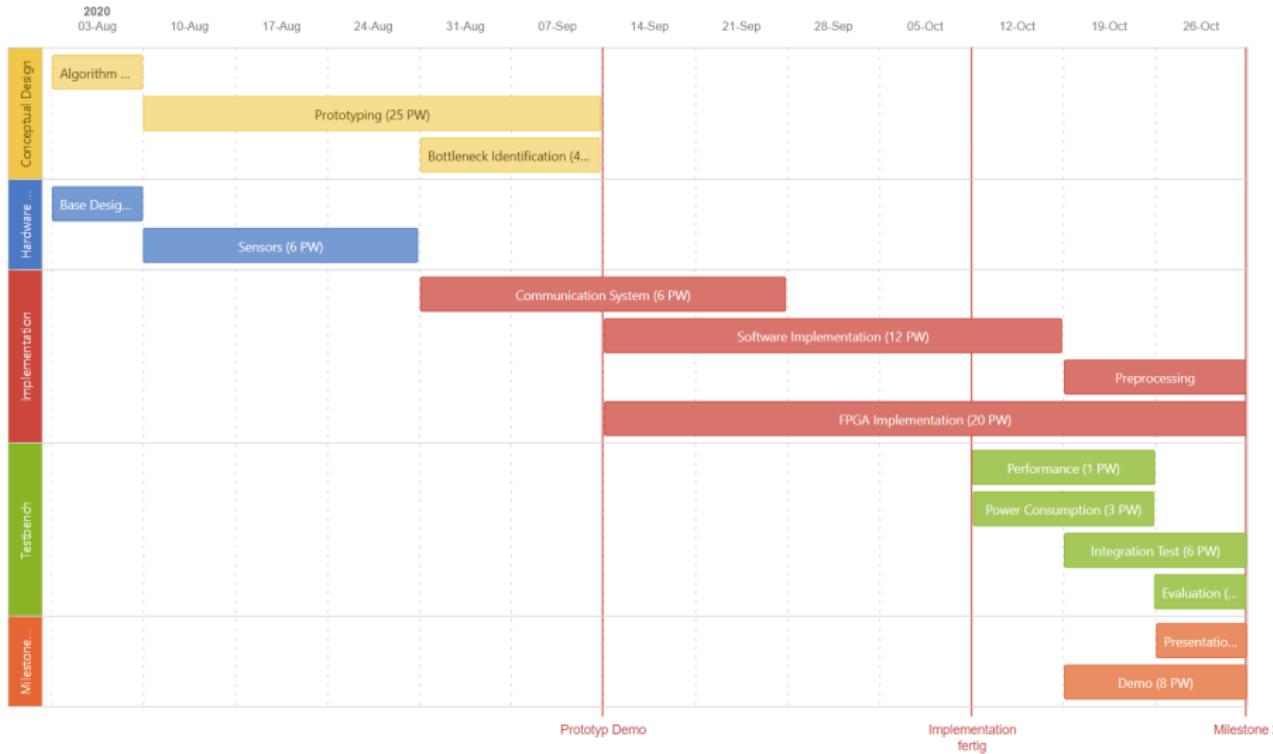
Verbesserungspotenzial

- Registrierung
 - Drift entfernen (aktuell 1cm/s in alle 3 Richtungen)
 - Komplett in Hardware (Overhead ist fast dreimal so hoch wie der eigentliche Aufruf)
 - Registrierung hat Probleme mit Bresenham-Approximation der TSDF-Werte
 - Stabilisierung durch trilineare Interpolation
- TSDF
 - Alternative zum Bresenham Algorithmus
 - Voraussetzung: Effiziente Umsetzbarkeit in Hardware

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
- Integrierung von GPS

Quellen

- [1] D. R. Canelhas, T. Stoyanov, and A. J. Lilienthal. "SDF Tracker: A parallel algorithm for on-line pose estimation and scene reconstruction from depth images". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 3671–3676. DOI: [10.1109/IROS.2013.6696880](https://doi.org/10.1109/IROS.2013.6696880).