

Forscherguppe SONARO

Smarte Objektübernahme und –übergabe für die
nutzerzentrierte mobile Assistenzrobotik

4. Workshop / Beiratstreffen
am 28.10.2020

www.sonaro-projekt.de



Agenda

13:00	Prof. Groß	Begrüßung der Teilnehmer und Einführung zu den Eckdaten des Projektes im 4. Halbjahr
13:10	Dr. Müller	Fortschrittsbericht und Ausblick zu den Arbeitspaketen von TUI-NIKR
13:40	M.Sc. Zhang	Fortschrittsbericht und Ausblick zu den Arbeitspaketen von TUI-QBV
14:05	Dr. Garten	Fortschrittsbericht und Ausblick zu den Arbeitspaketen von GFE
14:25	M.Sc. Schneider	Fortschrittsbericht und Ausblick zu den Arbeitspaketen von HSM
14:45	Prof. Groß	Diskussion mit dem Beirat zur Schwerpunktsetzung und Vorgehensweise

Vorstellung - SONARO Forschergruppe

TU Ilmenau,
FG Neuroinformatik und
Kognitive Robotik
TUI-NIKR



Prof. Groß



Dr. Müller



TECHNISCHE UNIVERSITÄT
ILMENAU

TU Ilmenau,
FG Qualitätssicherung und
Industrielle Bildverarbeitung
TUI-QBV



Prof. Notni



M. Sc. Zhang



TECHNISCHE UNIVERSITÄT
ILMENAU



Hochschule
Schmalkalden,
FG Eingebettete
Diagnosesysteme **HSM**



Prof. Wenzel



M. Sc. Schneider



Gesellschaft für
Fertigungstechnik und
Entwicklung Schmalkalden
e.V. **GFE**



Dr. Welzel



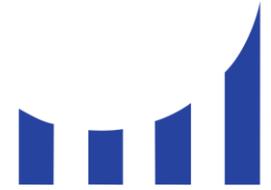
Dr. Garten

SONARO Unternehmensbeirat

- ✓ Dr. Trabert, Metralabs GmbH, Ilmenau
- ✓ Herr Sommer, SCS Robotik GmbH, Schmalkalden
- ✓ Herr Waldmann, Betriebsleiter HENKEL+ROTH GmbH, Ilmenau
- ✓ Herr Hofmann, Cluster für Fertigungstechnik & Metallbearbeitung
- ✓ Dr. Behling, Götting KG, Lehrte
 - Herr Buchanan, Vision & Control GmbH, Suhl
 - Honda Research Institute Europe
 - Hörisch Präzision GmbH

Vorstellung des Projektträgers

Thüringer Aufbaubank (TAB)



Thüringer Aufbaubank

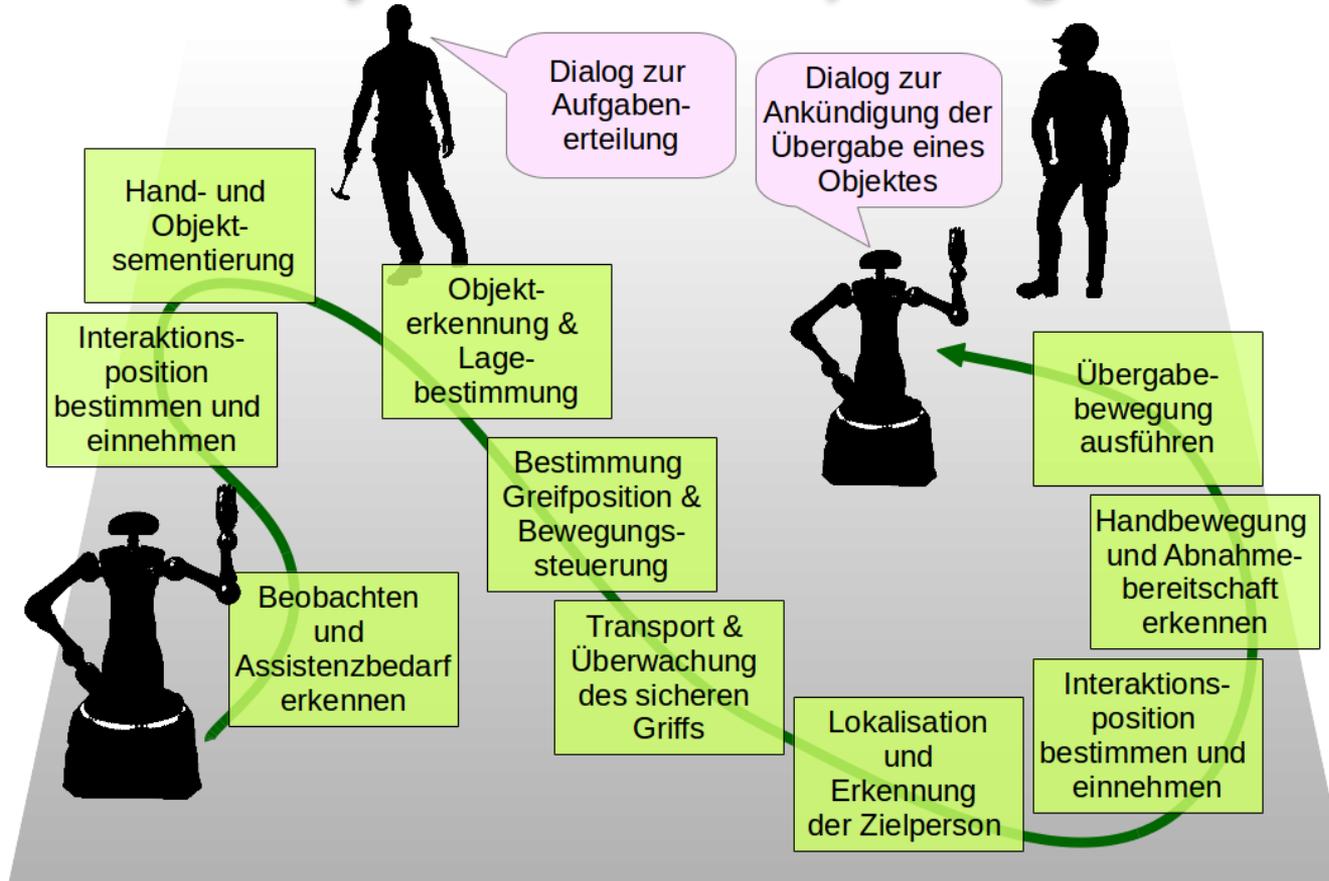


- Dr. Djambova
(fachlich inhaltliche Projektbetreuung)

Eckdaten der Forschergruppe

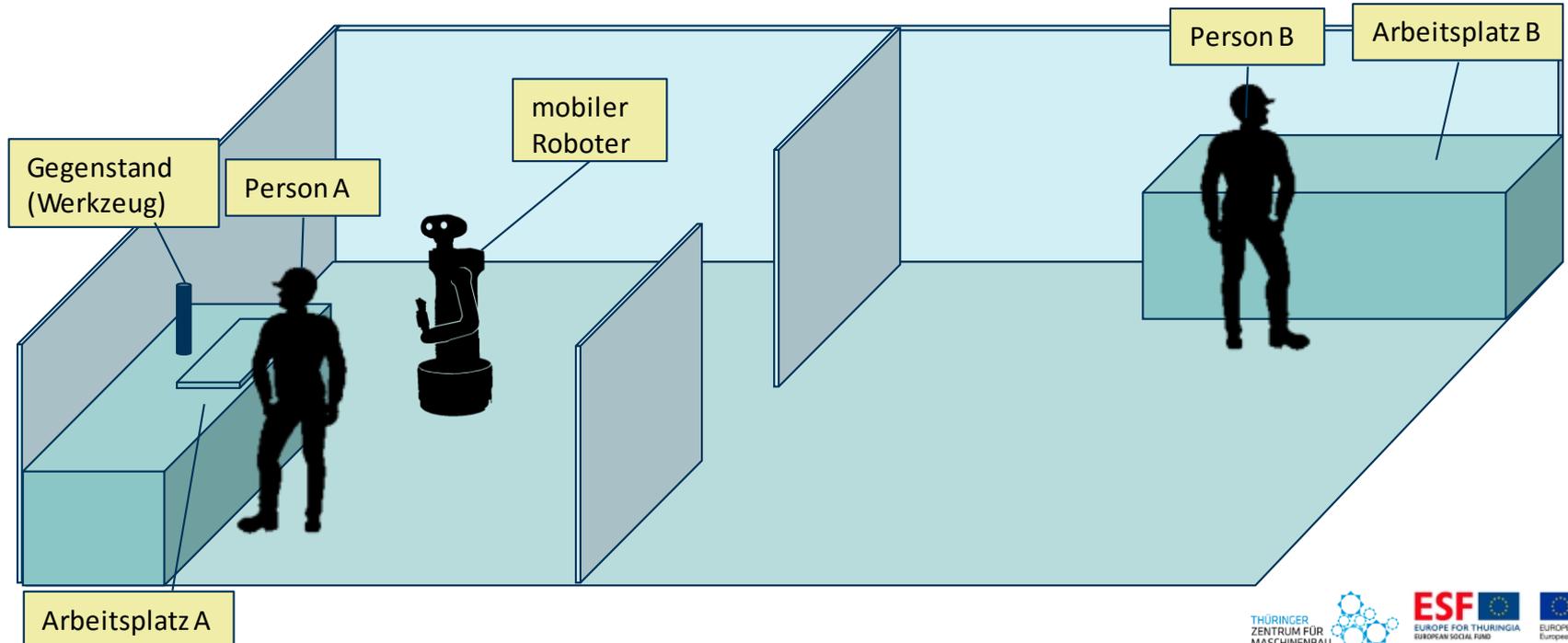
- ThZM als Antragsteller
- Koordinator: TU-Ilmenau FG NI&KR (Prof. Groß)
- Laufzeit 33 Monate seit 1.4.2019
- Fördersumme: 699.020 EUR
- 4 Mitarbeiter sind anteilig eingestellt
- ThZM stellt Mittel für Forschungstechnik (Roboter, Manipulatoren) zur Verfügung

Ablauf einer Objektübernahme/-übergabe



Demonstrationsszenario: Einsatz in der Produktion

- Zwei oder mehr Montageplätze an denen Werker Montagearbeiten ausführen
- **Bedarf:** Werkzeug/Werkstück wird durch den Roboter von Person A übernommen und zu Person B gebracht und übergeben



Studentische Arbeiten im Jahr 2020

- **TUI-NIKR:**



MA Markus Paschke: *Deep-Learning-Methoden zur Erstellung von, 3D-Modellen, TU-Ilmenau 2020*

- **BA Andreas John:** *Einbindung der Flexible Collision Library in einen Bewegungsplaner für das Greifen von Objekten*
- **Projektseminar** (zwei Bearbeiter): *Aufbau einer Testumgebung für robotisches Greifen*
- **MA Patick Stäblein:** *Integration eines Objekt-/Handtracking-Ansatzes in die reale Roboterapplikation*
- **MA Max Kogel:** *Visuelle Intenterkennung für die Mensch Roboter Kollaboration*

- **TUI-QBV:**



MA Jinggang Yan: *Untersuchung zur Nutzbarkeit des Convolutional-Neural-Networks für die echtzeitfähige Objektdetektion auf Basis von Farb- und 3D-Bilddaten.*

- **Projektarbeit Yujian Yuan, Xiaojiang Han und Jinxin Zhu:** *Untersuchung zum Verfahren der Kalibrierung und der Registrierung von 3D-Kamera und Thermokamera*
- **Projektarbeit Yang Li, Xiao Lin und Jingyu Wang:** *Die Entwicklung der effizienten teilweise automatischen Annotationsapplikation für die multimodale Punktwolke.*

- **HSM:**

- **Masterprojekt Sasi Kumar Selvaraj:** *Implementation for RTLS localisation in ROS and rviz*
- **Masterprojekt Stephan Hintz, Adrian Herrman, Nicolas Schmitt:** *Messfehlerbestimmung im 3D-Raum von UWB Signalen zur Indoor-Navigation mit RTLS*

Forscherguppe SONARO

Smarte Objektübernahme und –übergabe für die
nutzerzentrierte mobile Assistenzrobotik

Beiratstreffen am 28.10.2020

Vorstellung der Arbeiten von TU-Ilmenau NI&KR

Dr. Steffen Müller

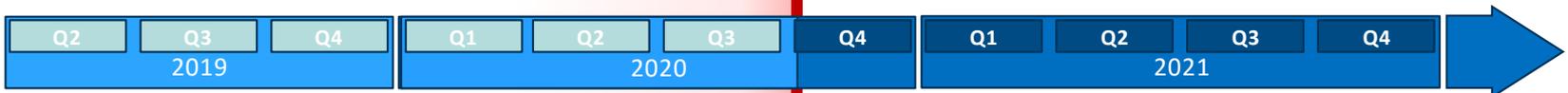
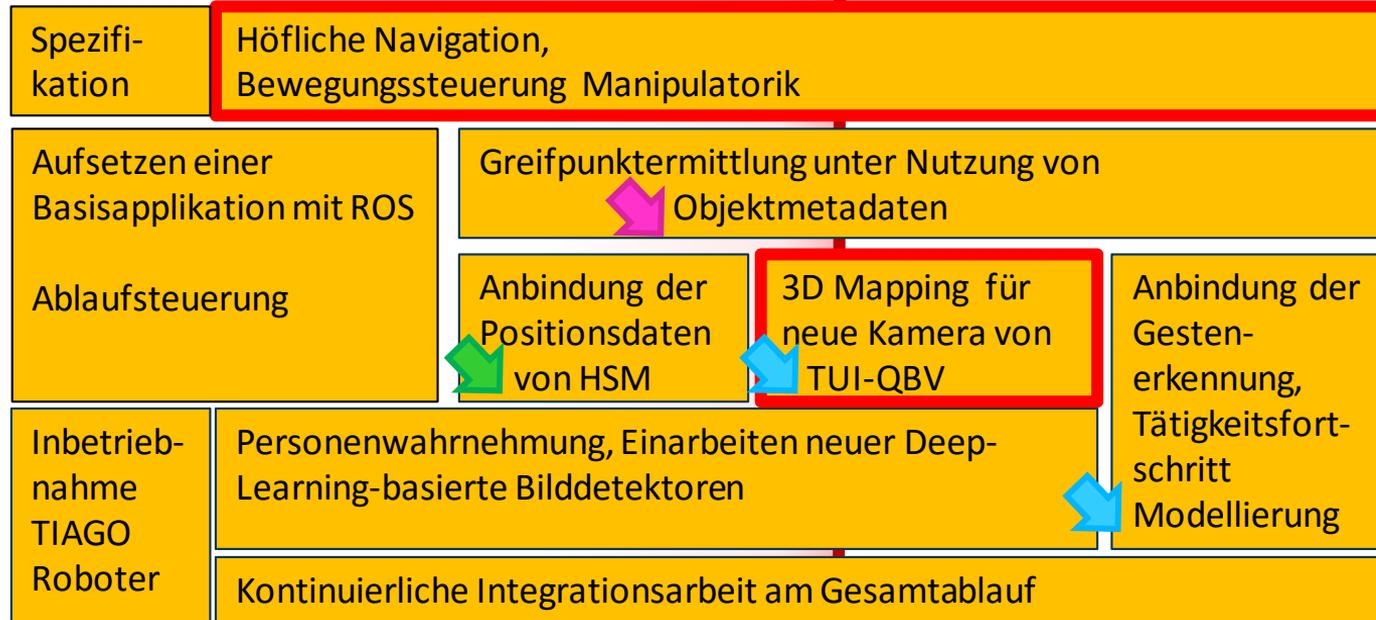
www.sonaro-projekt.de



Arbeitsplan für TUI-NIKR über die Projektlaufzeit

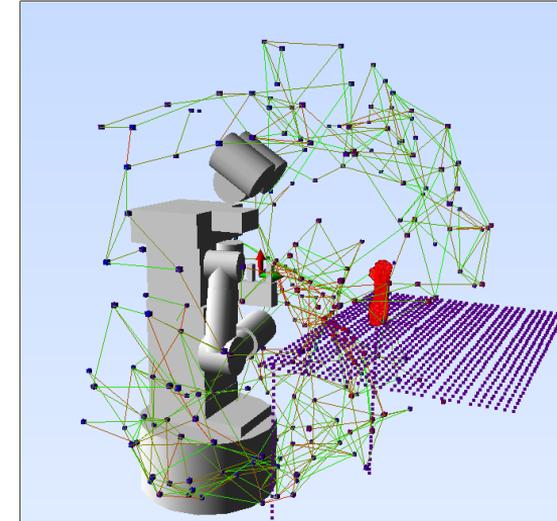
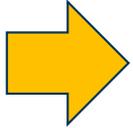
Zuarbeiten von:

-  HSM
-  TUI-QBV
-  GFE



Arbeitspaket Navigation & Bewegungssteuerung

- **Ausgangspunkt:**
 - Bewegungsplaner für dynamische Umgebung und Ziele für Tiago Roboter implementiert
 - **Offene Punkte:**
 - Integration der Sensormesswerte in aktuelles Modell der zu vermeidenden Hindernisse
 - Modellierung des gegriffenen Objekts nach dem Zugriff (Berücksichtigung als Teil des beweglichen Roboterarms für die Hindernisvermeidung)
 - Planung auf Graph bislang nur kürzester Pfad (keine Berücksichtigung von Rewards anderer Zielfunktionen oder mehrerer Zielposen)



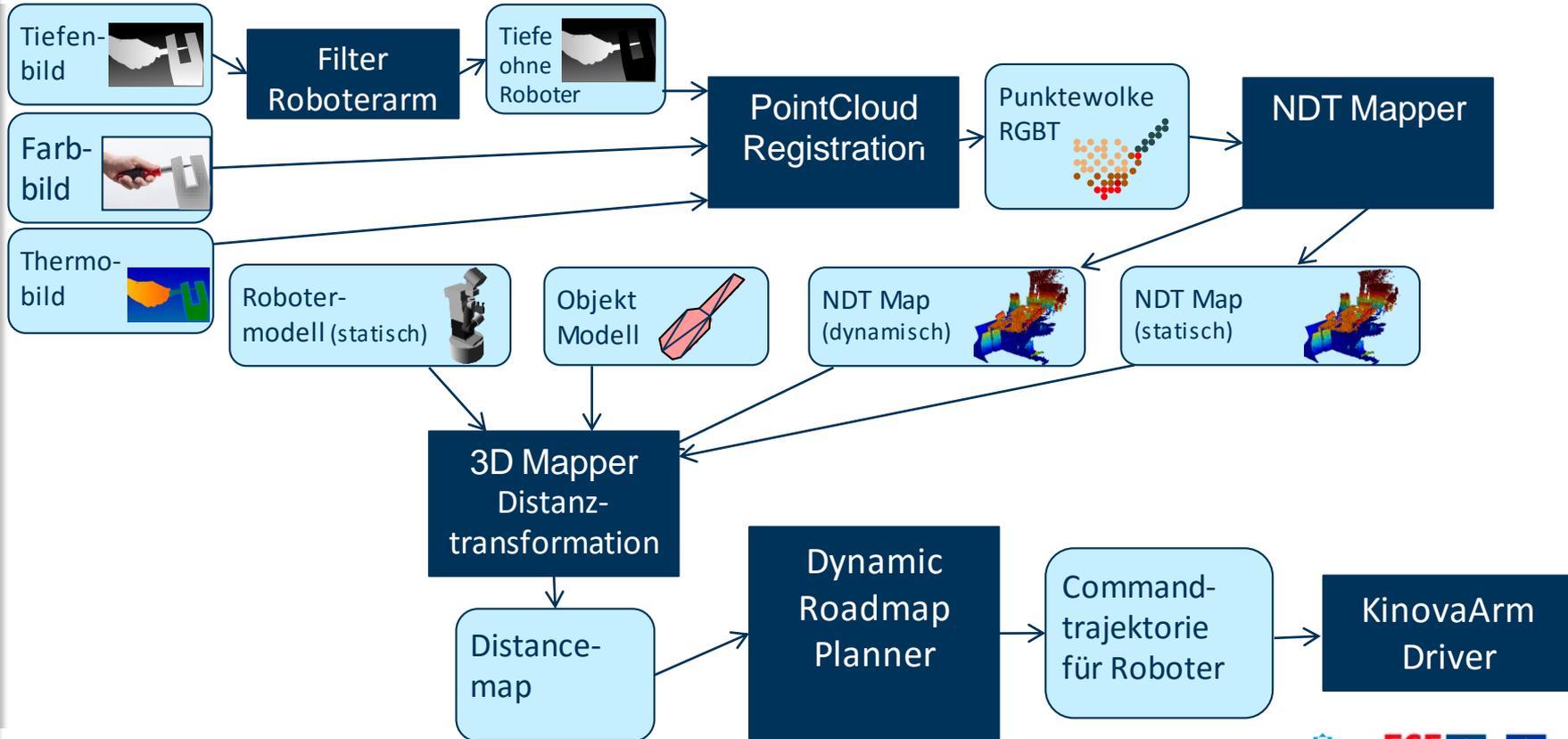
Bewegungssteuerung

- **Erreichter Stand:**

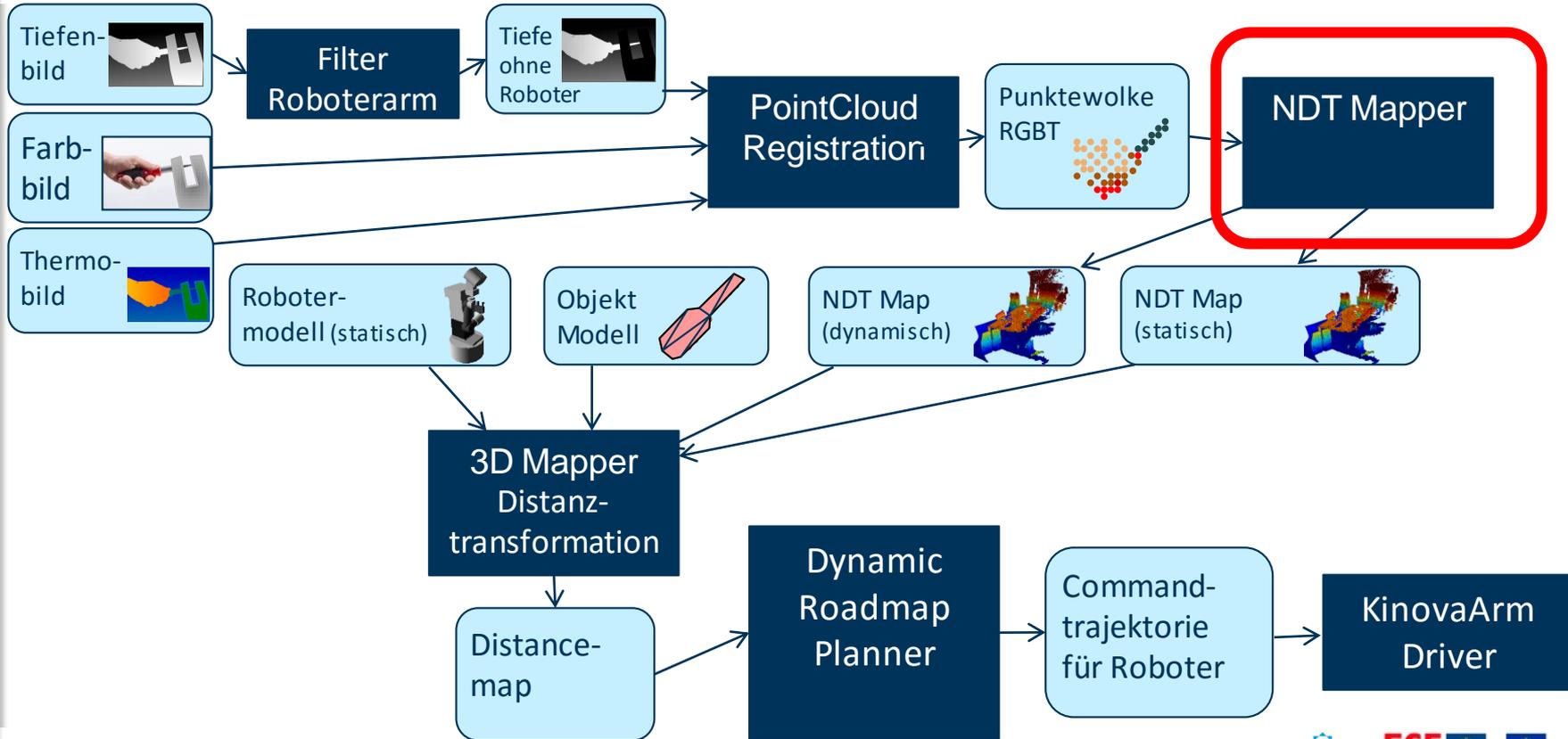
- Bewegungsplaner und Treiber für Kinova Arm angepasst
- Planung von 2 Armen im Jointspace möglich (12DoF)
- Integration NDT-Mapping zur Hindernismodellierung
 - Dynamische Hindernisse erfordern Vergessen
(Freibeweis nur bei Messung hinter Objekt möglich -> passives Vergessen)
 - Statische Karte nötig, da durch (Selbst-)Verdeckung Hindernisse nicht immer sichtbar
- Head-Scan Behavior (sich umschauchen und statische Karte der aktuellen Umgebung bauen)



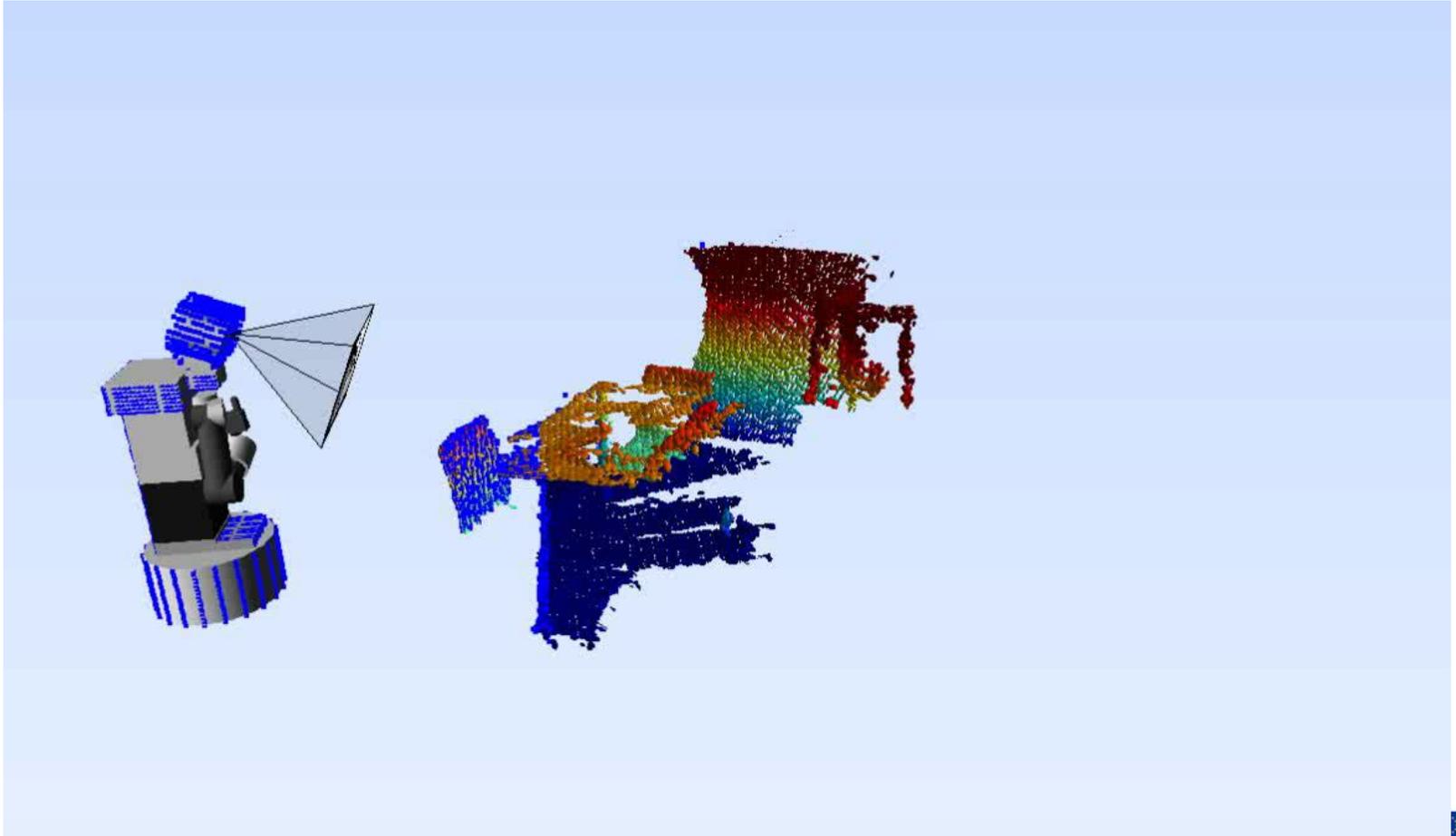
Wahrnehmung der Dynamischen Umgebung für die Bewegungssteuerung



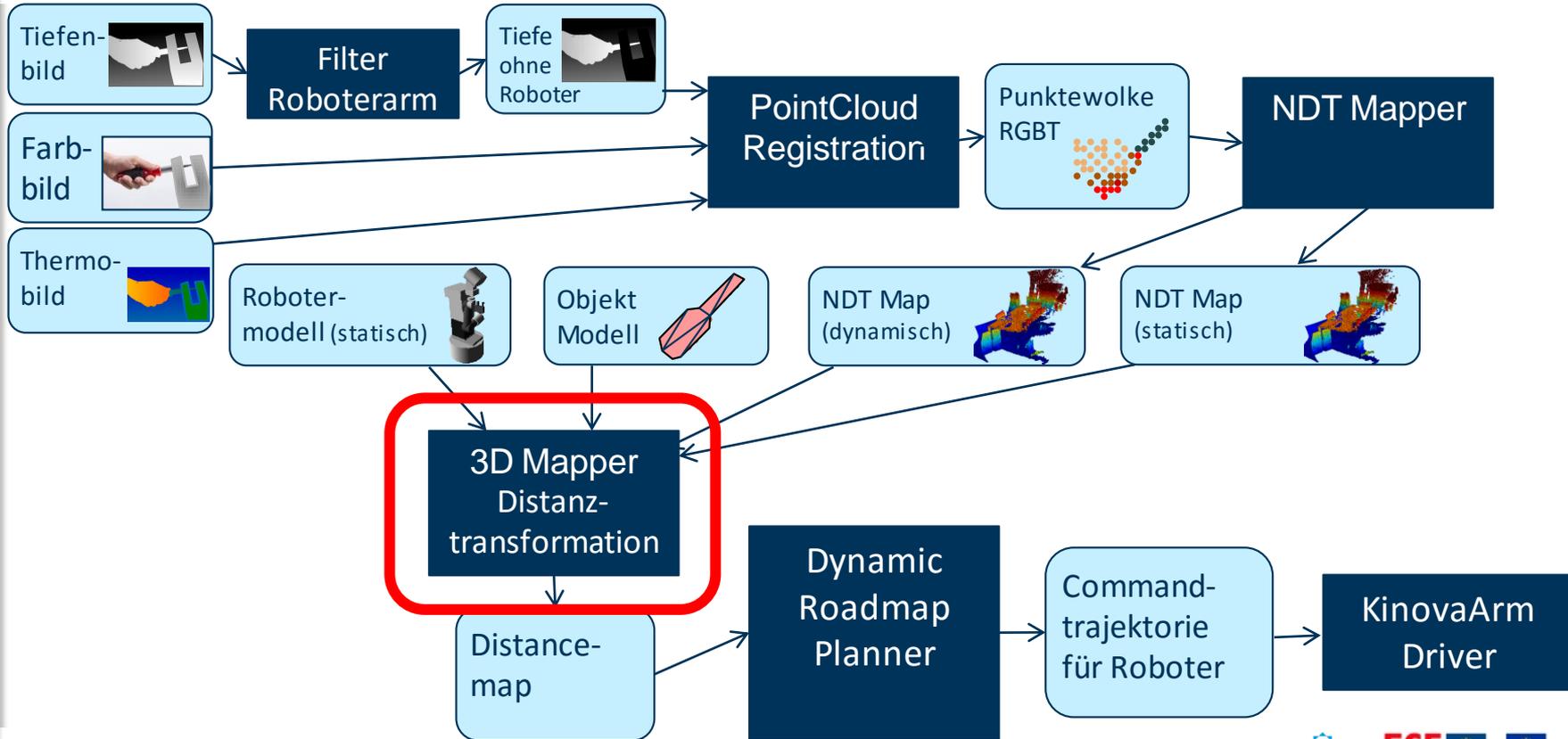
Wahrnehmung der Dynamischen Umgebung für die Bewegungssteuerung



Bewegungssteuerung (Head Scan Behavior)



Wahrnehmung der Dynamischen Umgebung für die Bewegungssteuerung



Bewegungssteuerung

- DistanceMap als Modell für die Hindernisvermeidung
 - Für jeden Raumvoxel ist der nächstgelegene Hindernispunkt hinterlegt, dies ermöglicht effiziente Distanzberechnung für Kollisionsvermeidung
 - Genauigkeit 3cm Grid, aber reellwertige Punktpositionen
 - Anfrage mit Kugeln > 3cm Durchmesser

(∞, ∞)	(∞, ∞)	(∞, ∞)	(∞, ∞)	$(4, 0)$
(∞, ∞)	$(1, 1)$	(∞, ∞)	(∞, ∞)	(∞, ∞)
(∞, ∞)				
(∞, ∞)	(∞, ∞)	(∞, ∞)	(∞, ∞)	$(4, 3)$
(∞, ∞)	(∞, ∞)	(∞, ∞)	(∞, ∞)	$(4, 4)$



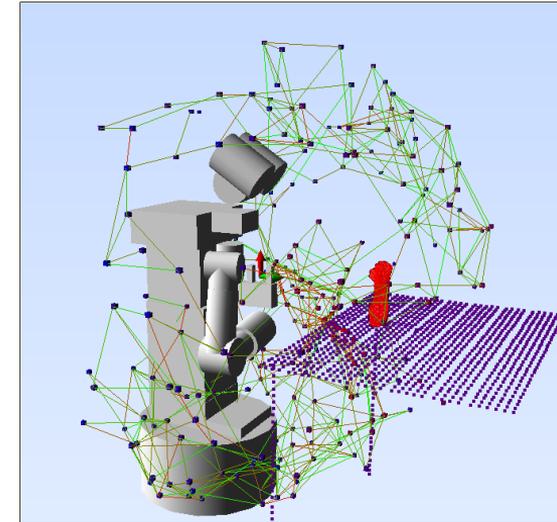
$(1, 1)$	$(1, 1)$	$(1, 1)$	$(4, 0)$	$(4, 0)$
$(1, 1)$	$(1, 1)$	$(1, 1)$	$(4, 0)$	$(4, 0)$
$(1,1)$	$(1, 1)$	$(1, 1)$	$(4, 3)$	$(4, 3)$
$(1, 1)$	$(1, 1)$	$(4, 3)$	$(4, 3)$	$(4, 3)$
$(1, 1)$	$(1, 1)$	$(4, 4)$	$(4, 4)$	$(4, 4)$

$(1,4)$

$$|(4,4)-(1,4)| = 3$$

Arbeitspaket Navigation & Bewegungssteuerung

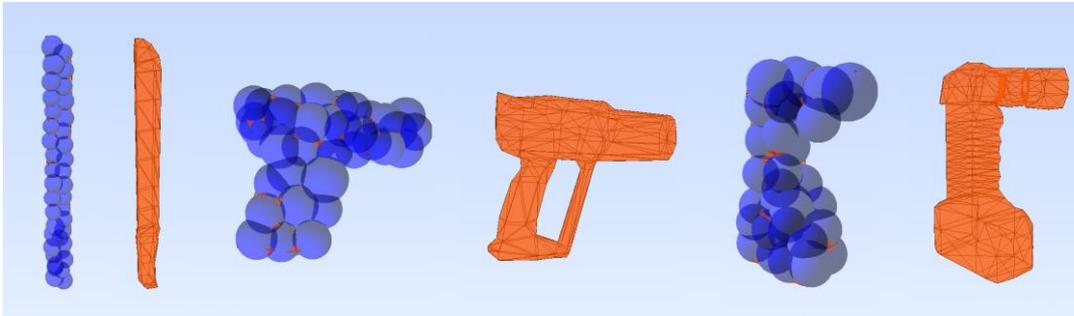
- **Ausgangspunkt:**
 - Bewegungsplaner für dynamische Umgebung und Ziele für Tiago Roboter implementiert
 - **Offene Punkte:**
 - Integration der Sensormesswerte in aktuelles Modell der zu vermeidenden Hindernisse
 - Modellierung des gegriffenen Objekts nach dem Zugriff (Berücksichtigung als Teil des beweglichen Roboterarms für die Hindernisvermeidung)
 - Planung auf Graph bislang nur kürzester Pfad (keine Berücksichtigung von Rewards anderer Zielfunktionen oder mehrerer Zielposen)



Greifbehavior

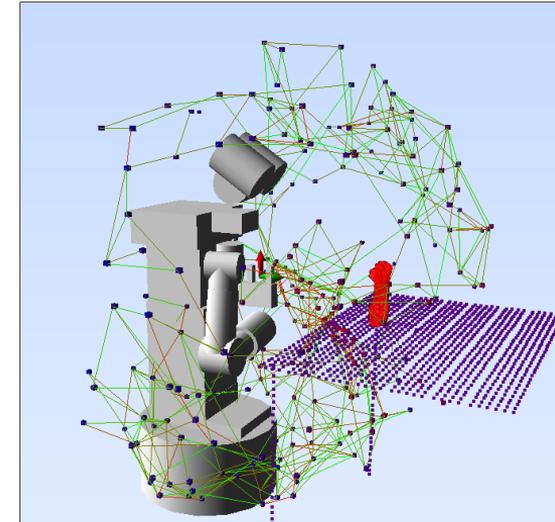
- Ablaufsteuerung des Zugriffs:
 - Greifer öffnen
 - Bewegung zu einem von n möglichen beweglichen Greifpositionen
 - Greifer schließen und Erfolgskontrolle
 - Objekt zur Roboterkollisionsgeometrie hinzufügen

- Modellierung des Objekts als „Wolke von Kugeln“ Clustern der Objektpunkte aus dem Mesh-Modell



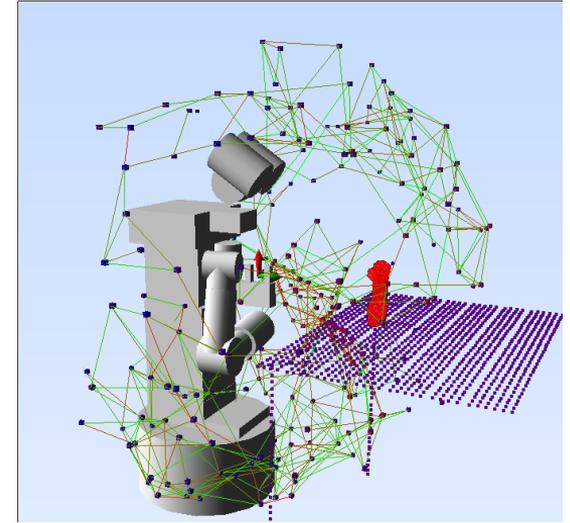
Arbeitspaket Navigation & Bewegungssteuerung

- **Ausgangspunkt:**
 - Bewegungsplaner für dynamische Umgebung und Ziele für Tiago Roboter implementiert
 - **Offene Punkte:**
 - Integration der Sensormesswerte in aktuelles Modell der zu vermeidenden Hindernisse
 - Modellierung des gegriffenen Objekts nach dem Zugriff (Berücksichtigung als Teil des beweglichen Roboterarms für die Hindernisvermeidung)
 - Planung auf Graph bislang nur kürzester Pfad (keine Berücksichtigung von Rewards anderer Zielfunktionen oder mehrerer Zielposen)



Planung im Dynamischen Roadmap-Graph (I)

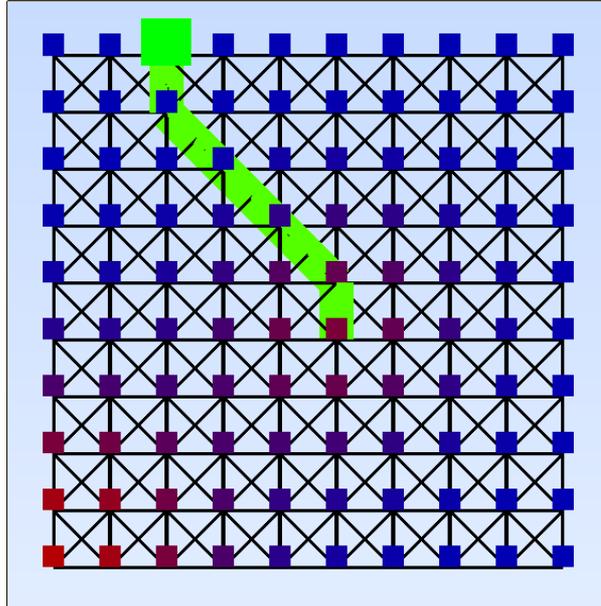
- Bewegungsplaner hält Graph mit kollisionsfrei erreichbaren Positionen im Joint Space
- Graphknoten und -kanten werden von verschiedenen Objectives bewertet (**Wegkosten, Zielabstand, Hindernisabstand**)
- Bislang Planung des kürzesten Weges im Graph mittels Dijkstra-Algorithmus
→ keine Optimierung bei mehreren möglichen Zielen
- **Lösungsansatz:** MDP (**Markov Decision Process**) mit Maximierung des diskontierten Rewards entlang der zukünftigen Trajektorie
- Bislang in der Literatur Lösung mit **Value Iteration** (dynamische Programmierung)
→ recht aufwändig $O(NM)$ (N Knoten, M Kanten)
- Neuentwicklung einer Planung ähnlich des Dijkstra-Algorithmus
→ $O(N \log(N) M)$



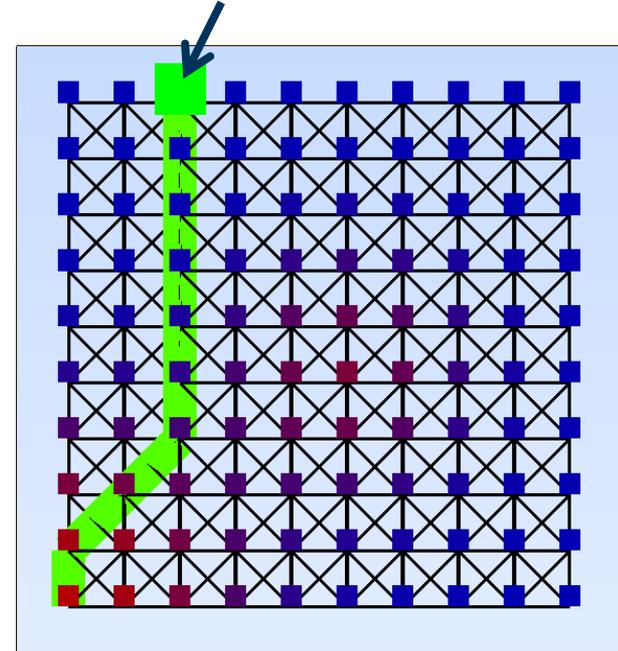
Planung im Dynamischen Roadmap-Graph (III)

- Beispiel mehrdeutige Ziele

■ wenig Belohnung
■ viel Belohnung

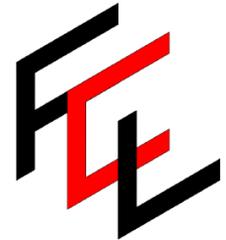


$\gamma = 0.6$



$\gamma = 0.9$

Arbeitspaket Navigation & Bewegungssteuerung

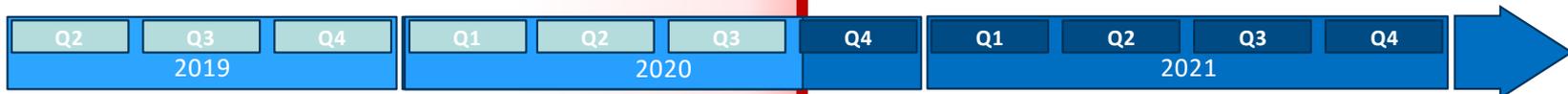
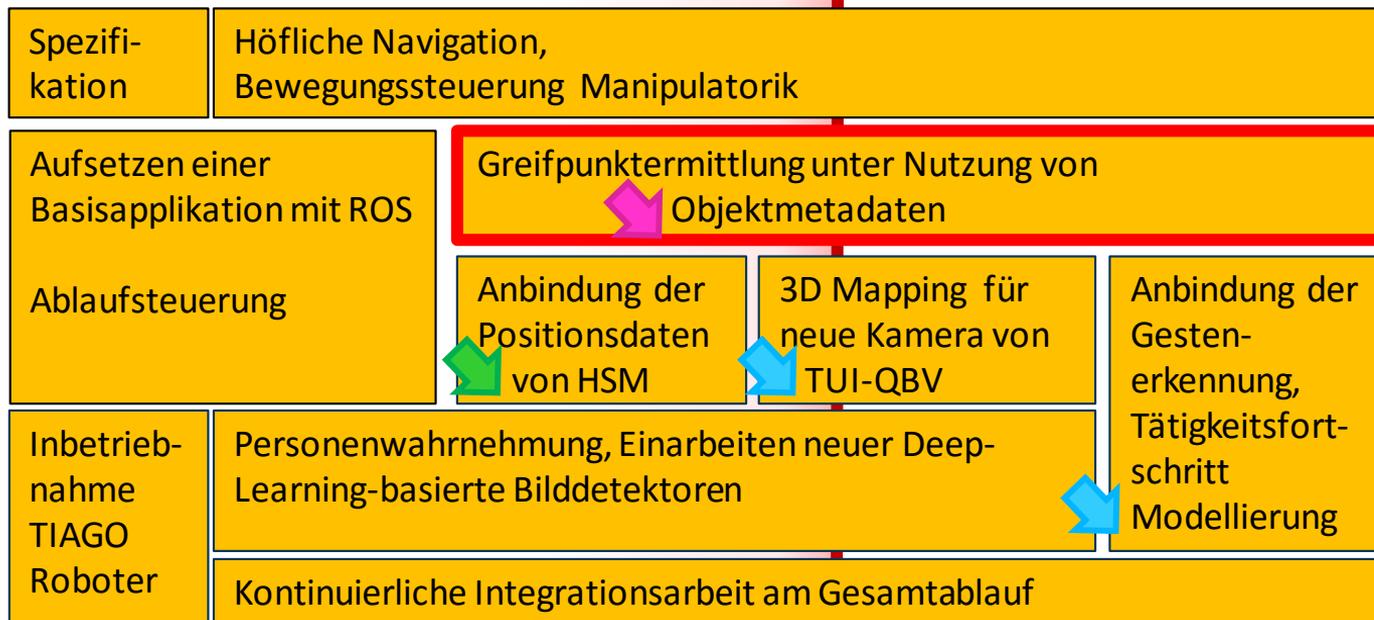


- **Ausblick / Weiteres Vorgehen:**
 - **Bachelorarbeit** Andreas John zur Verbesserung der Kollisionsberechnung im Bewegungsplaner
 - Bottleneck ist die **Kollisionsberechnung für jeden Graphknoten** in Echtzeit
 - **Flexible Collision Library** (Verwendet in ROS MoveIt)
soll verglichen werden mit der Eigenimplementierung → ggf. effizienter
 - Entwicklung eines Übergabe-Behaviors
 - Erfordert Tracking der Handposition als Ziel
-> siehe AP Personenwahrnehmung

Arbeitsplan für TUI-NIKR über die Projektlaufzeit

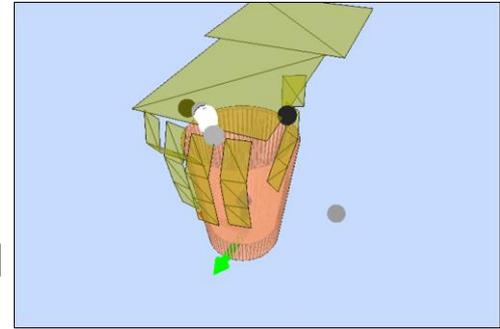
Zuarbeiten von:

-  HSM
-  TUI-QBV
-  GFE

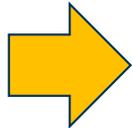


Greifpunktermittlung

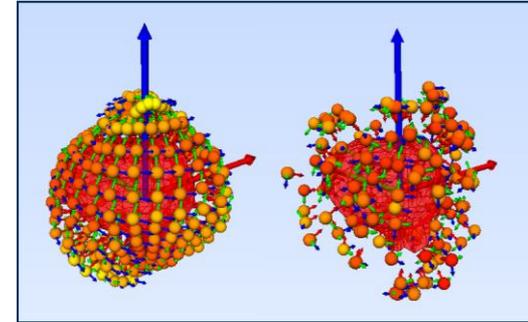
- **Ausgangspunkt:**
 - Bestehendes analytisches System zur Greifposenbewertung am Objekt als Mesh-Modell
 - Deep-Learning Verfahren zur Beschleunigung / Vervollständigung der Greifposenmenge



- **Offene Probleme:**



- Woher kommt das Mesh-Modell der Objekte?
- Übertragbarkeit auf Realwelt und anderen Greifergeometrien?
- Wie können Affordanzen berücksichtigt werden?



Objektrekonstruktion aus Single View Bildern (I)

- **MA Markus Paschke:**

„Deep-Learning-Methoden zur Erstellung von 3D-Modellen“,
TU Ilmenau, August 2020 (abgeschlossen)

- **Ziel:** Generierung von 3D Modellen zu unbekanntem Objekten mit der Robotersensorik (RGB-D Kamera)

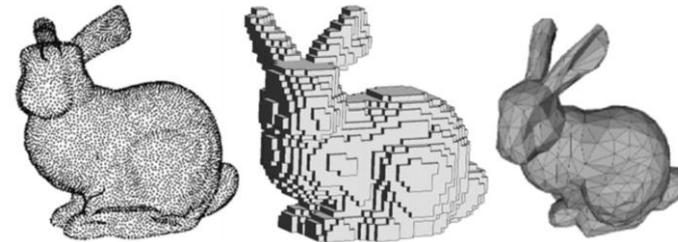
- **Ergebnisse:**

- Untersucht wurden verschiedene Verfahren
PointCloud, Octomap, Mesh (Pixel2Mesh)

- Einbeziehung von Tiefendaten verbessert Ergebnis signifikant
(ggf. nur auf Tiefendaten arbeiten)

- Erkannte Probleme:

- Tiefenausdehnung ungenau
- Bisher nur auf freigestellten Bildausschnitten trainiert

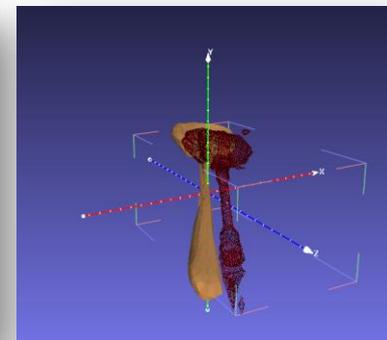
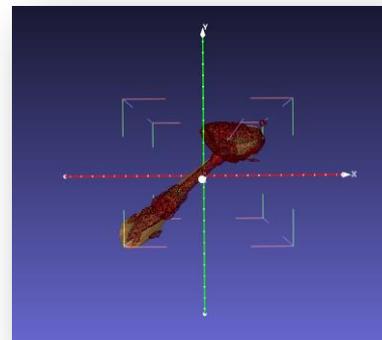
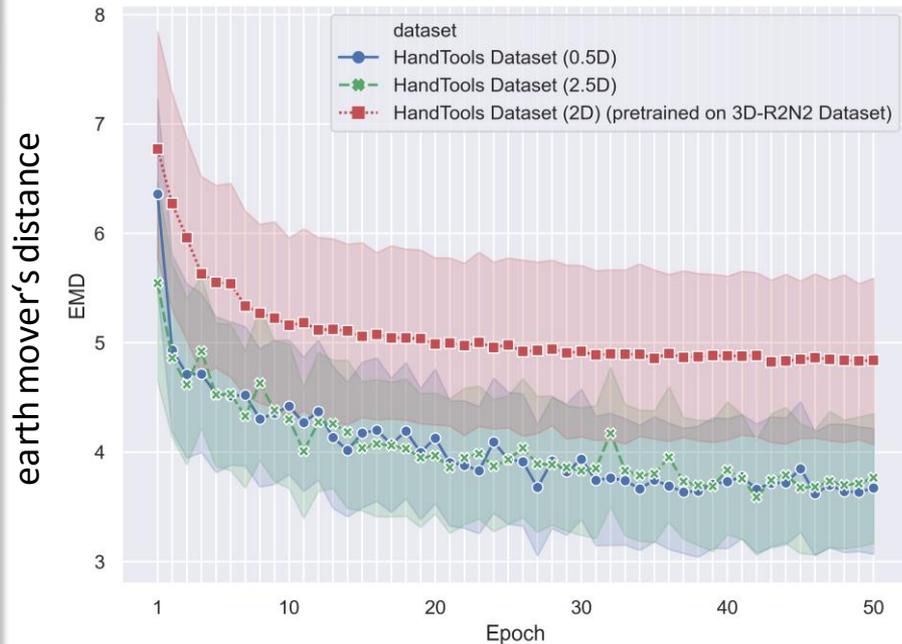


Objektrekonstruktion aus Single View Bildern (II)

- Vergleich der Rekonstruktionsgüte bei Verwendung von RGB, RGB-D oder nur D als Input

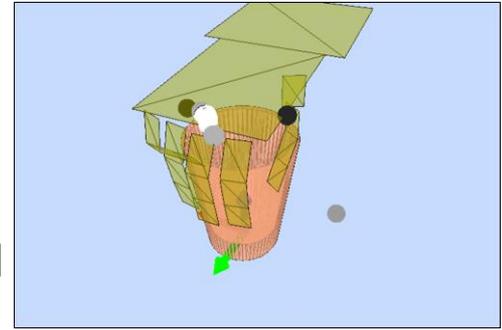


Input (RGB)



Greifpunktermittlung

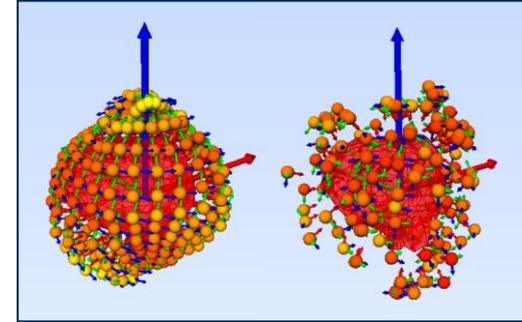
- **Ausgangspunkt:**
 - Bestehendes analytisches System zur Greifposenbewertung am Objekt als Mesh-Modell
 - Deep-Learning Verfahren zur Beschleunigung / Vervollständigung der Greifposenmenge



- **Offene Probleme:**



- Woher kommt das Mesh-Modell der Objekte?
- Übertragbarkeit auf Realwelt und anderen Greifergeometrien?
- Wie können Affordanzen berücksichtigt werden?



Erweiterung der analytischen Greifposenbewertung

- Implementierung zusätzlicher Constraints:
 - **Greifrichtung**
(Beschränkung auf Griffe von oben bzw. vom Roboter aus)
 - **Drehmoment** (forciert Griff über Schwerpunkt)

- Neue Greifermodelle können aus üblichen URDF Modellen zur Roboterbeschreibung generiert werden
 - Parametrierung der Kontaktflächen
 - Test an Kinova 3-Finger Greifer und Schunk SVH 5-Finger Hand

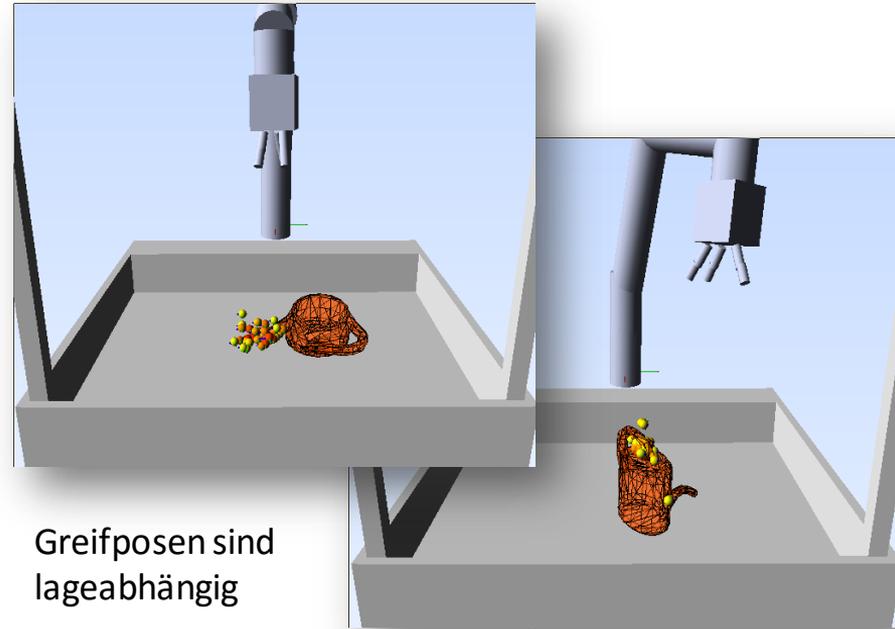
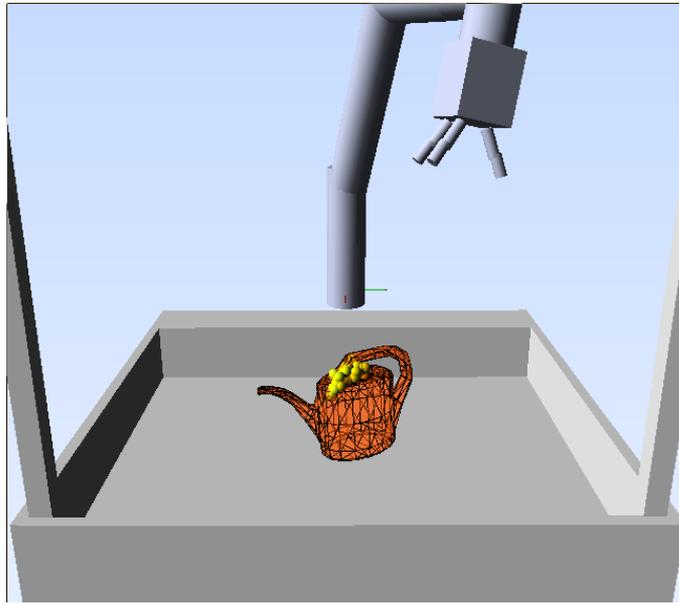


Teststand für automatisierte Greifversuche (I)

- Projektseminar
- **Ziel:** Greifposenqualität der Schätzer mit realen Erfolgsraten beim Greifversuch abgleichen
- Setup:
 - KinovaArm 6DoF 3-Finger Greifer
 - Top-View:
 - Asus Xtion Pro Live (bessere PointCloud)
 - Kinect 4 Azure (5 MPixel Farbbild, problematische Tiefendaten)
 - Hexacore Rechner mit Nvidia RTX 2060 Super



Teststand für automatisierte Greifversuche (II)



Verarbeitungsschritte:

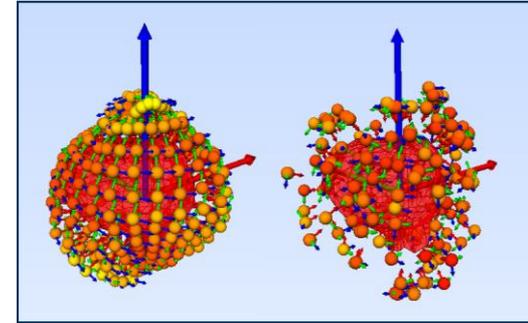
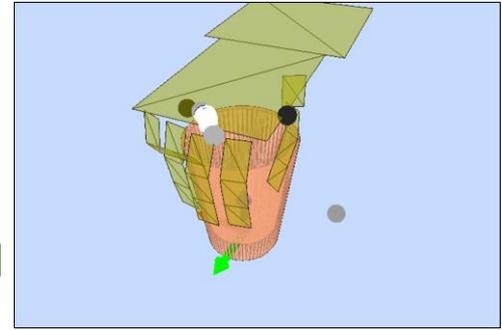
PointCloud Segmentierung → Modellfitting → Greifposenbestimmung → Bewegungsplanung

Teststand für automatisierte Greifversuche (III)



Greifpunktermittlung

- **Ausgangspunkt:**
 - Bestehendes analytisches System zur Greifposenbewertung am Objekt als Mesh-Modell
 - Deep-Learning Verfahren zur Beschleunigung / Vervollständigung der Greifposenmenge



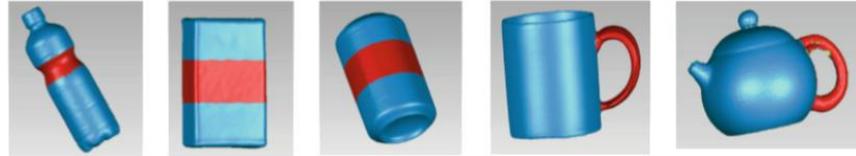
- **Offene Probleme:**
 - Woher kommt das Mesh-Modell der Objekte?
 - Übertragbarkeit auf Realwelt und anderen Greifergeometrien?
 - Wie können Affordanzen berücksichtigt werden?



Greifpunktermittlung

- **Ausblick / weiteres Vorgehen:**

- Nutzung von Affordanz-Labeln am Objektmodell im analytischen Grapsplaner

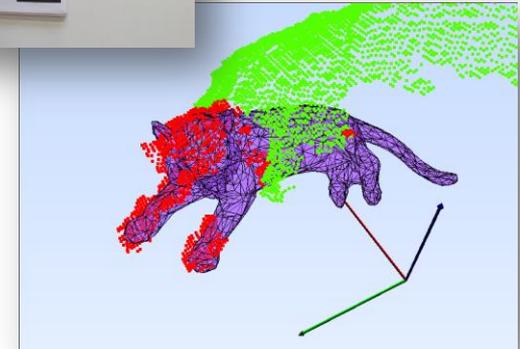
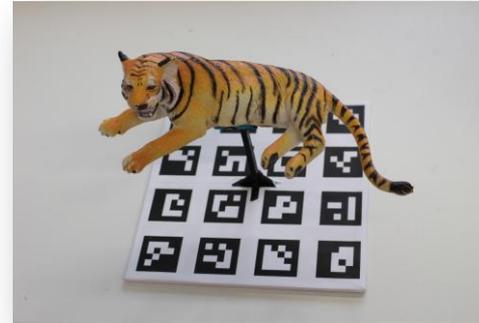


(Objekttextur enthält Label für Griff zur Übernahme oder Übergabe)

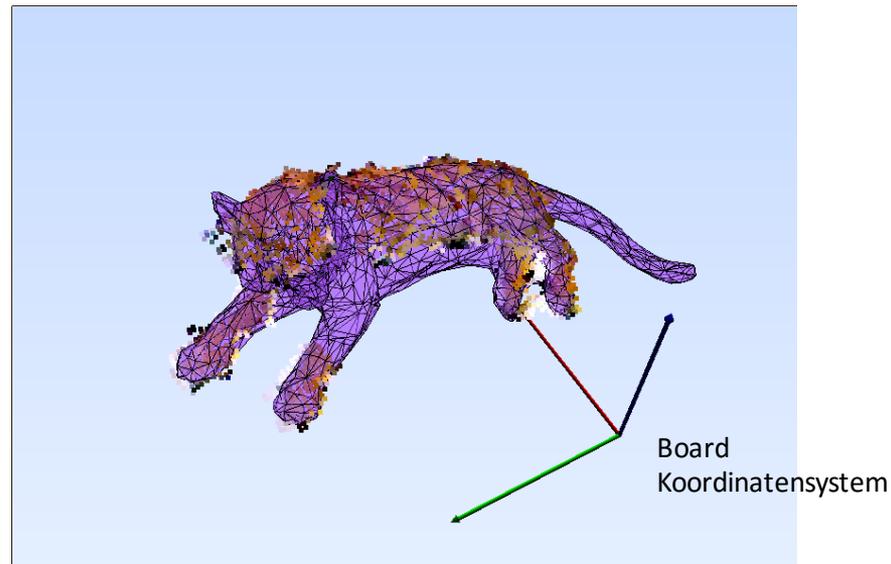
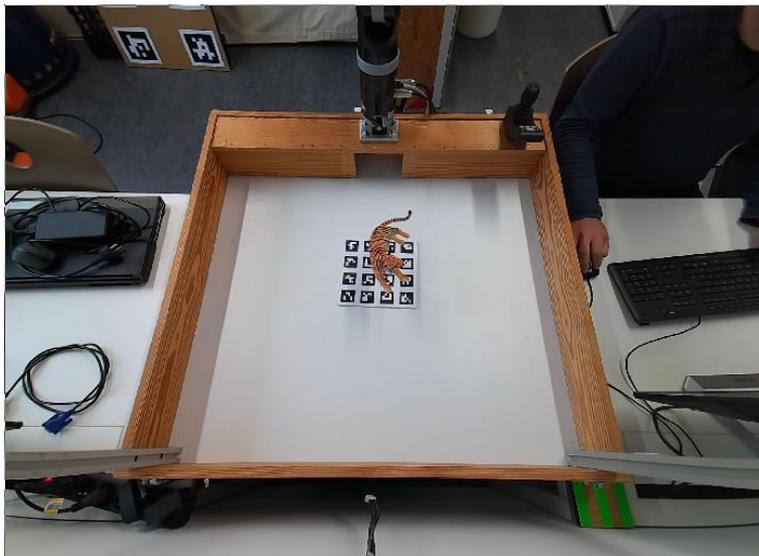
- Erweiterung der Deep-Learning Objektmesh-Schätzer um solche Labelausgaben pro Vertex

Trainingsdaten Aufnahme (I)

- Deep-Learning Verfahren für PointCloud-Segmentierung sowie für Objekt-Mesh Schätzung (siehe TUI-QBV) benötigen Trainingsdaten
- Manueller Labelaufwand soll minimiert werden -> automatische Segmentierung während der Datenaufnahme
- Vorgehen:
 - Bekanntes Mesh-Modell der Objekte
 - Tracking im 3D Raum durch externe ArUco-Marker (rigid Objects) im Farbbild
 - Aufnahme von Punktwolken mittels Tiefenkamera
 - Abstandsbasierte Segmentierung (Punkt zu Mesh – Abstand)
 - Geometrische Constraints: Clustergröße



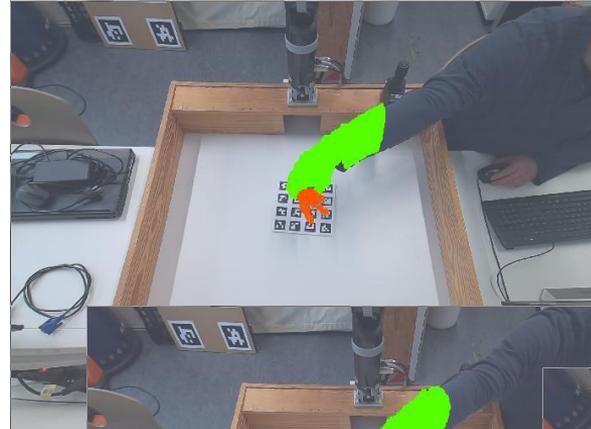
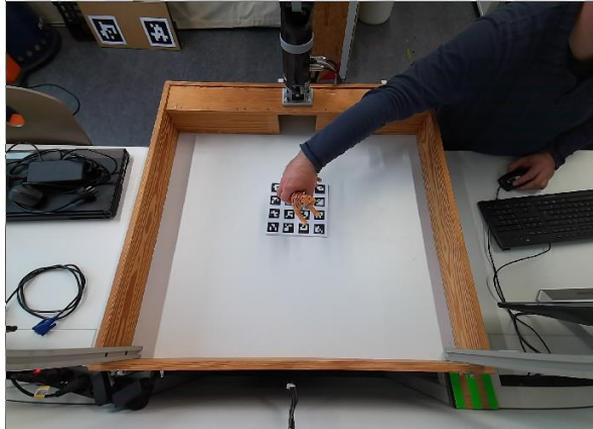
Trainingsdaten Aufnahme (II)



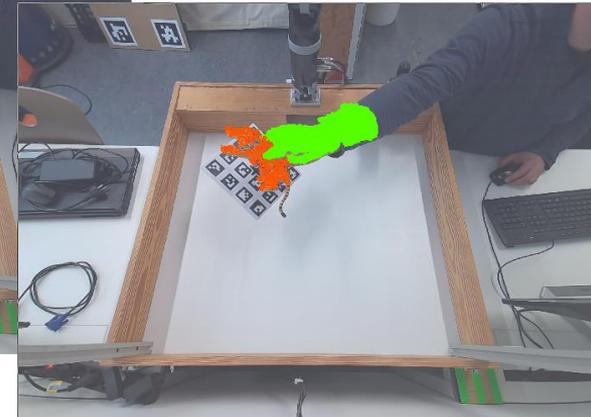
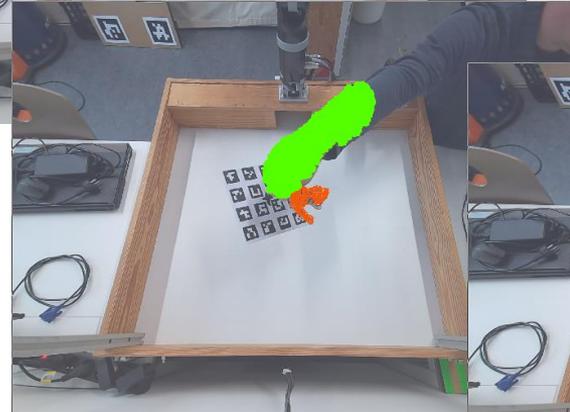
- Initialisierung für neues Objekt:
 - Aruco-Board wird über RGB Bild im 3D Raum getrackt
 - Objektmodell wird mittels ICP und der PointCloud am Board ausgerichtet

Trainingsdaten Aufnahme (III)

- Objekt kann gegriffen und mit Board bewegt werden
- Objektmesh bewegt sich mit Board und kann zur Segmentierung der Punktdaten genutzt werden



HandPointCloud und
ObjektPointCloud zurück
projiziert ins RGB Bild

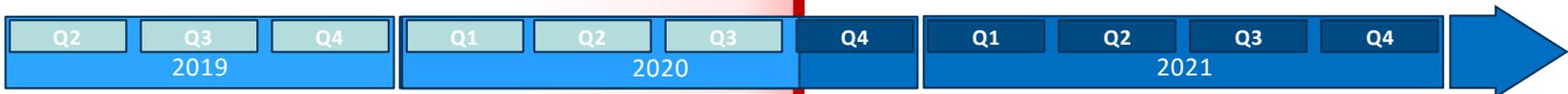
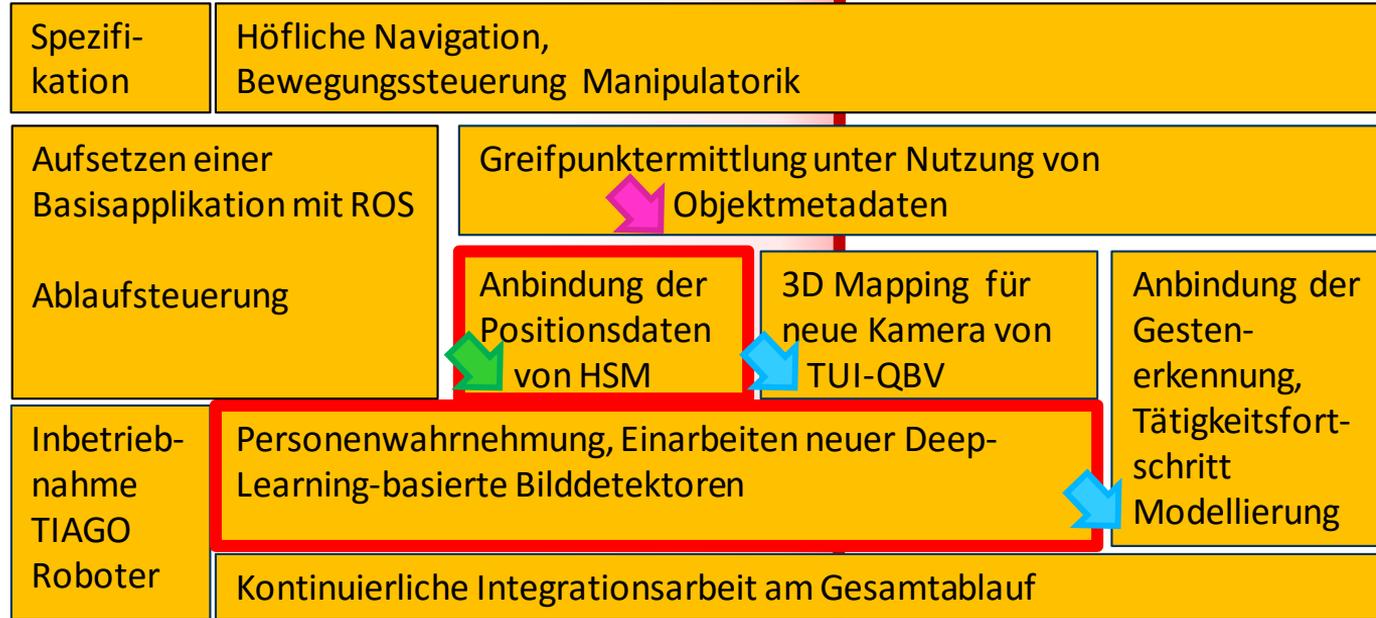


Für späteres Training wird
ein beliebiger Background
hinter die Pointclouds
eingefügt → Augmentation

Arbeitsplan für TUI-NIKR über die Projektlaufzeit

Zuarbeiten von:

- HSM
- TUI-QBV
- GFE

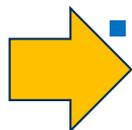


Personenwahrnehmung

- **Ausgangspunkt:**

- Modularer Personentracker
- Ermöglicht Tracking verschiedener Attribute
- Grundlage für Navigation behaviors für das Anfahren der Interaktionspartner

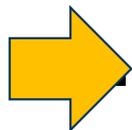
- **Erreichter Stand:**



- **Integration der RTLS-Flares (HSM)**

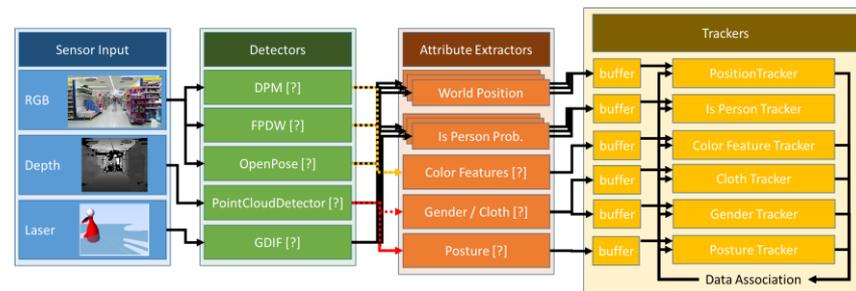
→ siehe Fortschrittsbericht HSM

- Ermöglichen auch eindeutige Identifikation der Interaktionspartner



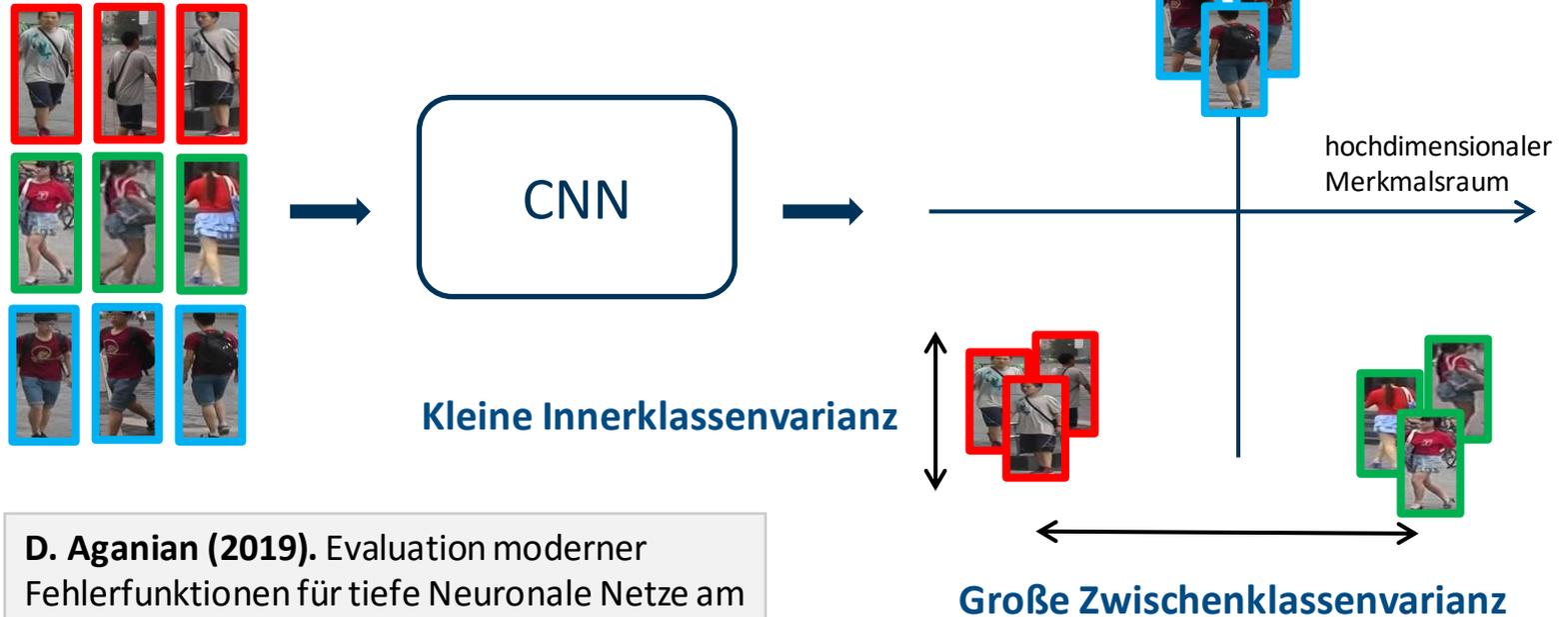
Wiedererkennung anhand der Ansicht im RGB Bild

- Integration Skeletttracking



Ansichtsbasierte Personenwiedererkennung

- Erlernen **diskriminativer Merkmalsvektoren** durch ein CNN
- Keine Ansichten rekonstruierbar



D. Aganian (2019). Evaluation moderner Fehlerfunktionen für tiefe Neuronale Netze am Beispiel der erscheinungsbasierten Personenwiedererkennung, *Masterarbeit*

Personenwahrnehmung

- **Ausgangspunkt:**

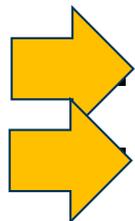
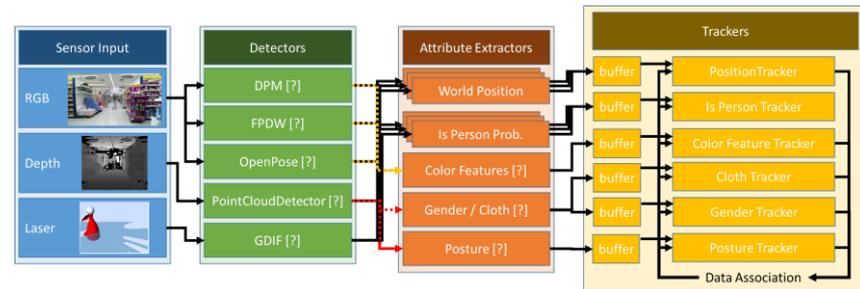
- Modularer Personentracker
- Ermöglicht Tracking verschiedener Attribute
- Grundlage für Navigation behaviors für das Anfahren der Interaktionspartner

- **Erreichter Stand:**

- **Integration der RTLS-Flares** (HSM)
 - siehe Fortschrittsbericht HSM
 - Ermöglichen auch eindeutige Identifikation der Interaktionspartner

Wiedererkennung anhand der Ansicht im RGB Bild

Integration Skelettracking



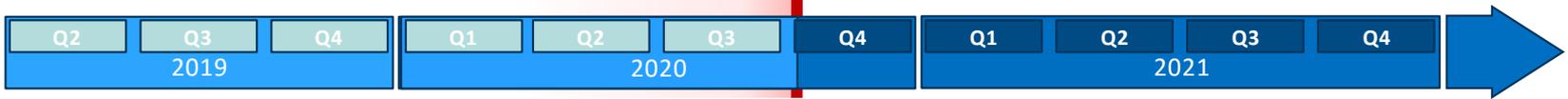
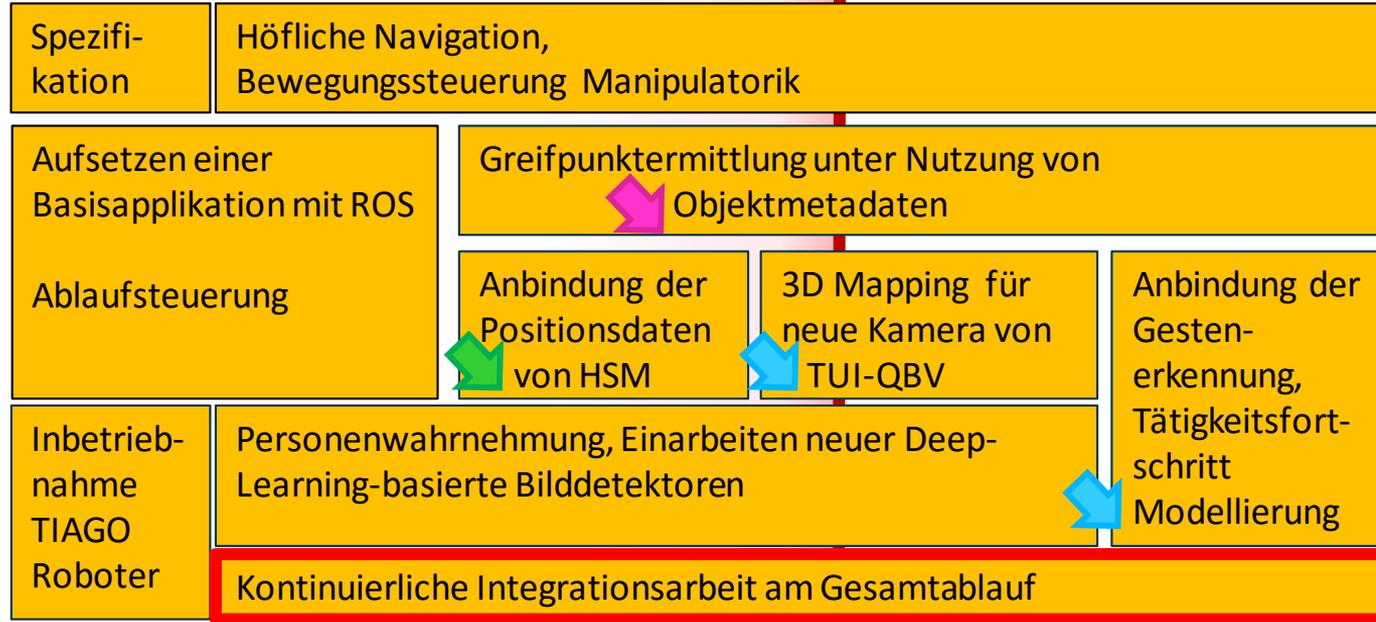
Skelettracking / Intenterkennung

- Pro Personenhypothese kann ein Skelettmodell getrackt werden -> Handposition für Übergabe und Übernahme
- **Masterarbeit** Max Kogel
„Integration von Skelettracking und Intenterkennung in den Personentracker“ (laufend)
 - Fusion verschiedener Skelettdetektoren aus unterschiedlichen Blickwinkeln zur Reduzierung von Verdeckungsartefakten
 - Skelette aus OpenPose oder Kinect 4 integrieren
 - Einfache Gestenerkennung basierend auf Skelettdaten

Arbeitsplan für TUI-NIKR über die Projektlaufzeit

Zuarbeiten von:

-  HSM
-  TUI-QBV
-  GFE



Kalibrierung der Kameras am Roboter

- **Ausgangspunkt:**
 - Methoden zur Kalibrierung der Extrinsischen Kameraparameter am Roboter durch Messfahrt in spezieller Umgebung mit Markern
 - Driftproblem mit Astra Orbbec Kamera im Roboter



- **Erreichter Stand:**
 - Kalibrierung der Punktwolken gegen das interne Robotermodell durch Kugel-segmentierung in der Punktwolke
 - CalibrationBehavior zum automatischen abfahren der nötigen Punkte
 - Kalibrierung der Roboter und der Greifbox schnell möglich

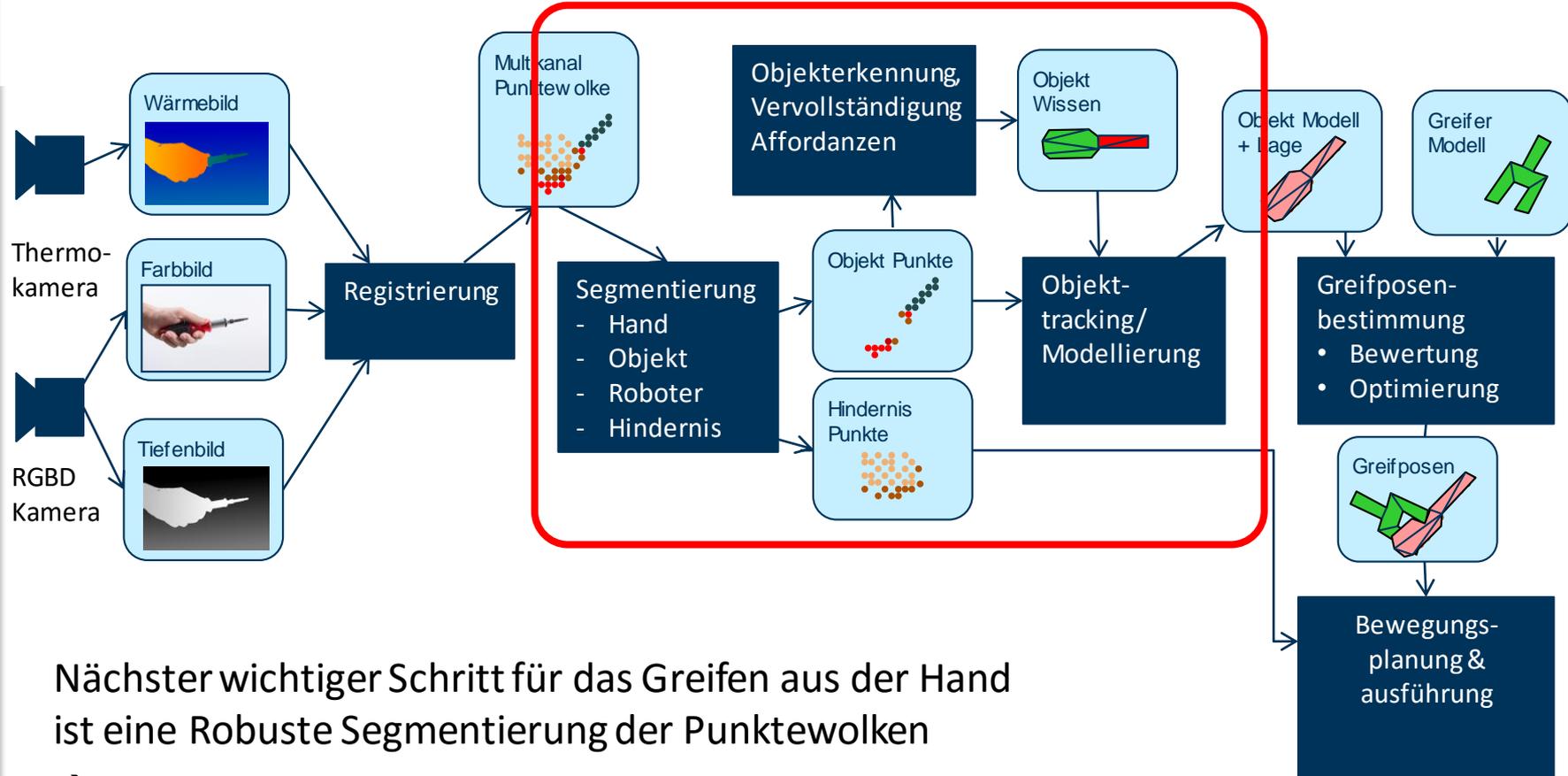


Integrationsarbeiten Hand-/Objektwahrnehmung



- Masterarbeit** Patrick Stäblein:
„Integration eines Objekt-/Handtracking-Ansatzes in die reale Roboterapplikation“ (laufend)
 - Entwicklung einer Pipeline von Deep-Learning Modulen zur Handdetektion im RGB Bild und anschließenden Modellierung von Hand und Objekt als Mesh-Modell im 3D Roboterkoordinaten

Ausblick weitere Integrationsarbeiten



Nächster wichtiger Schritt für das Greifen aus der Hand ist eine Robuste Segmentierung der Punktwolken

→Zuarbeiten von TUI-QBV

Forscherguppe SONARO

Smarte Objektübernahme und –übergabe für die
nutzerzentrierte mobile Assistenzrobotik

Beiratstreffen am 28.10.2020

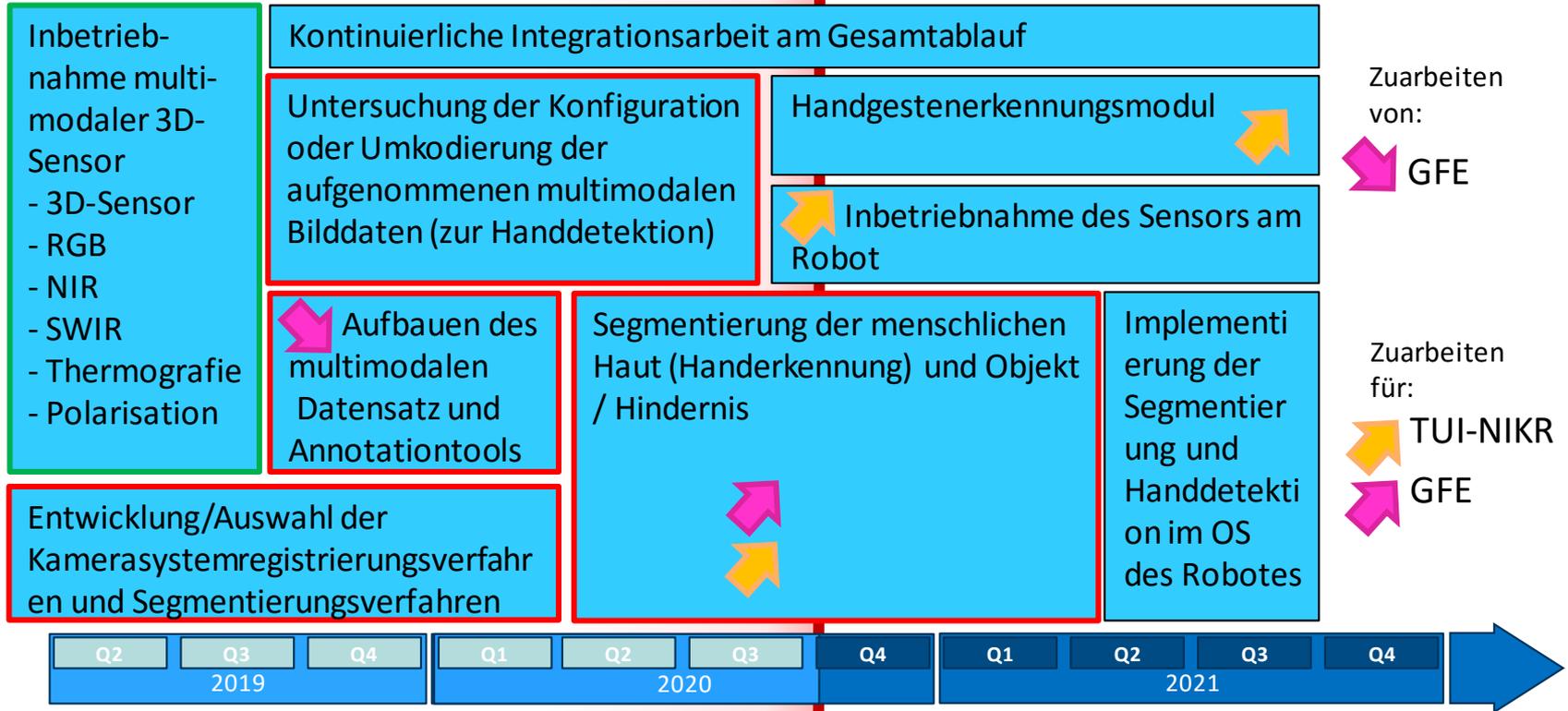
Vorstellung der Arbeiten von TU-Ilmenau QBV

M.Sc. Yan Zhang

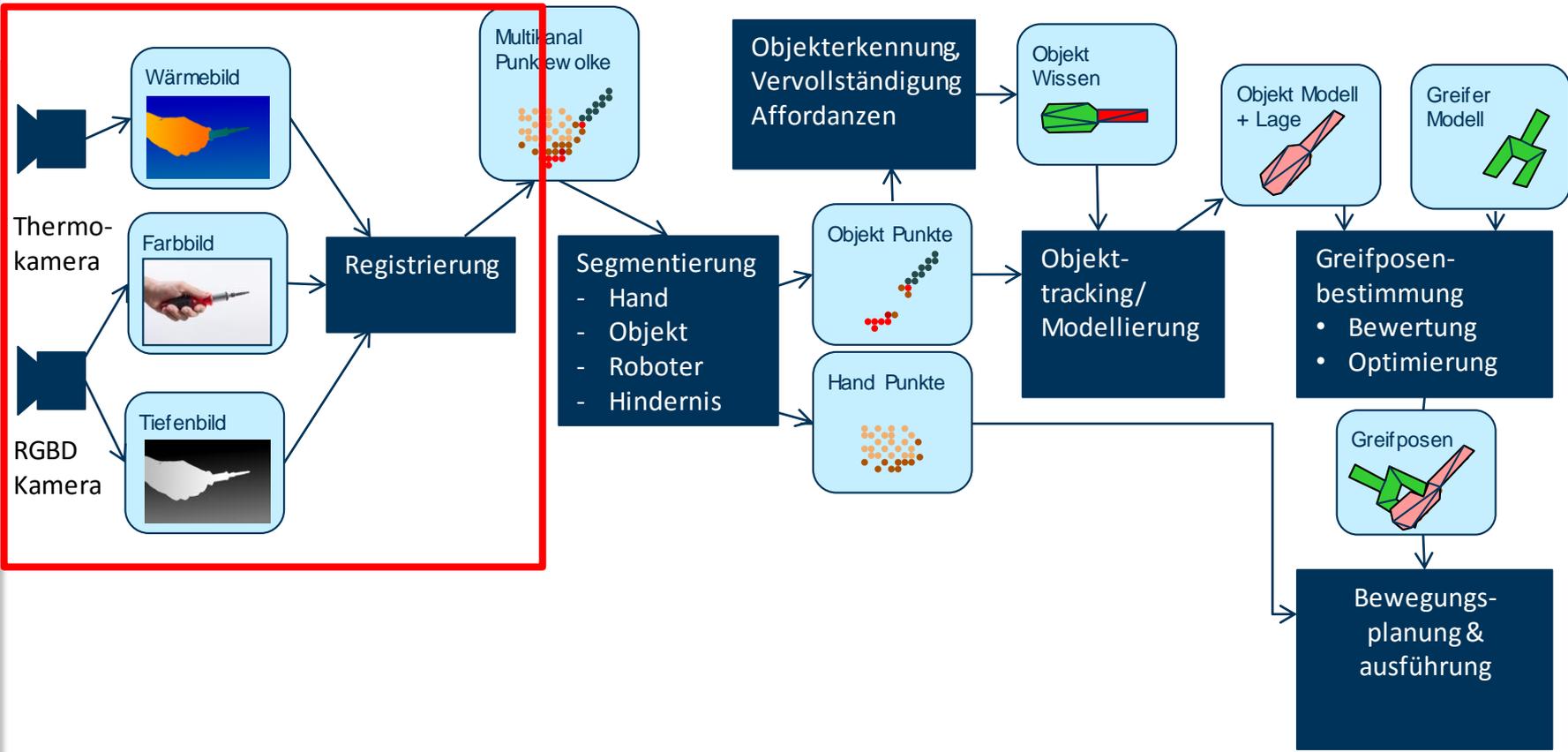
www.sonaro-projekt.de



Arbeitsplan für TUI-QBV über die Projektlaufzeit



Datenverarbeitungspipeline (a) für den Zugriff

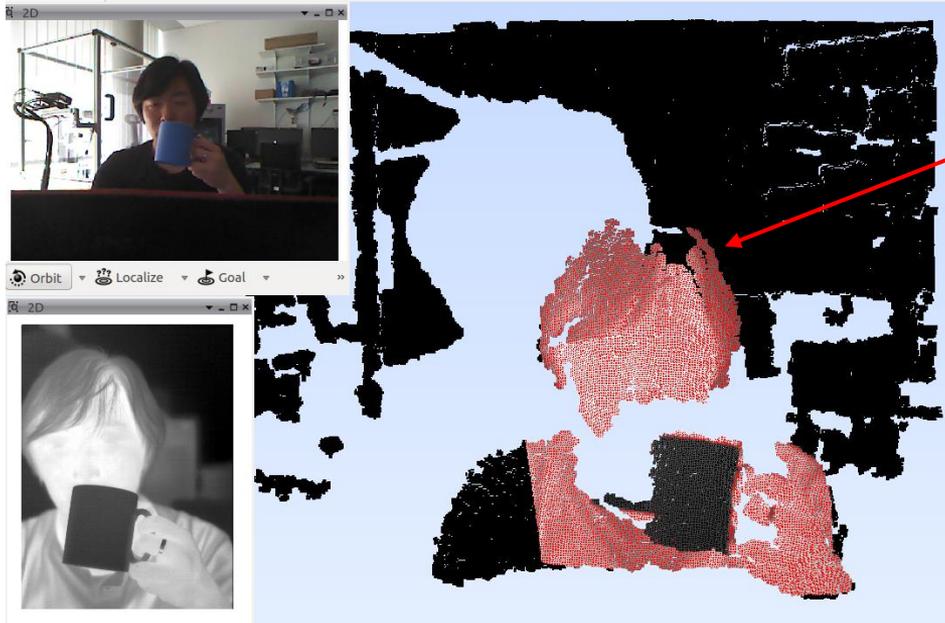


Aufbau des multimodalen 3D-Sensorsystems

- **Stand 05/2020 - Ausgangspunkt:**
 - Multimodales 3D-Sensorsystems aufgebaut
 - liefert verschiedene Informationen, z.B. Temperatur, Farbe und 3D-Mesh, um Menschen und Objekte sicherer segmentieren zu können.
 - Optimale Konfiguration des Sensorsystems:
 - RGB + Far-Infrared + 3D
 - Registrierungstools entworfen:
 - Thermo-Kamera -> 3D-Kamera

Optimierung Auswahl der Thermokamera

- **Multimodales System mit 3D-Sensor KinectV1 & Thermokamera I3system TE-V1**
 - Problem von I3system TE-V1: zu kleiner Öffnungswinkel



zu kleiner Sichtbereich



KinectV1



I3system TE V1

Alternative Thermokamera: I3system TE Q1

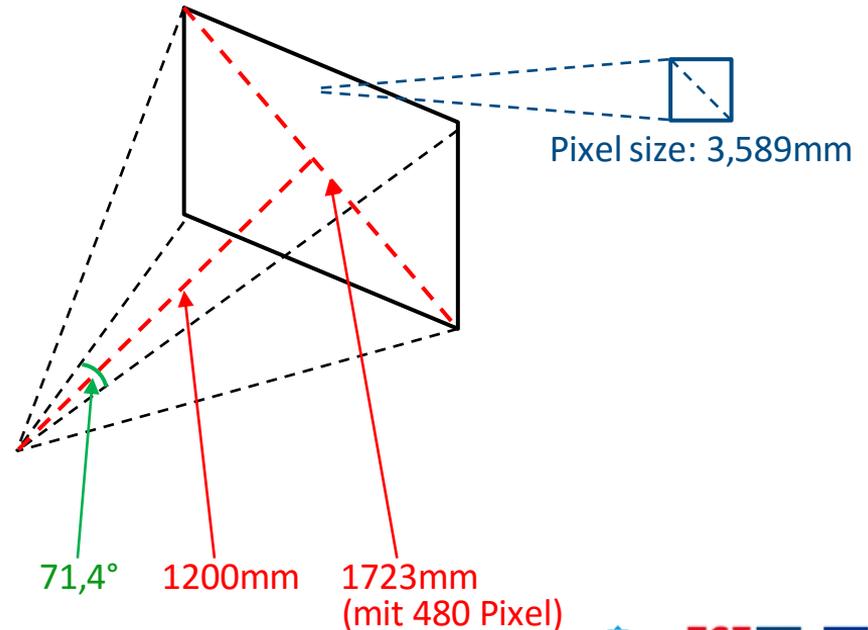
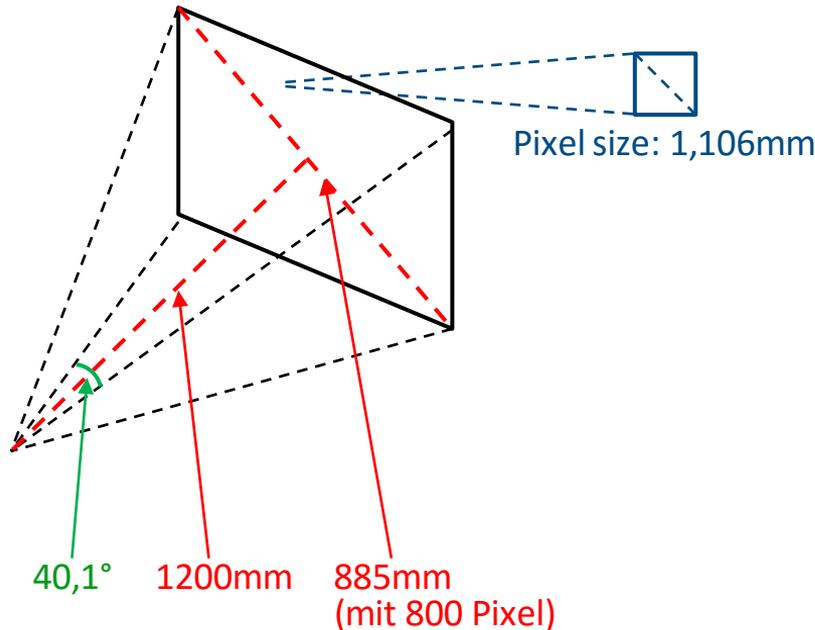
- I3system TE-V1**

- Auflösung: 640 x 480
- Öffnungswinkel: 40,1°



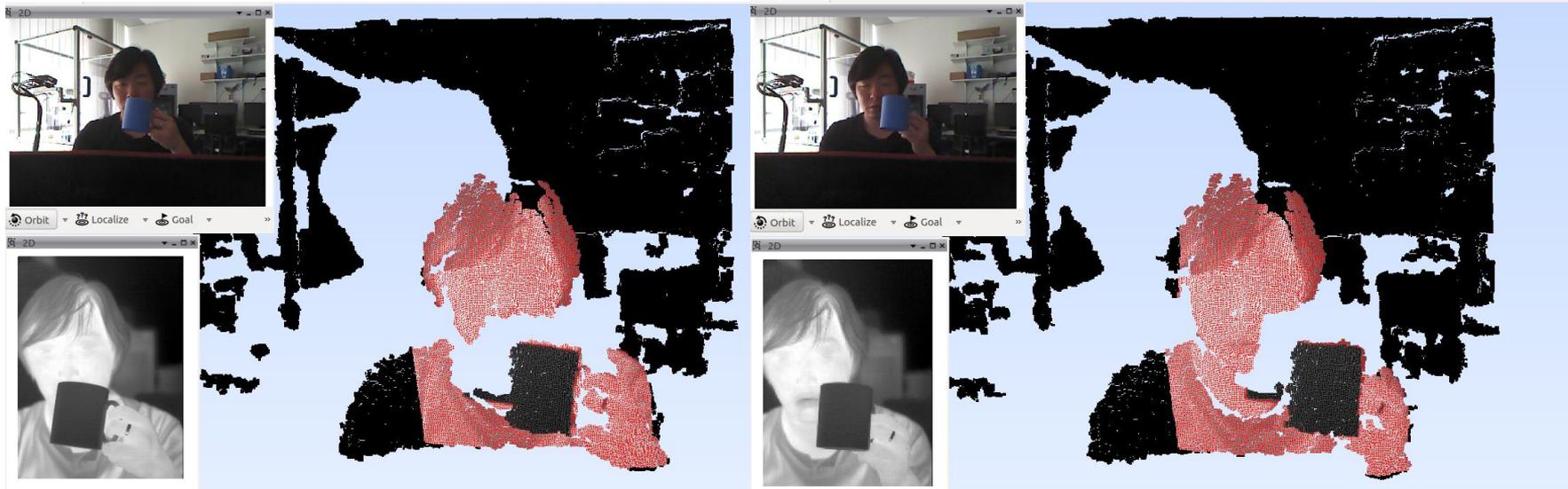
- I3system TE-Q1**

- Auflösung: 384 x 288
- Öffnungswinkel: 71,4°



Alternative Thermokamera: I3system TE Q1

- **Beachte I3system TE-Q1:** niedrigere Auflösung → Test der Einsetzbarkeit
- Test der Segmentierung mit unterschiedlichen Auflösungsstufen in dem das Thermobild von TE-V1 3-fach unterabgetastet wird

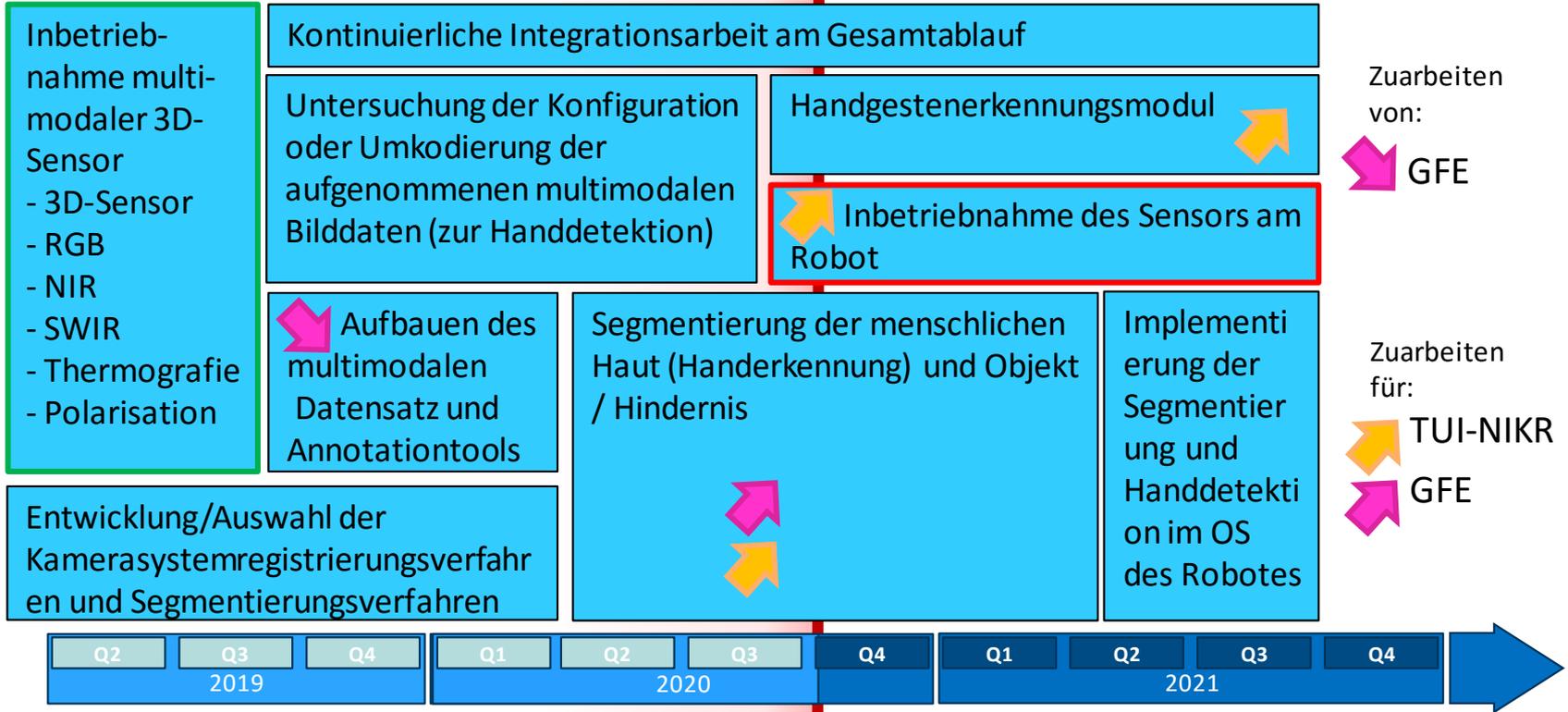


Originales Bild

Unterabgetastetes Bild

➔ Segmentierung mit TE-Q1 möglich

Arbeitsplan für TUI-QBV über die Projektlaufzeit



Integration des multimodalen 3D-Sensorsystems auf dem Tiago-Roboter

- Roboter wurde ausgestattet mit:
 - Kinect 4 Azure
 - I3Systems TE-Q1 Thermal Camera (384x288, 8-14 μ m)
 - Astra Orbbec (ASTRA S) im Kopf
- Registrierung aller Bilder für die Berechnung einer Punktwolke

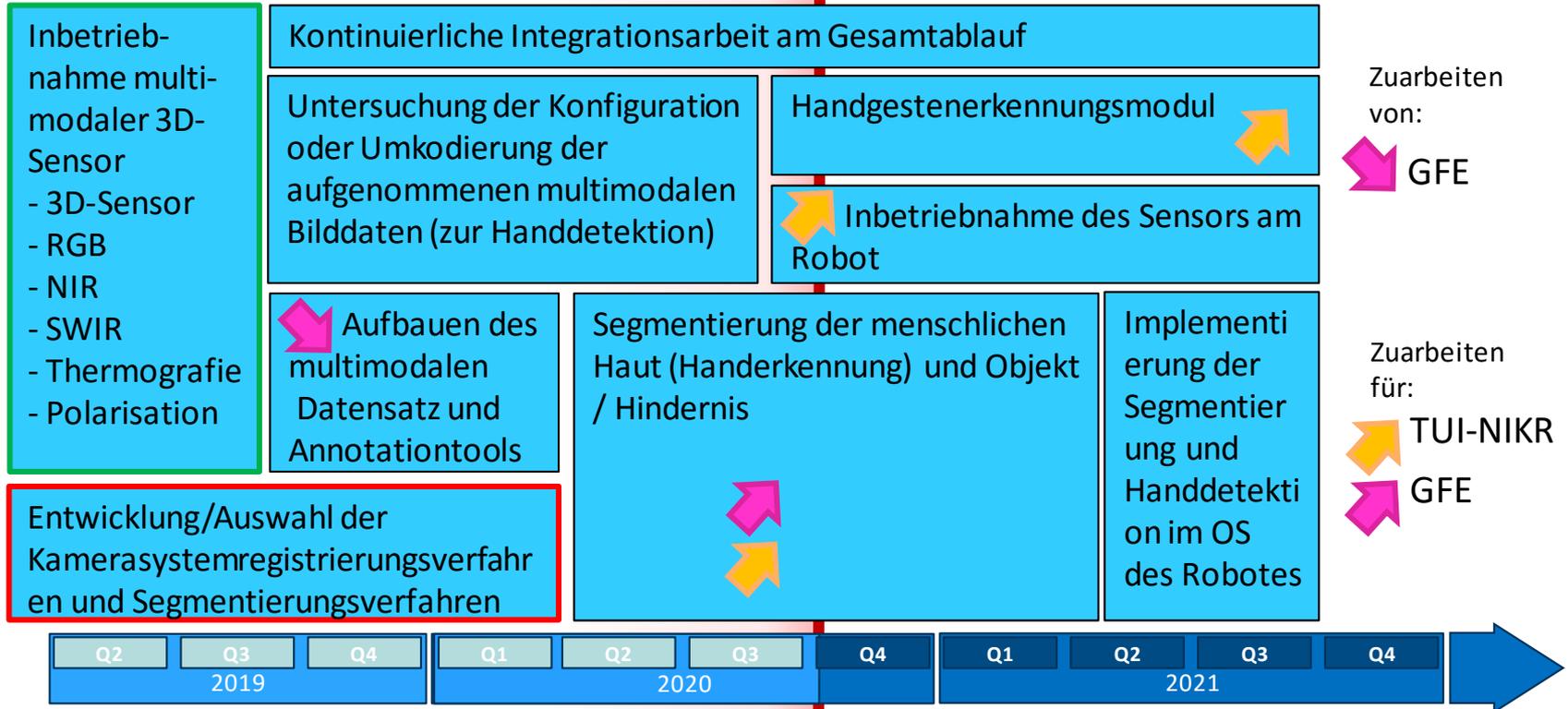


Milestones	Monat	Beschreibung
MS3	07/2020	Multispektraler 3D-Sensor ist betriebsbereit und auf Roboter integriert

Multimodales 3D-Sensorsystem auf Tiago-Roboter

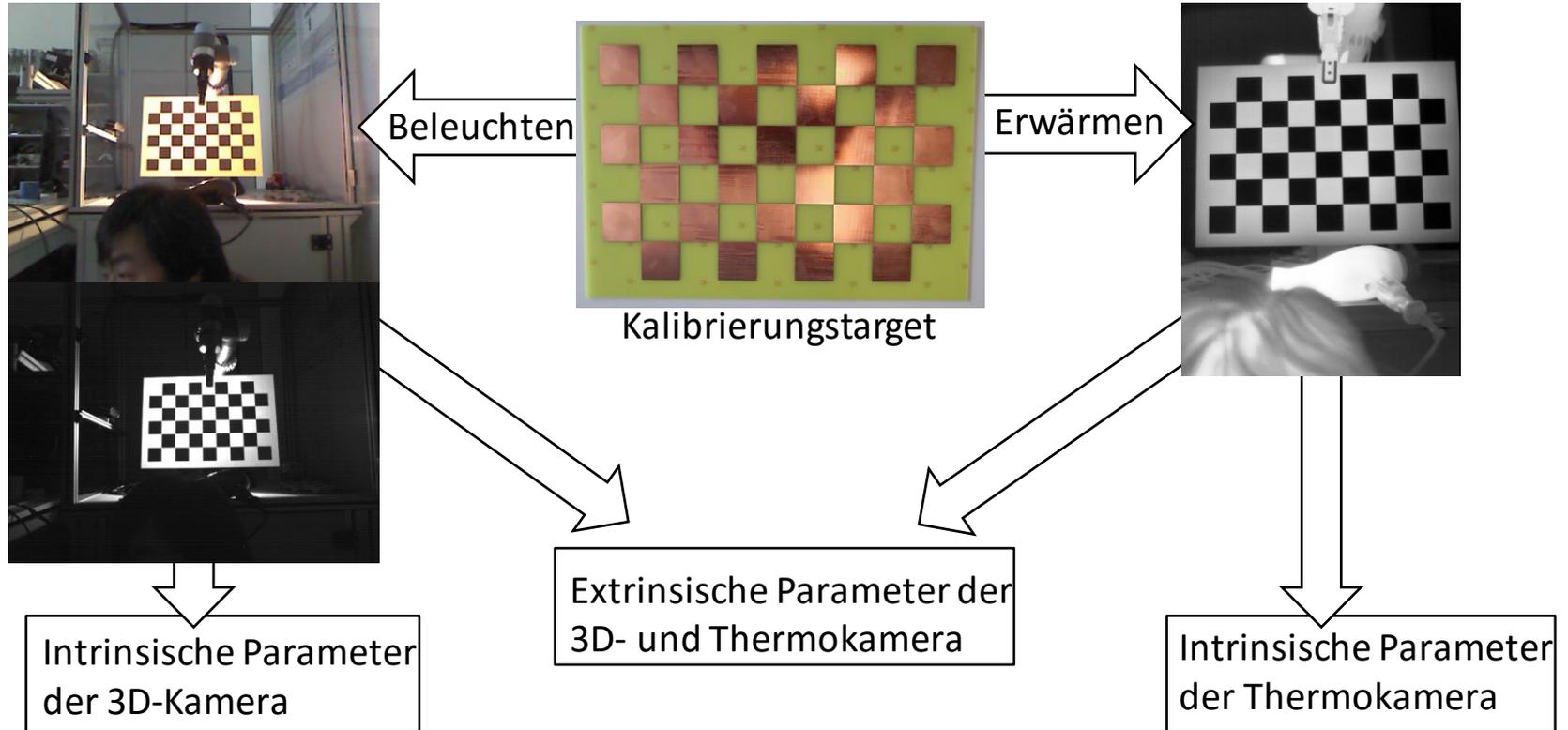


Arbeitsplan für TUI-QBV über die Projektlaufzeit



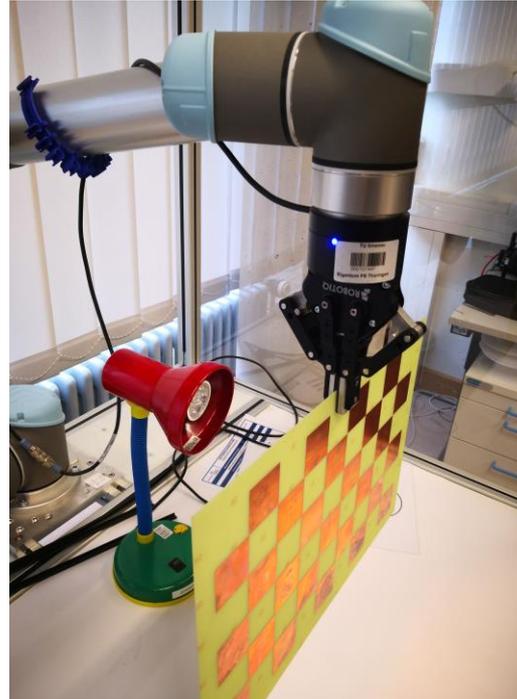
Registrierung des Kamerasystems

- Konzept der Kalibrierung der Extrinsic-Parameter der Kamerasystem



Registrierung des Kamerasystems

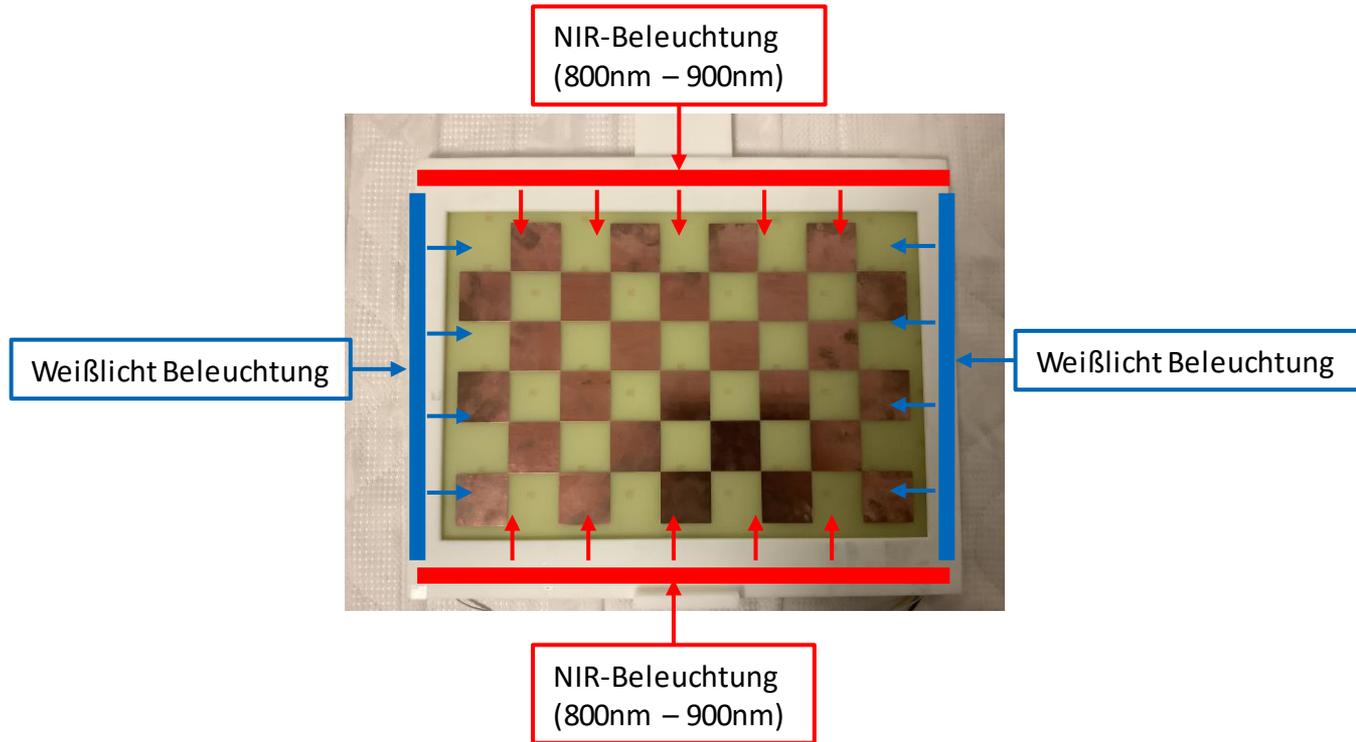
- **Problem:**
Zusatzbeleuchtung für
VIS-Bilder für
ausreichenden
Bildkontrast
notwendig. Bisher
wenig praktikabel.



Erweiterung der Kalibrierungstarget

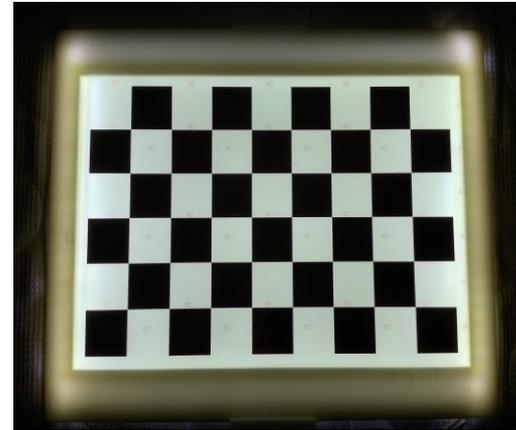
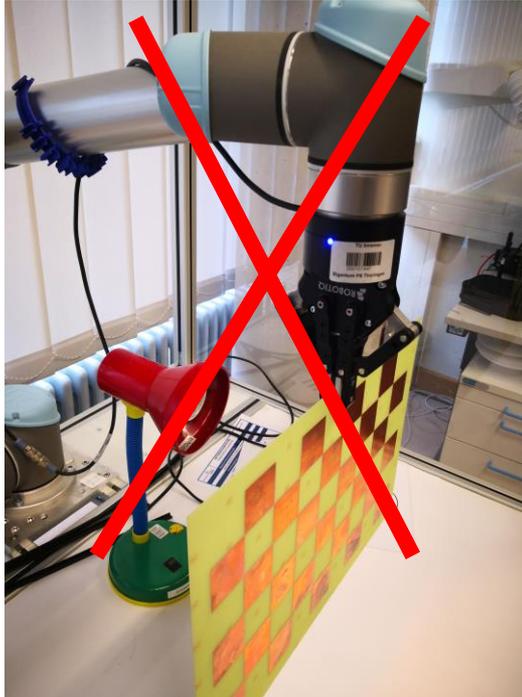
- Vorschlag:**

Kalibriertarget mit seitlicher Beleuchtung über LED-Zeilen.

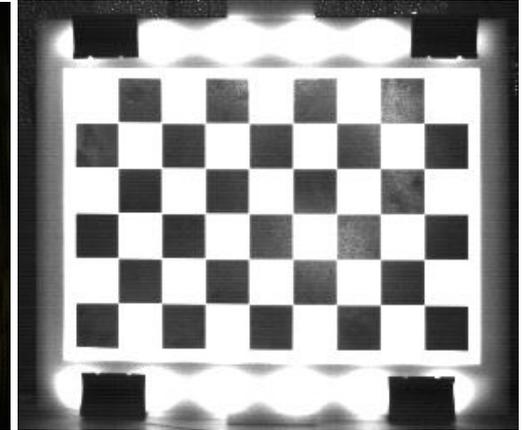


Erweiterung der Kalibrierungstarget

- Kalibrierungstarget mit eigene Beleuchtungen (Weißlicht-LED zum RGB-Kamera und Infrarot-LED zum NIR-Kamera)
 - Einfache Aufbauten -> viel bessere Qualität des Kalibrierungsbilds

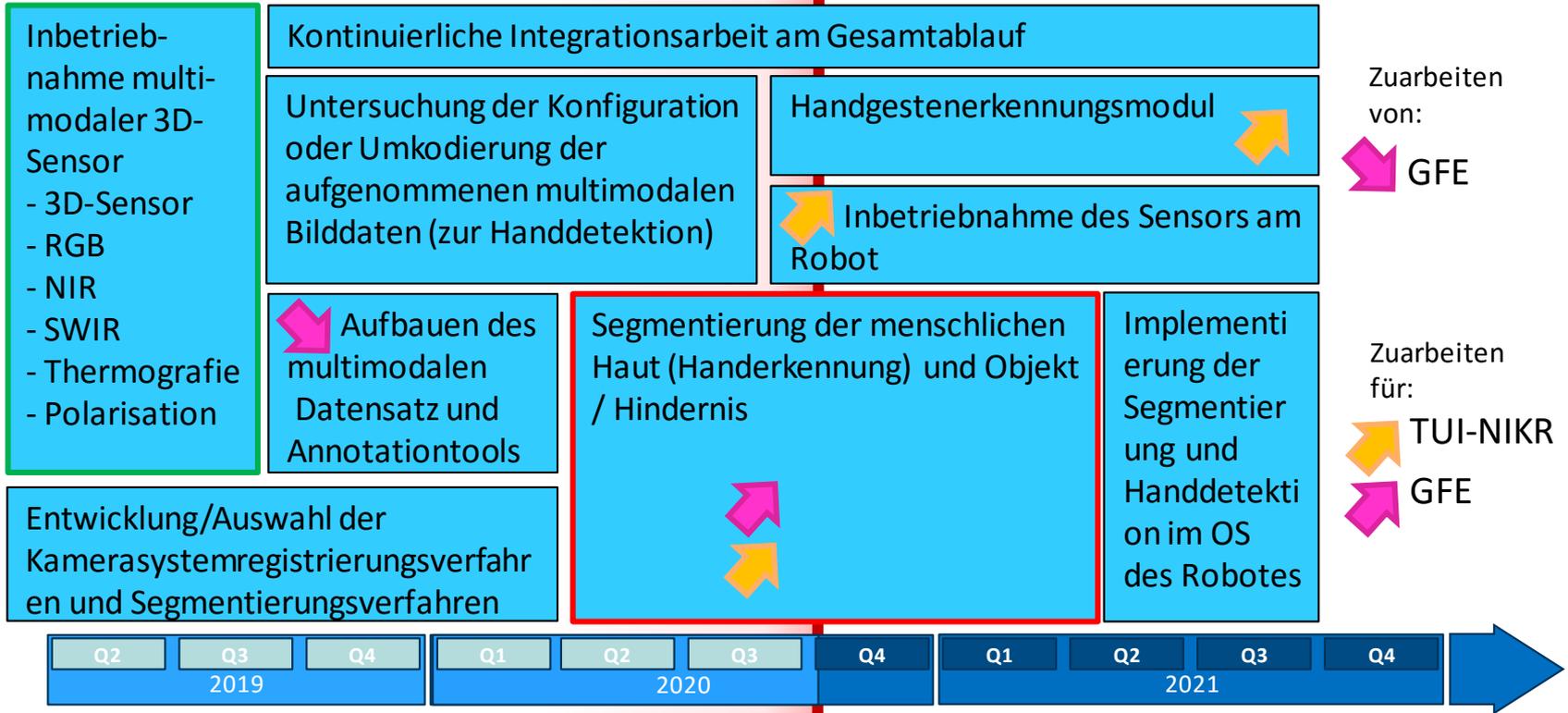


Farb-Kalibrierungsbild

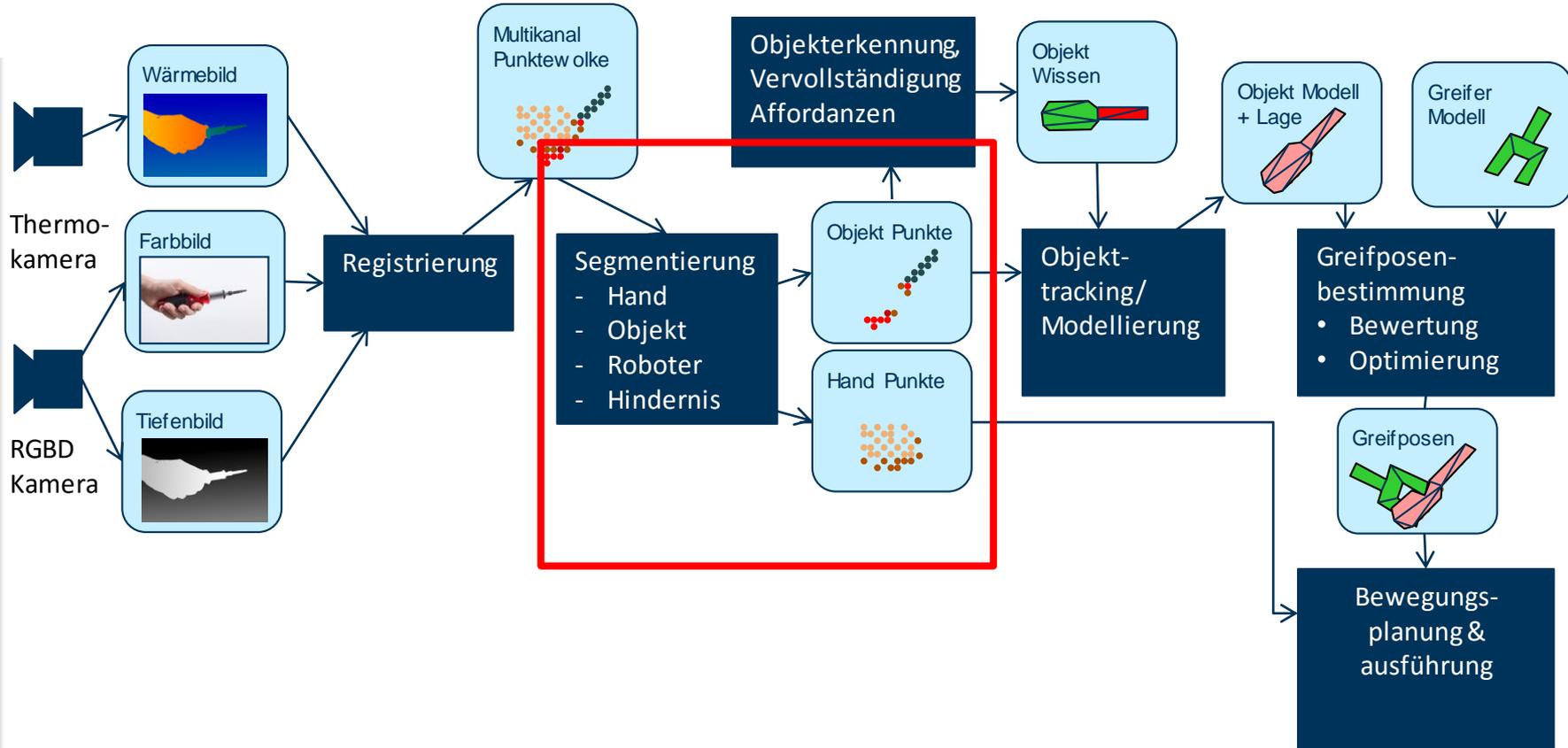


NIR-Kalibrierungsbild

Arbeitsplan für TUI-QBV über die Projektlaufzeit

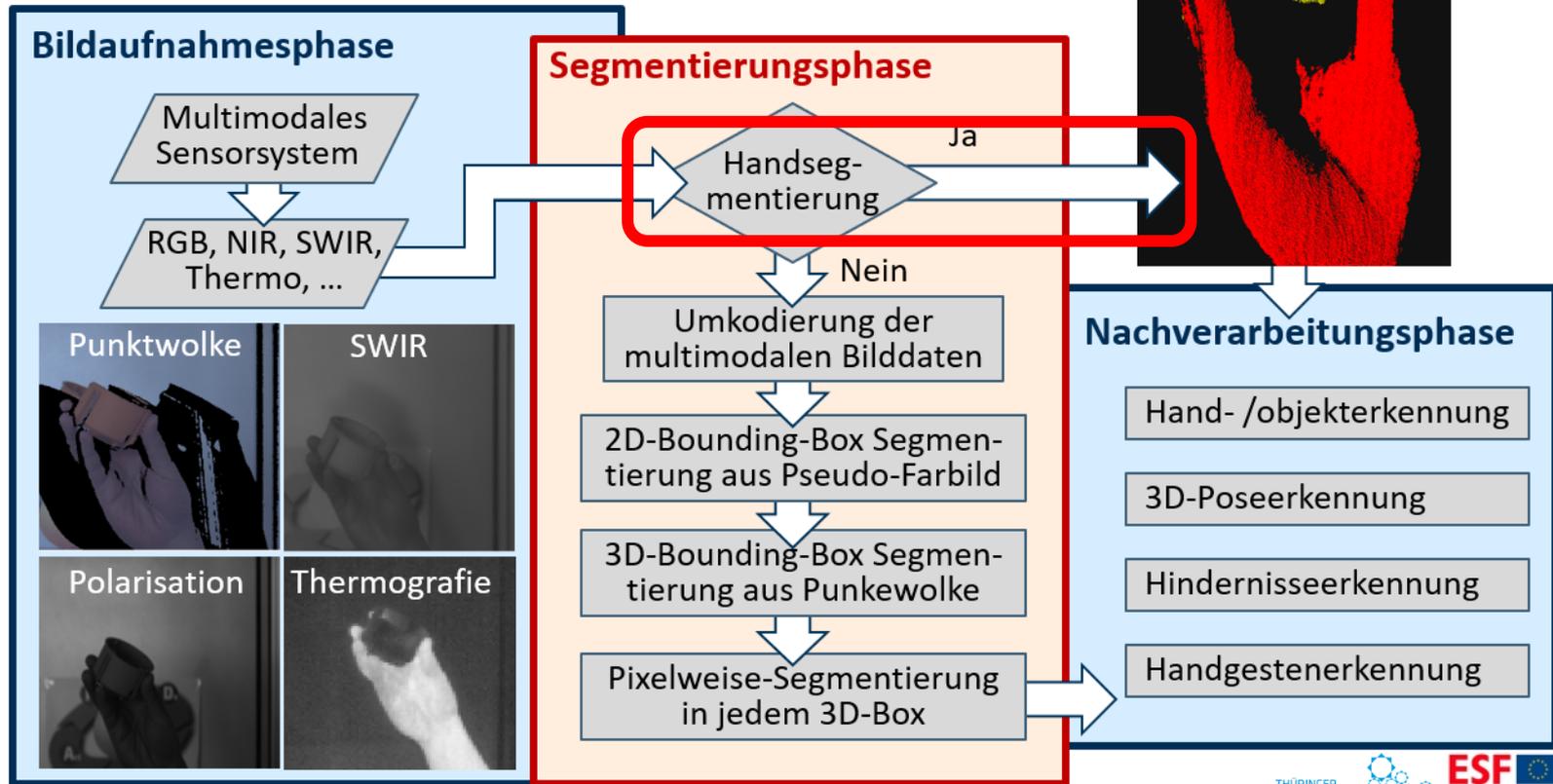


Datenverarbeitungspipeline (a) für den Zugriff



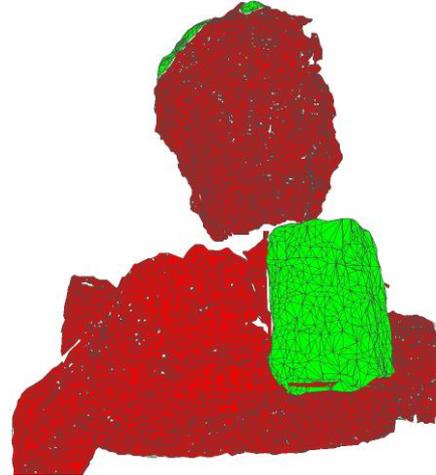
Segmentierung

- Konzept der Segmentierung



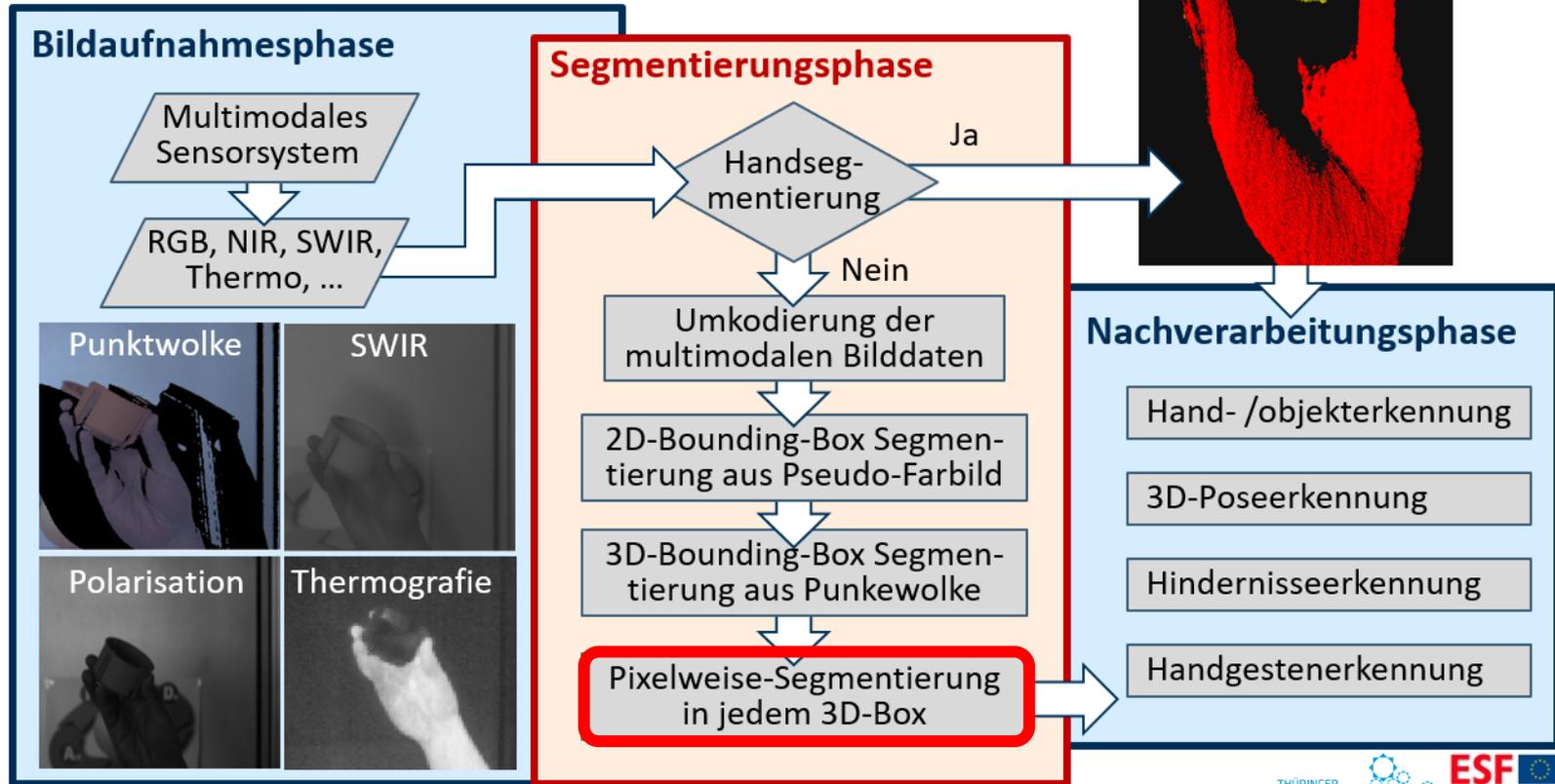
Segmentierung

- **Stand 05/2020 - Ausgangspunkt:**
 - Segmentierung basiert auf 3D-Punktwolke und 2D-Bild (RGB-Bild und Thermo-Bild)
 - Mit der Hilfe der Registrierung zwischen Thermokamera und 3D-Kamera kann die Segmentierung durch einen Temperaturschwellwert durchgeführt werden



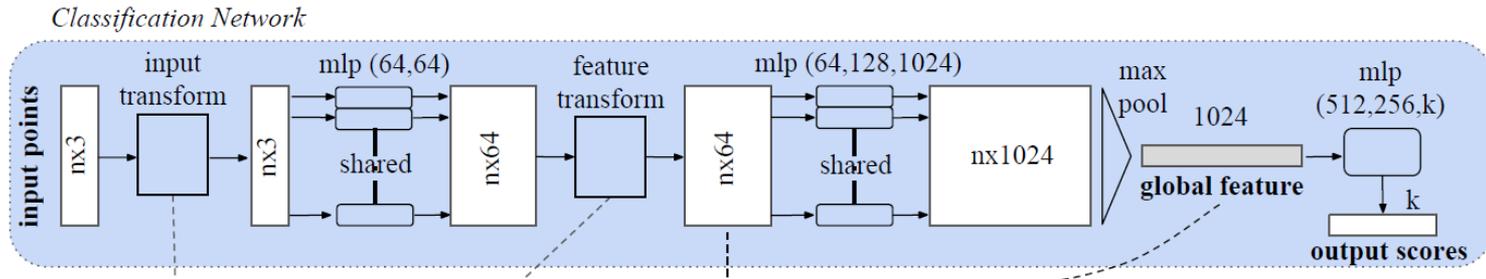
Segmentierung

- Konzept der Segmentierung



Neuronales Netzwerk für Punktwolkesegmentierung (PointNet, PointNet++)

- Ausgangspunkt

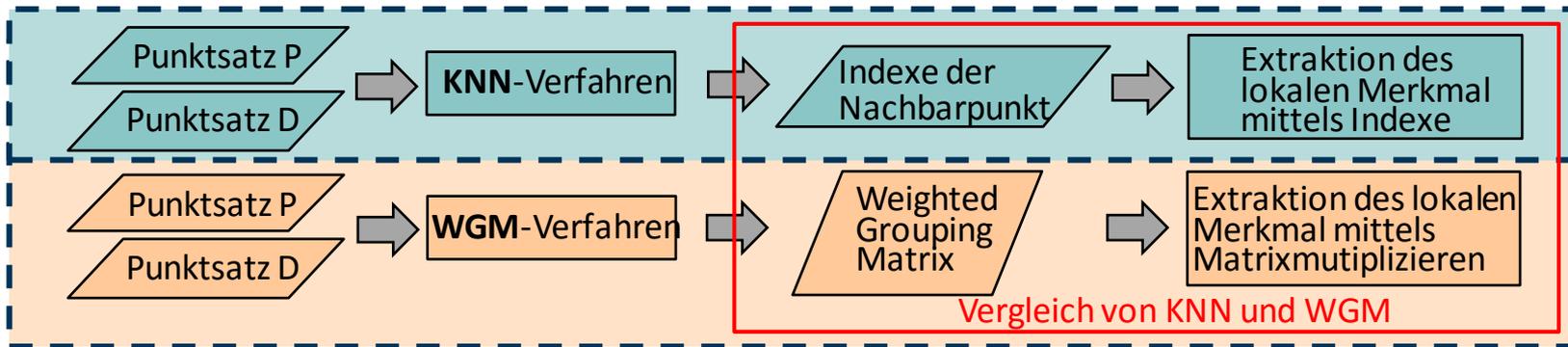


- **PointNet:** Keine Nachbarsuche -> kein lokales Merkmal wird extrahiert
- **PointNet++:** Mittels **KNN** (K-Nearest-Neighbor) wird Nachbarsuchen durchgeführt, aber die Zeitaufwand von **KNN** ist sehr groß.
- **Erstes Ziel:** Entwicklung eines effizienten Nachbarsuchverfahrens

➔ **Vorschlag: WGM-Verfahren**

Effizientes Nachbarsucheverfahren: Weighted Grouping Matrix

- Erreichter Stand: **WGM-Verfahren (Weighted Grouping Matrix)**
 - **Problemstellung:** Es gibt ein Pointset P mit Merkmalen und ein unterabgetastetes Pointset D . Für jeden Punkt in D soll lokales Merkmal aus P extrahiert werden.
 - **WGM** ist eine Gewichtsmatrix. Die Gewichte in **WGM** hängen von der Distanz der Punktpaare zwischen P und D ab.

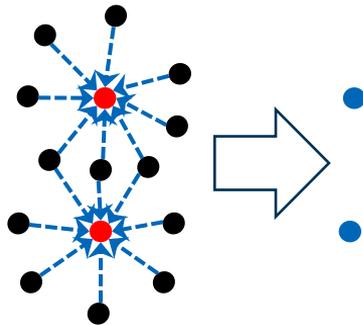


- **Zeitaufwand:** Für Pointsets: P (10k Punkte), D (10k Punkte)

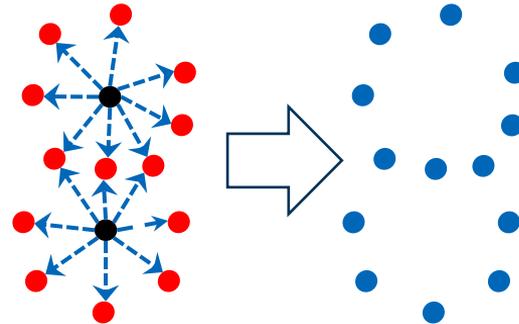
0,48 ms (**WGM**) vs 174 ms (**KNN**)

Verwendung der Weighted Grouping Matrix

- Einsatz der *WGM* in Segmentierungsnetzwerk für multimodale Punktwolken
 - (1) Extraktion der lokalen Merkmale
 - (2) Aggregation der lokalen Merkmale aus den gesampleten Nachbarn:



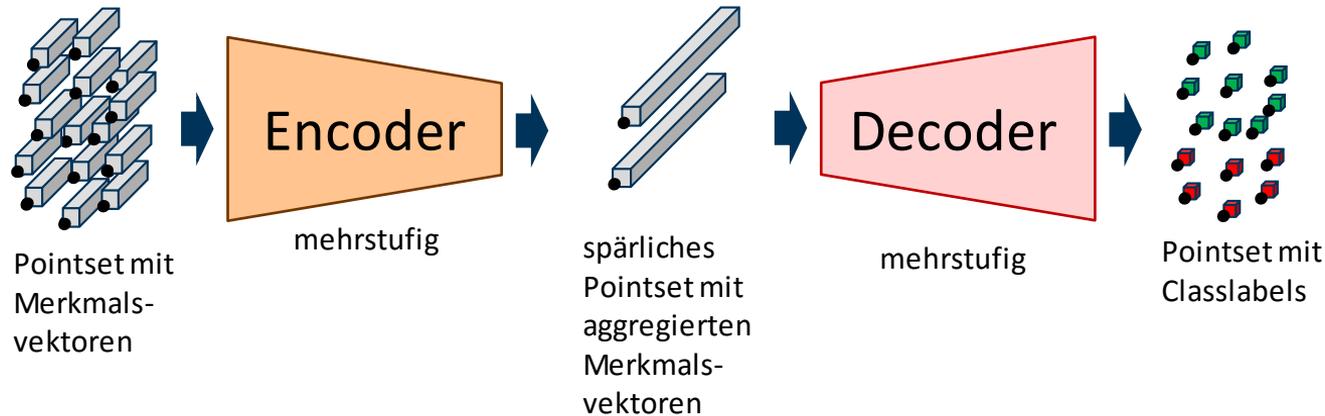
- (1)
- Pointset P
 - Sampled Pointset D
 - Lokales Merkmal beschreibt die Struktur der Punkte in der Nachbarschaft



- (2)
- Pointset P
 - Pointset D
 - Interpoliertes lokales Merkmal

Verwendung der Weighted Grouping Matrix

- Einsatz der *WGM* in Segmentierungsnetzwerk für multimodale Punktwolken
 - Segmentierung mittels typischer Encoder-Decoder Architektur

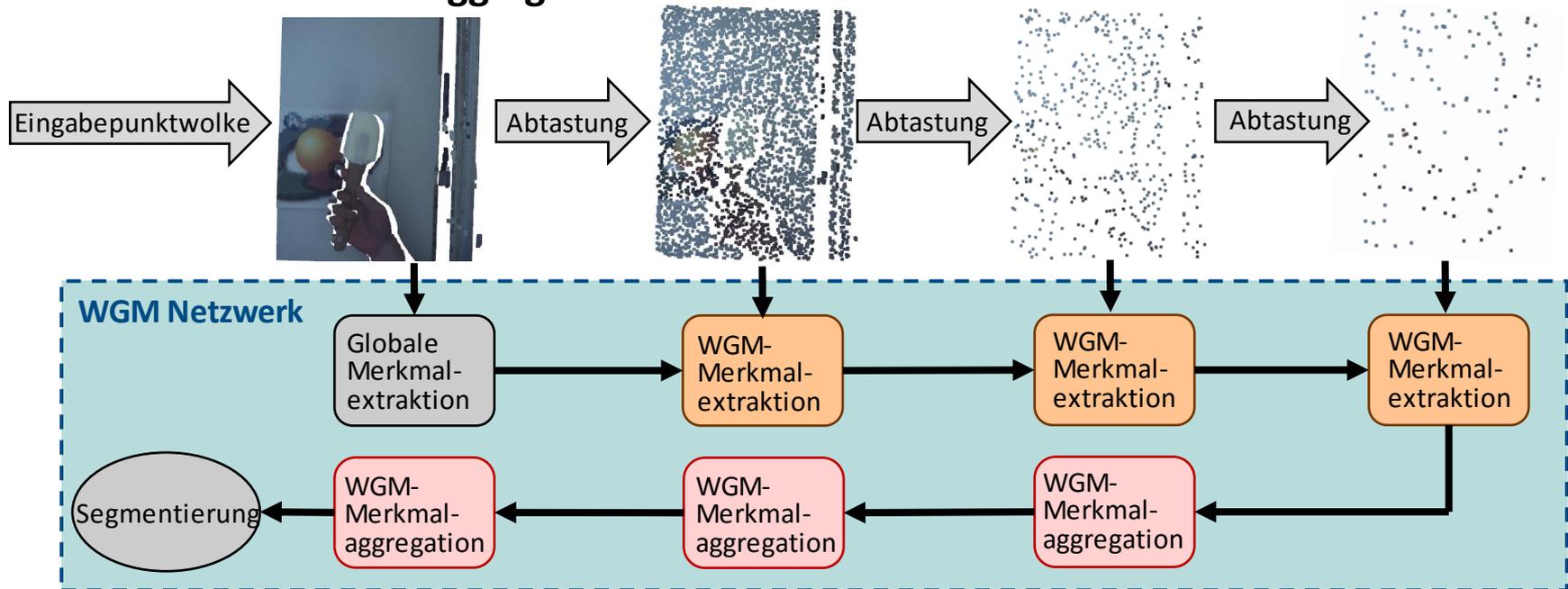


- Aggregation von Merkmalsinformationen aus der Nachbarschaft während Komplexität der Punktwolke abnimmt
- Anschließend Extrapolation der klassifizierten Merkmale der spärlichen Punkte auf die volle Auflösung

WGM Netzwerk

• Konzept des Netzwerks

- Eingabepunktswolke wird homogen in drei verschiedenen Auflösungen abgetastet
- Punktwolkensegmentierung mit der Hilfe von **WGM-Merkmalextraktion** und **WGM-Merkmalaggregation**



WGM Netzwerk

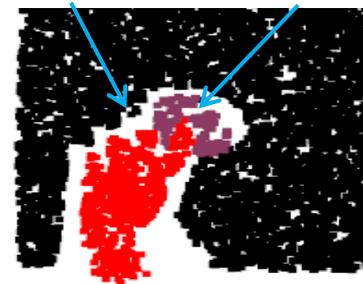
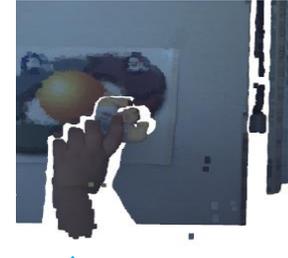
- Ergebnisse**

Segmentierung der multimodalen Punktcloud

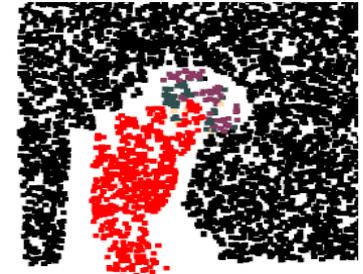
	Mean IoU (%)	Zeitaufwand (ms)
XYZ		
PointNet	34.1	-
PointNet++	56.5	-
ours	55.9	-
XYZRGB		
PointNet	57.2	-
PointNet++	70.9	-
ours	68.7	-
XYZRGBT		
PointNet	68.6	7.3
PointNet++	74.2	361.7
ours	73.1	7.9

IoU: Intersection over Union (IoU)

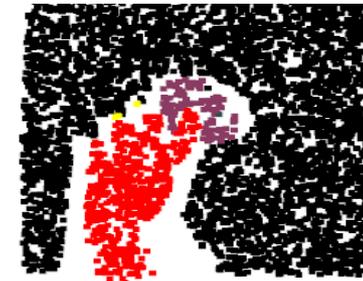
RGB point cloud



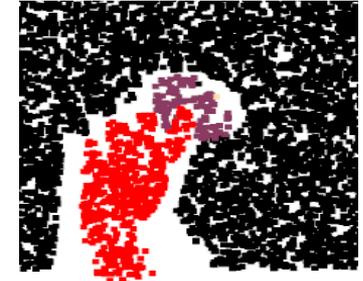
Grund truth



Segmentation on XYZ



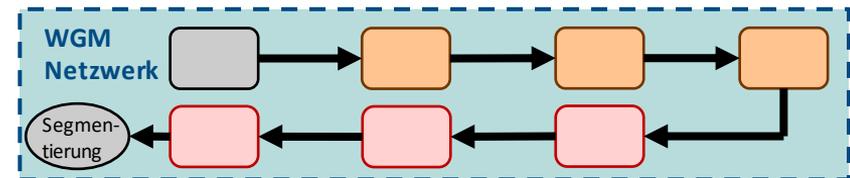
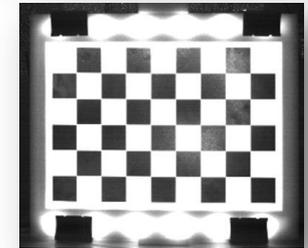
Segmentation on XYZRGB



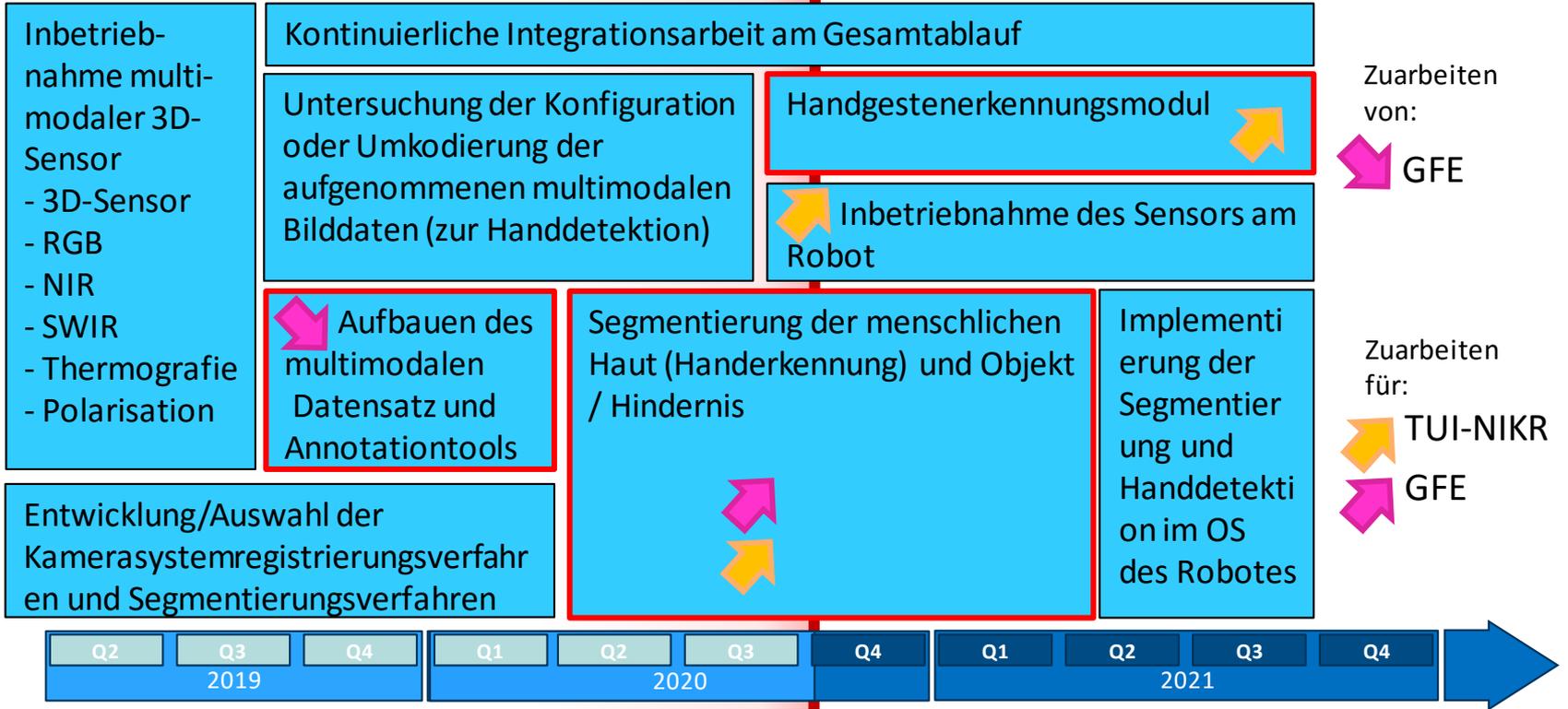
Segmentation on XYZRGBT

Zusammenfassung Ergebnisse

- Integration neues Thermokamerasystems auf dem Tiago-Roboter
- Neues Kalibriertarget für multimodalen Sensor
- Erweiterte Bild- oder Punktwolkesegmentierung mit neuronalen Netzwerk durch WGM-Verfahren



Ausblick der Tätigkeiten für TUI-QBV für 2021



Forscherguppe SONARO

Smarte Objektübernahme und –übergabe für die
nutzerzentrierte mobile Assistenzrobotik

Beiratstreffen am 28.10.2020

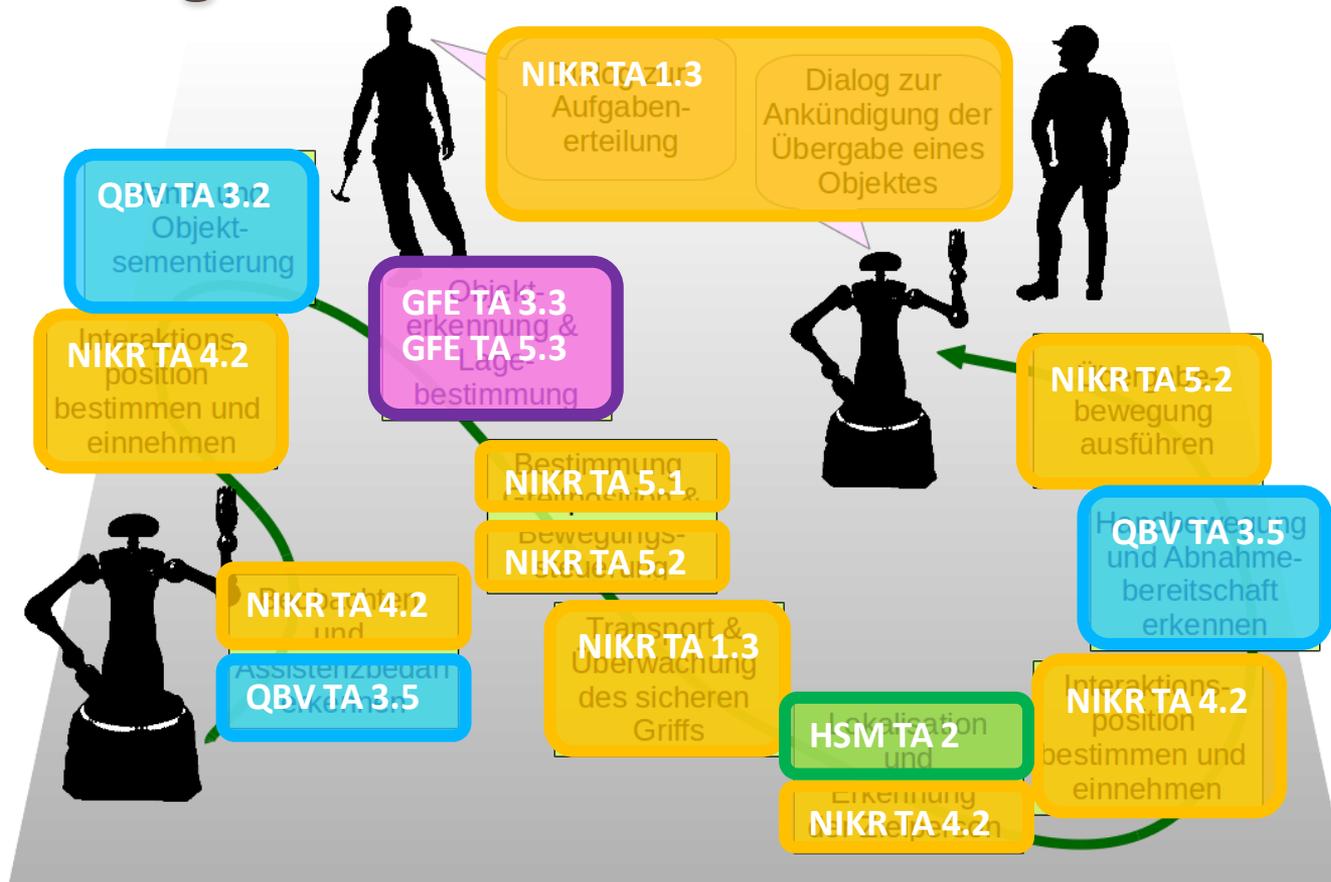
Vorstellung der Arbeiten von GFE Schmalkalden e. V.

Dr. Daniel Garten

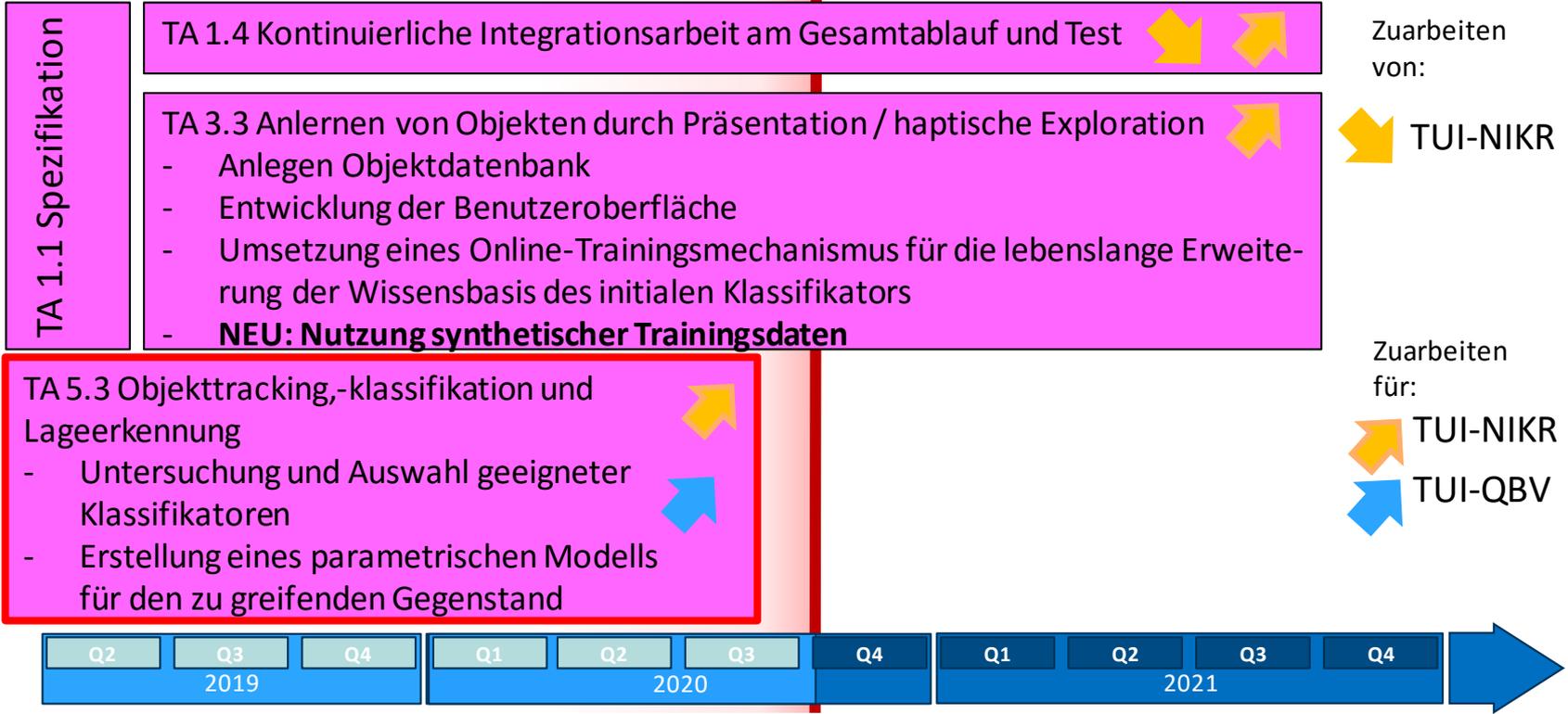
www.sonaro-projekt.de



Einordnung der GFE in SONARO



Arbeitsplan für GFE über die Projektlaufzeit



Zielstellung des Teilprojektes

- **AP 5.3: Objekttracking, -klassifikation und Lageerkennung**
 - Untersuchungen zur Auswahl geeigneter Verfahren der Objektlokalisierung (Deep-Learning, klassische maschinelle Lernverfahren oder Verfahren des direkten Vergleiches der segmentierten 3D-Punkte wie ICP - Iterative-Closest-Point)
 - Erstellung eines parametrischen Modell des zu greifenden Gegenstands
 - kontinuierliche Aktualisierung der Lage des Gegenstandes als Basis für das Greifen durch den Roboter
 - Entwicklung von Prognosemodellen für die Schätzung der zukünftigen Objektposition (z. B. Kalman-Filter)

Zielstellung des Teilprojektes

- **AP 3.3: Anlernen von Objekten durch Präsentation / haptische Exploration**
 - Anlegen Objektdatenbank
 - Entwicklung der Benutzeroberfläche
 - Umsetzung eines Online-Trainingsmechanismus für die lebenslange Erweiterung der Wissensbasis des initialen Klassifikators
- ⇒ Derzeitiger Ansatz [1] basiert auf synthetisch erzeugten Bildszenen (Basis bildet das 3-D-Visualisierungsprogramm Blender)
- ⇒ Erzeugung realer Bildszenen für das Anlernen tritt dabei in den Hintergrund

[1] Hasson, Yana et al.: Learning joint reconstruction of hands and manipulated objects, CVPR, 2019

Arbeitspaket Objekttracking,-klassifikation und Lageerkennung

- **Ausgangspunkt:** Klassifikatormodell zur Schätzung der Meshes von Hand und Gegenstand auf Basis eines einzelnen RGB-Bildes der Szene [1] an MIRA angebunden
- **Erreichter Stand:** Tiefendaten und 2-D-RGB-Daten sind registriert, Hand- und Objekt-Mesh sind visualisierbar, optimierte Abarbeitung der Erkennungsroutinen mit 2 bis 5 fps
- **Weiteres Vorgehen:** weiterer Test der Verfahren im simulierten Praxisumfeld

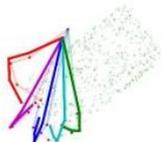
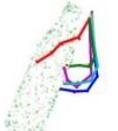
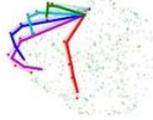
[1] Hasson, Yana et al.: Learning joint reconstruction of hands and manipulated objects, CVPR, 2019

Aktuelle Arbeiten

Hand-Joints
Objektpunkte

Hand-
Meshes

Objekt-
Meshes



Ergebnis:

Derzeitiger Ansatz liefert Hand-Joints sowie Hand- und Objekt-Meshes

Problem:

Greifende Hand verdeckt Teile des Objektes, Objektschätzung ist weniger stabil als Handschätzung

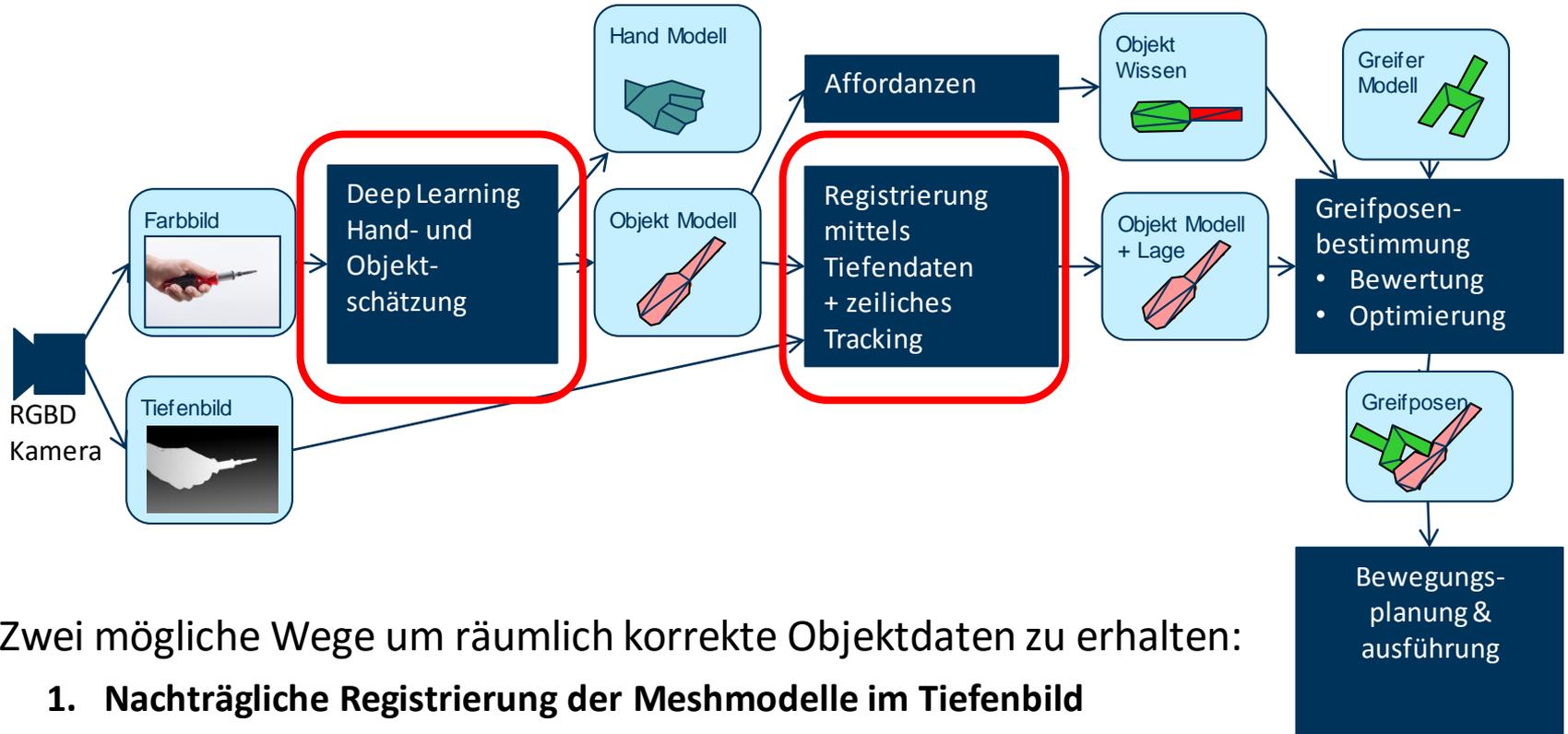
Ansatz:

Auffüllen fehlender Objektpunkte durch Clustering-Verfahren

Derzeitiger Arbeitsstand

1. Nachtrainieren des Verfahrens mit zusätzlichen Objekt-Meshes sowie Anpassung des Ergebnis-Meshes an die Daten des Tiefensensors zur Bestimmung der Objektlage in Roboter- bzw. Globalkoordinaten sind derzeitiger Forschungsgegenstand
2. Die Objekterkennung ist mit ca. 3 – 5 fps ausreichend schnell
3. Bei zu starker Neigung des Objektes sowie unscharfer Abbildung kommt es zu Erkennungsfehlern
4. Registrierung von RGB-Bilddaten auf denen die Objektlokalisierung erfolgt und 3-D-Daten des Tiefensensors ist umgesetzt
5. Ergebnis des Erkennungsprozesses sind Objekt-Meshes (Tracking, Lokalisation möglich) und keine Objektklassen
=> Erweiterung um Klassifikation ist nötig

Datenverarbeitungspipeline (b) für den Zugriff



Zwei mögliche Wege um räumlich korrekte Objektdaten zu erhalten:

1. Nachträgliche Registrierung der Meshmodelle im Tiefenbild
2. Erweiterung der Netzwerkarchitektur um Tiefenbilder als Eingang

Lokalisation von Hand- und Objekt-Mesh

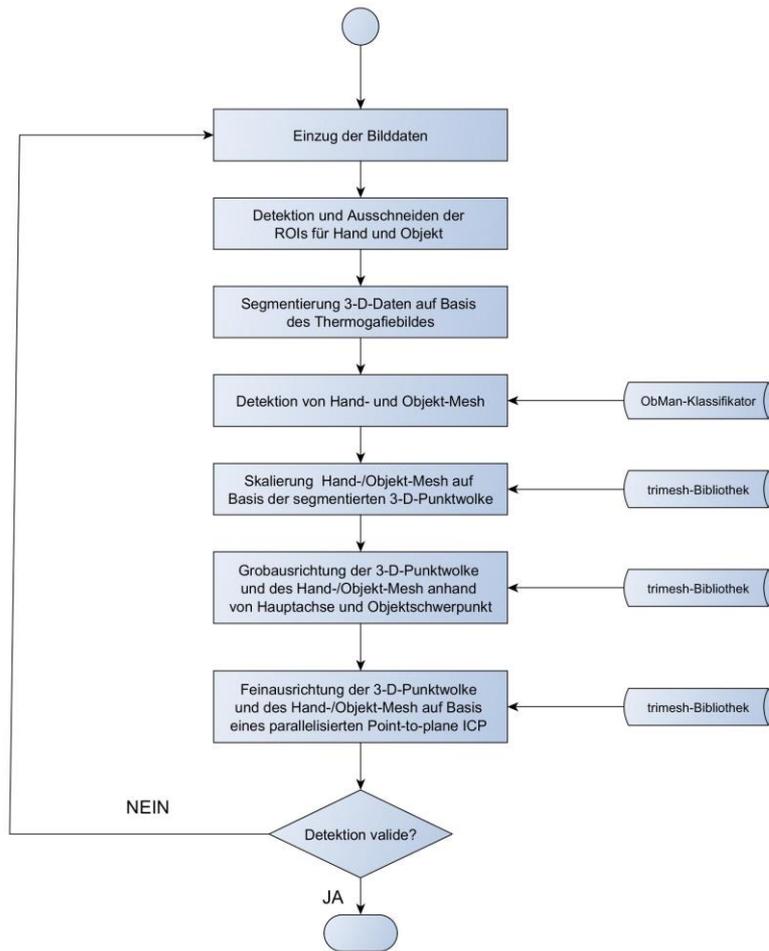
The screenshot shows the MIRA Center software interface. On the left, there is a 'Channels' tree view with folders for 'hardware', 'localization', 'maps', and 'robot'. Below it is a 'Visualization Control' panel with a table of properties:

Property	Value
General	
/Debugmag...	enabled
/Outimage	enabled
/hardware/d...	enabled

The main window is split into two views: a 2D depth map on the left and a 3D RGB live image on the right. A hand mesh is overlaid on the 3D image. Below the 2D view, a white box contains the text 'Tiefendaten' and 'Live-Bild'. Below the 3D view, another white box contains the text 'geschätztes Mesh von Hand und Objekt im RGB-Live-Bild'. On the right side, there are panels for 'Authorities' and 'Properties'. The 'Authorities' panel shows a tree view with 'ObjMa...' and 'hardw...'. The 'Properties' panel shows 'No properties available'. At the bottom, there is a 'Player' window with playback controls and a system tray showing system information like 'Time: 15:25:45.0', 'Memory: 2.1 GB', 'CPU: 337.5 %', 'In: 0.000 KBytes', 'Out: 0.000 KBytes'.

Problem: Tiefendaten und Meshes sind nicht zueinander registriert

Registrierungs-Pipeline (bisherige Untersuchungen)



Nachteile

- Hohe Laufzeit
- geringe Genauigkeit
- teilweise keine Konvergenz erzielt

⇒ Schätzung der Hand- und Objekt-Pose durch Klassifikatormodell

Lokalisation von Hand- und Objekt-Mesh

Live-Bild RGB-Sensor

Live-Bild mit Hand-Joints

Live-Bild Tiefensensor

Weltkoordinatensystem 3-D-Objekt

Channels

Name	Updates
hardware	
depth	
localization	
amcl	
maps	
MapFrame	0 (0 Hz)
robot	
arm_1_link	0 (0 Hz)
arm_2_link	0 (0 Hz)
arm_3_link	0 (0 Hz)
arm_4_link	0 (0 Hz)
arm_5_link	0 (0 Hz)
arm_6_link	0 (0 Hz)

Visualization Control

Property	Value
General	
Axes	<input type="checkbox"/> disabled
/ObjectMesh	<input checked="" type="checkbox"/> enabled

LogView

Level	Time	Message	Line	File	NS	Class	Function
WARNING	2020-Oct-22 13:41:11.009699	No Ogre Cg plugin found, some shaders will not be available. Install the corresp...	213	/home/dgart...			mira::OgreSi...

Properties Authorities

Time: 13:43:19.5 | Memory: 2.1 GB | CPU: 346.5 % | In: 0.000 KBytes | Out: 0.000 KBytes

Hands-Only-Detektion

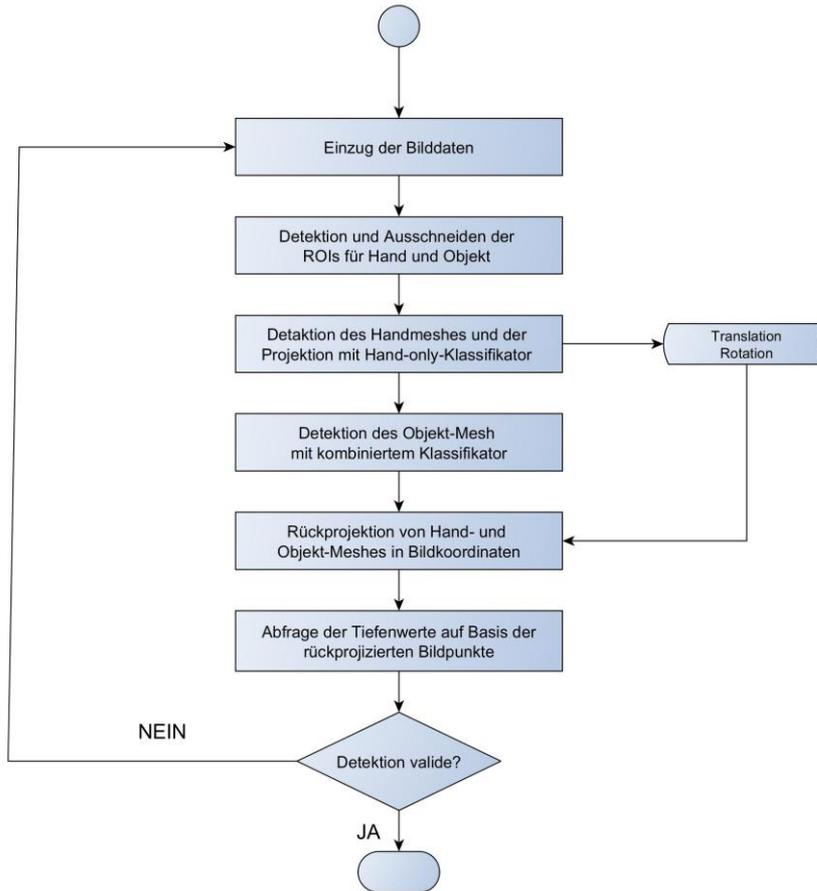
Klassifikator, welcher lediglich die Hand im Bild detektiert darüber hinaus jedoch die Parameter einer orthogonalen Projektion zur Rückprojektion der Hand in Kamerakoordinaten schätzt



Grundgedanken:

1. Handmodelle liefern aufgrund umfangreicher öffentlicher Trainingsdatensätze sehr stabile Ergebnisse
2. Schätzung der Lage im Raum auf Basis der Hand mit entsprechend trainiertem Modell möglich
3. Hand und Objekt weisen ähnliche Lage im Raum auf

Registrierungs-Pipeline (neuer Ansatz)



Vorteile gegenüber bisherigem Ansatz

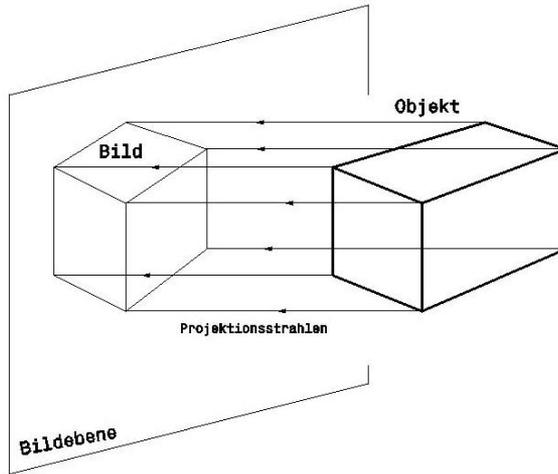
- geringere Laufzeit
- höhere Genauigkeit

Nachteile

- teilweise keine Konvergenz erzielt
- zwei Klassifikatormodelle nötig

Registrierungs-Pipeline (neuer Ansatz)

Hintergrund: Modell der orthogonalen Transformation des Hand-Meshes:



$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

mit (x, y) Bildkoordinaten
in Pixel

(X, Y) Koordinaten im
Einheitsmesh

Erkennung gültiger Hand- und Objekt-Meshes

Problem: Das neuronale Netz liefert kontinuierlich Hand- und Objekt-Meshes. Wenn keine Hand mit Objekt im Bild erkennbar sind entstehen Artefakte.

Lösungsansatz: Überwachung der Differenzen zwischen den Objekt-Meshes aufeinanderfolgender Frames sowie der Objektgrößen

Abfolge der Objekt-Mesh-Deformation beim Präsentieren eines Gegenstandes:

1. Größe des detektierten Objekt-Meshes nimmt im ersten Frame zu
2. Differenzen zwischen den Objekt-Meshes aufeinander folgender Frames nehmen ab
3. Größe des detektierten Objekt-Meshes nimmt nachdem der Gegenstand das Sichtfeld wieder verlassen hat wieder ab.

⇒ Anzeige der ermittelten Meshes nur, wenn Bedingung 1 initial erfüllt wurde und Bedingung 2 über mehrere Frames hinweg erfüllt ist bis Bedingung 3 eintritt

Stand der Integration in die Roboterumgebung MIRA

Beispielvideo: Erkennung eines präsentierten Mobiltelefones

The screenshot displays the MIRA Center software interface. On the left, there is a 'Channels' panel with a tree view showing hardware, localization, and robot components. Below it is the 'Visualization Control' panel with a 'Property' table:

Property	Value
General	
img->	enabled

The main area contains four windows: a top-left 2D camera view of a room, a top-right empty 2D window, a bottom-left 2D image processing view showing a binary mask of the room, and a bottom-right empty 3D window. A 'LogView' window at the bottom shows a warning message:

```

WARNING 2020-Oct-22 13:51:59.813910 No Ogre Crt plugin found, some shaders will not be available, install the corresp... 213 /home/dsart... mira-OgreSi...
    
```

The system tray at the bottom shows the time as 13:52:08.0, memory usage at 2.1 GB, and CPU usage at 356.3%.

Stand der Integration in die Roboterumgebung MIRA

Beispielvideo: Erkennung eines präsentierten Schraubendrehers

The screenshot displays the MIRA Center software interface. On the left, a 'Visualization Control' panel shows the 'img' property is 'enabled'. The main area contains four viewports: a top-left 2D camera view of a workshop, a top-right empty 2D view, a bottom-left 2D view showing a dark scene, and a bottom-right 3D view showing a blue plane. A 'Development' panel on the right shows a tree view with 'ObMa... OK', 'hardw...', and 'de...' nodes. At the bottom, a 'LogView' console displays a warning: 'No Ogre Cg plugin found, some shaders will not be available. Install the corresp...'. The system tray at the bottom shows the time as 14:52:03.7, memory usage at 2.1 GB, and CPU usage at 347.8%.

Forscherguppe SONARO

Smarte Objektübernahme und –übergabe für die
nutzerzentrierte mobile Assistenzrobotik

Beiratstreffen am 28.10.2020

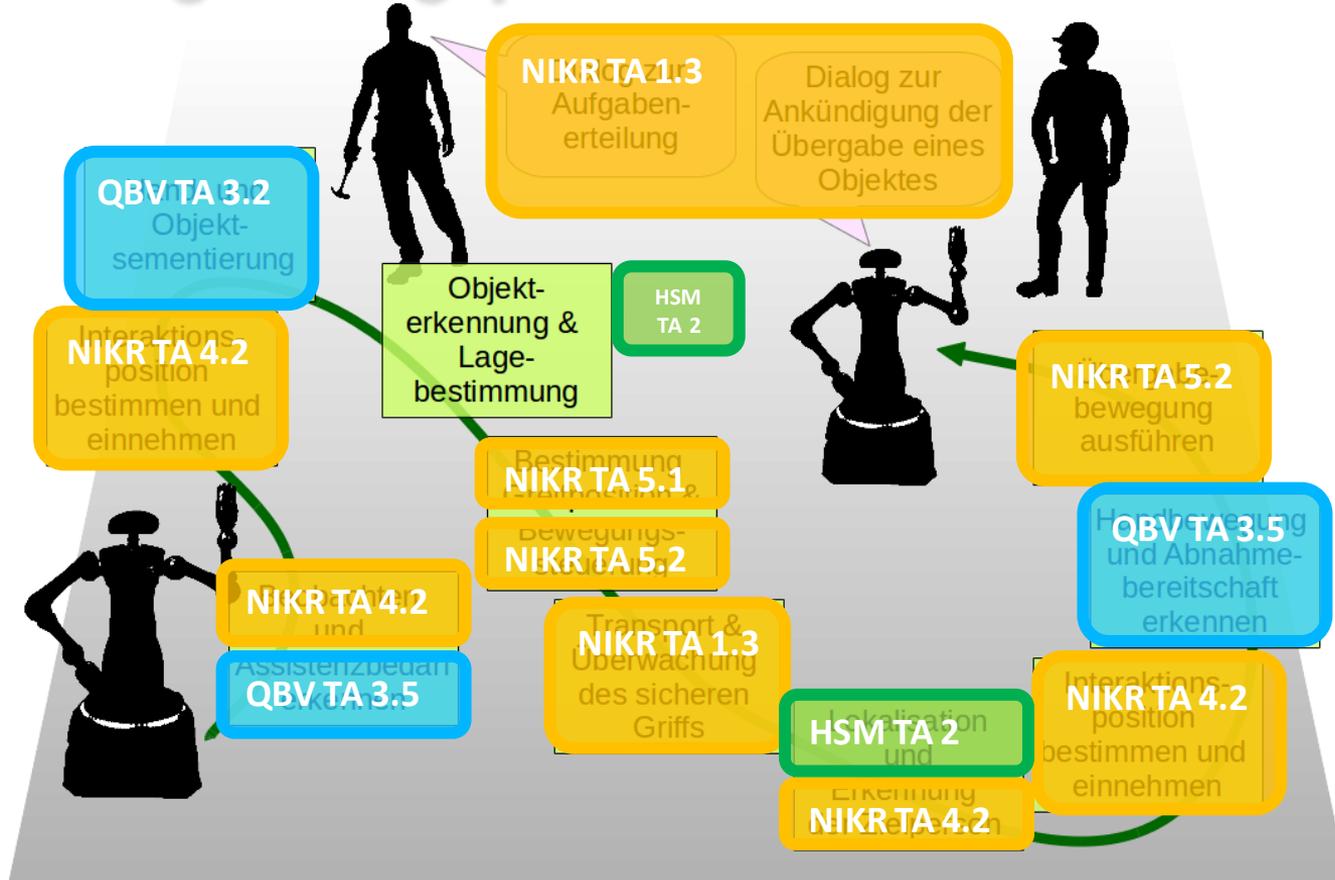
Vorstellung der Arbeiten von HSM FG EDS

M.Sc. Manuel Schneider

www.sonaro-projekt.de



Abdeckung durch geplante AP in SONARO



Zeitliche Planung über die Projektlaufzeit

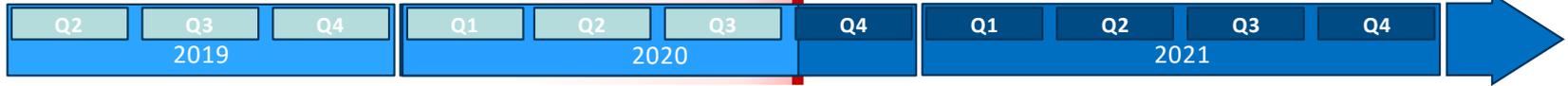
Anforderungs-
analyse,
Spezifikation und
Schnittstellen-
definition

Anbindung
an die
genutzten
Frameworks



Optimierung Hinderniserkennung

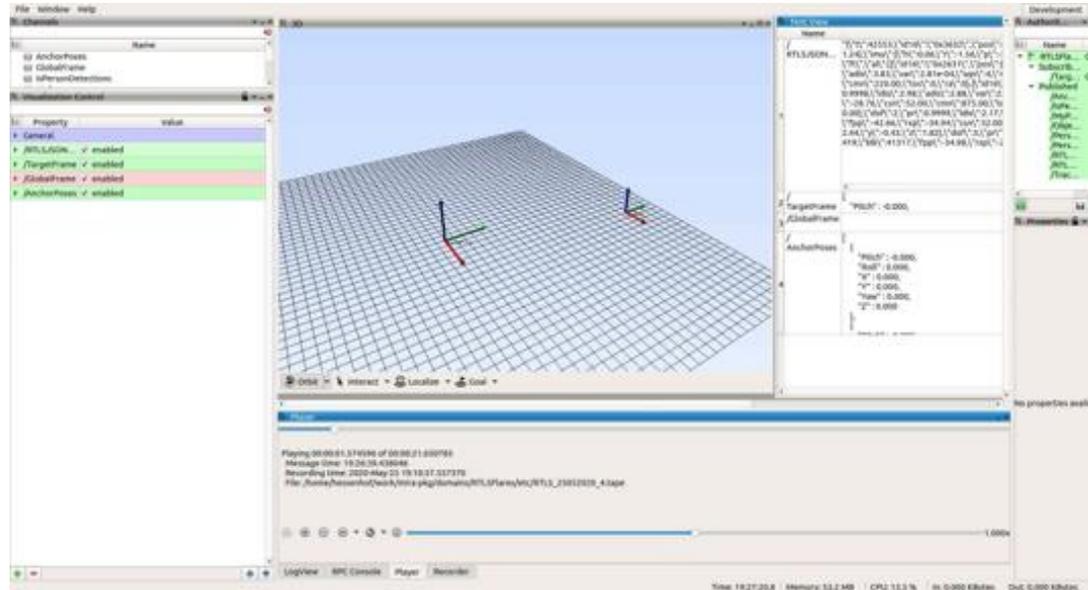
Hardwareoptimierung



Zuarbeiten
für:
 TUI-NIKR

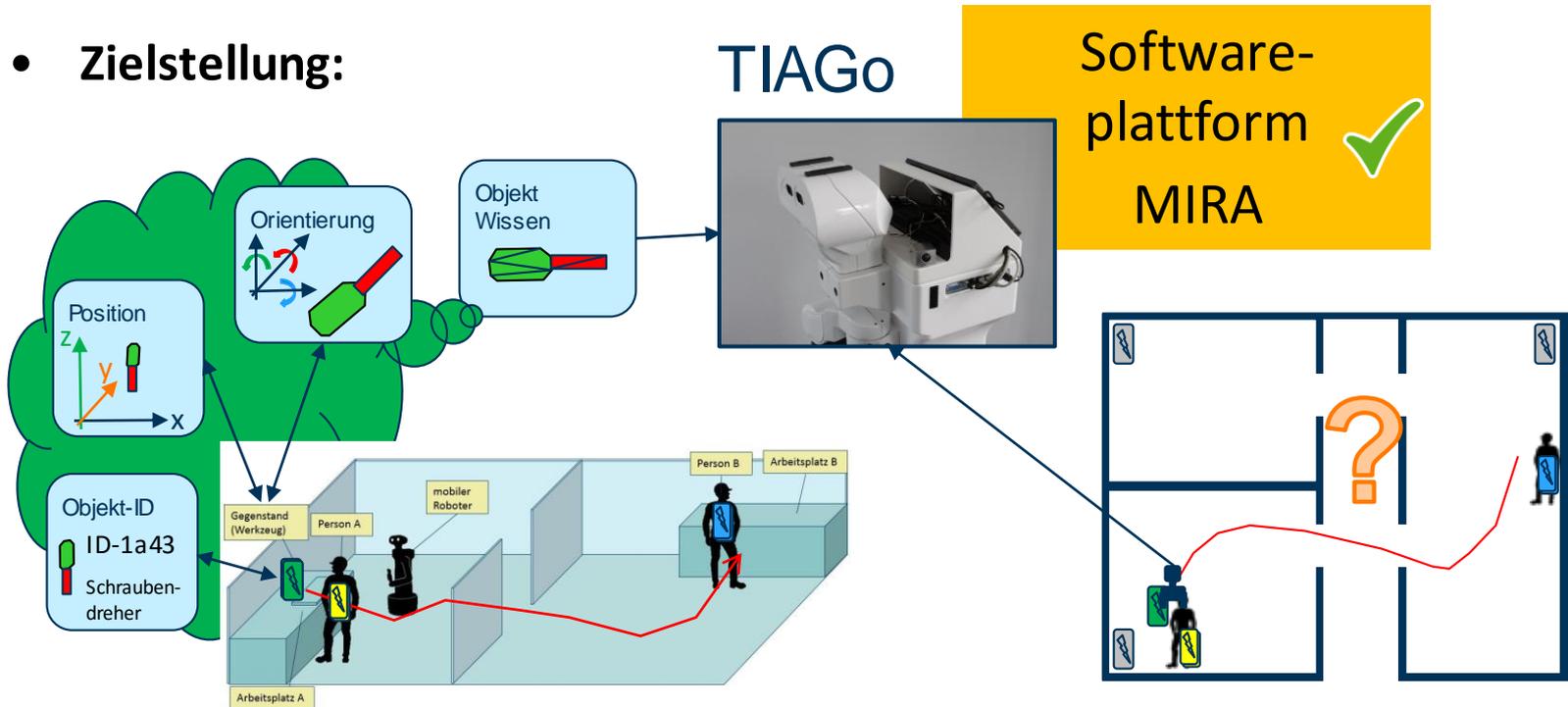
Anbindung an die genutzten Frameworks

- **Ausgangspunkt:**
 - RTLS Flares Funkortungssystem
 - Stationäre Flares bilden ein Koordinatensystem (Anchor)
 - Mobile Flares können sich bzgl. dieser Koordinaten lokalisieren



Anbindung an die genutzten Frameworks

- Zielstellung:



- Informationen über Zielobjekte

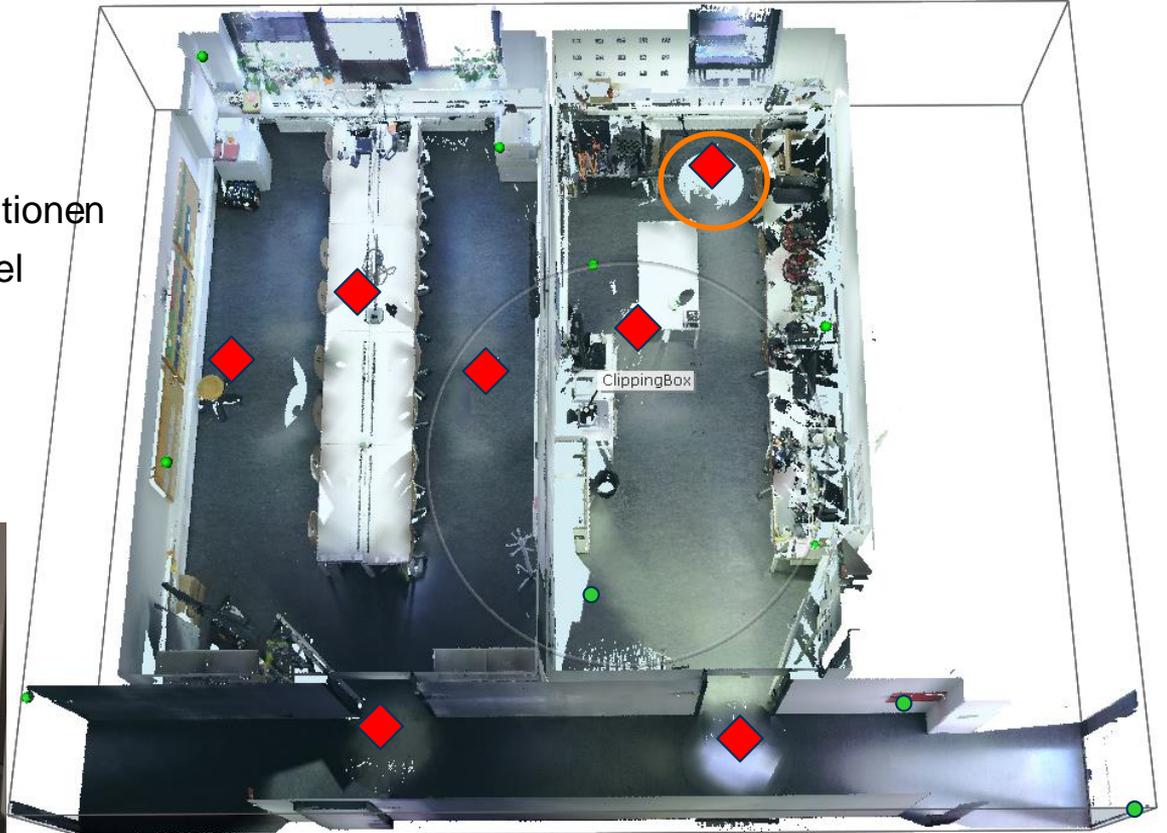
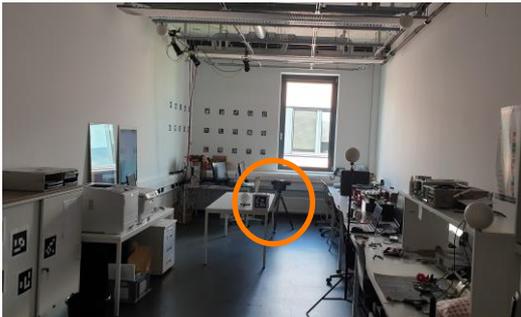
- Informationen über Zielpersonen / Umweltmodell

Umgebungserfassung mit 3D Messsystem FARO Focus

Ziel: Einmessung der Anchorpositionen



- ◆ Messpositionen
- Messkugel



Kontrolle des Testscans



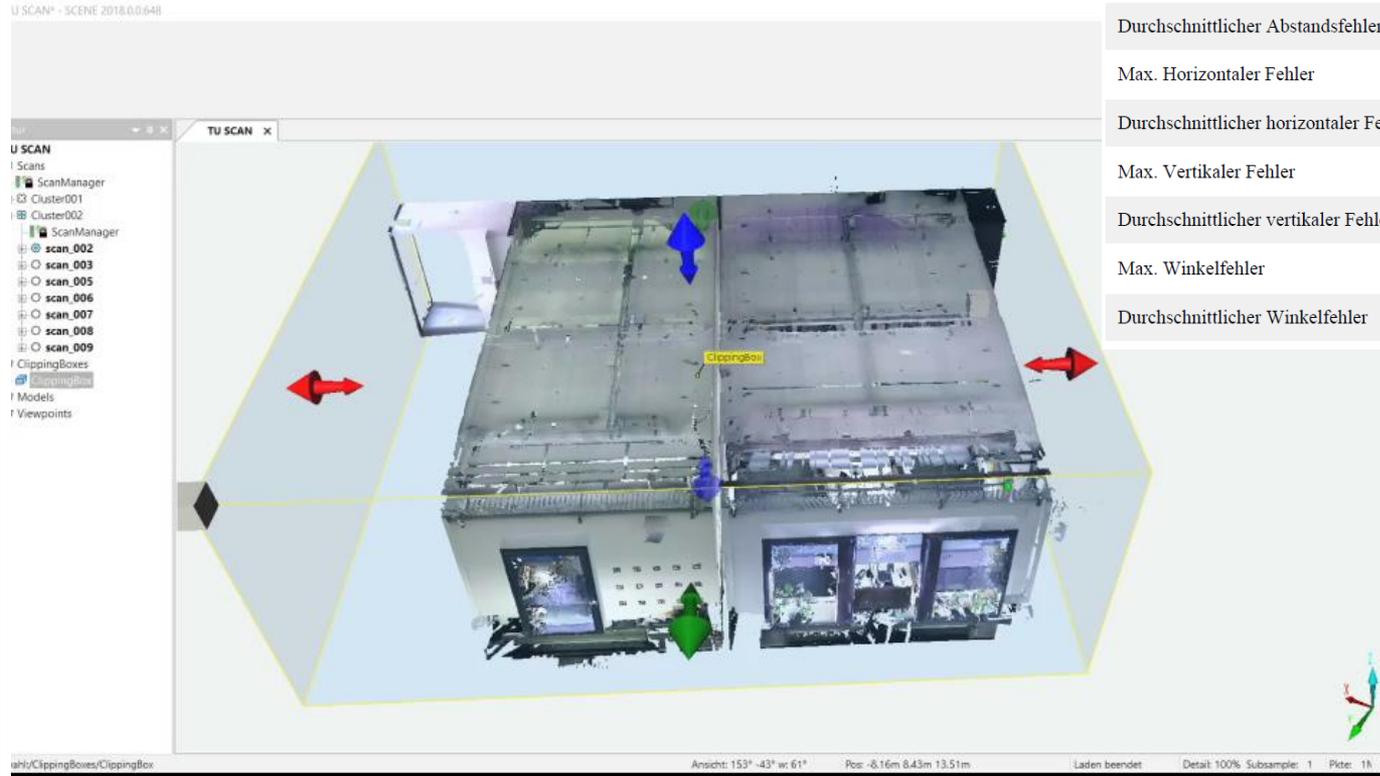
Ansicht: 138° -11° w: 74°

Pos: -6.16m -2.11m 1.72m

Laden beendet

Detail: 100% Subsample: 1 Pkte: 105

Ergebnis 3D Scan /Grundriss erstellen



Zielstatistiken

Max. Abstandsfehler	3.9 mm
Durchschnittlicher Abstandsfehler	2.1 mm
Max. Horizontaler Fehler	3.7 mm
Durchschnittlicher horizontaler Fehler	1.6 mm
Max. Vertikaler Fehler	2.3 mm
Durchschnittlicher vertikaler Fehler	1.0 mm
Max. Winkelfehler	-
Durchschnittlicher Winkelfehler	-

Abstandsfehler	< 20 mm	> 40 mm
Horizontaler Fehler	< 20 mm	> 40 mm
Vertikaler Fehler	< 20 mm	> 40 mm
Winkelfehler	< 0.5 deg	> 1.0 deg

Zeitliche Planung über die Projektlaufzeit

Anforderungsanalyse, Spezifikation und Schnittstellendefinition

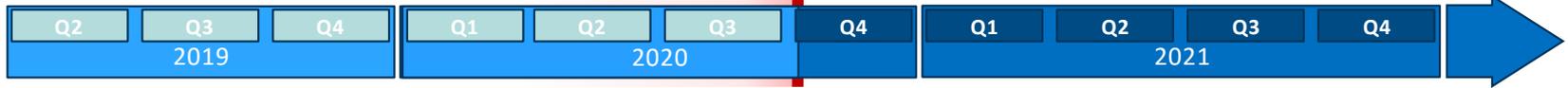
Anbindung an die genutzten Frameworks



Optimierung Hinderniserkennung

Hardwareoptimierung

Zuarbeiten für:
 TUI-NIKR



Veröffentlichungen / Studentische Arbeiten

- **Veröffentlichung:**

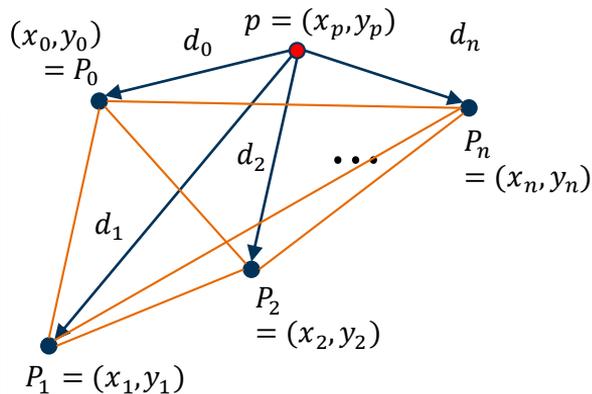
N. Fränzel, N. Greifzu, M. Schneider, A. Wenzel: *Robuste Lokalisierung in drahtlosen Sensornetzwerken*, Tag der Forschung, HSM-Print, 2020

- Simulativer Vergleich bei der Positionsberechnung mittels Least Squares Schätzung und Taylorreihen-Entwicklung
- Unter NLOS Bedingungen ist Least Squares Schätzung wahrscheinlich robuster

- **Masterprojekte:**

- Sasi Kumar Selvaraj: Implementation for RTLS localisation in ROS and rviz (begonnen)
- Stephan Hintz, Adrian Herrman, Nicolas Schmitt: Messfehlerbestimmung im 3D-Raum von UWB Signalen zur Indoor-Navigation mit RTLS (begonnen)

Robuste Lokalisierung in drahtlosen Sensornetzwerken



$$\begin{aligned}
 (x_p - x_0)^2 + (y_p - y_0)^2 &= d_0^2 & (1) \quad (x_p^2 - 2x_p x_0 + x_0^2) + (y_p^2 - 2y_p y_0 + y_0^2) &= d_0^2 \\
 (x_p - x_1)^2 + (y_p - y_1)^2 &= d_1^2 & (2) \quad (x_p^2 - 2x_p x_1 + x_1^2) + (y_p^2 - 2y_p y_1 + y_1^2) &= d_1^2 \\
 &\vdots & (1) - (2) & \\
 (x_p - x_n)^2 + (y_p - y_n)^2 &= d_n^2 & \cancel{(x_p^2 - 2x_p x_0 + x_0^2)} + (\cancel{y_p^2} - 2y_p y_0 + \cancel{y_0^2}) &\longrightarrow \\
 & & - (\cancel{(x_p^2 - 2x_p x_1 + x_1^2)} + (\cancel{y_p^2} - 2y_p y_1 + \cancel{y_1^2})) &= d_0^2 - d_1^2
 \end{aligned}$$

$$2x_p x_1 - 2x_p x_0 + 2y_p y_1 - 2y_p y_0 = d_0^2 - d_1^2 - x_0^2 + y_0^2 - x_1^2 - y_1^2$$

Least Squares Schätzung

$$2 \cdot \underbrace{\begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \\ \vdots & \vdots \\ x_{n-1} - x_0 & y_{n-1} - y_0 \end{pmatrix}}_M \underbrace{\begin{pmatrix} x_p \\ y_p \end{pmatrix}}_p = \underbrace{\begin{pmatrix} d_{p0}^2 - d_{p1}^2 - x_0^2 - y_0^2 + x_1^2 + y_1^2 \\ d_{p0}^2 - d_{p2}^2 - x_0^2 - y_0^2 + x_2^2 + y_2^2 \\ \vdots \\ d_{p0}^2 - d_{pn-1}^2 - x_0^2 - y_0^2 + x_{n-1}^2 + y_{n-1}^2 \end{pmatrix}}_b$$

$$p = \frac{1}{2} (M^T M)^{-1} M^T b$$

Taylorreihen-Entwicklung

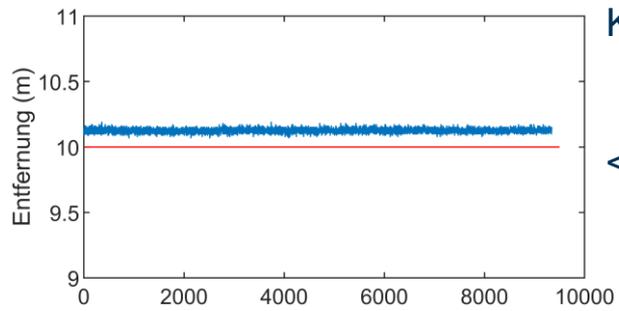
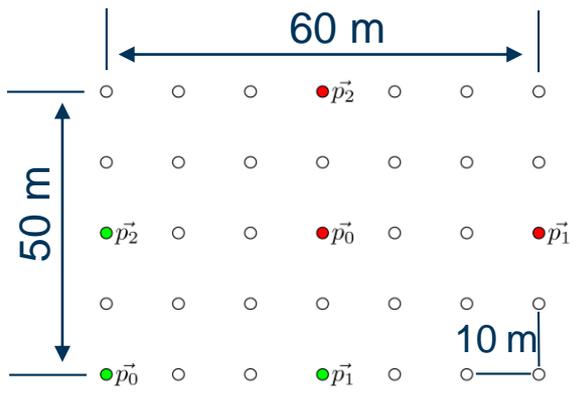
$$a_{i,1} \delta_x + a_{i,2} \delta_y = d_i - r_{v i} \quad ; \quad a_{i,1} = \frac{\partial r_i}{\partial x} = \frac{x_v - x_i}{r_{v i}} \quad ; \quad a_{i,2} = \frac{\partial r_i}{\partial y} = \frac{y_v - y_i}{r_{v i}}$$

$$\underbrace{\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ \vdots & \vdots \\ a_{n,1} & a_{n,2} \end{pmatrix}}_A \underbrace{\begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix}}_\delta = \underbrace{\begin{pmatrix} d_1 - r_{v 1} \\ d_2 - r_{v 2} \\ \vdots \\ d_n - r_{v n} \end{pmatrix}}_d$$

$$\delta = (A^T A)^{-1} A^T d$$

$$p = p_v + \delta$$

Robuste Lokalisierung in drahtlosen Sensornetzwerken



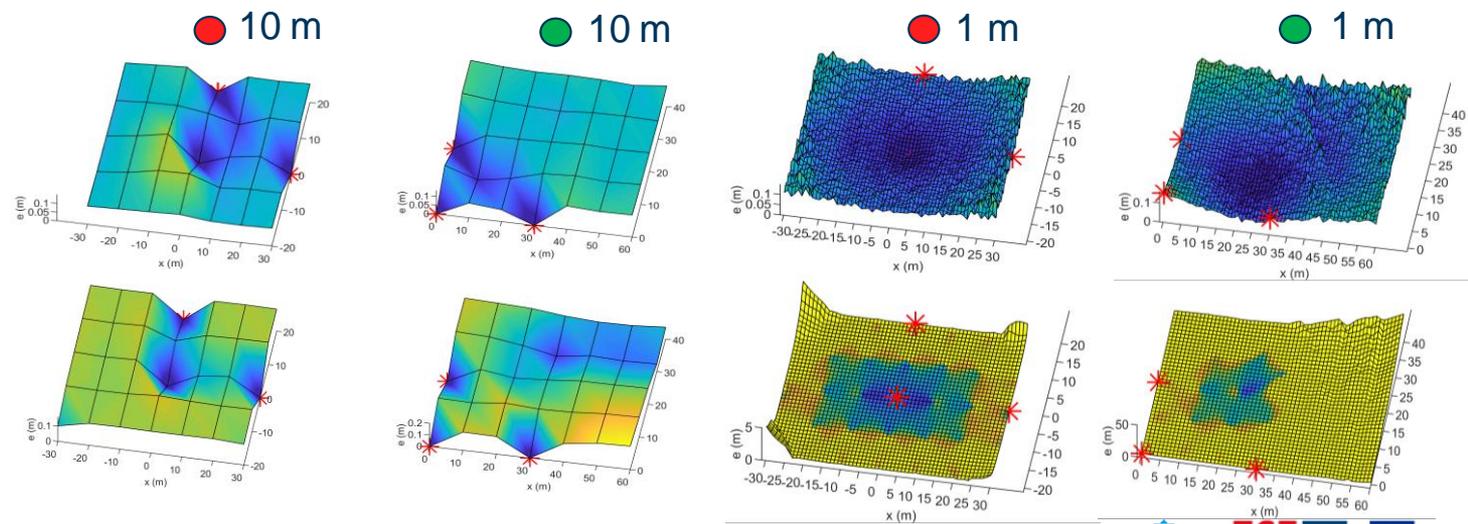
Kanalmodell

$$f(x|a, b) = \frac{1}{b^a \Gamma(a)} x^{a-1} e^{-\frac{x}{b}}$$

< Ergebnis aus 9342 Messungen

Least Squares
Schätzung ✓

Taylorreihen-
Entwicklung



Messpunkte RTLS festlegen / Demomessung im Labor

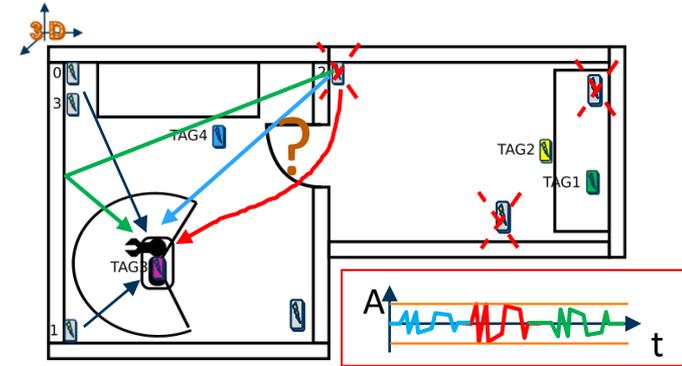
-  RTLS Anker (feste)
 -  Kugel Messung
 -  Kugel Tracker
 -  Lila Punkt Objekt
- (beweglich)



- Positionen für RTLS Anker festlegen
- Messstellen erhalten Scan-Kugeln

Hardwareoptimierung / Milestone erreicht

"al": [Anchorliste als JSON Array mit bis zu 31 Anchorobjekten
{	Anchorobjekt für jeden bekannten Anchor
"id16": "0x4e1a",	UDID des Anchors als JSON String
"pos": {	Positionsjekt des Anchors
"x": -34.16,	
"y": -2.38,	
"z": -1.48	
},	
"dof": 0,	Freiheitsgrad des Anchors, 0 bis 3 für die Basiszelle, 4 alle anderen
"pr": 0.998,	künstliches Hilfsmaß „Positionrate“ für die Güte der Positionsbestimmung des Anchors
"ldis": 22.87,	aktueller Abstand zum Anchor in m
"adis": 22.55,	Mittelwert des Abstands zum Anchor seit Reset in m
"var": 0.0712,	„Varianz“ des Abstands zum Anchor
"sqn": 5,	Sequenznummer, bezieht sich auf die Reihenfolge, in der die Anchor abgefragt wurden
"nom": 433,	Anzahl der Abstandsmessungen seit Positionsreset
"blk": 579591,	Zeitstempel des letzten „life sign“
"fpp": -32.54,	First Path Power (siehe nächstes Kapitel)
"rxp": -30.17,	Received Power (siehe nächstes Kapitel)
"csn": 104.00,	Channel Standard Noise
"cmn": 2652.00,	Channel Maximum Noise
"toc": 0,	Time out counter, wird erhöht, wenn Abstandsmessung fehlgeschlagen ist
"ra": 1.	Messanchor, 1 bedeutet, Daten sind valide und werden zur Positionsbestimmung verwendet



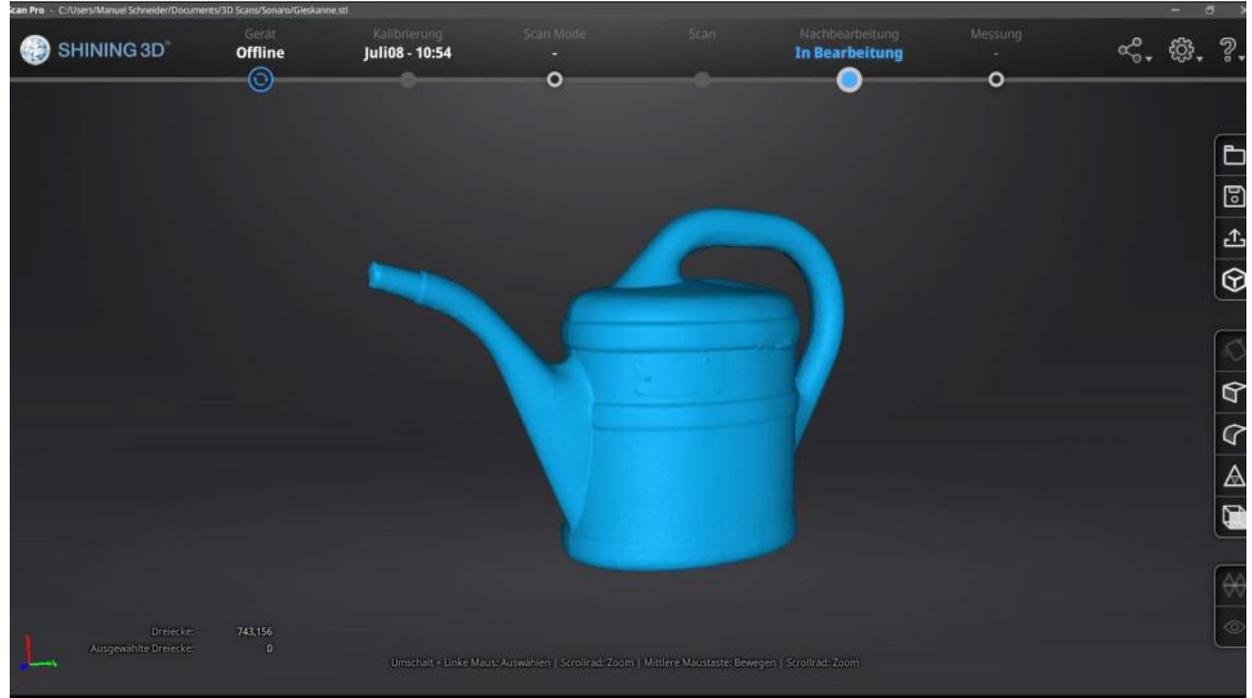
$$fpp = 10 \log_{10} \left(\frac{F_1^2 \cdot F_2^2 \cdot F_3^2}{N^2} \right) - A$$

$$rxp = 10 \log_{10} \left(\frac{C \cdot 2^{17}}{N^2} \right) - 121.74 \text{ dBm}$$

Milestones	Monat	Beschreibung
MS2	07/2020	Mobile RTLS-Flares sind funktionsfähig und in Demonstrator integriert

Weitere Tätigkeiten

- Erstellung 3D Datenbank
- 3D Handscanner EinScan Pro
- ca. 20 Gegenstände eingescannt



Weitere zeitliche Planung

Anforderungsanalyse, Spezifikation und Schnittstellendefinition

Anbindung an die genutzten Frameworks



Corona

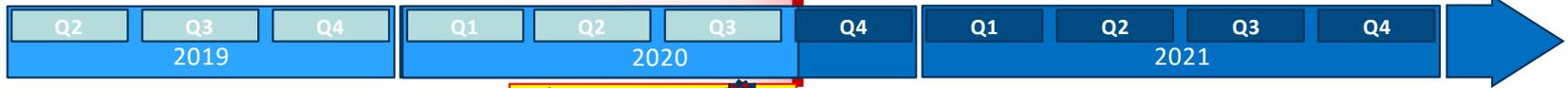
Hardwareoptimierung

Messszenarien mit Hinderniserkennung
Datenaufzeichnung, Entwicklung von Algorithmen zur Optimierung der Positionsschätzung

Optimierung Hinderniserkennung

Zuarbeiten für:
TUI-NIKR

Weitere Datenaufzeichnungen und Untersuchung der implementierten Funktionen zur Signalqualitätabschätzung



Corona

Forscherguppe SONARO

Smarte Objektübernahme und –übergabe für die
nutzerzentrierte mobile Assistenzrobotik

Beiratstreffen am 28.10.2020

Grobplanung für die restliche Projektlaufzeit /

Diskussion mit dem Beirat

www.sonaro-projekt.de



Präzisierung der Grobplanung/Zielfunktionalitäten für die kommenden zwei Jahre

Milestones	Monat	Beschreibung
MS1 	01/2020	Roboter betriebsbereit, kann Personen wahrnehmen und autonom navigieren
MS2 	07/2020	Mobile RTLS-Flares sind funktionsfähig und in Demonstrator integriert
MS3 	07/2020	Multispektraler 3D-Sensor ist betriebsbereit und auf Roboter integriert
MS4	01/2021	Präsentierte Objekte können von Hand getrennt, klassifiziert und im Raum getrackt werden
MS5	10/2021	Roboter kann im vollständigen Ablauf Gegenstände von Person A greifen und bei B wieder übergeben
MS6	12/2021	Die Leistungsfähigkeit des Systems ist evaluiert

Diskussion

- Wie verhält es sich mit der Reaktionsfähigkeit des Zugriffsplanes bei dynamischen Ortsveränderungen der zu greifenden Werkzeuge?
→ eigens entwickelter Planer kann im 4Hz Takt die Umgebungssituation neu bewerten und den Plan aktualisieren, Implementierung bislang nur auf CPU bietet noch potential für Optimierungen in kommerziellen Anwendungen
- Welche Hardwareanforderungen gibt es zur Realisierung der unterschiedlichen Abläufe in den Teilsystemen? Wo und wie wird gerechnet?
→ Ziel ist es, Alles onboard zu rechnen. Im Roboter ist derzeit ein standard quadcore PC sowie eine Nvidia Jetson GPU integriert und die demonstrierten Algorithmen laufen auf diesen Rechnern.
- Wie ist die finale Integration der Teilprojekte in die einheitliche Hardware vorgesehen?
→ Die Algorithmen müssen effizient genug laufen, damit alles auf den vorhandenen Ressourcen läuft.
- Was wird bereits zum Thema Maschinensicherheit getan? Wie sind die Zertifizierungspläne für zukünftige Systeme?
→ Sicherheitsaspekte werden lediglich durch die Entwicklung spezieller Sensorik für die Erfassung und Unterscheidung menschlicher Hände adressiert, allerdings noch nicht bzgl. Zertifizierbarkeit. Wir machen Grundlagenforschung.
- Planung des nächsten Workshops: Woche vom 12. April 2021