Experimental Setup for Investigation and Evaluation of a Mapping and Localization System

Versuchsaufbau zur Untersuchung und Evaluierung eines Kartierungs- und Lokalisierungssystem

Stefan Hensel*, Marin B. Marinov**, Max Schmitt*

* University of Applied Sciences Offenburg, Department for Electrical Engineering,
Badstraße 24, D-77652 Offenburg, Germany, stefan.hensel@hs-offenburg.de
** Technical University of Sofia, Faculty of Electronic Engineering and Technologies, Department of Electronics,
8, Kliment Ohridski Blvd., BG-1756 Sofia, Bulgaria, mbm@tu-sofia.bg

***Abstract* —** Through implementation and subsequent evaluation, the research-oriented visual-inertial mapping- and localization-framework maplab is analyzed. Mapping and localization are based on detecting different features in the environment. Next to the possibility to create single-case maps, the included algorithms allow merging maps to increase mapping accuracy and obtain large-scale maps. Furthermore, the algorithms can be used to optimize the collected data. The preliminary results show that maplab can be efficiently used for mapping, especially in rooms and small building environments. The possibility of merging maps can be used to increase the overall information of these maps, consequentially improving accuracy at a subsequent localization. However, large-scale mapping results in increased geometrical inconsistency.

***Zusammenfassung* —** Mit der Implementierung sowie einer anschließenden aussagekräftigen Evaluierung, soll das, visuelle-inertiale Kartierungs- und Lokalisierungssystem *maplab* analysiert werden. Hierbei basiert die Kartierung bzw. Lokalisierung auf der Detektion von Umgebungsmerkmalen. Neben der Möglichkeit der Kartenerstellung besteht ferner die Option, mehrere Karten zu fusionieren und somit weitreichende Gebiete zu kartieren sowie für weitere Datenauswertungen zu nutzen. Aufgrund der Durchführung und Bewertung der Ergebnisse in unterschiedlichen Anwendungsszenarien zeigt sich, dass maplab besonders zur Kartierung von Räumen bzw. kleinen Gebäudekomplexen geeignet ist. Die Möglichkeit der Kartenfusionierung bietet weiterhin die Option, den Informationsgehalt von Karten zu erhöhen, welches die Effektivität für eine anschließende Lokalisierung steigert. Bei wachsender Kartierungsgröße hingegen zeigt sich jedoch eine Vergrößerung geometrischer Inkonsistenzen.

I. Einführung und Motivation

Durch die Entwicklungen im Bereich roboterassistierter Systeme sowie der Steigerung der Leistungsfähigkeit von Algorithmen im Softwarebereich ist es möglich, unbemannte Fahrzeuge nahezu überall zu navigieren. Aufgrund des Entwicklungsprozesses ergibt sich ein breites Einsatzspektrum an möglichen, autarken Systemen, welche die Fähigkeit besitzen können, selbstständig zu handeln. Folglich entsteht die Möglichkeit, in Bereiche vorzudringen, die für einen Menschen nicht zu erreichen sind. Für die Navigation in unbekannten Gebieten ist es essentiell, ein genaues visuelles Abbild der umliegenden Umgebung zu besitzen. Hierfür sind Bildverarbeitungs- und Interpretationsalgorithmen notwendig, die aus einer realen Welt ein digitales Abbild erzeugen. Dies führt zu einer späteren Anwendung bereits bekannter Umgebungsinformationen. Durch die gegenwärtigen Entwicklungen im Bereich der Robotik ist es möglich, einfachste Systeme zur Kartierung und Lokalisierung zu entwerfen.

II. Materialien und Methoden

A. Das Framework maplab

Mithilfe der Analysen und Bewertungen dieser Arbeit soll eine Aussage als Nützlichkeit eines visuelles-inertiales Kartierungs- und Lokalisierungssystem erzeugt werden. Das Framework *maplab* wird von der ETH Zürich als Grundlage für weitere Forschungsprojekte im Bereich der Navigation von mobilen Roboterplattformen über die Software-Entwicklungsplattform GitHub zur Verfügung gestellt. Für die Evaluierung stehen eine Monochrom-Kamera sowie eine inertiale Messeinheit (engl. inertial measurement unit, IMU) bereit, für welche die notwendigen Softwarepakete vorhanden sind. Mithilfe der Konzipierung eines Systems mit diesen beiden Komponenten soll durch aussagekräftige Analysen sowie der Bewertungen der Ergebnisse eine Beurteilung als sog. SLAM-System (Simultaneous Localization and Mapping) erhalten werden. SLAM beschäftigt sich mit der Problematik der gleichzeitigen Kartierung und Lokalisierung in einer unbekannten Umgebung.

Zur Erprobung der Einsatzmöglichkeiten von *maplab* steht eine mobile Roboterplattform zur Verfügung, die zusätzlich mit einem Referenzsystem zur Kartierung ausgestattet ist. Diese Referenz soll überdies eine Erweiterung der Untersuchungsmöglichkeiten bieten.

Das Framework *maplab* ist ein visuelles-inertiales SLAM System. Es enthält eine Vielzahl an Werkzeugen, welche für die Lösung der SLAM-Problematik genutzt werden können. Diese umfassen [1]:

- Erstellung und Lokalisierung in Karten,
- Multi-Session Kartenkombinierung,
- Loop closing,
- Tiefenrekonstruktion,
- Visualisierung von Karten.

Das System *maplab* nutzt das Framework ROVIO. Die Erweiterung mit einem Lokalisierungsmodul führt zu ROVIOLI. ROVIO ist sozusagen der Grundbaustein für den Prozess der Detektion und Verfolgung von

Umgebungsmerkmalen. Um maplab zu implementieren und evaluieren, muss eine möglichst präzise Funktionsweise von ROVIO bzw. ROVIOLI gewährleistet sein.

Die Datenstruktur von maplab ist eine sog. VI-Map, welche die visuellen-inertialen Daten für die Kartierung beinhaltet. Jede dieser Karten kann mehrere *missions* enthalten, wobei jede aus einer Einzelaufnahme der Umgebung besteht. Die *missions* bestehen aus Knoten und Verbindungen, welche abstrakt in einem sog. Graph repräsentiert werden. Diese Darstellung beinhaltet eine Sammlung von Objekten, welche miteinander verbunden sein können, dargestellt als G¹V; Eº mit V als Knoten und E als Verbindungen. V werden als *vertices* oder *nodes* und E als *edges* bezeichnet [2].

Diese Knoten beinhalten eine Zustandsschätzung zu einem bestimmten Zeitpunkt. In diesem Zustand sind die Lage des Kamera-IMU-Systems, der Messfehler der inertialen Messeinheit sowie die Geschwindigkeit dieser für die inertialen Informationen enthalten. Ebenfalls gespeichert werden in diesem Zustand die

Bilder der Kamera für die visuellen Daten. Zu diesen Bilddaten gehören unter anderem die sog. BRISK oder FREAK Deskriptoren [1]. Deskriptoren besitzen die Aufgabe, die Region um den von einem Detektor als Kante ermittelten Punkt so zu beschreiben, dass dieser so eindeutig wie möglich dargestellt wird. Dies soll dazu führen, dass die Merkmale auch aus verschiedenen Blickwinkeln bzw. Aufnahmen als identischen Punkt erkannt werden [2].

Die Verbindung dieser Knoten findet durch das IMU *edge* statt. Dieses beinhaltet die Messungen der inertialen Einheit zwischen den *vertices*. Schließlich werden Umgebungsmerkmale detektiert, wenn mehrere *vertices* mit diesem verbunden sind. Für die spätere Möglichkeit einer *Loop Closure Detection* wird jenes Merkmal in demjenigen Knoten gespeichert, welcher zuerst eine erfolgreiche Detektierung durchgeführt hat [1].

Abb. 1 zeigt neben dem eben erläuterten Graphen $G(V, E)$ die verschieden aufgespannten Koordinatensysteme. F_G stellt das globale, durch die Gravitation festgelegtes Ursprungskoordinatensystem dar. Jeder Ursprung einer neuen Karte bzw. Session k liegt in F_{M_k}. F_{I_k} legt hierbei die Ausrichtung der IMU zum Zeitpunkt k fest.

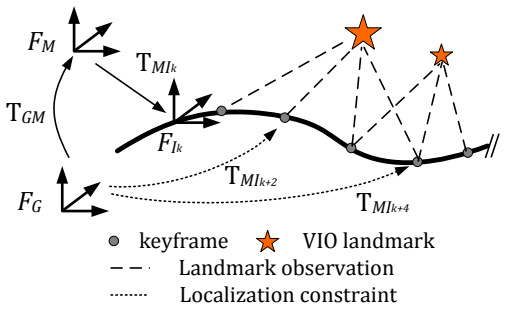

Abb. 1. Darstellung der Vorgehensweise zur Extrahierung und Verfolgung der Umgebungsmerkmale sowie die genutzten Koordinaten-bezugssysteme in *maplab* [1]

Um auf das Framework selbst sowie Karten zugreifen zu können, wird eine gesonderte Konsole bereitgestellt. Diese erlaubt es, den Benutzer diverse vorhandene Optimierungsalgorithmen auf Umgebungskartierungen anzuwenden und eine Steigerung der Kartierung bzw. Lokalisierung zu erreichen.

B. Vorgehensweise

Zur Durchführung einer qualitativ hochwertigen und aussagekräftigen Evaluation ist es notwendig, die Funktionsweise von *maplab* zu gewährleisten. Um im Vorfeld bereits mögliche Fehlerursachen deuten zu können, soll mit einem ersten Einblick in die dazugehörende Theorie, Wissen über SLAM sowie den einhergehenden mathematischen Hintergründen erlangt werden. Gleichzeitig müssen zunächst die notwendigen Softwarekomponenten für den Betrieb der Kamera, der inertialen Einheit sowie das Framework *maplab* implementiert werden.

C. Simultaneous Localization and Mapping (SLAM)

Durch die stetige Entwicklung und Verbesserung der Technologie mit einhergehender Leistungssteigerung elektronischer Komponenten ist es möglich, intelligente und autonome Systeme zu konzipieren. Für die Nutzung eben dieser Systeme ist es notwendig, dass diese ein genaues Abbild der Umgebung um sie herum besitzen, sodass weder Mensch, Tier, die Umgebung noch das System selbst schaden nehmen. Für den Erhalt eines Modells bzw. einer Karte der Umgebung ist es unumgänglich, eine der essentiellen Problematiken der Robotik zu lösen.

Die SLAM-Problematik behandelt die Schwierigkeit der Lokalisierung und Kartierung eines mobilen Roboters in einer unbekannten Umgebung mit der gleichzeitigen Positionierung dessen, relativ zu dieser Karte [2]. Vor allem wenn keine weiteren Navigationsmöglichkeiten wie GNSS zur Verfügung stehen, gewinnt die SLAM-Problematik an Bedeutung. Während es bereits zur Lösung der Problematik in einfachen Anwendungen kommt, können SLAM-Algorithmen durch herausfordernde dynamische Roboterbewegungen oder stark dynamische Umgebungen an ihre Grenzen gebracht werden [3]. Für den Erhalt einer Karte müssen mittels Sensoren die Struktur der Umgebung erkannt werden. Hierfür steht eine Vielzahl an möglichen Sensortypen zur Verfügung.

Mithilfe der Positionsbestimmung der Umgebungsmerkmale ist es schließlich möglich, eine Darstellung der Roboterumgebung zu erhalten und somit eine Karte zu gewinnen, welche auf verschiedene Arten wie beispielsweise der Lokalisierung verwendet werden kann [4]. Das grundlegende Problem innerhalb des SLAM ist nun sowohl die Bahn des Roboters sowie die Position aller Umgebungsmerkmale zu schätzen, ohne hierfür bereits Erkenntnisse über die wahre Position der Merkmale bzw. des Roboters selbst zu besitzen [5].

D. Robot Operating System (ROS)

Um die Kommunikation und den Datenaustausch zwischen der Kamera bzw. IMU und dem Computer zu gewährleisten, steht ein Robot Operating System, kurz ROS, zur Verfügung. Aufgrund der weiten Verbreitung und dem häufigen Einsatz in Roboterapplikationen stehen eine große Sammlung an Software-Tools und Bibliotheken bereit [6]. Diese erlauben es dem Benutzer notwendige Treiber für elektronische Sensoren einfach und unkompliziert zu installieren.

E. Robust Visual Odometry (ROVIO)

Für die präzise Navigation und Kontrolle von autonomen Robotersystemen wird eine hohe Bandbreite an Informationen von Position und Orientierung benötigt. Mithilfe der Fusionierung von Informationen aus optischen und inertialen Sensoren wird eine bestmögliche Schätzung der dynamischen Bewegungen des Systems erreicht [7]. Ebenso wird durch die Erweiterung der Kameradaten mit den Informationen aus einer inertialen Messeinheit eine erhebliche Steigerung der Robustheit sowie der Genauigkeit erzielt [8].

In dem Framework ROVIO wird die Struktur an „the standard visual-inertial EKF-SLAM formulation“ [1] angelehnt. Für die Erkennung der Umgebungseigenschaften ist

ein Filter notwendig, welcher die inertialen Messungen für die Vergrößerung dessen Zustands verwendet, während gleichzeitig die visuellen Informationen innerhalb der Filteraktualisierung berücksichtigt werden. Dieses Update ist jedoch nur dann möglich, wenn neue Umgebungsaufnahmen durch die Kamera bereitgestellt werden [7]. Das Detektieren von neuen Umgebungseigenschaften basieren auf dem sog. FAST-Kantendetektor, welcher eine hohe Anzahl an Positionen von möglichen Umgebungsmerkmalen liefert.

F. *ROVIOLI*

ROVIOLI ist die Erweiterung des Grundbausteins ROVIO für die Kartierungs- und Lokalisierungsfähigkeit von maplab. ROVIOLI kann in zwei grundsätzlichen Modi betrieben werden:

- VIO (Visual Inertial Odometry) Modus Erstellung einer Karte basierend auf den VIO Schätzungen.
- LOC (Localization) Modus Lokalisierung in einer bereits bekannten Karte mit bestmöglicher Reduzierung des Drifts relativ zu dieser Karte [1].

Für die Gewährleistung der Positionsfindung im LOC-Modus findet ein Abgleich zwischen den Deskriptoren der zweidimensionalen Bilder der aktuellen Aufnahme und der bekannten dreidimensionalen Karte (2D-3D) statt [9].

Diese resultieren in T_{GI_k}, die eine Transformation der aktuellen Lage zum Zeitpunkt k relativ zum Ursprungs-koordinatensystem F_G darstellt. Diese Übereinstimmungen werden durch das *Frame Localization* Modul durchgeführt. Anschließend werden diese Schätzungen mit den Odometriedaten fusioniert, um die Transformation T_{GM} zu erhalten, welche F_M auf F_G ausrichtet. F_M repräsentiert dabei den Ursprung einer Mission k (siehe Abb. 1). Die Ausgänge aller Module werden an den Baustein *Map Builder* geleitet, in welchem eine Synchronisation aller Module stattfindet. Gespeichert werden alle Daten in einer sog. VI-Map. In Abb. 2 wird der eben erwähnte Prozess genauer verdeutlicht.

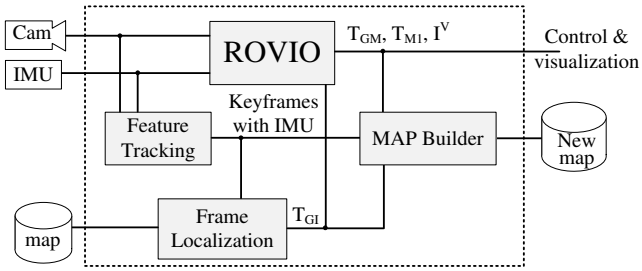

Abb. 2. Module und Datenfluss ROVIOLI [1]

III. Implementierung und Evaluierung des Systems

Für die Nutzung von *maplab* ist es notwendig, die einwandfreie Funktionsweise von ROVIOLI zu garantieren. Dies wird durch zwei grundlegende Tatsachen sichergestellt. Zum einen ist eine präzise Kalibrierung des gesamten Kamera-IMU-Systems notwendig, um eine genaue Aussage über die Positionierung der IMU gegenüber dem Kamerazentrum treffen zu können. Zum anderen ist eine Synchronisation zwischen den beiden elektronischen Komponenten Voraussetzung für die Merkmalsdetektion.

A. *Hardwaretechnische Synchronisation zwischen Kamera und IMU*

Mithilfe der verwendeten IMU der Firma *xsens* besteht die Möglichkeit der hardwaretechnischen Triggerung externer elektrischer Komponenten über ein sog. *SyncOut*-Signal. Durch dieses Signal ist es möglich, die Kamera in dem zur Verfügung stehenden Trigger-Modus zu betreiben. Anhand dieser Möglichkeit der Triggerung der Kamera durch eine externe Komponente wird gewährleistet, dass immer zu einem festgelegten Zeitpunkt ein Bild aufgenommen wird. Wie dem Datenblatt der Kamera zu entnehmen ist, muss diese aufgrund ihrer technischen Beschränkungen auf eine fallende Flanke getriggert werden.

In Abb. 3 ist der endgültige, interne Aufbau der Hardware, bestehend aus den drei Hauptkomponenten Adapterplatine, Kamera und IMU, präsentiert.

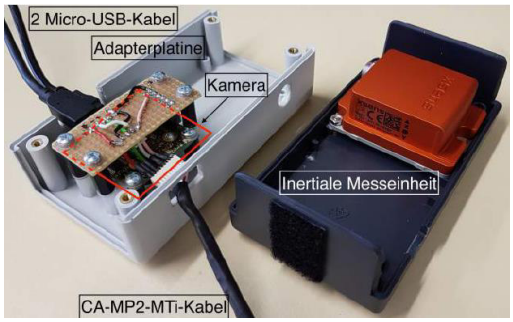

Abb. 3. Aufbau der Hardware des visuellen-inertialen Systems

B. *Softwaretechnische Synchronisation zwischen Kamera und IMU*

Bei Systemtests wurde deutlich, dass die alleinige hardwaretechnische Synchronisation von Kamera zur IMU nicht ausreicht, um ein stabiles System im Hinblick auf Vermeidung von Divergenzen zu garantieren.

Damit ROVIOLI Umgebungsmerkmale korrekt detektieren und verfolgen kann, ist es wichtig, dass innerhalb einer bestimmten Zeit alle notwendigen Daten für den unterlagerten Kalman-Filter vorhanden sind. Durch die ausgewählten Datenraten der elektronischen Komponenten beträgt dieses „Zeitfenster“ $t = 50\,ms$. Vor allem in Umgebungen mit einer geringen Beleuchtung ist die beschriebene Problematik der Asynchronität vorzufinden. Der Ansatz zum Lösen des Problems der automatischen Beleuchtungszeit ist auch hier erneut die Nutzung des Triggersignals der IMU. Dazu wird innerhalb des Treibers der IMU explizit der Zeitpunkt des $SyncOut$-Signals über den ROS Treiber der IMU bekannt gegeben. Dies ermöglicht es, den jeweiligen Kamerazeitstempel zu überschreiben und somit eine Unabhängigkeit von automatischen Belichtungszeiten zu gewährleisten.

Mithilfe dieser Erweiterungen des Kamera-Treibers ist es möglich, die Kamera mit einer automatischen Belichtungszeit zu betreiben und eine Umgebung trotz möglicher starker Belichtungswechsel zu kartieren bzw. sich innerhalb dieser zu positionieren. In Abb. 4 wird der endgültige schematische Aufbau des visuellen-inertialen Systems nach der hard- und softwaretechnischen Synchronisation dargelegt.

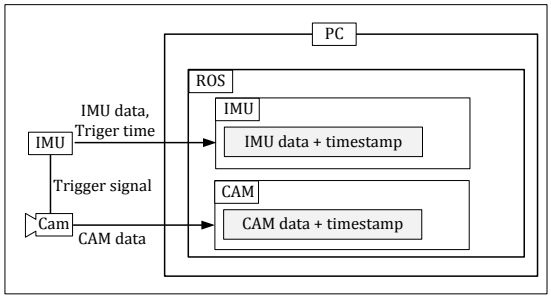

Abb. 4. Aufbau der Hardware des visuellen-inertialen Systems

C. *Evaluierung - Analyse und Bewertung der Distanzmessung mithilfe von maplab*

Mit dem Framework *maplab* wird die Möglichkeit geboten, die zurückgelegte Strecke nach der Durchführung einer Kartierung auszulesen. Um eine Bewertung der Genauigkeit der Längenmessung für folgende Untersuchungen berücksichtigen zu können, soll die Größe der Messabweichung bestimmt werden. Hierfür wird eine Teststrecke von 15m festgelegt, welche zur Verdeutlichung und Vergrößerung der Auswirkungen des Messfehlers zehnmal abgefahren bzw. abgelaufen wird. Dabei sollen zwei Messungen angestellt werden:

1. Messung der Strecke mit dem visuellen-inertialen System frei beweglich in der Hand
2. Messung der Strecke mit dem visuellen-inertialen System fest montiert auf einer mobilen Roboterplattform

Laut der GitHub-Webseite von ROVIOLI [10] ist eine mögliche Fehlerursache für die Entstehung von Divergenzen oder große Drift- bzw. Skalierfehler (bei der Extrahierung der Informationen aus einer Umgebungsaufnahme) die feste Fixierung des visuellen-inertialen Systems auf einer mobilen Roboterplattform. Die Ursache hierfür liegt in der fehlenden Information bezüglich der z-Achse der IMU, da diese aufgrund der festen Montage einen relativ konstanten Wert besitzt. ROVIOLI profitiert vor allem durch die ungleichmäßige, jedoch durchgängige Bewegung des gesamten Kamera-IMU-Systems. Der Effekt der fehlenden Achseninformation der IMU soll mithilfe einer konstanten Geschwindigkeit dieses Roboters verstärkt werden [10]. Da sich jedoch in [1] das Framework *maplab* als visuellen-inertiales SLAM-System versteht, soll im Folgenden die Eignung dieses Frameworks für mobile Roboterplattformen untersucht werden. Als Roboter steht hierfür der sog. „Husky“ [11] bereit, welcher mit einem externen, drahtlosen Controller gesteuert werden kann.

Mit den oben aufgelisteten Messungen sollen die Auswirkungen der festen Montage des visuellen-inertialen Systems untersucht werden. Um die äußeren Fehlereinflüsse zu minimieren, werden Hin- und Rückweg der $15\,m$ langen Teststrecke mit einer gleichbleibenden Orientierung des Kamera-IMU-Systems durchgeführt.

Mithilfe dieser Translationsbewegung sollen Fehlereinflüsse, entstehend durch eine Drehbewegung, minimiert sowie mit einer möglichst geradlinigen Bewegung reduziert werden. Dabei definiert d_{unopt} die Distanz vor der Durchführung des Optimierungsprozesses. Die Auswirkungen dieses Prozesses wird mit der neu errechneten Distanz d_{opt} dargestellt. Mit der Differenz d_{diff} lassen sich absolute und relative Fehler der Distanzmessung errechnen. Diese sind ein Maß für die Genauigkeit des Messvorgangs. Die eben erwähnten Größen errechnen sich durch:

$$d_{diff} = d_{unopt} - d_{opt}, \tag{1}$$

$$d_{abs} = d_{diff}, m - 150\,m, \tag{2}$$

$$f_{rel} = \frac{d_{abs}, m}{150\,m} \cdot 100\%. \tag{3}$$

Es ergeben sich nach einer erfolgreichen Kartierung folgende Messergebnisse:

TABELLE I. MESSERGEBNISSE DER STRECKENMESSUNG

Eigenschaft	d_{unopt}, m	d_{opt}, m	d_{diff}, m	d_{abs}, m	$f_{rel}, \%$
Mensch	162,84	152,70	10,14	2,70	1,8
Husky	203,90	148,86	55,04	-1,14	-0,76

Betrachtet man die Messwerte in TABELLE I vor und nach dem Optimierungsprozess, so sind erhebliche Differenzen in der Messung der zurückgelegten Strecken zu erkennen. Hierbei weicht die Distanzmessung durch den Husky um mehr als das fünffache gegenüber der Vermessung durch den Menschen ab. Jedoch reduzieren sich die Differenzen nach dem Optimierungsprozess der Streckenmessung erheblich. Werden nun die relativen Abweichungen analysiert, findet eine genauere Messung der Distanz unter Nutzung der Husky-Roboterplatform statt. Der relative Fehler der gemessenen Streckenlänge zur tatsächlichen Gesamtdistanz beträgt für die Messung auf der Roboterplattform lediglich 0,76 %. Durch die Möglichkeit der präzisen Manövrierung der Roboterplattform ist es annähernd möglich, die Ideallinie der geradlinigen Strecke abzufahren und somit einen minimalen Fehler, verursacht durch Abweichungen von der Ideallinie, zu erzeugen.

IV. ZUSAMMENFASSUNG UND AUSBLICK

Nach der Einarbeitung in das Framework maplab wurden zwei Problematiken ersichtlich, welche für eine erfolgreiche Implementierung und eine anschließende Evaluierung des visuellen-inertialen SLAM-Systems gelöst werden mussten. Für eine präzise Arbeitsweise und eine bestmögliche Schätzung der Position der Umgebungsmerkmale war es somit notwendig, die visuellen und inertialen Daten kontinuierlich und in gleichmäßigen, zeitlichen Abständen bereitzustellen. Aufgrund der gewollten variablen Belichtungszeit der Kamera fand eine Änderung der Übertragungsrate der visuellen Informationen statt, welche sich negativ auf den unterlagerten Extended Kalman-Filters ausgewirkt haben. Resultat dieser Asynchronität zwischen Kamera und IMU war neben der unzureichenden Positionsschätzung, eine Divergenz in der resultierenden Karte.

REFERENCES

[1] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski and R. Siegwart, "maplab: An Open Framework for Research in Visual-inertial Mapping and Localization," *CoRR,* vol. abs/1711.10250, 2017.

[2] P. Corke, *Robotics, Vision and Control,* Berlin: Springer, 2011.

[3] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. Leonard, "Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age," *IEEE Transactions on Robotics,* vol. 32, no. 6, 2016.

[4] R. Valencia and J. Andrade-Cetto, *Mapping, Planning and Exploration with Pose SLAM*, Vols. Springer Tracts in Advanced Robotics, 119, Springer, 2018.

[5] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms," *IEEE Robotics and Automation Magazine,* vol. 2, p. 1–9, 2006.

[6] "About ROS," 2019. [Online]. Available: https://www.ros.org/about-ros/. [Accessed 10 08 2019].

[7] M. Bloesch, S. Omari, M. Hutter and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 28 September–3 October 2015.

[8] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale and R. Siegwart, "A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[9] S. Lynen, T. Sattler, M. Bosse, J. A. Hesch and M. &. S. R. Pollefeys, "Get Out of My Lab: Large-scale, Real-Time Visual-Inertial Localization," in *Robotics: Science and Systems*, 2015.

[10] M. Fehr, "ROVIOLI Introduction.," [Online]. Available: https://github.com/ethz-asl/maplab/wiki/ROVIOLI-Introduction. [Accessed 28 06 2019].

[11] S. Hensel, M. Marinov and C. Kehret, "Design of a mobile platform for the evaluation of Localization and Mapping Algorithms," in *FDIBA Conference Proceedings*, Sofia, 2018.