DISS. ETH NO. 27038

Perception and Learning for Autonomous UAV Missions

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

Timo Hinzmann

MSc ETH in Robotics, Systems and Control

born on September 25, 1988
citizen of Germany

accepted on the recommendation of

Prof. Dr. Roland Siegwart, Examiner
Prof. Dr. Margarita Chli, Co-examiner
Prof. Dr. Shaojie Shen, Co-examiner

2020

To my family

Autonomous Systems Lab
Department of Mechanical and Process Engineering
ETH Zurich
Switzerland

Abstract

The progress in the research and development of Unmanned Aerial Vehicles (UAV) has been tremendous in the last decade, making drones a valuable tool to automate applications that are risky, monotonous, or even unachievable for human-crewed operations: UAVs promise to deliver medical supplies to remote places, count wildlife, or create overview images for damage assessment after natural catastrophes. Nevertheless, a large proportion of the research has concentrated on confined indoor spaces where motion capturing systems may provide nearly perfect state estimation, and a large variety of depth sensors is applicable. Instead, this dissertation focuses on the challenges that arise when the UAV leaves the controlled lab environment and has to cope with a limited payload capacity, noisy measurements, and vast unstructured scenes. Within the scope of this thesis, we combine machine and deep learning with computer vision to develop the core perception abilities that a robot needs for informed, autonomous decision-making.

A fundamental competence that an autonomous UAV requires is the ability to locate itself in large-scale outdoor environments under severe light conditions. Global Navigation Satellite Systems (GNSS) may be used for localization in specific applications but the provided accuracy is limited, and multi-pathing or dropouts may occur next to mountainsides or during operations close to the ground. In our first publication, we propose a navigation system, consisting of a single camera and an Inertial Measurement Unit (IMU), that makes accurate optimization-based state estimation computationally feasible by utilizing a sliding-window estimator. The visual-inertial solution is able to provide smooth pose estimates close to the ground enabling otherwise risky maneuvers such as landings, take-offs, or fly-bys. For larger-scale, geo-referenced localization, pre-existing maps generated from satellite or UAV imagery have immense potential, yet the appearance and environmental changes can be significant. The second publication introduces a framework to generate geo-referenced elevation maps and orthoimages for real-time robotic applications. The closely related third publication combines a rendering engine with a deep learning-based image alignment algorithm that estimates the geo-referenced six degrees of freedom (DoF) camera pose even under substantial environmental and illumination variations.

Furthermore, an autonomous UAV requires a reliable depth estimation to ensure environmental awareness and to detect and avoid unmapped or dynamic obstacles and navigate safely through unexplored, potentially cluttered environments. This capability becomes particularly crucial with the increasing amount of air traffic, induced by the surge of UAVs. However, for the vast majority of small-scale UAVs, available depth sensors are not deployable due to tight constraints on the payload,

dimensions, price, power consumption, and stringent requirements on range and resolution. Motivated by the caveats of depth-from-motion principles with only a single camera, we design a novel multi-IMU multi-camera system for long-range depth estimation, particularly suited for fixed-wing UAVs. The non-rigid multi-view stereo baseline is estimated using inertial measurements, visual cues, and is enhanced with deep learning.

The final part of this dissertation investigates autonomous landing site detection, deep learning-based human detection, and collaborative reconstruction, covering further essential perception capabilities, including scene understanding, object detection, and point cloud registration. Overall, this dissertation provides an extensive perception framework for UAVs, investigating crucial aspects for autonomous mission completion. The proposed algorithms are validated with realistic synthetic datasets, hardware-in-the-loop tests, or real-world experiments. Within the scope of the thesis, more than six different rotary-wing and fixed-wing platforms have been equipped with sensor systems and used in real-world missions. To accelerate the research in this field, source code of several algorithms and valuable datasets with different sensor modalities have been made publicly available to the community.

Keywords: Unmanned Aerial Vehicles; 2D Computer Vision; Machine Learning; Deep Learning; Image Alignment; Image Segmentation; Image Classification; 3D Computer Vision; 3D Reconstruction; Object Detection; Object Tracking; Object Localization; Multi-Camera Multi-IMU Systems; Visual-Inertial Navigation; Localization and Mapping; Reference View Rendering; Long-Range Depth Estimation.

Zusammenfassung

Im letzten Jahrzehnt konnten enorme Fortschritte in der Forschung und Entwicklung unbemannter Luftfahrzeuge (Unmanned Aerial Vehicles, UAVs) erzielt werden. Diese Fortschritte haben Drohnen zu einem unermesslichen Werkzeug für die Automatisierung von jenen Anwendungen gemacht, die für bemannte Einsätze zu risikoreich, monoton oder schlicht nicht realisierbar wären: Unbemannte Luftfahrzeuge sind in der Lage medizinische Hilfsgüter an entlegene Orte zu transportieren, den Wildbestand zu zählen oder Übersichtsbilder zur Schadensbeurteilung nach Naturkatastrophen zu erstellen. Gleichwohl hat sich ein grosser Teil der Forschung mit Experimenten in abgeschirmten Innenbereichen befasst, in denen beispielsweise sogenannte Motion-Capturing-Systeme eine nahezu perfekte Zustandsschätzung liefern können und gleichzeitig eine grosse Anzahl von Sensoren zur Tiefenschätzung Anwendung finden. Im Gegensatz dazu widmet sich die vorliegende Dissertation den Herausforderungen, die sich ergeben, wenn die Drohne die kontrollierte Laborumgebung verlässt und sich mit einer begrenzten Nutzlastkapazität, verrauschten Sensordaten und unstrukturierten Umgebungen konfrontiert sieht. Im Rahmen dieser Arbeit werden Elemente des maschinellen und tiefen Lernens (Machine, Deep Learning) mit computerbasierten Sehen (Computer Vision) kombiniert, um Fähigkeiten im Bereich der Umgebungswahrnehmung zu entwickeln, die der Roboter für eine informationsbasierte und autonome Entscheidungsfindung benötigt.

Eine autonome Drohne muss in der Lage sein sich selbst unter schwierigen Lichtverhältnissen in weiträumigen Aussengeländen zu orten. In bestimmte Anwendungen können hierfür Globale Navigationssatellitensysteme (GNSS) eingesetzt werden. Allerdings ist die Messgenauigkeit begrenzt und Mehrwegempfang (Multi-Pathing) oder komplette Messausfälle können in der Nähe von Berghängen oder bei bodennahen Einsätzen auftreten. In der ersten Publikation entwickeln wir ein Navigationssystem, das aus einer Kamera und einer inertialen Messeinheit (Inertial Measurement Unit, IMU) besteht und eine genaue, auf Optimierung basierende Zustandsschätzung durch die Verwendung eines Schiebefenster-Mechanismus (Sliding Window) rechnerisch möglich macht. Diese Kamera-IMU Lösung ist in der Lage genaue Schätzungen der Position und Orientierung in Bodennähe zu liefern und damit Flugmanöver wie Starts, Landungen und Tiefflüge zu ermöglichen. Für eine georeferenzierte Lokalisierung bietet bestehendes Kartenmaterial, welches aus Satelliten- oder UAV-Bildern erzeugt wurde, ein beträchtliches Potenzial. Allerdings müssen bei dieser Lokalisierungsart mögliche Licht- und Umgebungsveränderungen berücksichtigt werden. Die zweite Publikation stellt Verfahren vor, um georeferenzierte Höhenkarten und Orthobilder für Anwendungen im Robotikbereich in Echtzeit zu erstellen. Die damit verknüpfte dritte Publikation stellt einen lernbasierten Algo-

rithmus vor, der die Kamerabilder der Drohne mit denen eines Modells vergleicht, und daraus die georeferenzierte Kamerapose mit sechs Freiheitsgraden selbst unter erheblichen Umgebungsveränderungen schätzen kann.

Des Weiteren benötigt eine autonome Drohne eine verlässliche Tiefenschätzung zur Umgebungswahrnehmung um beispielsweise nicht kartierte oder dynamische Hindernisse zu detektieren und diesen auszuweichen und um ausserdem sicher durch unerkundete und potenziell unübersichtliche Umgebungen zu navigieren. Diese Fähigkeit wird mit der stetigen Zunahme von UAVs im Luftverkehr besonders wichtig. Für die überwiegende Mehrheit der kleinen unbemannten Drohnen sind die verfügbaren Tiefensensoren jedoch nicht einsetzbar, da die Nutzlast, die Abmessungen und der Stromverbrauch an Bord limitiert sind und gleichzeitig strenge Anforderungen an Reichweite und Auflösung gestellt werden. Basierend auf den Nachteilen der Tiefenschätzung mit nur einer Kamera entwerfen wir ein neuartiges Multi-IMU-Multi-Kamerasystem zur Tiefenschätzung über grosse Entfernungen, das sich besonders für unbemannte Flugzeuge mit Tragflächen eignet. Die veränderliche relative Position und Orientierung des Stereo-Kamera-Systems wird in Echtzeit mittels Inertialmessungen, Kamerabildern und Deep Learning geschätzt.

Der letzte Teil dieser Dissertation untersucht die autonome Landeplatzerkennung, die lernbasierten Menschenerkennung und die Rekonstruktion eines Gebiets durch die Zusammenarbeit mehrerer Drohnen. Durch diesen Teil werden weitere Gebiete des maschinellen Sehens wie Szenenverständnis, Objekterkennung und die Registrierung von Punktwolken abgedeckt. Insgesamt erarbeitet diese Dissertation ein umfassendes System zum maschinellen Sehen für Drohnen und ermöglicht dadurch eine autonome Missionsausführung. Die entworfenen Algorithmen werden mit realistischen synthetischen Datensätzen, Hardware-in-the-Loop-Tests oder Feldexperimenten validiert. Im Rahmen dieser Dissertation wurden mehr als sechs unbemannte Drehflügler und Flugzeuge mit Sensorsystemen ausgestattet und in echten Missionen eingesetzt. Um die Forschung auf diesem Gebiet weiter voranzutreiben, wurde der Quellcode mehrerer Algorithmen und aufgenommene Datensätze mit verschiedenen Sensormodalitäten öffentlich zugänglich gemacht.

Schlüsselwörter: Unbemannte Luftfahrzeuge; 2D-Computer-Sehen; maschinelles Lernen; tiefes Lernen; Bildausrichtung; Bildsegmentierung; Bildklassifizierung; 3D-Computer-Sehen; 3D-Rekonstruktion; Objekterkennung; Objektverfolgung; Objektlokalisierung; Mehrkamera-Multi-IMU-Systeme; visuell-inertiale Navigation; Lokalisierung und Kartenerstellung; Rendering von Referenzansichten; Tiefenschätzung über große Entfernungen.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Roland Siegwart, director of the Autonomous Systems Lab, ETH Zurich, who greatly advised and supported me as a tutor throughout my master's and doctoral studies. I sincerely appreciate being granted the opportunity to work in such a prestigious institute with unparalleled infrastructure and working atmosphere. A great thank you also to Prof. Margarita Chli and Prof. Shaojie Shen for co-supervising this dissertation.

During my time as a Ph.D. student at the Autonomous Systems Lab I had the opportunity to learn and receive advice from outstanding colleagues: Firstly, I would like to thank my lab-internal supervisors Dr. Gabriel Agamennoni, Dr. Igor Gilitschenski, Dr. Enric Galceran, Dr. Cesar Cadena, and Dr. Juan Nieto for their guidance and invaluable feedback throughout my doctoral studies. I would like to thank Dr. Paul Furgale and Dr. Mike Bosse who helped me in the first phase of my doctoral studies. A big thank you to my great former colleagues Dr. Jörn Rehder and Dr. Janosch Nikolic who taught me everything about calibration of cameras and inertial measurement units. Thank you also to Dr. Michael Burri, Dr. Markus Achtelik, Dr. Sammy Omari, and Dr. Philipp Krüsi.

I am proud that I have been part of ASL's fixed-wing team with whom I've spent countless hours working on fixed-wing UAVs in the lab and on the field learning about hardware, system integration, payload constraints, fixed-wing dynamics, control, path planning and more: Thank you to Dr. Philipp Oettershagen, Dr. Thomas Stastny, Amir Melzer, Thomas Mantel, Dr. Konrad Rudin, Prof. Kostas Alexis, Dr. Nicholas Lawrance, Florian Achermann, and David Rohr.

Furthermore, I am very grateful that I could be part of the mapping team where I could learn from excellent programmers and software engineers how to efficiently work in a team on a large-scale code base. A great thank you to: Dr. Simon Lynen, Dr. Thomas Schneider, Dr. Marcin Dymczyk, Dr. Mathias Bürki, and Dr. Marius Fehr. IT support and the secretary's office are essential to keep the lab running: Thank you to Stefan Bertschi, Michael Riner and our secretaries Luciana Borsatti and Cornelia Della Casa. Furthermore, I vastly benefited from the work of outstanding and hard-working Bachelor and Master students, *Zivis*, and safety pilots.

I also had the chance to work with excellent researchers and engineers from external institutes: Thank you for the great collaboration with the Jet Propulsion Laboratory in particular Dr. Roland Brockers and Dr. Larry Matthies, thank you to Dr. Lucas Teixeira and Prof. Margarita Chli from the Vision for Robotics Lab, Dr. Johannes Schönberger from the Computer Vision and Geometry Group, and to the Linköping team under Prof. Patrick Doherty.

My deepest gratitude, however, goes to my family for their unconditional love, help, and support – without them on my side I would have never reached this point.

March 29, 2021 *Timo Hinzmann*

Financial Support

The research leading to these results has received funding from the following entities:

- the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°285417 (ICARUS)
- the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°600958 (SHERPA)
- ArmaSuisse – Science and Technology under project number n°050-45
- the Swiss air rescue organization Rega and
- Aurora Flight Sciences (sponsored the UAV platform Skate).

Contents

Glossary

AS	Active Search 62
ASL	Autonomous Systems Lab 11
BOW	Bag-of-Words 62
CNN	Convolutional Neural Network 63, 64, 130, 131
DEM	Digital Elevation Map 64
DL	Deep Learning 5–7, 26–28, 64, 69, 106, 130, 132
DNN	Deep Neural Network 63
DoF	Degree of Freedom 6, 12, 14, 25, 26, 62, 63, 65, 66, 69, 106
EKF	Extended Kalman Filter 13, 14
FPN	Feature Pyramid Networks 131
GNSS	Global Navigation Satellite System 4, 6, 25, 26
GPS	Global Positioning System 11, 13, 62, 64, 66
GPU	Graphics Processing Unit 106, 107, 110
HOG	Histogram of Oriented Gradient 64, 131, 142, 143
ICLK	Inverse Compositional Lucas-Kanade 6, 7, 12, 14, 25, 64–66, 106, 108
ICP	Iterative Closest Point 28, 64
IMU	Inertial Measurement Unit 5–7, 13, 14, 22, 26, 64, 140, 141
IoU	Intersection over Union 133, 134
kNN	k-Nearest Neighbors 62
LiDAR	Light Detection And Ranging 4, 13, 27
LWIR	Long-Wave Infrared 7, 16, 27, 28, 130
MI	Mutual Information 64
ML	Machine Learning 5
MLE	Maximum Likelihood Estimation 63
NIR	Near-Infrared 27
PDF	Probability Density Function 138
PF	Particle Filter 137, 138
PnP	Perspective-n-Point 63
PVV	Probabilistic Vertex Voting 62

RADAR	Radio Detection And Ranging 27
RANSAC	Random Sample Consensus 62, 63
RF	Random Forest 27
SaR	Search and Rescue 7, 16, 23, 27, 130, 132, 142
SLAM	Simultaneous Localization And Mapping 4
SO	Ordnance Survey 64
SR	Systematic Resampling 138
SSD	Single Shot Detector 132
SVM	Support Vector Machine 131, 142, 143
SWE	Sliding Window Estimator 9, 11, 25, 26
TI	Thermal-Infrared 130, 131, 141
UAV	Unmanned Aerial Vehicle 3–7, 11–16, 25–28, 62–66, 69, 109, 130–135, 138, 140, 144
UTM	Universal Transverse Mercator 65, 137
VAV	Vote-and-verify 63
VI	Visual-Inertial 62
VLAD	Vector of Locally Aggregated Descriptors 63

Preface

This is a cumulative doctoral thesis and as such consists of the most relevant publications. The publications are grouped into three parts and attached at the end. In addition to the individual publications an overarching introduction is provided in Chapter 1. We start with explaining the relevance of this thesis, followed by the objectives and the approach taken to fulfill these. For each contributing publication we explain how it embeds into the overall goals of this thesis and highlight the relevance of the research work in Chapter 2. Furthermore, we show how each paper is related to our other publications. We close this thesis by a summary of the achievements and provide an outlook for future directions and research in Chapter 3.

Chapter 1

Introduction

The advances in the research and development of Unmanned Aerial Vehicles (UAVs) have been enormous in the last decade, making drones a ubiquitous and indispensable tool in various applications. UAVs promise to take over tasks that are dangerous,

Figure 1.1: Atlantik-Solar [197, 198]

Figure 1.2: Techpod [125]

Figure 1.3: senseSoar [157, 160]

Figure 1.4: Skate by Aurora [1]

Figure 1.5: Rmax by Yamaha [4]

Figure 1.6: Rega Drone [3]

Figure 1.7: A subset of UAVs used in this dissertation for conducting field experiments and data collection.

repetitive, or even unachievable for human-crewed operations. Already today, they deliver vaccines and medical supplies to remote places, inspect wind turbines or other industrial facilities, count wildlife and spot poachers, or create overview

images and metrical elevation maps for a first damage assessment after natural catastrophes like earthquakes or tsunamis.

However, a large proportion of the early research on UAVs has focused on experiments under lab conditions in structured indoor or close-to-ground outdoor environments. Often motion capture systems are employed, providing nearly perfect state information and enabling impressive demonstrations of control algorithms. Likewise, a large variety of depth sensor types such as Light Detection And Ranging (LiDAR), depth camera, and small-baseline rigid stereo are applicable and used to show safe navigation in unknown environments.

In contrast, the goal of this thesis is to identify and address the challenges that occur for small-scale Unmanned Aerial Vehicles operating in large-scale unstructured outdoor environments. A selection of rotary-wing and fixed-wing UAVs used in this dissertation for field experiments and data collection is shown in Fig. 1.7. This chapter continues by discussing the identified challenges in Sec. 1.1. An overview of how these challenges are addressed on the software and hardware level is presented in Sec. 1.2.

1.1 Motivation and Objectives

This section discusses the identified challenges and objectives of this dissertation.

Localization and Mapping A core requirement for an autonomous robot is the ability to localize itself in a pre-existing map or to perform Simultaneous Localization And Mapping (SLAM). For position estimation in open outdoor spaces, the Global Navigation Satellite System (GNSS) may be applicable. However, the accuracy obtained from consumer-grade GNSS receivers is limited, and dropouts or multi-pathing near a mountainside, urban canyons, or close to the ground make this sensor less reliable. Our objective in this module is to investigate sensor modalities and systems to localize and map large-scale environments that undergo substantial appearance changes.

Depth Estimation Robots require a reliable depth estimation in order to navigate safely through unexplored or dynamic environments. Ground robots or UAVs operating in confined spaces usually gain this environmental awareness with the help of small-baseline (e.g., 12 cm) stereo systems, depth cameras, or LiDARs. However, the vast majority of small-scale UAVs cannot be equipped with the sensors mentioned above due to the tight constraints on the payload, dimensions, power consumption, price, and application-specific requirements for range and resolution. In order to be able to fulfill missions close to the ground or perform landing and take-off maneuvers, the objective of this module is to design a light-weight, power-, and cost-efficient sensor system for long-range depth estimation.

Landing Site Detection With the increasing number of UAVs performing automated or autonomous missions, safety systems that can intervene in sudden

hardware or software failure become crucial components of the autopilot. For instance, fixed-wing UAVs may encounter motor failures during their autonomous mission in a previously unexplored environment. Therefore, the goal of this module is to design a vision-based system that is able to identify landing sites reliably. The objective is furthermore to focus on fixed-wing UAVs due to the increased complexity: The system needs to be capable of computing an optimized and feasible approach path while considering essential constraints such as local wind fields and obstacles.

Object Detection In many UAV missions, the goal is to detect specific objects and report their location to a human operator at the ground control station. Such applications include, for instance, wildlife counting, detection of poachers, or finding missing humans that are trapped in mountain areas. However, for the system to truly relieve the human operator, the false alarm rate needs to be minimized while not overlooking any true positives. Furthermore, the system is to be operational during day and night flights.

Collaborative Reconstruction Depending on the type of real-world UAV mission, it is sometimes beneficial or mandatory to have two or more UAVs collaborate to achieve a common goal. The UAVs might share the same or have carefully selected complementary capabilities. However, research often focuses merely on one robot due to the reduced complexity and cost for system integration and field experiments. In this context, the objective of this module is to experimentally validate a collaborative reconstruction framework with two robots equipped with a different set of sensors.

1.2 Approach

To accomplish our objectives, we group this thesis into three parts. The first two parts are considered base competencies that the robot needs for safe navigation. The third part show-cases three different UAV missions and its very individual challenges, consisting of landing site detection, objection detection, and collaborative reconstruction. All challenges have in common that the UAV operates in a large-scale, unprepared environment with distant landmarks. Furthermore, the problem is tackled with one or more optical cameras as the main exteroceptive sensor, enhanced by using Machine Learning (ML), Deep Learning (DL), and Inertial Measurement Units (IMUs) where applicable.

Part A: Localization and Mapping

The goal of Part A is to introduce localization and mapping methods that are particularly suited for UAVs operating in large-scale outdoor environments.

Visual-Inertial Localization and Mapping Due to the complementary characteristics, the fusion of sensor information from cameras and IMUs has de-facto

become a golden standard in robotic navigation. Also, for small-scale UAVs operating in large-scale outdoor environments, the combination of cameras and IMUs is compelling due to its low power consumption, weight, dimensions, and cost on the one hand and its information-rich data, on the other hand, enabling high-rate metric motion estimation. In this context, Paper I describes our monocular visual-inertial navigation system that is evaluated in real-world flights with a fixed-wing UAV. For the visual-inertial motion estimation, we rely on a trade-off between accuracy and computational costs: Optimization-based estimation is used instead of filtering to achieve a higher accuracy, but a sliding-window ensures that older states are marginalized out, keeping the number of variables to be optimized in each iteration at a minimum.

2.5D Mapping and DL-Based Localization Visual-inertial localization and mapping in indoor or confined environments usually requires a full 3D representation of the map in order to adequately model complex shapes. However, the memory and computational load can be reduced by using a 2.5D representation for flights in large-scale outdoor environments, particularly at higher altitudes. Paper II presents a survey on the real-time computation of georeferenced orthoimages and elevation maps with the help of a single camera and an autopilot equipped with a GNSS receiver. In Paper III, we propose the combination of a real-time 2.5D rendering engine with a learning-based six-Degree of Freedom (DoF) Inverse Compositional Lucas-Kanade (ICLK) algorithm to be able to localize in 2.5D orthoimages that are several years old and have undergone environmental and seasonal changes.

Part B: Depth Estimation

The objective of Part B is to propose and experimentally validate long-range depth estimation methods designed for small-scale UAV operations in large-scale outdoor environments. The introduced depth estimation methods rely predominantly on classical geometric considerations to ensure reliable depth estimates in every scenario.

Monocular Depth Estimation Accurate real-time estimation of a single moving camera allows creating multiple virtual stereo pairs for depth estimation. The scene can be geometrically reconstructed by rectifying the image pairs using the pre-calibrated camera intrinsics and online-estimated extrinsics and finding feature correspondences along the epipolar line. This underlying principle, commonly called depth-from-motion or structure-from-motion, is evaluated in real-world test flights in Paper IV. In this experiment, a monocular oblique setup is mounted on a fixed-wing UAV and pose priors from the autopilot are refined with visual cues in a smoothing-based sliding-window estimator.

DL-Based Non-rigid Multi-view Stereo The disadvantage of classical depth-from-motion approaches with a monocular setup is that the depth uncertainty is

dependent on the arrangement of the virtual stereo pair and hence on the flight trajectory. Deep learning methods attempt to overcome these shortcomings in approaches like depth from single image [71], single view depth completion with sparse depth priors [258], or multi-view pose and depth estimation [262, 266]. Instead, our approach is motivated by [250], which favors stereo to monocular vision, and by the required high guarantees on reliability in the airspace sector where false depth estimations may lead to a loss of the aircraft. Hence, to avoid sub-optimal or degenerate cases of the monocular setup, we design a novel wide-baseline multi-camera, multi-IMU system that allows long-range depth estimation independent of the flight trajectory. The proposed framework is outlined and experimentally validated in Paper V. To further increase the robustness of the vision-based alignment, Paper VI proposes a learning-based sparse ICLK algorithm. In contrast to end-to-end DL-based depth estimation methods, our framework merely enhances classical geometric considerations with learning-based image alignment.

Part C: UAV Missions

The objective of Part C is to show-case several realistic UAV missions and reveal and to address the underlying robotic challenges.

Landing Site Detection Paper VII approaches the complex task of finding a suitable landing spot for a fixed-wing UAV by splitting it into several sub-problems: Large homogeneous grass regions are found with the help of image segmentation and classification algorithms and evaluated based on their shape, area, and texture. After this rough 2D assessment, the potential landing spot candidates are analyzed further in 3D based on the slope and terrain roughness. The approach path is computed while taking into account the estimated local wind field, fixed-wing UAV dynamics, and obstacles obscuring the landing spot.

DL-based Human Detection Paper VIII describes our framework that is able to detect humans during night and day operation with the help of a sensor system consisting of an optical (RGB) and a Long-Wave Infrared (LWIR), i.e., thermal camera. In order to optimize the precision-recall of the detector, we compare various DL-based detectors and train them with extensive datasets recorded with different UAV platforms in realistic Search and Rescue (SaR) scenarios. The detections are automatically sent to the human operator at the ground control station, who can take further actions.

Collaborative Reconstruction The final paper, Paper IX, demonstrates how a heterogeneous fleet of UAVs can collaborate to reconstruct a scene of interest. We approach this task by equipping a high-flying fixed-wing UAV with an optical camera and a lower-flying, rotary-wing UAV with a laser scanner. During the mission, the UAVs are commanded with the help of an existing delegation framework [63, 66] to fully profit from the complementary capabilities and

characteristics of the robots. The generated dense laser point cloud is fused with the relatively sparse optical point cloud using a probabilistic point cloud registration algorithm [7] specifically designed to handle data originating from different sensor sources.

Chapter 2

Contribution

This chapter lists first-author scientific contributions from a selection of publications. The contributions are illustrated in the overview in Fig. 2.1. Every publication is presented by describing its context, listing its contributions, and pointing out interrelations to other publications. The chapter concludes with the full list of publications made during the doctoral studies and a list of supervised students.

2.1 Part A: Mapping and Localization

Paper I

Timo Hinzmann, Thomas Schneider, Marcin Dymczyk, Andreas Schaffner, Simon Lynen, Roland Siegwart, Igor Gilitschenski, "Monocular Visual-Inertial SLAM for Fixed-Wing UAVs Using Sliding Window Based Nonlinear Optimization". In *International Symposium on Visual Computing (ISVC)*, 2016.

Context

The estimation of robot states and landmarks is of paramount importance to be able to navigate and interact with the environment. In this work, we select a monocular camera and an inertial measurement unit as a sensor suite due to its complementary character, low payload weight, and rich data information. In contrast to previous research that mainly relied on filtering [186], Paper I uses smoothing and captures problem information like state variables to optimize and error terms in a factor graph.

Contribution

This paper introduces our Sliding Window Estimator (SWE), a light-weight real-time visual-inertial navigation system that smooths the states within a sliding-window

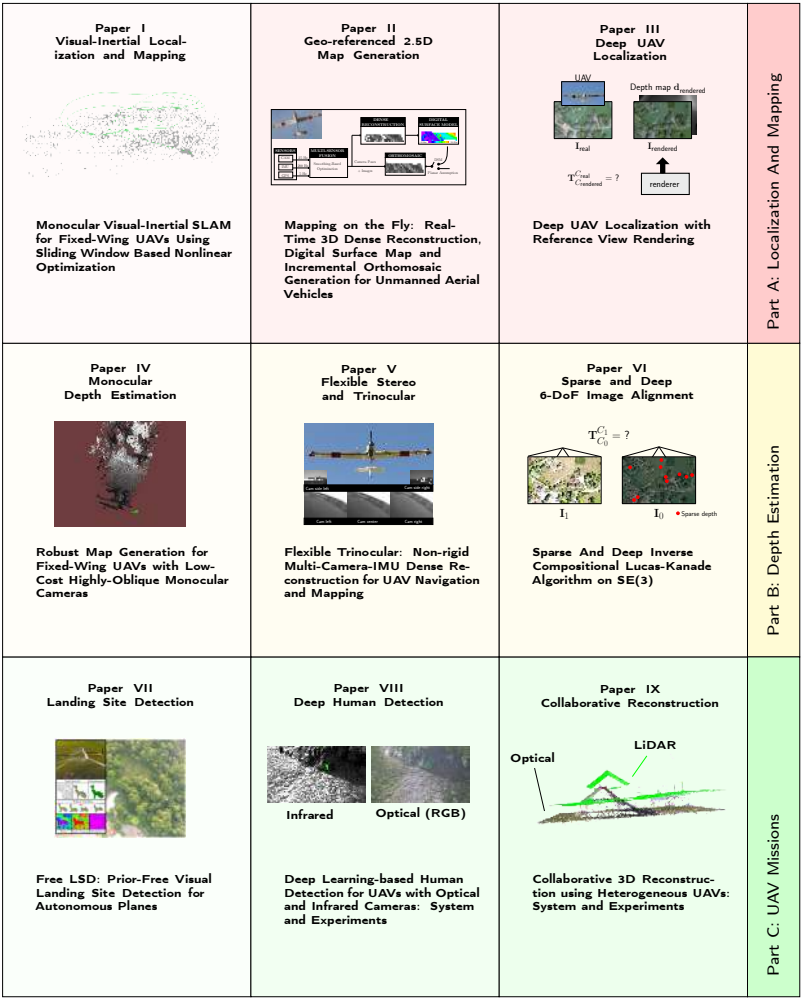


Figure 2.1: Overview of selected first-author scientific contributions.

and marginalizes out states that fall outside of that window. A commercially available fixed-wing UAV is employed as a carrier platform for the visual-inertial sensor rig, using a Global Positioning System (GPS) receiver to obtain a reference solution. Real-world experiments show that, while the GPS solution is noisy close to the terrain, the trajectory of the visual-inertial system is smooth, enabling, for instance, automated take-off and landing maneuvers.

Interrelations

The SWE was integrated into Autonomous Systems Lab (ASL)'s maplab [233] framework, where it was used as a front-end optimizer. Furthermore, the SWE is being employed for commercial purposes in an ETH Zurich spin-off.

Paper II

Timo Hinzmann, Johannes L. Schönberger, Marc Pollefeys, Roland Siegwart, "Mapping on the Fly: Real-Time 3D Dense Reconstruction, Digital Surface Map and Incremental Orthomosaic Generation for Unmanned Aerial Vehicles". In *Field and Service Robotics (FSR)*, 2018.

Context

Real-time generation of geo-referenced elevation maps and orthoimages is a crucial capability for UAVs operating in large-scale outdoor environments. Firstly, these maps can be used for geo-referenced localization of UAVs flying at high altitudes. Secondly, based on the maps, a rescue team can make informed decisions, e.g., in disaster scenarios like earthquakes. However, commercially available photogrammetric or pure structure-from-motion software often requires several hours to process a batch of images.

Contribution

This work introduces a framework to efficiently generate geo-referenced 2.5D elevation maps and orthomosaics for real-time robotic applications. The orthomosaics are computed incrementally, incorporating the 2.5D elevation map and considering the viewing angle. The performance is compared to homography, point cloud, and batch alternatives. Code and datasets are made publicly available to boost the progress in this field.

Interrelations

The output of this framework, a geo-referenced 2.5D orthomosaic and elevation map, can be used as input to Paper III for UAV localization.

Paper III

Timo Hinzmann and Roland Siegwart, "Deep UAV Localization with Reference View Rendering". In *CoRR*, 2020.

Context

Vision-based ego-pose estimation in vast outdoor environments entails a variety of challenges: During a flight, the same landmarks might be observed from very different scales and orientations. Classical matching approaches would require a robust scale- and rotation-invariant descriptor. Furthermore, landmarks in indoor environments are usually illuminated with an artificial light source, resulting in a similar appearance independent of the time of the day. However, UAVs operating in outdoor environments are facing significant seasonal and daytime illumination changes. Finally, representing large-scale environments by a set of 3D landmarks with descriptors is memory consuming, making a query computationally expensive.

Contribution

Firstly, Paper III introduces a rendering engine that takes as input a geo-referenced orthoimage, elevation map, distortion, and camera model and renders a reference view for a given camera pose. The reference view consists of a depth map and an optical image that is deemed to be similar in scale and orientation to the real-world image taken by the UAV. Secondly, we combine the rendering engine with a learning-based ICLK algorithm that returns the six-DoF geo-referenced pose that best aligns the reference and query image. Experiments demonstrate that this framework is able, for instance, to align images with a four-year time gap and extreme environmental differences. Source code and datasets are partially released to the community for further research.

Interrelations

The output from Paper II, a geo-referenced 2.5D orthomosaic and elevation map, can be used as input for localization.

2.2 Part B: Depth Estimation

Paper IV

Timo Hinzmann, Thomas Schneider, Marcin Tomasz Dymczyk, Amir Melzer, Thomas Mantel, Igor Gilitschenski, Roland Siegwart, "Robust Map Generation for Fixed-Wing UAVs with Low-Cost Highly-Oblique Monocular Cameras". In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

Context

Precise, real-time depth estimation is of paramount importance for UAVs, allowing them to operate autonomously in a previously unmapped environment. While ground robots, for instance, can be equipped with depth sensors and LiDARs to obtain depth information, small fixed-wing UAVs face unlike more immense challenges: Firstly, they have strict constraints on payload weight, dimensions, and power consumption. Secondly, the required range is not reachable by state-of-the-art sensors. Therefore, in this work, we rely on a monocular highly-oblique camera as the exteroceptive sensor.

Contribution

This work presents a light-weight framework for generating dense depth maps with a monocular camera setup in real-time by distributing the computational load on different processors: Geo-referenced pose priors are computed on a micro-controller by fusing GPS, IMU, magnetometer, and pressure measurements in an Extended Kalman Filter (EKF). The pose priors are then refined with visual cues on the on-board computer in a sliding-window estimator. The estimated poses can be used to integrate sparse landmarks into OctoMap [276] or used for image rectification and dense point cloud generation. The paper shows results from real-world flight experiments with a comparison of depth maps computed from planar and polar rectification.

Interrelations

The shortcomings encountered with monocular depth estimation by means of depth-from-motion motivated Paper V and Paper VI.

Paper V

Timo Hinzmann, Cesar Cadena, Juan Nieto, Roland Siegwart, "Flexible Trinocular: Non-rigid Multi-Camera-IMU Dense Reconstruction for UAV Navigation and Mapping". In *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

Context

Accurate long-range depth estimation is essential for fast-flying Unmanned Aerial Vehicles, enabling obstacle avoidance and local re-planning. In classical monocular depth-from-motion algorithms, camera poses are estimated and used to create a virtual stereo pair. However, with a front-looking or nadir camera setup, the epipole of this virtual stereo pair is likely to be close or inside the image, making this geometric constellation prone to return a depth map with high uncertainty. Therefore, in this work, we rely on a wide-baseline multi-stereo setup to reduce the depth uncertainty.

Contribution

Paper V proposes a novel multi-camera multi-IMU sensor system designed for long-range depth estimation on-board of fast-flying UAVs. Due to aerodynamics and structural constraints of the wings, the stereo baseline is non-rigid and needs to be estimated accurately in real-time. In order to achieve this, inertial measurements, visual cues, and a probabilistic wing model are fused in an EKF. Real-world experiments show the effectiveness of the approach in challenging scenarios and lighting conditions.

Interrelations

Paper V is motivated by the shortcomings encountered in Paper IV, where the quality of the depth map is influenced by the flight path of the UAV. The idea of wide-baseline visual-inertial flexible stereo was first introduced in our paper [123] where it was evaluated on synthetic datasets.

Paper VI

Timo Hinzmann and Roland Siegwart, "Sparse And Deep Inverse Compositional Lucas-Kanade Algorithm on SE(3)". In *CoRR*, 2020.

Context

In many approaches, an important factor for reliable depth estimation is a precise image alignment and relative pose estimation algorithm. For instance, classical methods require accurately estimated extrinsics to rectify the stereo pairs and to search for feature correspondences along the epipolar line. Additionally, depth information is needed to estimate the full six DoFs relative pose between the two cameras. However, in visual-inertial odometry pipelines often only sparse depth information is available. Lastly, a further challenge in the image alignment of two images in real-world scenarios is the potential appearance difference due to different imaging sensor characteristics, exposure times, or lens flares.

Contribution

Paper VI introduces SD-6DoF-ICLK, a learning-based ICLK algorithm for image alignment on SE(3) using sparse depth estimates. A synthetic dataset rendered with OpenGL [273] shows that SD-6DoF-ICLK outperforms the classical sparse 6DoF-ICLK algorithm by a large margin. The proposed SD-6Dof-ICLK is able to perform inference in 145 ms on a GeForce RTX 2080 Ti with input images at a resolution of 752×480 pixels.

Interrelations

Paper VI was motivated by the shortcomings experienced in Paper IV and Paper V. Note the subtle differences to Paper III: Paper III implements a learning-based

6-DoF-ICLK algorithm where one of the image pairs has access to the full depth map, obtained from the rendering engine. In contrast, Paper VI introduces a learning-based 6-DoF-ICLK algorithm where one of the image pairs has access to a sparse depth map, e.g., obtained from visual-inertial odometry.

2.3 Part C: UAV Missions

Paper VII

Timo Hinzmann, Thomas Stastny, Cesar Cadena, Roland Siegwart, Igor Gilitschenski, "Free LSD: Prior-Free Visual Landing Site Detection for Autonomous Planes". In *IEEE Robotics and Automation Letters (RA-L)*, 2018.

Context

A crucial capability of autonomous UAVs is to be able to mitigate suddenly occurring failure cases and to prevent the loss of the vehicle at all costs. For instance, UAVs may encounter motor failures during their autonomous mission. In this situation, the UAV is expected to be able to autonomously detect a suitable landing spot and compute an approach path in a potentially unexplored environment.

Contribution

Paper VII proposes a framework for landing site detection, which is able to find a suitable landing spot in an unknown environment. Our approach handles the complexity of the problem by dividing it into several subproblems: Landing spots are first scored based on their texture, shape, and area using terrain segmentation and classification. The quality of the landing spot candidates is then assessed further by considering slope, terrain roughness, obstacles obscuring the approach path, wind field, and vehicle dynamics. The effectiveness of the framework is validated based on synthetic and real-world datasets.

Paper VIII

Timo Hinzmann, Tobias Stegemann, Cesar Cadena, Roland Siegwart, "Deep Learning-based Human Detection for UAVs with Optical and Infrared Cameras: System and Experiments". In *CoRR*, 2020.

Context

In many UAV missions, the objective is to detect specific objects and to report their location to a human operator at the ground control station. Applications include wildlife counting, detection of poachers, or finding missing humans that are trapped in mountain areas – the later of which is addressed in Paper VIII. However, for the system to truly relieve the human operator, the false alarm rate needs to be minimized while not overlooking any true positives. Day and night operability is

required and motivates the combination of an optical with a LWIR, i.e, thermal camera. Moreover, adding the LWIR camera is beneficial for day flights due to its complementary image information.

Contribution

In Paper VIII, we present our deep learning-based human detection system combining optical (RGB) and LWIR image information. In extensive evaluations recorded during realistic SaR scenarios, we compare RetinaNet [171], YOLO [217] variants, and hand-crafted optical-infrared human detectors. Along with this paper, the recorded optical-infrared datasets with bounding box annotations and convenient annotation tools are made publicly available.

Interrelations

Paper VIII is a successor paper of our previous publications [151, 199] on optical-infrared human detection.

Paper IX

Timo Hinzmann, Thomas Stastny, Gianpaolo Conte, Patrick Doherty, Piotr Rudol, Marius Wzorek, Igor Gilitschenski, Enric Galceran, Roland Siegwart, "Collaborative 3D Reconstruction using Heterogeneous UAVs: System and Experiments". In *International Symposium on Experimental Robotics (ISER)*, 2016.

Context

In real-world search-and-rescue scenarios, UAVs are expected to collaborate seamlessly to achieve a common goal while taking advantage of the individual capabilities like maximum speed or mounted sensor system. In this context, Paper IX investigates the collaborative reconstruction of region of interests by delegating a fleet of heterogeneous UAVs.

Contribution

In Paper IX, a fixed-wing UAV with an optical camera and a rotary-wing UAV equipped with a laser scanner reconstruct an area selected by a ground control crew. The obtained optical and laser point-cloud are aligned to take advantage of the complementary character of the two sensor modalities. Extensive real-world experiments at different locations conclude that classical point-cloud alignment methods show an inferior performance compared to the probabilistic data association algorithm [7]. Datasets consisting of optical and dense point-clouds and the probabilistic data association algorithm are made publicly available to boost further research in this domain.

Interrelations

The delegation framework coordinates robots and their actions to achieve a common goal and is described in the companion paper [66]. The algorithm was used in the experiments for Paper IX, but is not claimed as part of the contributions.

2.4 List of Publications

This section provides a full list of publications and articles that have been (co-)authored during the doctoral studies. The publications are sorted by type and year.

Pre-Publication Articles

- T. Hinzmann and R. Siegwart. Sparse and deep inverse compositional Lucas-Kanade algorithm on SE(3). *CoRR*, 2020

- T. Hinzmann and R. Siegwart. Deep UAV localization with reference view rendering. *CoRR*, 2020

- T. Hinzmann, T. Stegemann, C. Cadena, and R. Siegwart. Deep learning-based human detection for UAVs with optical and infrared cameras: System and experiments. *CoRR*, 2020

Journals Articles

- T. Hinzmann, T. Stastny, C. Cadena, R. Siegwart, and I. Gilitschenski. Free LSD: Prior-free visual landing site detection for autonomous planes. *IEEE Robotics and Automation Letters*, 3(3):2545–2552, 2018

- P. Oettershagen, T. Stastny, T. Hinzmann, K. Rudin, T. Mantel, A. Melzer, B. Wawrzacz, G. Hitz, and R. Siegwart. Robotic technologies for solar-powered uavs: Fully autonomous updraft-aware aerial sensing for multiday search-and-rescue missions. *Journal of Field Robotics*, 35(4):612–640, 2018

- P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart. Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight. *Journal of Field Robotics*, 34(7):1352–1377, 2017

- P. Doherty, J. Kvarnstroem, P. Rudol, M. Wzorek, G. Conte, C. Berger, T. Hinzmann, and T. Stastny. A Collaborative Framework for 3D Mapping using Unmanned Aerial Vehicles. *Int. J. Comput. Vision*, 2016

Conference Publications

- T. Hinzmann, C. Cadena, and J. Nieto. Flexible trinocular: Non-rigid multi-camera-IMU dense reconstruction for UAV navigation and mapping. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1137–1142, 2019

- T. Hinzmann, T. Taubner, and R. Siegwart. Flexible stereo: Constrained, non-rigid, wide-baseline stereo vision for fixed-wing aerial platforms. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2550–2557. IEEE, 2018

- R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli. GOMSF: Graph-optimization based multi-sensor fusion for robust UAV pose estimation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1421–1428. IEEE, 2018

- M. Huber, T. Hinzmann, R. Siegwart, and L. H. Matthies. Cubic range error model for stereo vision with illuminators. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 842–848. IEEE, 2018

- T. Hinzmann, J. L. Schönberger, M. Pollefeys, and R. Siegwart. Mapping on the fly: Real-time 3d dense reconstruction, digital surface map and incremental orthomosaic generation for unmanned aerial vehicles. In *Field and Service Robotics*, pages 383–396. Springer, 2018

- M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart. Visual place recognition with probabilistic voting. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3192–3199. IEEE, 2017

- T. Hinzmann, T. Schneider, M. Dymczyk, A. Schaffner, S. Lynen, R. Siegwart, and I. Gilitschenski. Monocular visual-inertial SLAM for fixed-wing UAVs using sliding window based nonlinear optimization. In *International Symposium on Visual Computing*, pages 569–581. Springer, 2016

- J. Kümmerle, T. Hinzmann, A. S. Vempati, and R. Siegwart. Real-time detection and tracking of multiple humans from high bird's-eye views in the visual and infrared spectrum. In *International Symposium on Visual Computing*, pages 545–556. Springer, 2016

- T. Hinzmann, T. Schneider, M. Dymczyk, A. Melzer, T. Mantel, R. Siegwart, and I. Gilitschenski. Robust map generation for fixed-wing UAVs with low-cost highly-oblique monocular cameras. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3261–3268. IEEE, 2016

- P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, K. Alexis, and R. Siegwart. Perpetual flight with a small solar-powered UAV: Flight results, performance analysis and model validation. In *2016 IEEE Aerospace Conference*, pages 1–8. IEEE, 2016

- T. Hinzmann, T. Stastny, G. Conte, P. Doherty, P. Rudol, M. Wzorek, E. Galceran, R. Siegwart, and I. Gilitschenski. Collaborative 3D reconstruction using heterogeneous UAVs: System and experiments. In *International Symposium on Experimental Robotics*, pages 43–56. Springer, 2016

- J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart. Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016

2.5 List of Supervised Students

This section lists all student projects that have been (co-)supervised during the doctoral studies. For projects that resulted in a publication the citation is given. External supervisors are listed with affiliation at the time of the project. The projects are sorted by type and year.

Master Thesis

Master student, 6 months full time

- Pascal Schoppmann (2020): "Multi-Resolution Elevation Mapping and Safe Landing Site Detection for an Autonomous Rotorcraft", supervised by Roland Brockers (JPL), Michael Pantic, and Timo Hinzmann

- Sandro Berchier (2019): "Experimental Validation of State Estimation and Localization for Hybrid MAVs in Perceptually Degraded Environments", supervised by Abel Gawel, Timo Hinzmann, Luca Carlone (MIT), and Ali Agha (JPL).

- Andreea Lutac (2018): "Optimal Pose Selection for Aerial Dense Reconstruction and Localization", supervised by Timo Hinzmann and Rik Bähnemann.

- Tobias Stegemann (2018): "Deep Learning-based Human Detection in Optical and Thermal Aerial Imagery", supervised by Timo Hinzmann and Cesar Cadena.

- Ruben Mascaro (2017): "Graph-Optimization Based Multi-Sensor Fusion for Robust UAV Pose Estimation", supervised by Lucas Teixeira (Vision for Robotics Lab, ETH Zurich) and Timo Hinzmann.

- Adam Radomski (2017): “Robust Multirotor Precision Landing in Outdoor Environment”, supervised by Timo Hinzmann and Thomas Schneider.

- Marius Huber (2017): “Autonomous Rotorcraft Landing with Structured Light Stereo Vision”, supervised by Larry Matthies (JPL), Roland Brockers (JPL), Timo Hinzmann, and Thomas Stastny.

- Danylo Malyuta (2017): “Guidance, Navigation, Control and Mission Logic for Quadrotor Full-cycle Autonomy”, supervised by Roland Brockers (JPL), Thomas Stastny, and Timo Hinzmann.

- Ryen Elith (2017): “Design of a Radar System for Sense and Avoid Applications”, supervised by Amir Melzer and Timo Hinzmann.

- Ricardo Zurfluh (2016): “GNSS-Based Attitude Determination for Automatic Magnetometer Calibration Ricardo Zurfluh”, supervised by Timo Hinzmann and Amir Melzer.

- Julius Kümmerle (2016): “Real-time Detection and Tracking of Multiple Human Targets from Aerial Vehicles Using Thermal Imagery”, supervised by Timo Hinzmann and Anurag Vempati.

- Bastien Chatton (2016): “Thermal Updraft Prediction for a Fixed-Wing UAVs”, supervised by Philipp Oettershagen and Timo Hinzmann.

- Nicolas El Hayek (2016): “Ridge Lift Exploitation for Small Unmanned Fixed-Wing Aircraft”, supervised by Thomas Stastny, Philipp Oettershagen, and Timo Hinzmann.

- Mathias Gehrig (2016): “Robust and Efficient Loop-Closure Detection for Aerial Images”, supervised by Timo Hinzmann and Elena Stumm.

- Pavel Vechersky (2016): “Development of a Comprehensive, Hardware-in-the-Loop Simulation Environment for Fixed-Wing UAVs”, supervised by Timo Hinzmann, Thomas Stastny, and Philipp Oettershagen.

- Andreas Forster (2016): “Tightly Coupled GNSS Integration into a SLAM Framework”, supervised by Timo Hinzmann and Amir Melzer.

- Felix Renaut (2015): “Vision-Based Autonomous Landing of an Unmanned Fixed-Wing UAV”, supervised by Timo Hinzmann and Thomas Stastny.

- Andreas Schaffner (2015): “Demonstration of Visual Navigation with Fixed-Wing UAVs”, supervised by Timo Hinzmann, Gabriel Agamennoni, and Tim Dawson-Townsend (Aurora).

Semester Thesis

Master student, 3-4 months part time

- Felix Graule (2019): "Towards Robust Cross-Spectral Optical-Thermal SLAM onboard a fixed-wing UAV", supervised by Timo Hinzmann, Florian Achermann, and Nicholas Lawrance.
- Balazs Nagy (2017): "Constrained, Non-rigid, Wide-baseline Stereo Vision for Fixed-wing Aerial Platforms (Continued)", supervised by Timo Hinzmann.
- Rudolf Metzler (2017): "High-quality Ground-Truth Generation for Fixed-Wing UAVs", supervised by Guillaume Sébastien (ETHZ, IGP) and Timo Hinzmann.
- Jingwei Tang (2017): "Mutual-Information-Based Direct Visual-Inertial Odometry", supervised by Timo Hinzmann.
- Patrik Frey (2017): "Multi-Sensor Multi-State Constraint Kalman Filter for Fixed-Wing UAVs", supervised by Timo Hinzmann.
- Tim Taubner (2017): "Constrained, Non-rigid, Wide-baseline Stereo Vision for Fixed-wing Aerial Platforms", supervised by Timo Hinzmann and Thomas Stastny.

Courses

Master students, 1 semester part time.
All students listed below were supervised in the course "Perception and Learning for Robotics":

- Emilk Sempertegui, Cliff Li, and Giancarlo Di Biase in the course (2019): "Towards Semi-Supervised Learning for Human Detection from Aerial Images", supervised by Timo Hinzmann.
- Hrishikesh Gupta (2019): "Semantic Segmentation for Autonomous landing site detection using RGB Aerial Images", supervised by Timo Hinzmann.
- Adrian Ruckli and Alberto Pennino (2018): "Deep Learning Enhanced Human Detection using RGB and Infrared Imagery onboard of Fixed-Wing UAVs", supervised by Timo Hinzmann

Seminar

Bachelor or Master student, 3-4 months part time

- Jingwei Tang (2017): "Long Range Landmark Observations and their Influence on the Convergence of Bundle-Adjustment Problems", supervised by Timo Hinzmann.

Bachelor Thesis

Bachelor student, 3-4 months part time

- Laurent Braun (2016): "Robust Vision-Based Localization for Fixed-Wing UAVs", supervised by Timo Hinzmann.

Focus Project

6-8 Bachelor students, 1 year project full time

- Michael Arigoni (2015): "Control of a Wall Racing Robot for Agile Ground Maneuvers", supervised by Timo Hinzmann and Lennon Rodgers (MIT).

2.6 List of Released Software and Datasets

Within the doctoral studies, the author open-sourced or contributed to the following software packages:

Software (non-exhaustive list)

- **aerial_renderer:** A rendering program that, given a geo-referenced orthoimage, elevation map, camera pose, camera and distortion model, is able to render a depth, optical, or semantic image that can be used for training data generation or online map tracking (Paper III).
  `https://github.com/ethz-asl/aerial_renderer`
- **aerial_mapper:** A library containing basic code to generate geo-referenced dense point clouds, digital elevation models, and orthoimages (Paper II).
  `https://github.com/ethz-asl/aerial_mapper`
- **robust_point_cloud_registration:** A library containing the source code for point cloud registration using iterative probabilistic data association method [7]. It contains wrappers for other point cloud registration methods. All methods were experimentally validated in Paper IX.
  `https://github.com/ethz-asl/robust_point_cloud_registration`
- **kalibr:** A toolbox [220] for the intrinsic and extrinsic calibration of a or multiple cameras, the spatial-temporal calibration of a camera-IMU setup, and IMU-IMU calibration.
  `https://github.com/ethz-asl/kalibr`
- **aslam_cv2:** A library containing commonly used functions for computer vision algorithms, such as camera or distortion models.
  `https://github.com/ethz-asl/aslam_cv2`
- **maplab:** A framework [233] for visual-inertial mapping and localization.
  `https://github.com/ethz-asl/maplab`

Within the doctoral studies, the author released the following datasets:

Datasets (non-exhaustive list)

- Datasets consisting of optical and laser point clouds have been released in the context of paper Paper IX and the robust_point_cloud_registration repository.
  https://projects.asl.ethz.ch/datasets/doku.php?id=iser2016
- Datasets consisting of optical and infrared imagery, recorded during SaR missions for human detection has been released. The humans in the datasets are annotated by bounding boxes.
  https://projects.asl.ethz.ch/datasets/doku.php?id=jfr2020
- As part of Paper II and the aerial_mapper repository, we released synthetic and real-world images with corresponding ground-truth and estimated camera poses, camera intrinsics, and distortion parameters to encourage the development of new algorithms and to reproduce the results.
  https://github.com/ethz-asl/aerial_mapper
- In the context of Paper III and the aerial_renderer repository, we release orthoimages and elevation maps.
  https://github.com/ethz-asl/aerial_renderer

For the case that a link becomes unavailable, we encourage the reader to consult ASL's dataset website https://projects.asl.ethz.ch/datasets/, code base https://github.com/ethz-asl/, or to contact the authors. Further datasets and software packages might be released on ASL's dataset website or code base after the submission of this dissertation.

Chapter 3

Conclusion and Outlook

In this dissertation, we proposed and experimentally validated core components crucial for autonomous UAV that operate in unstructured large-scale outdoor environments. This chapter summarizes our key findings and outlines future research directions. Particularly the incorporation of deep learning has the potential to increase the robot's overall scene understanding and autonomy.

3.1 Part A: Localization And Mapping

Part A of this dissertation proposed and experimentally evaluated different localization and mapping approaches suited for UAVs navigating in large-scale outdoor environments: Paper I introduced SWE, a light-weight visual-inertial navigation system that uses a sliding-window to make a trade-off between accuracy and computational costs. The system is tested on a small-scale fixed-wing UAV and achieved superior pose estimation results close to the ground when compared to a consumer-grade GNSS receiver, enabling otherwise risky automated or autonomous maneuvers such as landings, take-offs, or fly-bys.

The mapping and geo-referenced ego-pose estimation in large-scale environments represented by a 2.5D map is covered by the closely related Paper II (mapping) and Paper III (localization): Paper II discusses methods for the generation of geo-referenced 2.5D elevation maps and orthomosaics for real-time robotic applications. Paper III combines a rendering engine with a learning-based ICLK algorithm to estimate the geo-referenced six-DoF camera pose: As input, the renderer requires a geo-referenced orthoimage, elevation map, distortion model, camera intrinsics, and camera pose and is able to render a depth and textured image (e.g., optical or semantical depending on the layer). In the experiments, the learning-based ICLK algorithm was, for instance, capable of robustly aligning images with a time gap of four years and extreme appearance changes. Both, Paper II and Paper III, make parts of the source code and datasets publicly available (cf. Sec. 2.6). Particularly the

rendering-based reference view generation and the learning-based image alignment have the potential to spark further research in this area.

Learning-based image alignment, loop-closure detection, and optimization The experiments in Paper III demonstrate that learning-based image alignment and optimization algorithms are able to outperform classical, non-trainable methods by a large margin. However, in existing localization and mapping pipelines, the vision modules for sequential or exhaustive image alignment, loop closure detection, or pose estimation still rely predominantly on non-trainable methods. Replacing these modules by trainable counterparts may greatly robustify the performance of the pipeline and make them more suitable for real-world applications under severe conditions (cf. [256]). Examples are the SIFT-based [176] matching and vertex voting algorithm in colmap [234, 235], the time-sequential Lucas-Kanade feature tracking method in the SWE (cf. Paper I), or the loop-closure detection algorithm dBoW2 [94] that is built up on FAST [225] and BRIEF [40] and used for re-localization in VINS-mono [213]. Furthermore, in future work, the DL-architecture should not only predict the state but also the state uncertainty. For instance in Paper III, the uncertainty of the six-DoF camera pose would be useful for data fusion or failure detection.

3.2 Part B: Depth Estimation

The second part of this dissertation investigated depth estimation methods applicable to small-scale UAVs operating in large-scale outdoor environments: Paper IV implements and evaluates a classical depth-from-motion algorithm with a single moving camera. The camera is in a highly-oblique[1] configuration to be able to detect objects well in advance. To achieve the goal of real-time on-board depth map computation, the publication proposes to distribute the computational load on different boards: Geo-referenced pose priors are obtained from the GNSS-equipped autopilot. These pose priors are then refined on the on-board computer using re-projection factors from the optical camera. The experiments demonstrate accurate relative pose estimation that can be used for virtual stereo pair generation. In particular planar rectification [91] with subsequent block matching along the epipolar lines returned high-quality depth maps. However, the major caveat of this method remains: the quality of the depth maps is determined by the flight path.

Deep learning methods attempt to overcome these shortcomings [71, 258, 262, 266]. Instead, and supported by [250] that positions in favor of stereo vision, we presented a multi-IMU multi-camera system that utilizes the available wide baseline to geometrically reduce the depth uncertainty (cf. Paper V). In contrast to end-to-end DL-based depth estimation methods our approach enhances classical geometric considerations with learning-based image alignment (Paper VI).

[1] Oblique defines the camera mounting angle in between down-looking (nadir) and front-looking.

With the development of miniaturized sensors that are able to provide long-range and sufficient resolution, we see a potential to fuse different sensor modalities:

Fusion of Additional Depth Sensors The integration of Radio Detection And Ranging (RADAR) and LiDAR systems into small-scale rotary-wing or fixed-wing UAVs has been problematic due to tight constraints on power consumption, payload weight, dimensions, budget, and required range and resolution. However, the development of new and miniaturized sensors promise to make the integration of RADAR and LiDAR feasible. Once this technology is available, a fusion of the optical depth map with the RADAR or LiDAR point cloud would be beneficial. While RADAR and LiDAR depth estimates are expected to have a lower depth uncertainty, the optical cameras are better suited for specific materials (e.g., water), provide texture, high resolution, and a wide field of view.

(Near-)Infrared cameras For night flights or flights during dawn or dusk, the visible light cameras could be replaced with Near-Infrared (NIR) or LWIR cameras.

3.3 Part C: UAV Missions

Part C, the final part of this dissertation, show-cases three UAV missions: Landing site detection (Paper VII), human detection (Paper VIII), and collaborative scene reconstruction (Paper VIII).

Paper VII introduced our framework capable of finding a suitable landing spot in previously unexplored environments. The algorithm is subdivided into a rough 2D and a more computationally expensive 3D evaluation step. The vision front-end segments and classifies the potential landing spot and analyses it based on texture, shape, and area. In a second step, 3D metrics such as slope, terrain roughness, and other hazards are taken into account. The approach path is computed based on the estimated wind field, obstacles in the vicinity of the path, and fixed-wing UAV dynamics.

DL-based Semantic Segmentation In Paper VII, large homogeneous regions are segmented by subsequently applying a Canny Edge detector [42] and a thresholded Distance Transform [76]. The segmented regions are then classified into grass or not grass using a binary Random Forest (RF) classifier. While this is a relatively simple task, DL-based semantic segmentation could further increase the precision and recall rate for the extracted landing spot candidates and provide a better scene understanding.

Paper VIII presents our DL-based human detection system for UAVs that combines an optical camera (RGB) with a LWIR camera. The paper concludes that RetinaNet [171] and YOLOv3 [217] both outperformed the hand-crafted optical-infrared human detector with RetinaNet showing superior performance over YOLOv3. Along with the paper, we make the annotated optical-infrared datasets of realistic SaR missions publicly available.

Generic Object Detection The proposed framework in Paper VIII can readily be trained to detect other objects, including wildlife, poachers, vehicles, or ships. For all of these examples, the combination of optical and infrared camera would be beneficial to the overall detection performance.

Human Detection with Hyper-spectral Camera The experiments presented in Paper VIII demonstrate the beneficial complementary character of combining an optical with a LWIR camera. For instance, persons standing in the shadow are often well visible in the infrared spectrum due to the temperature difference. However, it is challenging to detect the human with the optical camera due to the limited dynamic range that becomes apparent in scenes that contain both shadows and strongly illuminated areas. Building up on these experiences, we propose to investigate the usage of a multi- or hyper-spectral camera. Depending on the camera type, this multi- or hyper-spectral camera could completely replace the optical and LWIR camera. The main advantages would be the following: Firstly, instead of only relying on the visible-light and infrared spectrum, a hyper-spectral camera gives access to hundreds or thousands of bands. With the help of DL, one could identify the combination of bands yielding the best detection performance. Secondly, in contrast to the optical-infrared setup, the hyper-spectral camera is one physical unit. Transformations between the cameras could be avoided and, depending on the imaging sensor and exposure mechanism, the multiple bands could be directly overlaid and passed to a neural network in form of a multi-channel image.

The final paper, Paper VIII, demonstrates the collaborative scene reconstruction with a fixed-wing UAV equipped with an optical camera and a rotary-wing UAV with a laser scanner. The sparser optical point cloud, obtained from a photogrammetric reconstruction, and the denser laser point cloud are fused with a point cloud registration algorithm. The experiments demonstrate the competitive performance of the probabilistic data association algorithm [7] compared to Iterative Closest Point (ICP) and other classical point cloud registration methods. Within the scope of this publication, we made the optical-laser datasets and the probabilistic data association algorithm, that can be best summarized as a probabilistic variant of the classical ICP algorithm, publicly available (cf. Sec. 2.6).

3.4 Concluding Remarks

We would like to conclude this chapter with two important remarks that apply to all parts described above:

Benchmark Datasets Benchmark datasets with accurate ground-truth are an essential factor to boost research in a particular area: Firstly, the datasets increase the comparability of different algorithms. Secondly, the datasets encourage more researchers to work on a specific topic, especially if they do not have

access to certain robotic platforms or sensor systems. Within the scope of this thesis, we made several datasets (cf. Sec. 2.6) publicly available to the community.

Synergistic Effects For publications, it is common practice to minimize the number and the scope of assumptions in an attempt to keep the contributions generic and usable by as many researchers as possible. However, combining the modules described in this dissertation can further increase the robot's scene understanding and autonomy. To give one example, the proposed non-rigid stereo system (Paper V and Paper VI) can be used for the landing site detection module (Paper VII), enabling a quicker evaluation of slope, terrain roughness, and nearby obstacles.

Part A

LOCALIZATION AND MAPPING

Paper

Monocular Visual-Inertial SLAM for Fixed-Wing UAVs Using Sliding Window Based Nonlinear Optimization

Timo Hinzmann, Thomas Schneider, Marcin Dymczyk, Andreas Schaffner, Simon Lynen, Roland Siegwart, Igor Gilitschenski

Abstract

Precise real-time information about the position and orientation of robotic platforms as well as locally consistent point-clouds are essential for control, navigation, and obstacle avoidance. For years, GPS has been the central source of navigational information in airborne applications, yet as we aim for robotic operations close to the terrain and urban environments, alternatives to GPS need to be found. Fusing data from cameras and inertial measurement units in a nonlinear recursive estimator has shown to allow precise estimation of 6-Degree-of-Freedom (DoF) motion without relying on GPS signals. While related methods have shown to work in lab conditions since several years, only recently real-world robotic applications using visual-inertial state estimation found wider adoption. Due to the computational constraints, and the required robustness and reliability, it remains a challenge to employ a visual-inertial navigation system in the field. This paper presents our tightly integrated system involving hardware and software efforts to provide an accurate visual-inertial navigation system for low-altitude fixed-wing unmanned aerial vehicles (UAVs) without relying on GPS or visual beacons. In particular, we present a sliding window based visual-inertial Simultaneous Localization and Mapping (SLAM) algorithm which provides real-time 6-DoF estimates for control. We demonstrate the performance on a small unmanned aerial vehicle and compare the estimated trajectory to a GPS based reference solution.

Published in:
International Symposium on Visual Computing (ISVC), 2016

1 Introduction And Related Work

Unmanned aerial vehicles, and in particular fixed-wing UAVs, are powerful agents in scenarios where a large area needs to be scanned in a short amount of time. In recent years the interest in employing UAVs for applications such as industrial inspection, surveillance as well as agricultural monitoring has drastically increased due to the low acquisition and maintenance cost when compared to conventional aircraft. Today, GPS still remains the central source of 3-degrees of freedom (DoF) navigational information for the majority of these applications. Its limitations in certain cases, however, have motivated the exploration of alternative sources of 3-DoF signals such as ultra-sonic, laser, cameras and beacons. Due to the low cost and weight of the sensor suite motion estimation based on data from cameras and inertial-measurement units (IMU) has become a common choice to obtain a metric 6-DoF estimate of the body motion in real-time. There exists a large body of robotic research focusing on probabilistic fusion of visual and inertial data over which the publications of Nerurkar et al. [191] and [161] give an excellent overview. All these approaches perform simultaneous localization and mapping by jointly minimizing errors from reprojecting triangulated 3D-landmarks and integrating the measurements from the IMU. The currently best performing algorithms employ either (nonlinear) optimization [161, 164, 191], filtering [114, 130, 165, 183, 186] or combinations of both [187].

Optimization based approaches potentially achieve higher accuracy due to the ability to limit linearization errors through repeated linearization of the inherently nonlinear problem. This is particularly relevant in visual-inertial navigation systems (VINS). Here the relinearization of the IMU based propagation limits errors caused by inaccurate linearization points of IMU biases. Nevertheless, solving the covariance form of the problem in a recursive estimator is a common choice due to the lower computational requirements. Only recently formulations which allow "online mapping" in inverse form have shown real-time performance [161, 191]. To retain real-time calculations these algorithms approximate the problem using one of the following approaches:

- Key-frame based concepts as proposed e.g. in [159, 191]. These approaches, however, require an involved marginalization strategy and the constraints imposed by integrating the IMU measurements can theoretically span an indefinite duration.

- Fixed-lag smoothing or sliding window approaches that consider time successive states within a fixed time interval in the optimization problem but *discard* measurements outside of the sliding window. Cf. discussion in e.g. [242] of how this leads to loss of information regarding the variable interaction.

- Incremental smoothing and mapping methods such as iSAM2 [141] that optimize over all robot states and landmarks, making use of all available correlations.

In this paper we combine the last two approaches such that poses and associated landmarks older than the constant smoother lag are *marginalized* to ensure a smooth pose estimate while keeping the problem size bounded. In contrast to [48], we only employ a short term sliding window in factor graph formulation. Furthermore, we use a tightly-coupled integration approach where the error originating from pre-integrated IMU measurements as well as the reprojection errors can be jointly optimized and thus all correlations within the optimization window are considered [86]. Our contributions lie in the modifications to increase robustness and to lower computational requirements which allow operation onboard a fixed-wing UAV. We furthermore discuss the hardware setup of our modular sensor pod and thereby demonstrate accurate SLAM on a platform with limited dimensions and payload capabilities.

2 Methodology

This section introduces the coordinate frames, states, problem statement as well as the proposed visual-inertial estimation framework.

2.1 Coordinate frames

We distinguish between the camera frame $\mathcal{F}_C$, the body frame $\mathcal{F}_B$ as well as the global frame $\mathcal{F}_G$. Furthermore, the body frame $\mathcal{F}_B$ is assumed to be identical to the IMU frame $\mathcal{F}_I$ as illustrated in Fig. 4.1.

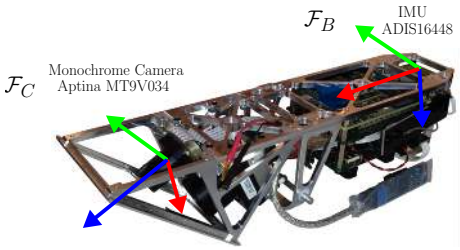


Figure 4.1: Camera and body coordinate frame of the sensor processing unit.

2.2 State vector

The SLAM problem seeks to estimate the robot states $\mathbf{x}_R$ as well as the set of landmarks $\mathbf{x}_L$ with $\theta = \{\mathbf{x}_R, \mathbf{x}_L\}$. The robot state is defined as

$$\mathbf{x_R} := \left[\mathbf{p}_B^{G^\top}, \mathbf{q}_B^{G^\top}, \mathbf{v}_B^{G^\top}, \mathbf{b}_g^\top, \mathbf{b}_a^\top\right]^\top \in \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^9 \tag{4.1}$$

and comprises the robot pose and velocity in the global frame as well as the gyroscope and accelerometer biases.

2.3 SLAM as maximum a posteriori (MAP) problem

We seek to maximize the joint probability $p(\mathbf{x}_R, \mathbf{x}_L, \mathbf{z})$ which is equivalent to minimizing the negative log-likelihood of the robot states $\mathbf{x}_R$ and set of landmarks $\mathbf{x}_L$ given the measurements $\mathbf{z}$:

$$\begin{aligned}\theta^* &:= \arg\max_\theta p(\mathbf{x}_R, \mathbf{x}_L|\mathbf{z}) = \arg\max_\theta p(\mathbf{x}_R, \mathbf{x}_L, \mathbf{z}) \\ &= \arg\min_\theta \{-\log p(\mathbf{x}_R, \mathbf{x}_L, \mathbf{z})\} = \arg\min_\theta \{-\mathcal{L}(\mathbf{x}_R, \mathbf{x}_L, \mathbf{z})\} \ ,\end{aligned} \tag{4.2}$$

where $\mathcal{L}$ represents the log-likelihood function.

2.4 Factor graph formulation

The SLAM problem can be transformed into a factor graph $\mathcal{G} = \{\mathcal{F}, \theta, \mathcal{E}\}$ which is a bipartite[1] graph consisting of a set of unknown random variables to be estimated θ and a set of factors $\mathcal{F}$. In our case, the variables θ consist of the robot states $\mathbf{x}_R$ and the set of landmarks $\mathbf{x}_L$. The factors $\mathcal{F}$ are given by prior knowledge or measurements $\mathbf{z}$. The edges connect variable and factor nodes. The factor graph uses a factorization of the function $f(\theta) = \prod_i f_i(\theta_i)$ which we seek to maximize, i.e. we want to calculate

$$\begin{aligned}\theta^* &= \arg\max_\theta \{f(\theta)\} = \arg\min_\theta \{-\log f(\theta)\} \\ &= \arg\min_\theta \left\{-\log \prod_i \exp\left(-\frac{1}{2}||z_i - h_i(\theta_i)||^2_{\Sigma_i}\right)\right\} \\ &= \arg\min_\theta \left\{\sum_{i=1}^{I} ||z_i - h_i(\theta_i)||^2_{\Sigma_i}\right\} \\ &= \arg\min_\theta \left\{\sum_{i=1}^{I} \mathbf{e}_i^T \Sigma_i^{-1} \mathbf{e}_i\right\} = \arg\min_\theta \left\{\sum_{i=1}^{I} \mathbf{e}_i^T W_i \mathbf{e}_i\right\} \ ,\end{aligned} \tag{4.3}$$

where we used the fact that the measurement covariance matrix Σ is the inverse of the information matrix W and the Mahalanobis distance is defined as $||\mathbf{e}||^2_\Sigma := \mathbf{e}^T \Sigma^{-1} \mathbf{e}$.

[1] I.e. every value node is *always* connected to one or multiple factor nodes and vice versa [108].

Factor and error term definitions

Only the factors and error terms are presented in this section. For the derivation of the Jacobians and information matrices we refer to [85, 157, 161].

Reprojection factor We define the reprojection factor as

$$f_{reproj}(\mathbf{x}_R, \mathbf{x}_L) = \mathbf{d}(\mathbf{e}_{reproj}) \propto \exp\left(-\frac{1}{2}||\mathbf{e}_{reproj}||^2_{\mathbf{W}^{-1}_{reproj}}\right) , \tag{4.4}$$

where $\mathbf{d}(\cdot)$ denotes a cost function, $\mathbf{e}_{reproj}$ is the reprojection error and $\mathbf{W}_{reproj}$ is the corresponding information matrix. We adopt the reprojection error formulation from [85]

$$\mathbf{e}^{i,j,k}_{reproj}(\mathbf{x}_r, \mathbf{x}_l, \mathbf{z}) = \mathbf{z}^{i,j,k} - \mathbf{h}_i(\mathbf{T}^{C_i}_{B_k}\mathbf{T}^{B_k}_{G}\mathbf{l}^G_j) \in \mathbb{R}^2 , \tag{4.5}$$

where $\mathbf{z}^{i,j,k}$ is the observation of landmark j in camera i at camera frame k in image coordinates and $\mathbf{h}_i(\cdot)$ is the camera projection model. The transformation from the body to the camera frames $\mathbf{T}^C_{B_k}$ was obtained by offline calibration and we assume a rigid sensor rig, i.e. $\mathbf{T}^C_{B_k} \equiv \mathbf{T}^C_B$.

IMU factor The IMU measurements are pre-integrated and summarized in a single relative motion constraint connecting two time-consecutive poses as described in [43, 81]. The factor evaluates the residuals and Jacobians for the pose, velocity and IMU biases of the previous and current state.

2.5 Sliding window estimator (SWE)

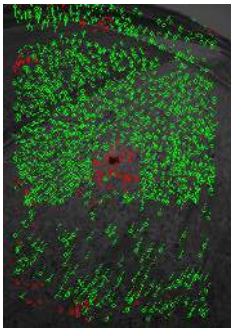
The outline of the sliding window estimator is presented in Algorithm 1. The most relevant parts of the framework include feature extraction, landmark initialization and graph optimization: Feature tracks are generated using a Lucas-Kanade tracker with gyroscope-based feature prediction for speed-up and feature bucketing as well as two-point translation-only RANSAC outlier rejection as visualized in Fig. 4.2a.

Landmark initialization

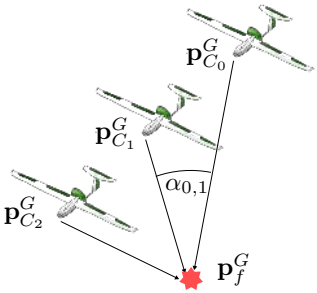
Only well constrained features are used as potential landmarks. We define the quality of a landmark observation by the number of observations, the minimum and maximum distance and by the angle between the incident rays:

$$\begin{aligned}
d_{min} &= \min(d_{min}, ||\mathbf{p}^G_{C_i} - \mathbf{p}^G_f||_2), \ i \in [0, M) \\
d_{max} &= \max(d_{max}, ||\mathbf{p}^G_{C_i} - \mathbf{p}^G_f||_2), \ i \in [0, M) \\
\alpha_{min} &= \min(\alpha_{min}, \cos^{-1}(||\mathbf{p}^G_{C_i} - \mathbf{p}^G_f||_2 \cdot ||\mathbf{p}^G_{C_j} - \mathbf{p}^G_f||_2))
\end{aligned}$$

with $i \in [0, M)$, $j \in [i+1, M)$ and where M represents the number of landmark observations. The basic idea as well as the used heuristics during flight are presented in Fig. 4.2b.



(a) The image visualizes the output of the vision front-end: Inlier feature tracks are shown in green. Tracks classified as outliers by RANSAC are visualized in red - here mainly due to the UAV's shadow violating the static feature assumption.



(b) Definition of well-constrained landmarks illustrated by the incident rays of three landmark observations. The following heuristics are used for the UAV SKATE datasets: $M_{min} = 30$, $d_{min} = 0.1\,\text{m}$, $d_{max} = 100\,\text{m}$, $\alpha_{min} = 0.5\,\text{deg}$.

Figure 4.2: Vision front-end: Feature tracking and landmark initialization.

After triangulating the feature tracks the estimated landmark locations are validated by back-projecting the landmarks in the frames from which they were observed. The landmark is only inserted as a state if it lies in the visible camera cone. That is, e.g. landmarks that are triangulated behind the cameras are rejected.

Graph optimization

Every factor and value node is associated with a marginalization strategy:

- Robot states: Marginalized after the graph update if the corresponding value node falls outside the sliding window.

- Landmarks: Landmark nodes are marginalized before the graph update if

  1. *opportunistic feature:* the feature track has been terminated in the previous camera frame due to the temporal condition or the feature has not been re-detected. Opportunistic features ensure that every frame is connected to *sufficient* reprojection factors.

2. *persistent feature:* the feature has not been re-detected in the previous camera frame. Persistent features are included in the factor graph to reduce temporal drift.

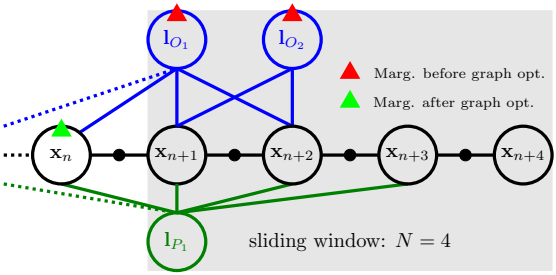


Figure 4.3: Marginalization strategy: At step $k = n+3$ the VI-node is propagated and the graph is augmented with the new state $\mathbf{x}_{n+4}$. Due to the temporal condition (TC) the value node $\mathbf{x}_n$ is to be marginalized. Also, the opportunistic landmark nodes $\mathbf{l}_{O_1}$ (TC) and $\mathbf{l}_{O_2}$ (not re-detected) are to be marginalized.

A toy example of the marginalization strategy with a sliding window of $N = 4$ frames is presented in Fig. 4.3. In order to marginalize variable nodes the factors associated with these variables are identified and removed. The marginalized factors are replaced by the linearized factors which corresponds to a transformation into a Bayes tree. Before marginalization, the variables in the graph need to be reordered to preserve sparsity: For this purpose we use the column approximate minimum degree heuristic (COLAMD) [54]. The graph is linearized using Cholesky factorization and optimized using Levenberg-Marquardt. For marginalization and optimization of the factor graph we use the *Georgia Tech Smoothing and Mapping* (GTSAM) [58] framework.

Estimator initialization

For coarse gravity alignment, the accelerometer readings $\mathbf{a}^B$ are averaged over a short static period before take-off. The calculations are performed in axis-angle notation:

$$(axis, angle)_B^G = (-W_g \times \|\bar{\mathbf{a}}^B\|, \cos^{-1}(W_g^T \|\bar{\mathbf{a}}^B\|)) \tag{4.6}$$

with $W_g = \begin{bmatrix}0.0 & 0.0 & 1.0\end{bmatrix}^T$. As noted in [86], a purely visual SLAM problem has six degrees of freedom (DoF). The visual-inertial problem has only four DoF since gravity renders two additional DoF observable (around world x- and y-axis). Based on these calculations the initial transformation $T_{B_0}^G$ is estimated and the estimator switches to the routine described in Algorithm 1.

Algorithm 1 Estimator pipeline (running)

1: Extract opportunistic and persistent feature tracks:
   - Lucas-Kanade feature tracker
   - Gyroscope measurement integration for feature prediction
   - Feature bucketing
   - Two-point RANSAC outlier rejection
2: Initialize landmarks:
   - Check if landmark is well constrained
   - Triangulate feature tracks
   - Check if triangulated landmark location is in visible camera cone
3: Add landmarks to factor graph (Fig. 4.4, b)
4: Detect stationary phases:
   - Add velocity priors if stationary phase detected (take-off, landing)
5: Update factor graph
   - Apply marginalization strategy
   - Linearize and optimize factor graph
   - Detect and remove outliers
6: Pre-integrate the IMU measurements and propagate the current VI-node
7: Augment the factor graph with the propagated VI-node (Fig. 4.4, c)
8: Add the IMU factor to the factor graph (Fig. 4.4, c)

3 Experimental Results

In this section we outline the hardware setup and present the results generated by the real-time visual-inertial navigation framework.

3.1 Sensor and processing unit

The sensor and processing unit used for the experiment is shown in Fig. 4.1: It is equipped with an Aptina MT9V034 grayscale global shutter camera which is able to record images at up to 60 fps with a resolution of 752×480 pixels. The MEMS inertial measurement unit (IMU) ADIS 16448 measures angular velocities as well as linear accelerations. The camera and IMU are integrated into an ARM-FPGA-based Visual-Inertial (VI) sensor system [195] that time-synchronizes IMU and camera on a hardware level. The Intel Atom CPU has access to the synchronized visual-inertial data stream. All components are mounted on an aluminium frame that guarantees a rigid camera-IMU transformation throughout the flight. For more details we refer to [196]. The camera intrinsics (i.e. focal length and principal point) as well as extrinsics (i.e. camera-IMU transformation T_C^B) are estimated offline using the calibration tool *Kalibr* [86].

3.2 Platform: Aurora SKATE

For demonstrating visual-inertial navigation, the sensor & processing unit is mounted on the small fixed-wing UAV SKATE. The light flying wing shown in Fig. 4.5 was manufactured by *Aurora*, features two independently articulated propulsion

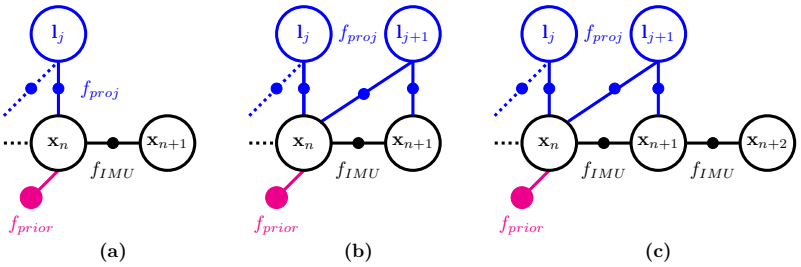


Figure 4.4: Factor and value node insertion into the factor graph. **(a)** Initial setup at $k = n + 1$. **(b)** Insertion of landmark values and reprojection factors. **(c)** Pre-integration of IMU measurements and graph augmentation.

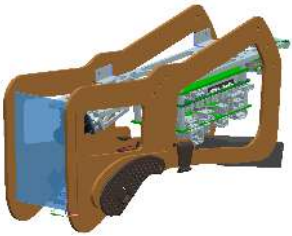
units and is controlled by an elevator control surface. Using thrust vectoring it is able to rapidly transition between vertical and horizontal flight and thus provides high agility and maneuverability. Different control modes (auto, manual) can be set by the hand-held remote control and allow for easy and safe operation of the vehicle. Note that the presented flight was performed in manual mode. The original processing unit shown in Fig. 4.5 is replaced by an interface specifically designed for this carrier. The interface mounts the sensor pod safely, powers it, is rigid and easily removable at the same time.

3.3 Simultaneous state estimation and sparse map generation

This section presents the state estimates and sparse point-cloud generated by the proposed nonlinear estimation framework based on datasets recorded by the small unmanned aerial vehicles SKATE. The test site is characterized by flat terrain with few three-dimensional structure which makes the underlying scale estimation challenging.

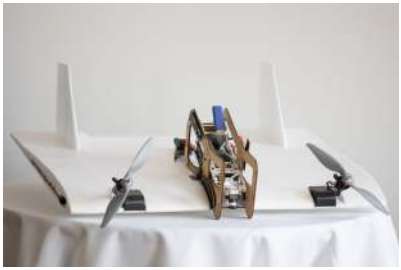
Fig. 5.4 shows the state estimates of the robot's position, velocity and IMU biases. The estimated altitude after the vehicle has landed is used as a first validation method. For this flight we registered a final altitude of around 3.1 m which demonstrates the limited translational drift. Fig. 4.7 represents the sparse point-cloud and the estimated trajectory flown by the fixed-wing UAV. In particular take-off and landing spot show a high point density which enables automatic take-off and landing procedures. To obtain a further quantitative accuracy measurement, an error is defined as the Euclidean distance to the GPS position obtained from a uBlox LEA-6H sensor. For the complete trajectory, from take-off to landing, the RMSE amounts to 14.6 m with respect to the aligned GPS based reference trajectory. Both trajectories are shown in Fig. 12.16: Especially at low altitudes, the GPS measurements show large jumps due to multi-path effects while our proposed visual-

(a) SKATE by Aurora in factory configuration. The head piece contains batteries and two down-looking cameras.



(b) CAD model of sensor & processing unit as well as its power interface to SKATE. The sensors consist of an IMU and a gray-scale camera.

(c) Sensor & processing unit mounted on SKATE. The measurements from the GPS receiver are used to generate a reference trajectory.



(d) Sensor & processing unit mounted on SKATE.

Figure 4.5: Hardware modifications made to SKATE to allow for accurate and time-synchronized visual-inertial localization and mapping.

inertial navigation system produces smooth estimates, including the take-off and landing phase. The runtime evaluated for the SKATE mission is shown in Table 10.1 and was performed on an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz. The profiling indicates that a camera stream of 23 Hz can be processed by the estimator in real-time.

4 Conclusion and Future Work

This paper presented an inverse-form visual-inertial navigation system that fuses inertial measurements and visual cues into a sliding window estimator applied to a small fixed-wing unmanned aerial vehicle. The state estimates of the robot's pose, velocity, IMU biases, and landmarks which were generated from a flight are

presented and the trajectory is compared to a GPS based reference solution. The

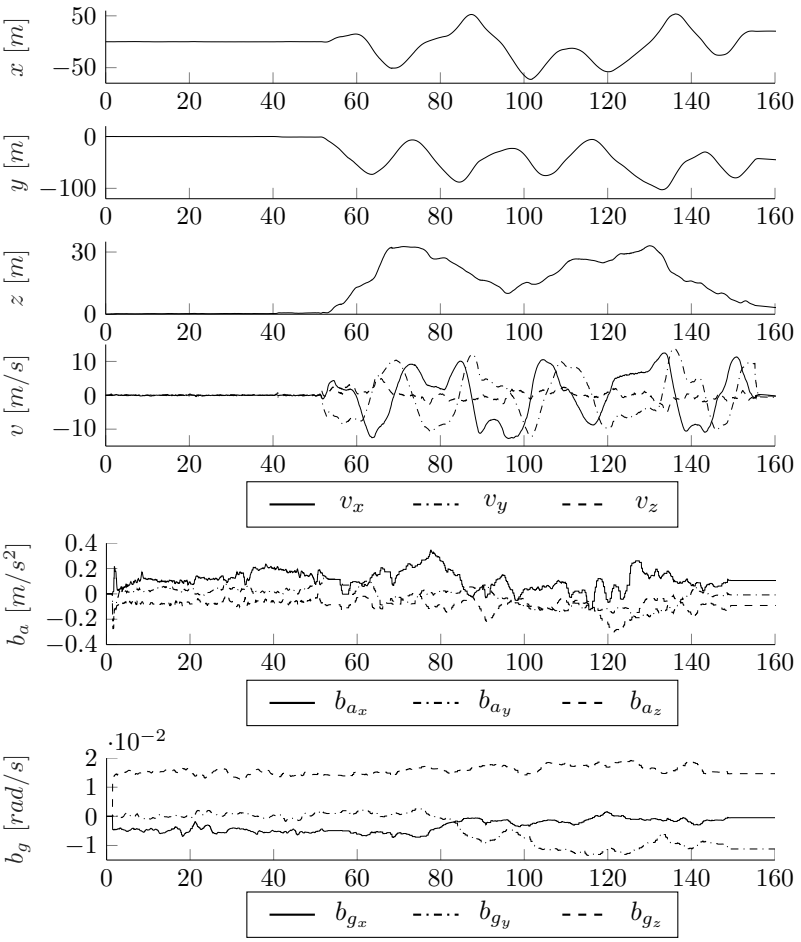
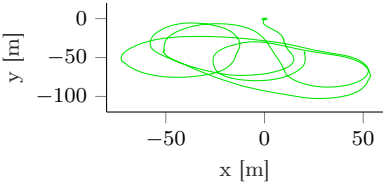
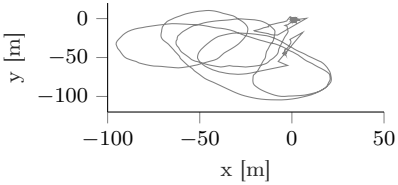


Figure 4.6: State estimates of robot's position, velocity, accelerometer as well as gyroscope biases.

Figure 4.7: Side-view of the generated sparse point-cloud and the estimated trajectory flown by the UAV. The right figure shows take-off and landing spot with increased point-cloud density.



(a) Top-view of the estimated trajectory in local vision frame. The position is set to zero during initialization. The trajectory shows smooth estimates during take-off and landing phase.



(b) Top-view of the GPS trajectory in local UTM coordinates where the first GPS measurement defines the origin. Especially at low altitudes, the GPS measurements show large jumps due to multi-path effects.

Figure 4.8: Comparison of estimated trajectory and GPS based reference trajectory.

	mean $[ms]$	std. dev. $[ms]$
Feature tracking*	19.976	7.267
Landmark initialization	0.269	0.113
State augmentation	3.455	2.497
Variable reordering (COLAMD)	0.207	0.079
Marginalization	4.321	0.935
Graph optimization	27.370	17.325
Frame processing	42.249	21.805

Table 4.1: Runtime in ms. for one frame. (*Runs in a separate thread and does not count towards total frame processing time.)

experiments showed that the locally consistent sparse point-cloud can be employed for static obstacle avoidance in terms of automatic take-off and landing for fixed-wing UAVs. Future work in the context of visual-inertial SLAM for fixed-wing UAVs will comprise further drift reduction by means of vanishing point detection,

horizon tracking and loop closure detection. Furthermore, the experiments showed that the accuracy of the reference trajectory needs to be enhanced, e.g. by using DGPS, to allow for a more expressive quantitative evaluation.

5 Acknowledgements

The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°600958 (SHERPA) and was sponsored by Aurora Flight Sciences. The authors thank Thomas Schneider and Marcin Dymczyk for the support regarding the software implementation.

Paper II

Mapping on the Fly: Real-Time 3D Dense Reconstruction, Digital Surface Map and Incremental Orthomosaic Generation for Unmanned Aerial Vehicles

Timo Hinzmann, Johannes L. Schönberger, Marc Pollefeys, Roland Siegwart

Abstract

The reduced operational cost and increased robustness of unmanned aerial vehicles has made them a ubiquitous tool in the commercial, industrial and scientific sector. Especially the ability to map and surveil a large area in a short amount of time makes them interesting for various applications. Generating a map in real-time is essential for first response teams in disaster scenarios such as, e.g., earthquakes, floods, or avalanches or may help other UAVs to localize without the need of Global Navigation Satellite Systems. For this application, we implemented a mapping framework that incrementally generates a dense georeferenced 3D point cloud, a digital surface model, and an orthomosaic and we support our design choices with respect to computational costs and its performance in diverse terrain. For accurate estimation of the camera poses, we employ a cost-efficient sensor setup consisting of a monocular visual-inertial camera rig as well as a Global Positioning System receiver, which we fuse using an incremental smoothing algorithm. We validate our mapping framework on a synthetic dataset embedded in a hardware-in-the-loop environment and in a real-world experiment using a fixed-wing UAV. Finally, we show that our framework outperforms existing orthomosaic generation methods by an order of magnitude in terms of timing, making real-time reconstruction and orthomosaic generation feasible onboard of unmanned aerial vehicles.

Published in:
Field and Service Robotics (FSR), 2018

1 Introduction

A fast and precise overview of an area is important for first aid teams in disaster scenarios such as earthquakes, floods, or avalanches. In particular, digital surface models (DSM) and orthomosaics are essential tools to support the human operator in quick decision-making. An orthomosaic gives a broad overview of the surroundings and helps the human operator to find regions of interest. Furthermore, orthomosaics enable every agent with a camera to infer its own absolute pose by employing feature extraction or image matching. The orthomosaic can therefore be used to localize the robot and other unmanned aerial vehicles (UAVs) while solely relying on an image stream [280]. An orthomosaic image is obtained by correcting aerial images for perspective and camera distortion using the information about the camera intrinsics and camera poses such that the generated image is true to scale and corresponds to a map projection throughout the image. The task of true orthorectification requires a three-dimensional model of the scenery. This is necessary in order to appropriately map intensities observed by the perspective camera to their location with respect to the orthographic camera. The DSM represents the three-dimensional model in form of a height map and furthermore helps to detect changes in elevation or to plan robot or human missions. The literature distinguishes between a DSM and a digital terrain model (DTM). The DSM includes the earth's surface and all objects such as buildings and trees on top of it. In contrast, the DTM models the bare earth's surface. In this publication, we are only interested in generating DSMs.

2 Related Work

The literature for creating overview images can be roughly categorized into panorama and mosaic generation where we utilize the distinction from [278, p. 12]: "Panorama is an extension of field of view (FOV) while mosaic is an extension of point of view (POV)". The mosaic generation can be divided into forward projection, using e.g. homographies or dense point clouds, and backward projection, using e.g. ray tracing in combination with grids or triangle meshes. An overview of the categories is given in Fig. 5.1. In this publication, we describe and compare a homography-based and point cloud-based forward projection, as well as a batch, and incremental grid-based backward projection approach by analyzing the advantages and disadvantages in particular with respect to their real-time capabilities. All of the approaches above are incorporated in our end-to-end mapping framework (cf. Fig. 11.1) that tightly couples IMU odometry, GPS position and visual cues in a smoothing-based estimator and thus does not detach state estimation from orthomosaic generation. In summary, we claim the following contributions:

- A real-time incremental end-to-end dense reconstruction and orthomosaic generation framework for UAVs that tightly couples state estimation and seamless mosaic generation.
- Most importantly, we propose an incremental grid-based orthomosaic generation algorithm that is suitable for real-time applications in arbirary terrain

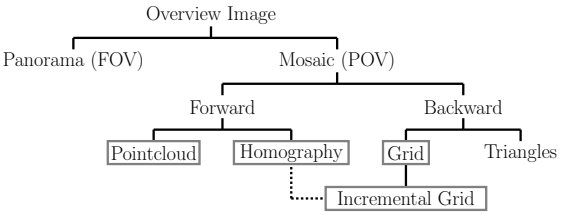


Figure 5.1: Categories for generating an overview image. In this paper, we analyze a homography-based and point cloud-based forward projection, as well as a batch and incremental grid-based backward projection approach.

by considering the surface model and best viewing angle. We validate its performance on a synthetic and real-world dataset with respect to homography-based, point cloud-based, and batch alternatives.

- We open-source our framework `aerial_mapper` consisting of all described DSM and orthomosaic generation approaches. Our framework augments the efficient and versatile `grid_map` library [74] with utilities for georeferenced mapping from aerial views.

2.1 Panorama Generation

Many approaches exist to generate a panoramic view given a set of images by applying a homography. Brown et al. presents in [38] an approach to robustly stitch a set of unordered images to a seamless panorama assuming rotations only around the optical axis. The main steps consist of feature extraction, matching in feature space, applying RANSAC and then computing the homography and applying bundle adjustment. Steedly et al. [253] build up on [38] and predict overlapping images more efficiently by utilizing the fact that the video stream is not unordered. Agarwala et al. [8] generate a multi-viewpoint panorama of a street using a homography and Markov Random Field (MRF) optimization. Laganière et al. [153] use homographies to generate bird-eye views for teleoperation of a robot. All of the approaches have in common that they focus on obtaining seamless and visually appealing panoramas or bird-eye views and are not concerned about georeferencing or georeferencing errors. However, stitching using only feature correspondences leads to error accumulation and distorted maps when directly applied to UAVs as demonstrated e.g. in [277, p. 20]. The same is true when only the first image is georeferenced and the subsequent images are incrementally stitched to this reference image.

2.2 Mosaic Generation

Forward Projection

In UAV applications, where we are rather interested in generating a seamless and georeferenced mosaic, additional sensor measurements are used to obtain camera pose measurements or estimates: Hemerly et al. [111] describe the process of obtaining a single georeferenced image using a UAV. Olawale et al. [201] recover the camera intrinsics and extrinsics using GPS and manually collected ground control points in combination with the commercial photogrammetric software (*Agisoft*) and generate an orthomosaic. Yahyanejad et al. present in [277, 278] the results of homography-based image mosaicing from sensor data recorded on board of a rotary-wing UAV with a down-looking camera.

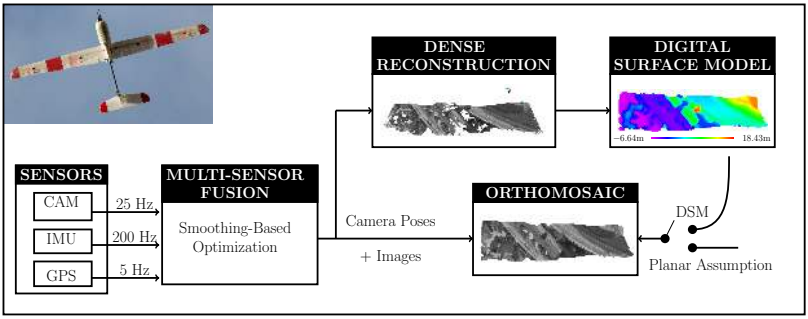


Figure 5.2: System overview: The IMU, camera, and GPS measurements are fused in a smoothing-based optimizer. The optimized camera poses and images are used as input for the dense reconstruction and orthomosaic generation. The DSM is updated incrementally from the 3D dense georeferenced point cloud. The orthomosaic is computed via an incremental backward grid-based approach while employing the DSM or a planar assumption and considering the optimal viewing angle.

As presented, many approaches use a camera pose estimate and an image as input and then apply a robust but costly feature detection and matching algorithm. For instance, [277, 278] assume noisy IMU and GPS measurements and deal with this by designing a quality function that finds a trade-off between geo-referencing error and seamless stitching. In contrast, we do not detach state estimation and orthomosaic generation but fuse GPS and IMU measurements as well as feature tracks in a consistent smoothing-based state estimation. The small offset between images in combination with gyroscope measurements enables fast and subpixel-accurate Lucas-Kanade feature tracking (KLT) [177]. The use of KLT is also supported by the findings in [14] claiming that KLT achieves the best quantitative results

in the context of orthomosaic generation. Our philosophy is that accurate and efficient estimation of the camera poses is the backbone of consistent dense 3D reconstruction and seamless orthomosaic generation. Furthermore, the literature was previously not concerned about presenting runtime results and [14], [278] deplore lack of quantitative performance measures. We tackle this absence of information by presenting the runtime of all methods and an open-source Gazebo-based HIL environment [88] capable of generating synthetic datasets.

Backward Projection

Note that none of the homography-based forward projection approaches presented in the previous section employ a DSM as input. In contrast, in order to generate true orthomosaics, [194] employ triangle-based backprojection, also known as ray tracing, in combination with a DSM. Our backprojection approach is very similar to [194] but we utilize a grid of squares to simplify the raytracing process. Furthermore, we present a novel incremental grid-based orthomosaic generation approach to speed up the computation.

3 Methodology

The methodology section follows the data flow illustrated in Fig. 11.1: Sec. 3.1 presents the smoothing-based GPS-IMU-Vision fusion. Given the input images and corresponding optimized camera poses, a dense georeferenced point cloud can be generated using planar rectification, as demonstrated in Sec. 3.2. Sec. 3.3 presents how this dense point cloud can be used to generate a DSM by employing inverse distance weighting (IDW). Finally, Sec. 3.4 presents our approaches to generate an orthomosaic from a stream of images, optimized camera poses, and DSM using (a) forward projection and (b) backward projection.

3.1 Multi-Sensor Fusion

In this section, we present the core elements of our proposed multi-sensor fusion framework. We distinguish three coordinate systems: the global frame $\mathcal{F}_G$, the camera frame $\mathcal{F}_C$, and the body frame $\mathcal{F}_B$. To avoid unnecessary conversions due to the vision-based fusion, we choose the Universal Transverse Mercator (UTM) coordinate system where $\mathbf{p}_B^G$ expresses easting, northing, and elevation. We seek to estimate the robot states $\mathbf{x}_R$ as well as the set of landmarks $\mathbf{x}_L$. We define the robot state as: $\mathbf{x_R} := \begin{bmatrix}\mathbf{p}_B^G & \mathbf{q}_B^G & \mathbf{v}_B^G & \mathbf{b}_a & \mathbf{b}_g\end{bmatrix}$ where the orientation, position and velocity of the body frame expressed in global coordinates are denoted with $\mathbf{q}_B^G$, $\mathbf{p}_B^G$ and $\mathbf{v}_B^G$. The remaining state vector consists of accelerometer bias $\mathbf{b}_a$ and gyroscope bias $\mathbf{b}_g$.

Vision Front-End

FAST features [23] are extracted from every input image and tracked from frame to frame using KLT with subpixel refinement. To speed up the tracking process and to avoid outliers, we use the gyroscope of the IMU to predict the location of the pixel in the subsequent image. Furthermore, we employ feature bucketing to guarantee uniformly distributed features across the image for improved vision-based motion estimation.

Smoothing-Based State Estimation

For sensor fusion and pose estimation we use the incremental smoothing and mapping algorithm *iSAM2* [140]. For the employed reprojection residual, we refer to [163]. Every reprojection factor has a Cauchy M-Estimator associated with it to reduce the influence of outliers[1]. The IMU measurements are preintegrated and summarized in a single relative motion constraint connecting two time-consecutive poses as described in [83]. The residual and Jacobian of the GNSS position factor is calculated by "lifting" the residual: $\mathbf{e} = \tilde{\mathbf{t}}_B^G - \mathbf{t}_B^G$, $\frac{\partial \mathbf{e}}{\partial \delta t} = -\frac{\partial}{\partial \delta t}\left(\mathbf{t}_B^G + \mathbf{R}_B^G \delta t\right) = -\mathbf{R}_B^G$ where $\tilde{\mathbf{t}}_B^G$ is the measured position transformed to UTM coordinates [2]. All measurements are inserted into the factor graph once they become available. For every measurement, the factor graph is augmented by a state node. To estimate the initial position, orientation as well as accelerometer biases, at the beginning of every experiment, the plane is kept level for few seconds. During this time, the GPS position measurements are averaged to determine the initial position. The averaged accelerometer readings are used for coarse gravity alignment and bias estimation. After take-off is detected, the vision measurements are incorporated into the factor graph. Note that in this publication only open-loop SLAM was employed, i.e. no loop closures or inter-matches were included in the factor graph. Albeit we did not experience any inconsistencies in the generated dense reconstruction or orthomosaics we consider to integrate an online loop-closure or map-tracking module in future work to guarantee the global consistency of the map.

3.2 Dense Reconstruction

Given the optimized camera poses of our monocular camera rig, a virtual stereo-pair is generated using planar rectification [92]. The dense point cloud is then computed by applying efficient stereo block matching. Note that planar rectification assumes that the epipoles of a virtual stereo-pair are outside the field of view which is fulfilled for our fixed-wing UAV with down-looking camera due to the approximately fronto-parallel motion with respect to the ground.

[1] The Cauchy weight is $k^2/(k^2 + e^2)$, where e is the residual and k is a constant set to 3.0.

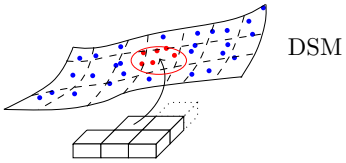
[2] Note that we neglect the translational offset between GNSS antenna and IMU since for our setup this corresponds to few centimeters.

3.3 Digital Surface Map Generation

The georeferenced dense point cloud serves as input for the digital surface map. The algorithm consists of a for-loop that iterates over all affected cells in the grid. A fast kd-tree[3] implementation returns the set of nearest points N found

Algorithm 2 Grid-Based DSM
p_r : Radius of the kd-tree used for interpolation.
λ : Factor to increase the interpolation radius.

```
1: function DSM(POINT CLOUD, CAMERA POSES)
2:     cells ←identifyAffectedCells(T_C^G)
3:     for c : cells do
4:         N ← {}
5:         while N == {} do
6:             p_r ← λ · p_r
7:             N ← kd-tree(x_c, y_c, p_r)
8:         end while
9:         Apply interpolation methods
10:        (Optional:) Height-to-color mapping
11:    end for
12: end function
```



within the interpolation radius p_r. Next, inverse distance weighting (IDW) is applied as interpolation method. IDW intuitively determines the cell's height by using a linearly weighted combination of the nearest neighbors, where the weight corresponds to the inverse distance to the cell center, thus giving higher weight to points that are closer to the cell center. An adaptive interpolation radius is utilized (cf. Alg. 2) that is guaranteed to return an interpolated height value in sparse regions and still keeps a high level of detail in dense regions.

3.4 (Ortho-)Mosaic Generation

In this section, we present the implemented approaches for computing an (ortho-)mosaic while focusing on the proposed incremental grid-based orthomosaic generation.

Homography-Based Mosaic (Forward Projection)

A perspective homography $\mathbf{H}$ is computed which relates the border pixel coordinates of the image to points on the ground surface. The homography is then applied to the undistorted input image and the transformed single rectified image is blended with the overall mosaic using feathering. The pseudo code of the algorithm and the results are presented in Alg. 3 and Fig. 5.5, respectively.

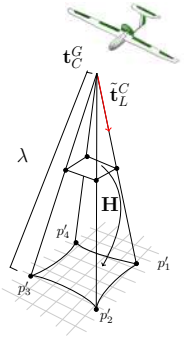
Grid-Based Orthomosaic (Backward Projection)

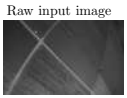
The grid-based orthomosaic generation in *batch* formulation iterates over all cells and, for every cell, queries the corresponding height from the DSM layer. An

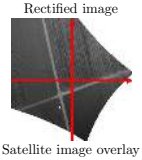
[3] *nanoflann*: nano fast library for approximate nearest neighbors.

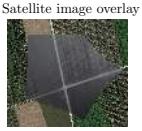
Algorithm 3 Homography-Based Mosaic
w : Image width in pixel. h : Image height in pixel.

1: **function** MosaicHomography(image, camera pose, camera intrinsics)
2: $p_1 \leftarrow (0,0)$, $p_2 \leftarrow (w,0)$, $p_3 \leftarrow (w,h)$,$p_4 \leftarrow (0,h)$
3: undistort(image)
4: // Obtain ground points.
5: **for** $i = 1:4$ **do**
6: // Computing the scale.
7: $\lambda_i \leftarrow -(z_C^G - h_{ground})/(\mathbf{R}_C^G \tilde{\mathbf{t}}_L^C)_z)$
8: // Computing the ground position [UTM].
9: $p_i' \leftarrow \mathbf{t}_C^G + \lambda_i \mathbf{R}_C^G \tilde{\mathbf{t}}_L^C$
10: **end for**
11: $\mathbf{H} \leftarrow$computeHomography$(\mathbf{p}, \mathbf{p}')$
12: $image_{transf.} \leftarrow$ applyHomography$(H, image)$
13: mosaic $\leftarrow$ iterativeBlending$(image_{transf.})$
14: **end function**









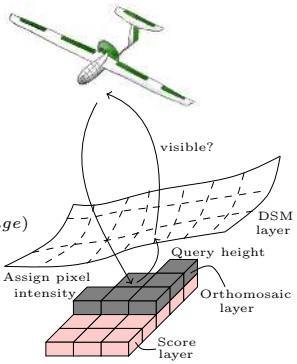
additional for-loop iterates over all images and, given the corresponding camera pose and camera intrinsics, checks if the cell is within the visible camera cone. Since every cell is usually observed from several camera frames, the question poses which is the ideal pixel intensity value to be assigned to the cell of the orthomosaic. Various mosaic strategies exist [194]. We propose to extract the pixel intensity from the image where the corresponding camera pose is the closest to nadir. This elevation angle is defined as the observation vector from the camera to the cell center. Instead of performing these operations on all cells in the grid, our proposed *incremental* formulation (Alg. 4) identifies the subset of cells that needs to be updated, as illustrated in green in Fig. 5.5. The cells are identified by projecting the border-pixels of the current image onto the plane defined by the minimal elevation obtained from the DSM (cf. Alg. 3). All cells which are potentially visible given the current camera pose are obtained by min/max operation. Depending on the image distortion, a tighter approximation could be achieved using e.g. the Bresenham algorithm [35]. Only those cells which show a higher elevation angle than currently stored and which are visible from the current camera configuration are updated.

Point Cloud-Based Orthomosaic (Forward Projection)

In contrast to the approach described in Section 3.4 one can directly use the dense 3D reconstruction of the environment (cf. Section 3.2) to generate an orthomosaic view and hence avoid the costly backprojection step. The proposed point cloud-based orthomosaic generation approach closely follows Alg. 3 but instead of the height we compute the IDW of the pixel intensity.

Algorithm 4 Incremental Grid-Based Orthom.

```
1: function IncrementalOrthomosaicGrid(image, camera pose, camera intrinsics)
2:     cells ←identifyAffectedCells(T_C^G)
3:     for c : cells do
4:         z_c ← dsm(c)
5:         if visibility(x_c, y_c, z_c, T_C^G) then
6:             score ← computeScore(x_c, y_c, z_c, T_C^G)
7:             if score > score(c) then
8:                 (u, v) ← backproject(x_c, y_c, z_c, T_C^G, image)
9:                 ortho(c) ←pixelIntensity(u, v, image)
10:            end if
11:        end if
12:    end for
13: end function
```

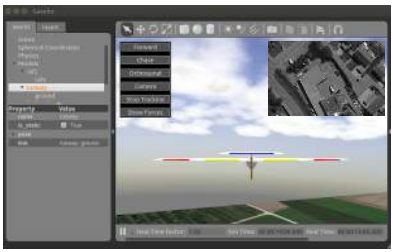


4 Platform And Sensors

For our experiments, we use *Techpod* (cf. Fig. 11.1), a small unmanned research plane with a wingspan of 2.60 m. The IMU *ADIS16448*, and the grayscale camera *Aptina MT9V034* of the sensor pod are hardware-synchronized using a VI-Sensor [195] and run at 200 Hz and 25 Hz, respectively. The camera *Aptina MT9V034* has a focal length of 2.8 mm, a sensor diagonal of 1/3 inch, and a resolution of 752×480 pixels. The *ublox LEA-6H* GPS receiver and the pressure sensors are connected to the *Pixhawk* autopilot running a real-time EKF [160].

5 Simulation Experiments

The Gazebo-based HIL environment was used to validate the DSM and orthomosaic generation is illustrated in Fig. 5.3. The aerodynamic coefficients and mounted sensors closely model our UAV *Techpod*. Fig. 5.6 shows the results from a simulated single scan line at a relatively low altitude of 50 m above the mesh of *Pix4D*'s *cadastre* [2] dataset. For this experiment, 429 images are rendered at a frame rate of 20 Hz and each image is associated with the ground truth pose. Fig. 5.6(a) shows the coordinate system of the last camera pose and the point cloud generated by the planar rectification algorithm using every 10 th image. In this experiment, we deliberately do not use every frame for the dense reconstruction to underline the framework's potential to handle sparse regions or holes in the point cloud. The DSM layer, which is given in Fig. 5.6(b), is generated by applying IDW with an initial radius of 5 m to the dense point cloud. Fig. 5.6(c) depicts the incremental grid-based orthomosaic in which the pixel intensity is queried from the first camera



(a) Fixed-wing UAV and synthetic image.

(b) Output of *QGroundControl* in HIL mode.

Figure 5.3: Gazebo-based HIL environment for fixed-wing UAVs

that is in line of sight of the respective cell. In contrast, Fig. 5.6(d) shows the incremental grid-based orthomosaic where the pixel intensities are obtained from the camera frame with the view closest to nadir. The corresponding elevation angles between selected camera pose and orthomosaic cell are shown in Fig.5.6(f), (g). In particular in regions with a small altitude to terrain height ratio (e.g. tree in center) one can observe that the nadir-view approach renders an improved orthorectified view and avoids double object mapping. As Fig.5.6(e) illustrates, the result of our nadir-view approach is in accordance with the orthomosaic generated by *Pix4D*, for which we used the same georeferenced images as input. The homography approach is not shown since the underlying flat plane assumption results in the predicted large orthomosaic distortions.

6 Real-World Experiments

In this section, we present the results obtained from the semi-autonomous flight at an altitude of 100 m above ground (cf. Fig. 5.4). The dense point cloud and orthomosaics are presented in Fig. 5.5. In contrast to the previous experiment, we present the output of the incremental grid based on a flat DSM. Due to the high altitude to terrain height ratio in this experiment, the assumption does not introduce measurable orthomosaic inconsistencies with respect to *Google* imagery (cf. Fig. 5.5b). The homography-based orthomosaic with applied feathering, shown in Fig. 5.5a, can handle an image stream of up to 57 Hz. Given the measured runtime in Table 10.1, the combination of dense reconstruction and point cloud-based orthomosaic is even slightly faster than the homography-based approach. The caveat of the former is that, due to image distortions at the outer regions, a smaller field of view will be covered by the virtual stereo pair (cf. Fig. 5.5c). Both the homography- and point cloud-based approach outperform the methods presented in [278] by an order of magnitude. Note that the variants proposed in [278] do

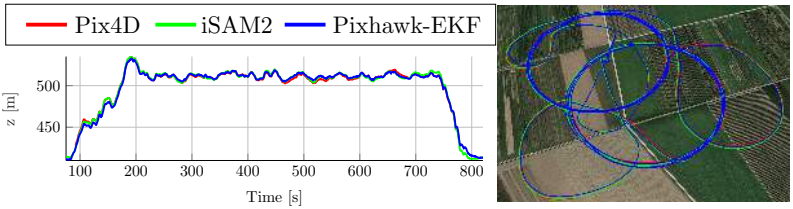
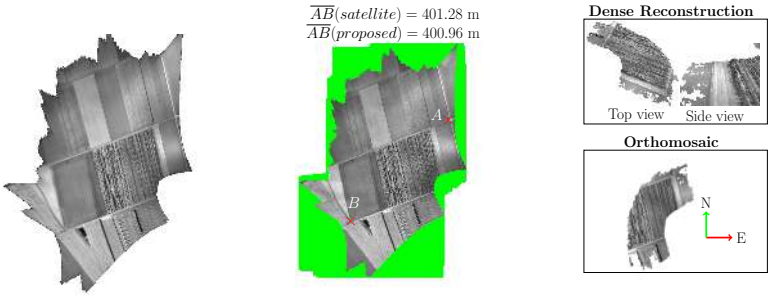


Figure 5.4: Comparison of *Pix4D*, *Pixhawk*-EKF [160] and *iSAM2*-based estimation.



(a) Homography-based orthomosaic (Section 3.4)

(b) Incr. grid orthomosaic (Section 3.4, first view)

(c) Point cloud-based orthomosaic (Section 3.4)

Figure 5.5: Mapping results based on the *iSAM2* state estimates using a fixed-wing UAV.

not generate a DSM and thus visual artifacts are introduced into the orthomosaic when the planar assumption is violated. The proposed incremental grid-based approach speeds up the computation by a factor of 10 compared to the batch variant. Both, the batch and incremental grid approach are several magnitudes faster than the triangle mesh implementation [194]. However, the implementation in [194] also performs color matching and identifies obscured pixels during the ray casting process adding up to a runtime of 62 minutes per image.

7 Conclusion

In this publication, we demonstrated that incremental end-to-end dense reconstruction and orthomosaic generation for UAVs is feasible in real-time allowing, for instance, advanced autonomous missions of UAV fleets by relying on orthomosaic-

	Time/image	Total time	# images	Resol.	CPU	Type	Impl.
Homography (Sec. 3.4)	17.4 ms	4.33 s	249	-	2.8 GHz	Forw.	C++
Dense rec. (Sec. 3.2)	16.7 ms	0.384 s	23/249	-	2.8 GHz	Forw.	C++
Point cloud (Sec. 3.4)	0.2 ms	0.51 s	249	10 m	2.8 GHz	Forw.	C++
Point cloud (Sec. 3.4)	0.79 ms	1.97 s	249	1 m	2.8 GHz	Forw.	C++
Point cloud (Sec. 3.4)	31 ms	7.72 s	249	0.1 m	2.8 GHz	Forw.	C++
"Position" [278]	0.47 s	17.31 s	37	n/a	2.66 GHz	Forw.	Matlab
"Pose" [278]	0.5 s	18.33 s	37	n/a	2.66 GHz	Forw.	Matlab
"Image" [278]	12.41 s	459.2 s	37	n/a	2.66 GHz	Forw.	Matlab
"Hybrid" [278]	3.68 s	136.28 s	37	n/a	2.66 GHz	Forw.	Matlab
Grid (batch) (Sec. 3.4)	0.43 s	107.41 s	249	10 m	2.8 GHz	Backw.	C++
Grid (batch) (Sec. 3.4)	1.73 s	430.27 s	249	1 m	2.8 GHz	Backw.	C++
Grid (incr.) (Sec. 3.4)	171 ms	42.6 s	249	1 m	2.8 GHz	Backw.	C++
Triangle Mesh [194]	62 min	620 min	10	0.15 m	2.8 GHz	Backw.	C#, Matl.

Implemented Proposed

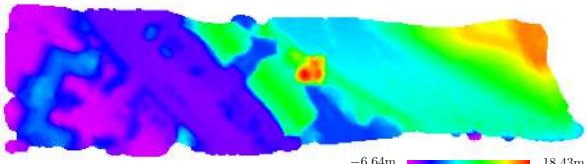
Table 5.1: Runtime results for dense reconstruction and orthomosaic generation.

based localization only. We highlight the characteristics of our implemented orthomosaic generation approaches in particular with respect to runtime and the influence of the flight altitude to terrain height ratio: The advantage of homography-based orthomosaic generation is the seamless blending, the fast computation and the optimal integration of all pixels but is only suited for planar scenery or, alternatively, high flight altitudes. The benefit of the point cloud-based orthomosaic is the lowest computation time among the evaluated methods, the seamless blending and the direct way of considering the surface elevation. However, depending on the dense reconstruction algorithm the area of coverage is smaller and sparse regions can only be overcome by interpolating nearby point intensities potentially leading to incorrect orthomosaics. Our proposed backward incremental grid-based orthomosaic is suited for arbitrary terrain, renders a true orthomosaic by considering the surface model and optimal viewing angle and still achieves real-time performance.

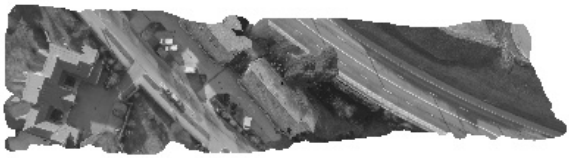
8 Acknowledgements

The research leading to these results has received funding from ArmaSuisse and the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°600958 (SHERPA). The authors thank Andreas Jäger and Sammy Omari for the implementation of the planar rectification algorithm, Lucas Pinto Teixeira for the synthetic image rendering pipeline, and Thomas J. Stastny for comments that greatly improved the publication.

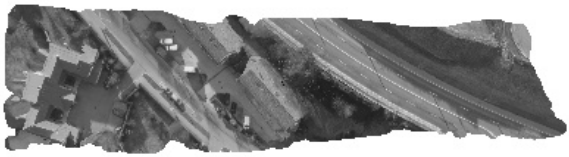
(a) Dense Reconstruction



(b) DSM (grid resolution: 0.5 m)

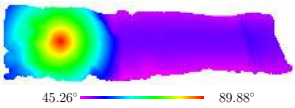


(c) Incremental Grid-Based Orthomosaic (First view; grid res.: 0.5 m)

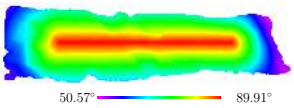


(d) Incremental Grid-Based Orthomosaic (Max. elevation angle; grid res.: 0.5 m)

(e) Pix4D Orthomosaic



(f) Observation angle (First view)



(g) Observation angle (Max. elevation angle)

Figure 5.6: Simulation results for dataset *cadastre* [2].

Paper III

Deep UAV Localization with Reference View Rendering

Timo Hinzmann and Roland Siegwart

Abstract

This paper presents a framework for the localization of Unmanned Aerial Vehicles in unstructured environments with the help of deep learning. A real-time rendering engine is introduced that produces optical and depth images given a six Degrees-of-Freedom (DoF) camera pose, camera model, geo-referenced orthoimage, and elevation map. The rendering engine is embedded into a learning-based six-DoF Inverse Compositional Lucas-Kanade (ICLK) algorithm that is able to robustly align the two images. To learn the alignment under environmental changes, the architecture is trained using maps spanning multiple years at high resolution. The evaluation shows that the deep 6DoF-ICLK algorithm outperforms its non-trainable counterparts by a large margin. To further support the research in this field, the real-time rendering engine and accompanying datasets are released along with this publication.

Published in:
CoRR, 2020

1 Introduction

GPS measurements are commonly fused in aerial navigation systems to guarantee long-term stable navigation. However, numerous reasons exist for not fully relying on GPS measurements, for instance, flights in naturally GPS-denied areas (e.g., mountainside), intentionally jammed GPS signals from external sources (e.g., by military), unintentionally jammed GPS signals by on-board sources (e.g., USB 3.0 [168]), or complete GPS receiver failures. On the other hand, the increasing availability of highly-accurate, high-resolution geo-referenced terrain mosaics, and elevation data allows its tight coupling into a visual-inertial navigation system for UAVs. The map data is standardized, stored efficiently in the form of images, and can either be acquired from satellite data or generated from geo-referenced images using commercial (e.g., Pix4Dmapper[1]) or open-source software [121]. Another advantage of localizing within the map, instead of relying purely on GPS, is that it ensures that the UAV pose and the map are consistent, which makes aerial navigation close to the static terrain safer. However, this raises the challenge of robust image registration under potential environmental and seasonal changes that we want to address in this paper.

2 Related work

The related work section is subdivided into the subproblems (a) image retrieval and (b) local image alignment. (a) and (b) are further divided into general localization and UAV specific localization methods. The remainder of the paper concentrates on the local image alignment, also referred to as map tracking algorithm. The map tracking algorithm requires a coarse pose initialization, which can be obtained from one of the image retrieval algorithms from Sec. 2.1.

2.1 Image Retrieval and Global Localization

Definition: Given a set of (geo-referenced) database images, retrieve the N best-matching database images for a query image.

Traditional image retrieval methods can be categorized in Bag-of-Words (BOW) and image voting approaches: FAB-MAP [51] is one of the most famous representant of the BOW approach. Proposed by Galvez-Lopez et al., dBoW2 [94] is built up from FAST [225] and BRIEF [40] features with focus on quick real-time query. It is used by the Visual-Inertial (VI) state estimator VINS-mono [213] for re-localization.

Many vertex or image voting approaches with different voting strategies exist: Probabilistic Vertex Voting (PVV) is a scoring scheme proposed by Gehrig et al. [100], that uses BRISK [158] descriptor projection, approximate k-Nearest Neighbors (kNN) search, and scoring based on a binomial distribution. The strategy significantly outperformed the baseline Active Search (AS) [232] in an aerial dataset. The 6-DoF pose is computed with the Random Sample Consensus (RANSAC) and

[1] https://www.pix4d.com/

Perspective-n-Point (PnP) algorithm based on the identified matches. Schönberger et al. proposed Vote-and-verify (VAV) [235], a voting scheme that, in contrast to pure bag-of-words approaches, features spatial verification but aims at being more efficient than standard RANSAC hypothesize-and-verify approaches while maintaining their accuracy. The image retrieval approach computes a vocabulary and visual words from training images. Database images are then used for indexing. For every query, the 6-DoF pose is computed with RANSAC-PnP. This image retrieval approach is part of the incremental structure-from-motion pipeline colmap [234]. Jegou et al. proposed the Vector of Locally Aggregated Descriptors (VLAD) [137] descriptor that uses local image features and produces a compact representation for global indexing. This approach constructs a visual dictionary from the descriptors, computes VLAD descriptors from the visual dictionary, and then creates an index for lookup. As a first Deep Neural Network (DNN)-based global image retrieval method, Arandjelovic et al. proposed NetVLAD [12], a trainable Convolutional Neural Network (CNN) architecture with VLAD layer as main component. Proposed by Sarlin et al., HF-Net [230] is a hierarchical network based on image retrieval with global descriptors and subsequent 6-DoF pose estimation with local features. The approach compresses the Superpoint [61] and NetVLAD layers using a knowledge distillation scheme. Shetty et al. [239] use a CNN-based scene localization network for image retrieval and a camera localization network for pose regression. However, this approach is not returning the full 6-DoF pose and results in relatively high position and orientation errors.

Several frameworks exist that explicitly address the challenge of environmental changes between query and database image, with the majority of works designed for automotive or, in general, ground-based applications: Bürki et al. [39] propose an appearance-based map management system that summarizes multiple missions and improves the query performance while limiting the accumulation of data. This method is well suited if a location is revisited multiple times with the same sensor setup. Other approaches make use of semantics to improve invariance to appearance changes: Lindsten et al. [172] estimate the geo-referenced 2D position of a UAV by applying environmental classification to both the terrain mosaic and UAV image stream and using subsequent rotation-invariant template matching for alignment. Yu et al. propose VLASE [281], the aggregation of semantic edges into a feature for subsequent VLAD description for ground vehicle localization. For ground-aerial localization, X-View [98] creates a multi-view graph of semantic observations but ignores the size and shape of the segments. Garg et al. introduce LoST [96], a local semantic tensor combining optical and semantic cues. Schönberger et al. [236] make use of a 3D semantic descriptor, assuming access to a depth image for every query image.

The majority of methods discussed above treat the queries independently and not as a video stream, with the following exceptions: Medina et al. [185] evaluate different strategies to aggregate image retrieval votes from a video stream to infer the camera's location but do not consider motion between the frames. X-View [98] performs graph matching between the query and database graph using Maximum Likelihood Estimation (MLE), neglecting the existence of potentially

multiple hypotheses. In contrast, Doherty et al. [62] detect objects of the same semantic class (e.g., cars) and models the multiple data association possibilities with graph-based multi-modal inference [84].

2.2 Local Image Alignment and Map Tracking

Definition: Given a query and a target image, compute the transformation that best aligns both images.

Hand-crafted features and descriptors have been used to match images with small appearance variations: Koch et al. [146] use SIFT [176] features with outlier handling to match nearby aerial images. A combination of correlation and SIFT-like feature matching is employed in [202] to match images recorded of the same terrain but on different days. Surber et al. [255] use weak GPS priors for initialization and direct BRISK matching for visual localization of a UAV in a previously flown mission. Mutual Information (MI) methods have shown to be more robust to appearance changes [204, 280], but come with higher computational costs due to the Hessian-based iterative optimization.

Higher-level features are employed, for instance, by Patterson et al. in [205]. Ordnance Survey (SO) layers with roads and building outlines are used as a reference map. During the flight, the UAV image stream is converted to the same format, and correlation-based matching is applied to locally align it to the reference map and to estimate the 2D position of the UAV. Shan et al. [238] compute the correlation between UAV image and reference map for global localization. After initialization, a particle filter, Histogram of Oriented Gradient (HOG) features, monocular camera, IMU, and barometer are used for local alignment. Han et al. [109] extract lines and apply ICP for reference alignment. Similarly, Son et al. [251] extract building information from satellite images and match the lines to the ones extracted from the live UAV image stream.

Other methods rely on matching of Digital Elevation Maps (DEMs): Sim et al. [243] estimate the geo-registered UAV position by relative motion estimation from image sequences and absolute position estimation from matching the online computed elevation map to the existing elevation map. Grelsson et al. [107] compute the global pose by aligning a geo-referenced 3D model with a local height patch computed from motion stereo. These approaches assume access to a reliable depth map at run-time and the terrain variation to be discriminative (e.g., urban).

The following approaches have been presented that use rendering for reference view generation: Conte and Doherty use correlation-based template matching fused using a Kalman and Point Mass Filter [49]. Altimeter readings are required for scaling. Chiu et al. [47] claim map views are rendered from a 3D model but keep the vision module as a black box and focus on the graph-based optimization.

Substantial improvements have been achieved with the introduction of DL and CNNs: Aznar et al. [13] use a simple CNN for visual alignment of UAV and satellite image but requires a compass and altimeter for operation. More recent feature and descriptor approaches, such as Superpoint [61], D2-Net [70], and tracking-based methods like deep ICLK variants [104, 105, 180] are able to align two images

even under extreme viewpoint and appearance changes. However, the approach of Goforth et al. [104, 105] does not simultaneously compute the alignment and six-DoF geo-referenced pose, and [180] assumes the availability of depth images for both the target and the reference image, as it is, for instance, the case when using depth cameras.

In this context, this paper introduces a rendering engine that takes geo-referenced orthoimages and elevation maps as input and renders optical and depth images given the six-DoF camera pose and desired camera model. The rendering engine is combined with a learning-based six-DoF-ICLK algorithm that is able to align images under severe appearance variations and only requires one depth image per image pair.

The remainder of the paper is structured as follows: Sec. 3 introduces the rendering engine that is used for offline training data generation and online reference view generation. Sec. 4 embeds the rendering engine in the learning-based ICLK framework for online six-DoF localization. The paper concludes with our experiments, a conclusion, and suggestions for future research directions.

3 Geo-referenced Renderer

This section introduces `aerial_renderer`, a rendering program that uses geo-referenced orthoimages (e.g., optical mosaics of different years) and elevation maps to render a metric depth map and texture images given a query camera pose $\mathbf{T}_C^W$, camera intrinsics and distortion parameters. The renderer serves two purposes in this paper:

1. **Offline training data generation and augmentation:** Parameter inputs are the six-DoF query pose in geo-referenced[2] coordinates, camera intrinsics and camera distortion parameters. The data outputs are the rendered texture images (e.g., from all available optical orthoimages) with the corresponding depth map.

2. **Online reference image generation for map tracking (Sec. 4):** Given a pose estimate $\mathbf{T}_C^W$, a reference image is generated, which is deemed to be close in scale and rotation to the real world image captured by the UAV. This has the advantage that the image alignment algorithm *merely* needs to be invariant to seasonal or environmental changes. Additionally, for the reference image, a dense depth map is rendered essential for six-DoF alignment. In the online mode, the most recent texture and elevation layers are used for rendering.

4 Map Tracking

Fig. 6.1 visualizes our proposed six-DoF `aerial_map_tracker`, combining the geo-referenced rendering engine `aerial_renderer` with a learning-based 6DoF-ICLK

[2] e.g., Universal Transverse Mercator (UTM)

algorithm. The map tracker is agnostic to the UAV platform, i.e., it may be used

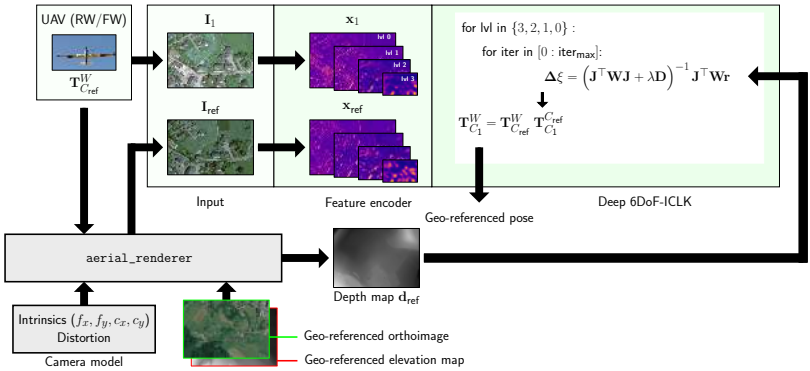


Figure 6.1: Map tracking framework with Deep 6DoF-ICLK.

by rotary-wing (RW) or fixed-wing (FW) UAVs. In the first step, the current camera pose prior of the UAV, denoted by $\mathbf{T}_{C_{\text{ref}}}^{W}$, is passed to the rendering engine. During initialization, this pose may come from a GPS-equipped autopilot, or from one of the coarse global localization algorithms described in Sec. 2.1. Once initialized, this prior is obtained from the ICLK's last iteration. For the query pose $\mathbf{T}_{C_{\text{ref}}}^{W}$, the rendering engine returns the optical image $\mathbf{I}_{\text{ref}}$ and depth map $\mathbf{d}_{\text{ref}}$, given the pre-calibrated camera intrinsics and distortion parameters as well as geo-referenced orthoimage and elevation map. Both, the real optical image taken by the UAV $\mathbf{I}_1$ and the rendered optical image $\mathbf{I}_{\text{ref}}$ are color normalized and fed as single views to the convolutional feature encoder proposed in [181]. The encoder returns four pyramidal layers, ranging from level 0 (752×480 pixels) to level 3 (94×60 pixels). The ICLK algorithm continues to iterate over all levels, starting with the lowest resolution (level 3) up to the original resolution image. For every iteration of the ICLK the six-DoF relative pose is incrementally updated using Levenberg-Marquardt [181, 182]:

$$\Delta\xi = \left(\mathbf{J}^\top\mathbf{W}\mathbf{J} + \lambda\mathbf{D}\right)^{-1}\mathbf{J}^\top\mathbf{W}\mathbf{r} \tag{6.1}$$

where the Jacobian $\mathbf{J}$ and residual $\mathbf{r}$ are functions of the depth map $\mathbf{d}_{\text{ref}}$. The diagonal matrix $\lambda\mathbf{D} = \lambda\text{diag}\left(\mathbf{J}^\top\mathbf{W}\mathbf{J}\right)$ is used for regularization of the Hessian matrix with $\lambda = 1e^{-6}$. The weight matrix $\mathbf{W}$ is learned by the convolutional M-Estimator introduced in [181]. Once the ICLK reaches a stopping criteria (e.g., max. number of iterations) the relative transformation matrix $\mathbf{T}_{C_1}^{C_{\text{ref}}} \in \text{SE}(3)$ is

computed from $\Delta\xi \in \mathrm{se}(3)$ and used to output the desired map-aligned pose $\mathbf{T}_{C_1}^{W}$:

$$\mathbf{T}_{C_1}^{W} = \mathbf{T}_{C_{\mathrm{ref}}}^{W}\, \mathbf{T}_{C_1}^{C_{\mathrm{ref}}}. \tag{6.2}$$

5 Experiments

Implementation The renderer `aerial_renderer` is implemented in OpenGL (C++) [273]. Every pixel of the orthoimage is converted into two triangles, i.e., six vertices, inside the vertex shader program. The `aerial_map_tracker` is implemented in pytorch [203] and adopted from [181]. The `aerial_renderer` is embedded in a ROS [252] wrapper to interface with `aerial_map_tracker`.

Figure 6.2: From left to right: Four locations in Switzerland are selected to extract training, validation and testing data. Training, validation and testing data is extracted for every location from nearby but non-overlapping areas. Each of these areas consists again of orthoimages and elevation maps of all available years.

Datasets Fig. 6.2 visualizes the four locations in Switzerland from where training, validation, and testing data is extracted. For each of these four locations, two nearby but non-overlapping areas are selected for training/validation and testing, respectively. The training and validation set are split 80 % to 20 %. Each of these areas consists again of orthoimages and elevation maps of all available years. All year-to-year combinations are considered for training (e.g., 2002 to 2006, 2010, 2013).

Results Tab. 6.1 and 6.2 present the quantitative results in form of the 3D End-Point-Error (EPE) [166, 181], angular and translational error. The EPE is used as training loss function for all experiments. Tab. 6.1 evaluates the alignment for the case that $\mathbf{I}_1$ and $\mathbf{I}_{\mathrm{ref}}$ are rendered from the same year, i.e., same orthoimage. In contrast, Tab. 6.2 shows the error metrics for two different years (2006, 2010).

First of all, one can see that the deep 6DoF-ICLK algorithm performs on average the best for both cases and in all error metrics, outperforming the variants without trainable parameters independent of the number of iterations. Secondly, for the same year, the non-trainable 6DoF-ICLK is able to reduce the EPE by a factor

	EPE					Ang. Error					Transl. Error				
	mean	stdev	median	min	max	mean	stdev	median	min	max	mean	stdev	median	min	max
init	13.88	6.40	12.78	5.53	35.07	0.06	0.03	0.07	0.02	0.12	13.11	5.43	13.56	3.97	25.58
no NN (20)	7.83	8.28	7.18	0.08	35.08	0.05	**0.04**	0.05	**0.00**	**0.11**	8.70	**6.96**	8.62	0.22	**26.99**
no NN (50)	3.73	**8.19**	**0.05**	**0.01**	**34.67**	0.03	**0.04**	**0.00**	**0.00**	0.12	4.46	7.21	**0.18**	**0.01**	29.21
NN (20)	**3.19**	8.55	0.58	0.10	37.82	**0.02**	**0.04**	0.01	**0.00**	0.18	**4.37**	8.64	1.63	0.24	38.74

Table 6.1: The error metrics End-Point-Error (EPE), angular and translational error evaluated on the test set for **the same year** (2006, 2006).

of 2.1 if the number of iterations is increased by a factor of 2.5, i.e., implying convergence. However, for a different year, the EPE increases by a factor of 1.22

	EPE					Ang. Error					Transl. Error				
	mean	stdev	median	min	max	mean	stdev	median	min	max	mean	stdev	median	min	max
init	13.88	6.40	12.78	5.53	35.07	0.06	0.03	0.07	0.02	0.12	13.11	5.43	13.56	3.97	25.58
no NN (20)	14.14	**7.87**	13.90	1.41	**39.48**	0.08	**0.04**	0.08	0.01	**0.16**	13.85	**6.96**	12.25	3.20	**24.69**
no NN (50)	17.29	10.57	18.18	**0.65**	44.14	0.11	0.07	0.09	**0.00**	0.22	19.22	10.80	18.10	**1.20**	47.48
NN (20)	**7.65**	8.61	**5.75**	1.20	40.87	**0.07**	0.05	**0.06**	0.01	0.21	**10.85**	7.18	**8.96**	2.88	28.70

Table 6.2: The error metrics End-Point-Error (EPE), angular and translational error evaluated on the test set for **different years** (2006, 2010).

when increasing the number of iterations from 20 to 50, which is a larger error then at initialization, i.e., implying divergence. The deep 6DoF-ICLK algorithm on the other hand, is able to reduce the EPE well below the initial error in both experiments.

The qualitative results are visualized in Fig. 6.3, presenting four samples taken from the test set (cf. Fig. 6.2). The first and second row show the framework input image_0 (i.e., $\mathbf{I}_{\text{ref}}$, from year 2006) and image_1 (i.e., $\mathbf{I}_1$, from year 2010) respectively. In the first and, in particular, the last column, one can see significant changes in the infrastructure. The third row visualizes the overlay of image_0 and image_1, given the initialized transformation, which is set to identity in all experiments. The fourth and fifth row show the overlay for the non-trainable 6DoF-ICLK after 20 and 50 iterations, respectively, followed by the deep 6DoF-ICLK in the sixth row. The last row shows the overlay given the ground-truth relative transformation. All methods are tagged with EPE and the number of iterations. Overall, the conclusions that can be drawn are consistent with the quantitative analysis described above: In all samples, the EPE for the non-trainable 6DoF-ICLK rises with an increasing number of iterations, and in 3 out of the 4 samples it is higher than the initial error. Instead, the deep 6DoF-ICLK is able to decrease the EPE in all cases drastically.

Runtime The `aerial_renderer` runs at over 2500 fps on a GeForce RTX 2080 Ti (12GB) for map sizes shown in Fig. 6.2. The runtime of the 6DoF-ICLK algorithm is given in Tab. 10.1 in ms, tested on the same GeForce RTX 2080 Ti: It takes the classical 6DoF-ICLK algorithm (*no NN*) approximately 200 ms to align one image-pair when stopping after 20 iterations, which is comparable to the average runtime of the deep 6DoF-ICLK algorithm with the same number of allowed iterations.

Increasing the number of allowed iterations of the classical 6DoF-ICLK algorithm by a factor of 2.5 increases the runtime by a factor of 2.26.

	Runtime [ms]				
	mean	stdev	median	min	max
no NN (20)	199.30	6.07	199.72	185.97	214.77
no NN (50)	450.32	12.26	446.63	421.90	476.78
NN (20)	206.85	3.67	207.51	197.99	213.31

Table 6.3: Runtime of the 6DoF-ICLK algorithm in ms.

6 Conclusion

This work presents a framework for UAV localization in unstructured environments with the help of DL. Firstly, this paper introduces `aerial_renderer`, a real-time rendering engine that uses geo-referenced orthoimages and elevation maps to render depth and optical images given a six-DoF camera pose $\mathbf{T}_C^W$ and camera model. Secondly, `aerial_map_tracker` is presented, which combines the geo-referenced rendering engine `aerial_renderer` with a learning-based 6DoF-ICLK algorithm. The evaluation shows that the deep 6DoF-ICLK outperforms its non-trainable counterparts by a large margin. An interesting alley for future work would be to *hallucinate* the reference image via view synthesis, taking into account important factors like daytime, season, sun position to reduce the environmental or seasonal change and further ease the image alignment process [11, 142]. Furthermore, we motivate the use of `aerial_renderer` to accelerate the employment of DL for UAV localization in unstructured environments. For instance, one could create infrared orthomosaics and create infrared-optical or infrared-semantical training data for image alignment.

Figure 6.3: Qualitative evaluation of four samples from the test set shown in Fig. 6.2. The input images image_0 and image_1 are from the years 2006 and 2010, respectively.

Part B

DEPTH ESTIMATION

Paper IV

Robust Map Generation for Fixed-Wing UAVs with Low-Cost Highly-Oblique Monocular Cameras

Timo Hinzmann, Thomas Schneider, Marcin Tomasz Dymczyk, Amir Melzer, Thomas Mantel, Igor Gilitschenski, Roland Siegwart

Abstract

Accurate and robust real-time map generation onboard of a fixed-wing UAV is essential for obstacle avoidance, path planning, and critical maneuvers such as autonomous take-off and landing. Due to the computational constraints, the required robustness and reliability, it remains a challenge to deploy a fixed-wing UAV with an online-capable, accurate and robust map generation framework. While photogrammetric approaches have underlying assumptions on the structure and the view of the camera, generic simultaneous localization and mapping (SLAM) approaches are computationally demanding. This paper presents a framework that uses the autopilot's state estimate as a prior for sliding window bundle adjustment and map generation. Our approach outputs an accurate geo-referenced dense point-cloud which was validated in simulation on a synthetic dataset and on two real-world scenarios based on ground control points.

Published in:
International Conference on Intelligent Robots and Systems (IROS), 2016

1 Introduction

Recent years have shown an increasing interest in using UAVs to enable applications such as industrial inspection, surveillance, and agricultural monitoring at much lower cost than conventional aircrafts. In contrast to rotary-wing UAVs, fixed-wing UAVs can cover large areas in a short amount of time. Even some solar-powered fixed-wing UAVs have recently demonstrated a flight endurance of several days [196]. These properties make fixed-wing UAVs the ideal platform for mapping missions but also requires a robust framework that efficiently processes the large amount of recorded data. For obstacle avoidance and path planning, for instance, the information needs to be available in the range of milliseconds and seconds respectively. But also for search and rescue or surveillance missions, the map needs to be available as soon as possible.

Photogrammetric approaches for UAV mapping missions have shown impressive results in the last decades [72, 222] leading to several commercial products such as Pix4D. However, these approaches are usually processed off-line and are formulated as a batch optimization problem. Furthermore, they often assume a nadir-looking camera, the landmarks to lie approximately on a common ground plane, and are unable to make use of all sensor measurements.

A major advantage of state-of-the-art SLAM approaches [73, 163] is the capability of efficiently generating a map and simultaneously localizing the robot within this map based on visual and inertial measurements. However, this advantage can also turn into a drawback when a low-cost camera is involved and the landmark-distance to inter-keyframe baseline becomes too high which results in a high uncertainty of the landmark locations. A degradation of the map leads to a degradation of the state estimation and vice versa. For fixed-wing UAVs that are loitering above the same scenery, altitude drift of the estimated robot position becomes visible in form of a map consisting of stacked landmark layers.

To avoid these effects of scale ambiguity at high altitudes, our method decouples state estimation and map generation and is based on the indirect Extended Kalman Filter implementation presented in [160]. In this regard, our approach is similar to the one presented by Irschara in [133] where weak position and orientation priors speed up the structure from motion calculation. However, the latter considers all information simultaneously whereas our approach is designed to run onboard of the UAV and to compute a dense point-cloud once the camera poses and images are available. An overview of our framework is given in Fig. 11.1:

The state estimates of the PX4 autopilot are used as camera pose priors for feature tracking, triangulation, and are refined in a sliding window bundle adjustment scheme. Although the framework is presented for this specific setup it can be employed with any sensor or state estimation setup which provides camera pose priors. To the best of our knowledge, this paper represents the first approach to generate a dense map onboard of a fixed-wing UAV by decoupling state estimation and mapping in a fixed-lag smoothing formulation. In sum, we present a flight-tested real-time capable mapping framework with the following benefits:

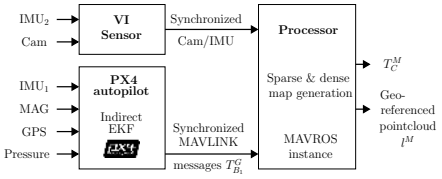


Figure 7.1: Framework overview for state estimation and map generation.

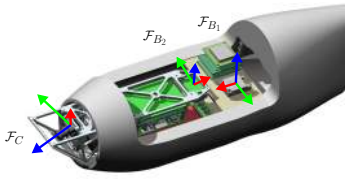


Figure 7.2: Interior of the UAV platform Techpod[1].

- No assumptions on the structure, such as ground plane assumption.
- In contrast to most photogrammetric approaches we do not assume that the camera is mounted with a nadir view. In our case the camera is mounted with a highly-oblique view such that it can be used for obstacle avoidance and map generation at the same time.
- Robust mapping with a low-cost monochrome camera by decoupling of state estimation and map generation.
- The computational cost is kept bounded due to the sliding window bundle adjustment scheme that keeps only a subset of the camera poses in the optimization problem. The cost for state estimation and map generation is distributed on different boards.

The remainder of this paper is structured as follows: Section 2 presents the methodology for generating a dense point-cloud based on camera pose priors. In Section 3, the sliding window bundle adjustment is validated on a synthetic dataset and compared to the result of the batch bundle adjustment. The small unmanned research plane used for the real-world experiments is described in Section 4. Section 5 presents real-world experiments, which are based on two datasets recorded onboard of the fixed-wing UAV and includes the results of the sparse and dense mapping framework as well as landmark accuracy validation. The paper concludes with Section 6.

2 Methodology

The outline of the mapping algorithm is presented in algorithm 1. The most relevant parts include feature tracking, feature track triangulation, bundle adjustment and dense reconstruction.

[1] The PX4 autopilot as well as its GPS receiver, magnetometer, IMU and pressure sensors are rigidly mounted. The sensorpod is a modular unit that is used in several UAVs.

Algorithm 5 Map generation

1: For every image, retrieve initial robot pose estimate from EKF and compute camera pose (2.1, 2.2)
2: Extract feature tracks (2.3):
- Lucas-Kanade feature tracker
- Gyroscope measurement integration for feature prediction
- Feature bucketing
- Two-point RANSAC outlier rejection

3: Initialize landmarks (2.3):
- Check if landmark is well constrained
- Apply Gauss-Newton triangulation (inverse depth) [186] with ground plane initialization
- Check if landmark location is in visible camera cone

4: Perform sliding window bundle adjustment (2.3)
5: Add resulting optimized camera poses to dense reconstruction pipeline (2.4)
6: Insert landmarks into octomap (2.5)

2.1 EKF-based autopilot

The Pixhawk PX4 auto-pilot performs an indirect EKF-based state estimation as presented in [160]. The Kalman filter uses the linear acceleration and angular rates measurements for propagation of the state equations. The dynamic and static pressure, GPS velocity and position as well as 3D magnetometer measurements are used for the Kalman Filter state update. The estimated states consist of sensor (gyroscope and accelerometer) biases, wind field, three-dimensional airspeed as well as the IMU's attitude and position in WGS84 coordinates. For more details about the state estimation framework we refer to [160].

2.2 Transformations

The EKF estimates the position and orientation of IMU_1 in a global coordinate system. However, for mapping we need to compute the camera pose with respect to the global mapping coordinate system denoted by M. The transformation chain is stated in equation 7.1 and involves the camera (denoted with C), the IMU_1 of the autopilot which is rigidly mounted in the fuselage (denoted with B_1), as well as the IMU_2 mounted on the sensor pod (denoted with B_2) as shown in Fig. 7.2.

$$\mathbf{T}_C^M = \mathbf{T}_G^M \mathbf{T}_{B_1}^G \mathbf{T}_{B_2}^{B_1} \mathbf{T}_C^{B_2} \tag{7.1}$$

with

$\mathbf{T}_C^{B_2}$: Transf. of the camera w.r.t the sensorpod IMU

$\mathbf{T}_{B_2}^{B_1}$: Transf. of sensorpod IMU w.r.t the autopilot IMU

$\mathbf{T}_{B_1}^G$: Transf. of the autopilot IMU w.r.t global system

$\mathbf{T}_G^M$: Transf. of the global system w.r.t mapping system

The transformation of the camera with respect to the IMU of the sensorpod as well as the camera intrinsics and distortion are calibrated with the standard Kalibr stack [86].

The rotation and translation between the IMU of the sensorpod and the IMU of the autopilot is computed with an extension of Kalibr which was proposed by Rehder and Nikolic [220] and capable of precise IMU-IMU transformation estimation. Finally, the position estimate coming from the autopilot is transformed from WGS84 coordinates into a metric UTM coordinate system. In a last step, the orientation needs to be aligned with the UTM coordinate system: In our case, the IMU assumed north direction to be x. However, in UTM coordinates, easting is x, northing is y. Consequently, the transformation matrix $\mathbf{T}_M^G$ results in a 90 ° rotation around the z-axis:

$$\mathbf{T}_M^G = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7.2}$$

The transformations were validated by solving an absolute perspective n-point problem (PNP), as proposed by Kneip and implemented in OpenGV [144]. The landmark positions in UTM coordinates were obtained from geo-referenced satellite images, the feature location in the image were hand-labeled as shown in Fig. 7.3.

Figure 7.3: Left: Landmark positions in UTM coordinates. **Right:** Hand-labeled feature positions (green) and reprojected landmark locations (red) given the camera pose estimated by the PNP.

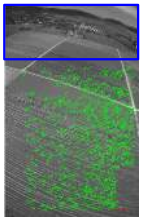


Figure 7.4: Feature tracking results showing the inlier set (green), outlier set (red) as well as horizon mask (blue).

2.3 Sparse point-cloud generation

The sparse map generation is presented in algorithm 1. The most relevant parts include feature tracking, feature track triangulation and sliding window bundle adjustment.

Feature Tracking

In a first step, features are extracted based on the approach proposed by Shi and Tomasi [240]. The features are then tracked with a Lucas-Kanade feature tracker. The gyroscope measurements of the sensorpod IMU[2] are integrated between successive frames to predict the feature location in the next frame and to constrain the search window and limit the computational cost. Feature bucketing ensures uniformly distributed feature observations across the image. Eventually, a 2-point relative translation-only RANSAC problem [145] is solved to identify and reject outliers. Fig. 7.4 shows the feature tracks classified by RANSAC as inliers in green and outliers in red. As visualized in blue, a mask can be used to avoid extracting features close to the horizon as the observation rays to the corresponding landmarks are almost parallel which results in high landmark position uncertainties. Alternatively, a horizon tracker or the evaluation of the observation angles can be used to reject not well constrained landmark observations.

Feature Triangulation

The often used direct triangulation approach may suffer from local minima [133]. For better numerical stability we use the Gauss-Newton triangulation where the landmark location is in inverse-depth parametrization as proposed in [186]: The basic idea is that the position of the j-th feature observed in the i-th camera frame can be expressed in terms of the n-th camera frame:

$$\begin{aligned}\mathbf{p}_{f_j}^{C_i} &= \mathbf{C}(\bar{q}_{C_n}^{C_i})\mathbf{p}_{f_j}^{C_n} + \mathbf{p}_{C_n}^{C_i} \\ &= Z_j^{C_n}\left(\mathbf{C}(\bar{q}_{C_n}^{C_i})\begin{bmatrix}\alpha_j & \beta_j & 1\end{bmatrix}^\top + \rho_j \mathbf{p}_{C_n}^{C_i}\right) \\ &= Z_j^{C_n}\begin{bmatrix}h_{i1} & h_{i2} & h_{i3}\end{bmatrix} \end{aligned} \tag{7.3}$$

with $\begin{bmatrix}\alpha_j & \beta_j & \rho_j\end{bmatrix}^\top = \begin{bmatrix}\frac{X_j^{C_n}}{Z_j^{C_n}} & \frac{Y_j^{C_n}}{Z_j^{C_n}} & \frac{1}{Z_j^{C_n}}\end{bmatrix}^\top$

Equation 7.3 is expressed in inverse depth parametrization and α, β and ρ are the minimization variables. We provide the complete pseudo-code in the appendix. The Gauss-Newton triangulation shows a good trade-off between accuracy and calculation time. Since this is an iterative approach, the algorithm needs to be initialized appropriately. In particular, an educated guess for the landmark position in terms of the first camera frame needs to be set. For this, we assume that the landmark is located on an approximated ground plane.

Sliding Window Bundle Adjustment

The visual bundle adjustment considers only the last N camera poses to retain real-time processing, where N denotes the size of the sliding window. The bundle

[2] In theory, also the autopilot IMU could have been used for feature tracking. However, the camera and IMU measurents are time-synchronized on the FPGA and come in with a higher rate (200 Hz instead of 100 Hz).

adjustment is formulated in a factor graph and optimized with *Georgia Tech Smoothing and Mapping* (GTSAM) [56, 83]. The insertion of the factor nodes, value nodes, the marginalization strategy as well as outlier rejection is explained based on the sample factor graph shown in Fig. 7.5: Once a new pose-image pair is available, the camera pose is inserted in the factor graph and initialized with the estimate from the Extended Kalman Filter. To constrain the factor graph, each camera pose is associated with a prior factor based on the mean and standard deviation from the EKF. The triangulated landmarks are inserted as value nodes and connected to the individual camera poses via reprojection factors. The reprojection error formulation is adopted from [163]:

$$\mathbf{e}_r^j = \mathbf{z}^j - \mathbf{h}(\mathbf{T}_M^C \mathbf{l}_j^M) \tag{7.4}$$

where $\mathbf{h}(\cdot)$ stands for the camera projection and $\mathbf{z}^j$ is the feature measurement of landmark j in image coordinates.

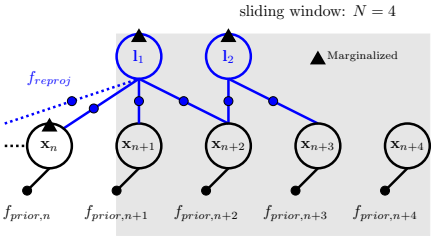


Figure 7.5: Sliding window bundle adjustment in factor graph formulation.

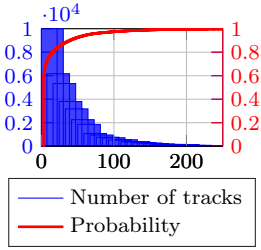


Figure 7.6: Feature track histogram for 20 fps. 93 percent of the feature tracks contain less than 50 observations.

Every reprojection factor has a Cauchy M-Estimator to reduce the influence of outliers[3]. The factor graph is optimized, visual measurements that are still identified as outliers are rejected and the graph is optimized again. After the graph optimization, all landmarks and all camera poses (i.e. the oldest one) outside of the sliding window are marginalized. The selection of the sliding window size N constitutes a trade-off between computational costs and accuracy of the state estimates. The window size should be set after evaluating the expected length of the feature tracks to ensure that the majority of observations of the feature tracks are included as reprojection factors in the optimization problem. Fig. 7.6 shows the feature track length histogram for the dataset I presented in Sec. 5. The empirical cumulative distribution function illustrates that around 97 percent of all feature tracks contain less than 100 observations which corresponds to a 5.0 s

[3] The Cauchy weight is $k^2/(k^2 + e^2)$, where e is the residual and k is a constant set to 3.0.

sliding window. Choosing a too small value for the sliding window results in a shorter baseline of first-to-last camera pose in the feature track and consequently in less constrained landmarks as well as camera poses.

2.4 Dense point-cloud generation

The image stream and the corresponding optimized camera poses are then used as input for the dense reconstruction algorithms. The dense reconstruction approaches can be divided into rectification-based and patch-based methods [89, 90]. The rectification-based methods seem more suited for our purpose since they are computationally more efficient. In general, the goal of rectification is to transform an arbitrarily arranged stereo pair into a rectified virtual stereo pair, in which epipolar lines become collinear and horizontal. In the rectified stereo pair the correspondence search becomes easier since matches lie on horizontal lines of the rectified images and efficient blockmatching algorithms can be employed [127, 147]. The output of the dense reconstruction module is a geo-referenced dense point-cloud which is colored by pixel intensities. The next section shortly describes the planar and polar rectification procedures and their advantages and disadvantages based on our camera configuration.

(a): Nadir **(b): Highly oblique**

Figure 7.7: (a): Nadir (down-looking) camera configuration. The epipoles are at infinity for the forward moving UAV and both, planar and polar rectification can be used. **(b):** Highly-oblique camera configuration. The epipoles might be close or even within the images. In this case the planar rectification fails and the polar rectification algorithm needs to be used for dense reconstruction.

Planar rectification

One of the properties of the virtual ideal stereo camera is, that it has parallel optical axes, which are perpendicular to the baseline. This ensures that the epipoles are at infinity and hence epipolar lines become collinear and parallel. The nadir (down-looking) camera configuration inherently fulfills these requirements for planar rectification as shown in Fig. 7.7. However in case of an oblique camera, the epipoles might be close or even within the images depending on the concrete motion of the fixed-wing UAV. In that case, pixels around the epipole project to infinitely far away points on the rectified image plane, which would result in extremely large and distorted rectified images. For implementation details we refer to [91].

Polar rectification

In contrast to planar rectification, the approach proposed by [208] can be employed for generic camera motion. The approach takes directly the pixel intensities along corresponding epipolar lines and inserts them into the same row of the rectified images. The rectified images are finally built up by circularly scanning the original images around the epipoles. A detailed description can be found in [208]. Compared to the planar rectification, the polar rectification algorithm is computationally more involved and produces more outliers. To benefit from both approaches, we keep a small buffer of frames and compute the geometry of the current camera frame to the camera frames in the buffer. If the epipoles are close or within all stereo pair combinations, we use polar otherwise we employ planar rectification.

2.5 Octomap interface

The generated point-cloud is inserted into OctoMap [275]. Two possible ways to insert the point-cloud exist:

- Endpoint-only: Only encodes the occupied space by inserting the landmarks directly.
- Ray-casting: Encodes the occupied and free space in a probabilistic manner by casting rays from the camera position to the landmark location which is useful for obstacle avoidance and path-planning. However, the calculation costs increases with the distance to the landmark.[4]

The point-cloud generated onboard of the fixed-wing UAV could then be visualized directly on the ground-control computer if the UAV is in WiFi range.

3 Simulation

To validate the performance of the sliding window bundle adjustment under realistic conditions it was simulated based on a real-world point-cloud[5] and camera poses recorded by the presented fixed-wing UAV. The nominal flight altitude in the simulation is around 150 m. The camera intrinsics are identical to the ones used for the real-world datasets. The camera positions are disturbed with Gaussian white noise $\mathcal{N}(0, (0.5\,\mathrm{m})^2)$; the feature observations with $\mathcal{N}(0, (0.2\,\mathrm{pixel})^2)$. The minimal and maximal track length were set to 2 and 50 respectively.

The absolute position errors in x of the disturbed input, the result of the sliding window bundle adjustment and of the batch bundle adjustment are shown in Fig. 7.8. The accompanying data is given in Table 7.1.

[4] For a nominal flight altitude of around 150 m and several hundred landmarks per image ray-casting becomes computationally costly. Compare performance discussion in [275].

[5] Dataset *cadastre* of the Pix4D example datasets.

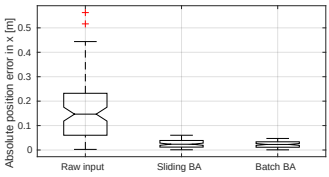


Figure 7.8: Performance comparison of sliding window bundle adjustment and batch bundle adjustment.

	$\mu(\lvert e_x \rvert)$	$\sigma(\lvert e_x \rvert)$
Input	0.1610	0.1177
Sliding	0.0246	0.0150
Batch	0.0231	0.0135
	$\mu(\lvert e_y \rvert)$	$\sigma(\lvert e_y \rvert)$
Input	0.1605	0.1155
Sliding	0.0163	0.0221
Batch	0.0128	0.0073
	$\mu(\lvert e_z \rvert)$	$\sigma(\lvert e_z \rvert)$
Input	0.1431	0.0973
Sliding	0.0210	0.0165
Batch	0.0192	0.0045

Table 7.1: Mean μ and standard deviation σ of the camera position errors in meter. The batch bundle adjustment performs only slightly better compared to the sliding window approach.

As expected, the batch bundle adjustment performs slightly better than the sliding window bundle adjustment since the whole factor graph is available for optimization.

4 Platform

The small unmanned research plane Techpod was used for the experiments presented in this paper. It has a wingspan of 2.60 m, classic T-tail configuration and is equipped with one propeller. The sensor pod and PX4 auto-pilot are placed inside the fuselage as shown in Fig. 7.2 and allow autonomous mission execution such as GPS-waypoint following. The technical specifications of the sensor and processing unit are listed in Table 7.2. As exteroceptive sensor it features an Aptina MT9V034 monochrome global shutter camera capturing images with a resolution of 752×480 pixels at up to 60 fps. It is rigidly mounted with an oblique field of view of around 45 deg. For measuring angular velocities and angular accelerations, the sensor pod is equipped with a MEMS inertial measurement unit (IMU). The camera and IMU are integrated into an ARM-FPGA-based Visual-Inertial (VI) sensor system [195] allowing hardware-synchronized IMU and camera data. An Intel Atom CPU (four cores at 1.92 GHz) is connected to the VI sensor system and the PX4 auto-pilot. All components are mounted on an aluminium frame that guarantees a rigid camera-imu transformation throughout all flight scenarios.

More technical details about the deployed camera is presented in the appendix.

5 Real World Experiments

The mapping pipeline was evaluated based on two datasets, denoted with Set I and Set II, which were recorded by the fixed-wing UAV presented in Section 4. Set I is characterized by a nominal altitude of 100 m to 150 m and a flat scenery as shown in Fig. 7.3. The sparse point-cloud generated during this flight is shown in Fig. 7.9.

Sensor pod	Monochrome camera: IMU Thermal camera Processing board Processor	Aptina MT9V034 ADIS16448 FLIR Tau 2 Kontron COMe-mBT10 Intel Atom (4 cores, 1.91 GHz)
Auto-pilot	IMU GPS Processor RAM	ADIS16448 uBlox LEA-6H Cortex M4F (168 MHz) 192 kB

Table 7.2: The sensor and processing unit, nicknamed sensor pod, and the PX4 Pixhawk auto-pilot used for the experiments.

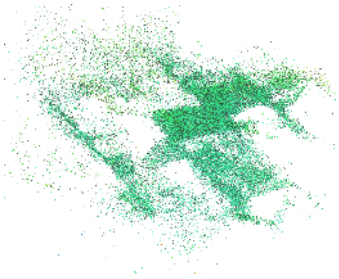


Figure 7.9: Sparse point-cloud visualized in octomap, colored by height: $2264 \times 1473 \times 245\,\mathrm{m}^3$ with 264793 landmarks, leaf size is 1 m.

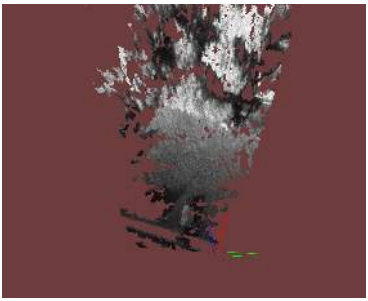


Figure 7.10: Geo-referenced point-cloud generated with planar rectification based on two views. The inter-keyframe baseline is 1.35 m. Both epipoles are outside of the images.

The landmarks are inserted into OctoMap once they leave the sliding window and are colored by height. The estimated landmark locations agree with the ground control points obtained from satellite data as presented in Section 5.1.

Set II was recorded during the final demonstration of the ICARUS FP7 project in March-en-Famenne, Belgium. The nominal altitude of around 70 m is lower than in Set I as can be seen from Fig. 7.11 and 7.12. Due to the 3-dimensional structure, it is used to present the results of the dense reconstruction pipeline.

5.1 Landmark accuracy validation

For fixed-wing UAVs it remains a challenge to obtain high-quality ground-truth data of the camera poses. Hence, we assess the quality of our approach based on ground control points obtained from satellite images. Care was taken to use permanent and easily identifiable ground control points suchs as house corners or road crossings. The process of obtaining the ground control point is as follows: The optimized landmark is obtained from the mapping pipeline where every landmark is associated with the corresponding image, keypoint and 3D position. The keypoint location in the image is visualized and the location is labeled in the satellite image. The results for Set I and II are presented in Table 7.3.

	$\mu(\lvert e_x \rvert)$	$\sigma(\lvert e_x \rvert)$	$\mu(\lvert e_y \rvert)$	$\sigma(\lvert e_y \rvert)$	$\mu(\lvert e_z \rvert)$	$\sigma(\lvert e_z \rvert)$
Set I	7.9840	7.7916	3.7913	2.7068	6.3316	0.2593
Set II	2.2571	0.7771	4.5677	1.4708	3.2798	2.0935

Table 7.3: Landmark accuracy validation based on two real-world datasets. The Table shows the mean μ and standard deviation σ of estimated landmark location to ground control points in meter. For each set we gathered 10 ground control points.

Overall, Set II achieves a higher accuracy compared to Set I which can be explained by the different flight altitudes. At higher flight altitudes, the same orientation, distortion or intrinsics error result in a larger landmark position error when the ray is projected on the ground. Possible error sources of the accuracy assessment include inaccuracies of the satellite images and of the ground control point labeling process.

5.2 Dense reconstruction

In this section, the output of the dense reconstruction module is presented to underline the accuracy of the bundle adjusted camera poses. Fig. 7.11 visualizes the rectified images and disparity map generated by the planar rectification algorithm.

Feature correspondences can be searched across horizontal lines as illustrated in the top row of Fig. 7.11. The point-cloud and virtual stereo pair are visualized in Fig. 7.10.

The generated landmarks are colored by the pixel intensities of the original image and are geo-referenced in UTM coordinates. The inter-keyframe baseline is 1.35 m, the epipoles are located at $e_1 = [990.483,\ 302.183]$ and $e_2 = [1005.53,\ 289.519]$. From visual inspection, the planar rectification approach generates a consistent point-cloud with few outliers.

This is in particular true in the well observed region below the inter-keyframe baseline.

Analogously, the images rectified by the polar rectification approach, the disparity map as well as the generated point-cloud are shown in Fig. 7.12. Compared to the results of the planar rectification, the point-cloud generated by polar rectification

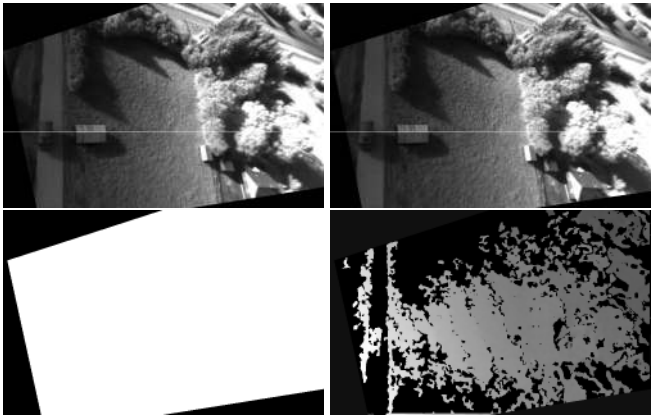


Figure 7.11: Planar rectification: The images on the top show the rectified images of the virtual stereo pair. The images on the bottom shows the employed mask and the disparity image seen from the virtual stereo rig.

shows decisively more outliers. As a conclusion, the planar rectification outperforms the polar rectification in terms of accuracy and runtime as presented in Table 7.5. Therefore, the usage of the planar rectification is to be maximized to produce consistent point-clouds. Nevertheless, polar rectification is used in cases that the planar rectification fails due to the geometry of the virtual stereo pair.

5.3 Runtime

The preliminary runtime results of the mapping pipeline are shown in Table 10.1. The validation was performed on an Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz for reference. The runtime results look promising and future work will include the optimization of the pipeline in order to make it run in real-time onboard of the fixed-wing UAV.

	mean $[ms]$	std. dev. $[ms]$
Feature tracking*	19.129	5.631
Feature triangulation	0.525	0.266
Bundle adjustment	26.571	17.511

Table 7.4: Runtime in ms for one frame. (*Runs in a separate thread and does not count towards total frame processing time.)

The runtime for the planar and polar densification step is shown in Table 7.5. The

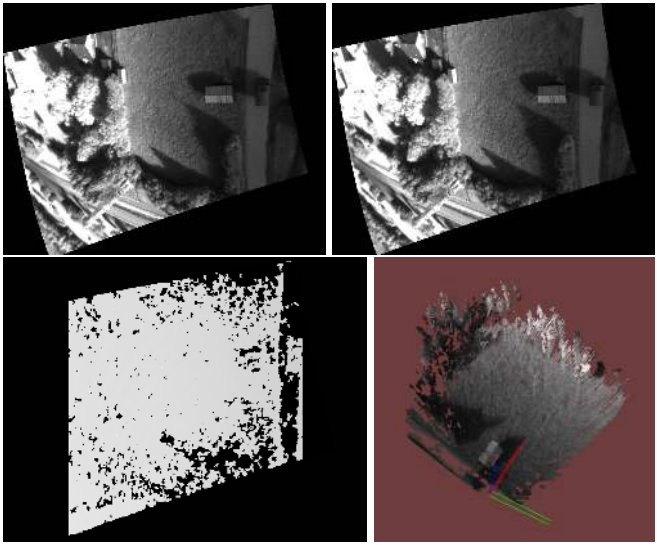


Figure 7.12: Polar rectification: The images on the top show the rectified images of the virtual stereo pair. The image on the bottom left visualizes the disparity image seen from the virtual stereo rig. On the right is the geo-referenced point-cloud generated with polar rectification based on two views.

increased complexity of the polar rectification algorithm is revealed in the runtime. The computational costs of both approaches can be decreased by downsampling the original image if necessary.

6 Conclusions

In this paper we presented a robust mapping framework that generates dense, geo-referenced point-clouds based on camera pose priors. The novelty of this approach lies in the use of a state-of-the-art smoothing and mapping framework in combination with camera pose priors to *iteratively* obtain depth estimates of the environment onboard of a fixed-wing UAV. The sliding window bundle adjustment was validated on a realistic synthetic dataset and achieved comparable results to the full batch bundle approach. Real world experiments were performed on a small fixed-wing UAV equipped with a low-resolution monochrome camera which was mounted with a highly-oblique view. The experiments underlined the generality of the approach that does not make any assumptions on the observed structure: The landmark

	Planar rectific. $[ms]$	Polar rectific.$[ms]$
Rectification	6.89	15.31
Correspondence search	14.54	24.13
Point-cloud generation	4.08	6.96
Frame processing	25.51	46.41

Table 7.5: Average runtime (200 runs) per image in ms for the planar and polar rectification. The original image size (752×480) was used for the rectification.

accuracy which was validated with ground control points performed equally good for a flat as well as for a 3D-structured scenery. The dense reconstruction pipeline considers the oblique view by checking the geometry of the stereo pairs and either performs planar or polar rectification. A rigorous outlier rejection in several layers of the pipeline ensures an accurate dense point-cloud.

The proposed framework is used as a robust backbone for autonomous UAV missions where obstacle avoidance and path-planning as well as autonomous landing and take-off are required. As a next step, we seek to improve the map coverage by feeding the map information to the path planning and control algorithms. In future work, the free space is to be computed in a more efficient manner, building up on the OctoMap implementation for ray-casting. Furthermore, for the results shown in this paper a constant camera rate was used. By using a camera view selection algorithm, the computational cost could be decreased.

Appendix

6.1 Gauss-Newton iterative multi-view triangulation

In this section, the pseudo-code for the iterative Gauss-Newton triangulation from subsection 2.3 is presented in Algorithm 2. The notation used in the pseudo-code of the Gauss-Newton triangulation is adopted from [186].

6.2 Camera parameters

Monochrome camera	Resolution n_x, n_y	752, 480
	Pixel size x	6 μm
	Lense f	2.8 mm
	Max. rate	60 Hz
	Delpoyed rate	20 Hz

Table 7.6: Camera parameters.

Algorithm 6 Gauss-Newton triangulation
T : Precision threshold
r : Measurement residual
J : Measurement Jacobian
$h_{m,i}$: Observation of the feature in camera i
$h_{p,i}$: Predicted observation of the feature in camera i
M : Number of observations for the feature

function GaussNewton(α_{init}, β_{init}, ρ_{init}, $iter_{max}$, T)
  $\alpha \leftarrow \alpha_{init},\ \beta \leftarrow \beta_{init},\ \rho \leftarrow \rho_{init}$
  while $\|r_{last} - r_{current}\|_2 > T$ and $iter < iter_{max}$ **do**
    for $i = 1$: all camera views M **do**
      ${}^{C_n}\mathbf{C}_{C_i} \leftarrow {}^{C_n}\mathbf{C}_G \cdot {}^{G}\mathbf{C}_{C_i}$
      ${}^{C_i}\mathbf{p}_{C_n} \leftarrow {}^{C_i}\mathbf{C}_G\,{}^{G}\mathbf{p}_{C_n} - {}^{C_i}\mathbf{C}_G\,{}^{G}\mathbf{p}_{C_n}$
      $h_{p,i} \leftarrow {}^{C_n}\mathbf{C}_{C_i} \cdot \begin{bmatrix}\alpha & \beta & 1\end{bmatrix}^\top + \rho \cdot {}^{C_i}\mathbf{p}_{C_n}$
      $h_{m,i} = \frac{1}{Z_{C,m}} \begin{bmatrix}X_{C,m} & Y_{C,m}\end{bmatrix}^\top$
      $r_i \leftarrow h_{p,i} - h_{m,i}$ ▷ Residuals in camera coordinates
      $r \leftarrow \begin{bmatrix}r & r_i\end{bmatrix}^\top$ ▷ Stack residuals
      $J_p \leftarrow \frac{1}{h_{p,i}(3)} \begin{bmatrix}1 & 0 & -\frac{h_{p,i}(1)}{h_{p,i}(3)} \\ 0 & 1 & -\frac{h_{p,i}(2)}{h_{p,i}(3)}\end{bmatrix}$ ▷ J. persp. camera
      $J_\alpha \leftarrow {}^{C_n}\mathbf{C}_{C_i} \begin{bmatrix}1 & 0 & 0\end{bmatrix}^\top$
      $J_\beta \leftarrow {}^{C_n}\mathbf{C}_{C_i} \begin{bmatrix}0 & 1 & 0\end{bmatrix}^\top$
      $J_\rho \leftarrow {}^{C_i}\mathbf{p}_{C_n}$
      $J_i \leftarrow J_p \begin{bmatrix}J_\alpha & J_\beta & J_\rho\end{bmatrix}$
      $J \leftarrow \begin{bmatrix}J & J_i\end{bmatrix}^\top$ ▷ Stack Jacobians
    end for
    $\Delta \leftarrow (J^\top J)^{-1} J^\top r$ ▷ $J = 2M \times 3$, $r = 2M \times 1$
    $\begin{bmatrix}\alpha & \beta & \rho\end{bmatrix}^\top \leftarrow \begin{bmatrix}\alpha & \beta & \rho\end{bmatrix}^\top - \Delta$
  end while
  ${}^{G}\mathbf{p}_f \leftarrow \frac{1}{\rho}\,{}^{G}\mathbf{C}_{C_n} \begin{bmatrix}\alpha & \beta & 1\end{bmatrix}^\top + {}^{G}\mathbf{p}_{C_n}$
end function

The camera field of view is given by

$$FOV = 2\tan^{-1}\left(\frac{n_{x,y}x}{2f}\right) \tag{7.5}$$

which computes to 77.72 ° and 54.43 ° for the horizontal and vertical FOV respectively given the camera parameters presented in table 7.6.

The ground sampling distance is then given by

$$GSD = xhf^{-1} \tag{7.6}$$

The image footprint is given by:

$$IFP = GSD \begin{bmatrix}n_x & n_y\end{bmatrix}^T \tag{7.7}$$

For the nominal flight altitude of $h = 200\,\mathrm{m}$, one pixel corresponds to $0.43\,\mathrm{m}$ on the ground and the image foot print is $322.29 \times 205.71\,\mathrm{m}^2$. N.b. that these equations are correct for the ideal case of a nadir-looking camera.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Commission's Sevenths Framework Programme (FP7/2007-2013) under grant agreement n°285417 (ICARUS) and n°600958 (SHERPA). Furthermore, the authors thank Jörn Rehder and Janosch Nikolic for their support in terms of the IMU-IMU and IMU-Camera calibration framework and Andreas Jäger and Sammy Omari in terms of the dense reconstruction algorithms.

Paper V

Flexible Trinocular: Non-rigid Multi-Camera-IMU Dense Reconstruction for UAV Navigation and Mapping

Timo Hinzmann, Cesar Cadena, Juan Nieto, Roland Siegwart

Abstract

In this paper, we propose a visual-inertial framework able to efficiently estimate the camera poses of a non-rigid trinocular baseline for long-range depth estimation on-board a fast moving aerial platform. The estimation of the time-varying baseline is based on relative inertial measurements, a photometric relative pose optimizer, and a probabilistic wing model fused in an efficient Extended Kalman Filter (EKF) formulation. The estimated depth measurements can be integrated into a geo-referenced global map to render a reconstruction of the environment useful for local replanning algorithms. Based on extensive real-world experiments we describe the challenges and solutions for obtaining the probabilistic wing model, reliable relative inertial measurements, and vision-based relative pose updates and demonstrate the computational efficiency and robustness of the overall system under challenging conditions.

Published in:
International Conference on Intelligent Robots and Systems (IROS), 2019

1 Introduction

For environmental awareness and autonomous mission execution in previously unknown terrain unmanned aerial platforms depend on high-quality depth measurements and ideally have access to a globally consistent map. For ground robots and rotary-wing Unmanned Aerial Vehicles (UAVs), small baseline rigid stereo rigs or depth cameras have shown to be sufficient to enable autonomous operation. However, for small fixed-wing UAVs flying at 15 m/s, like the platform shown in Fig. 10.1, off-the-shelf sensor solutions fail for various reasons: Monocular depth from motion (high depth uncertainty due to epipole), small baseline rigid stereo (limited range and depth uncertainty), laser (limited range, heavy, and expensive), depth cameras (limited range), radar (not yet miniaturized).

In this paper, we present our approach to increase the environmental awareness of fixed-wing UAVs to efficiently estimate the time-varying relative pose between multiple cameras while only relying on inexpensive visual-inertial sensors. In contrast to [123], in this paper we propose a trinocular setup where the center camera is rigidly mounted with respect to the autopilot located inside the fuselage to simplify the generation of a geo-referenced and globally consistent map. Additionally,

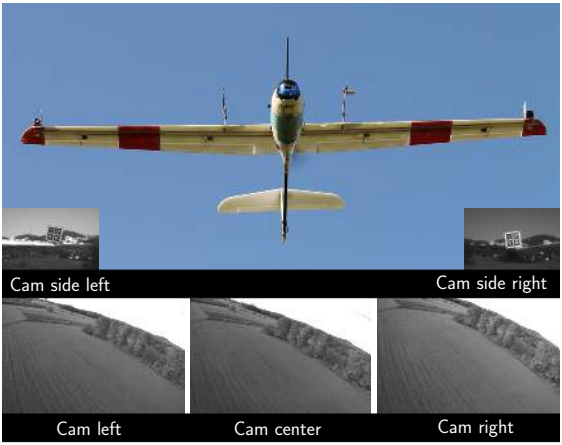


Figure 8.1: Fixed-wing UAV platform *Techpod* equipped with the proposed trinocular visual-inertial sensor setup. The side cameras and April tags are used for the identification of the wing model.

the trinocular setup allows to take advantage of both the full baseline for long-range depth estimates (left to right wing tip camera; approx. 2.4 m for our platform shown in Fig. 10.1) and half the baseline (e.g. left to center camera) for obstacles

nearby or for landing procedures. We estimate the non-rigid time-varying baseline

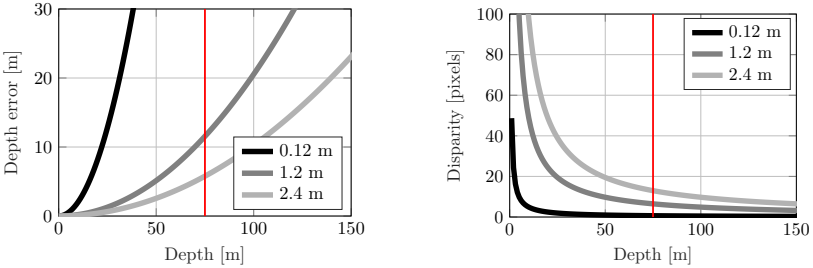


Figure 8.2: Theoretical depth error and disparity for different stereo baselines in meters. The red vertical line marks the distance that our UAV platform can cover within 5 s. Further parameters: focal length of 2.8 mm, pixel size of 6 µm, assumed resolvable disparity of one pixel.

between the center and left/right camera with Extended Kalman Filters (EKFs) that fuse relative visual-inertial measurements with a calibrated wing model that further constrains the relative baseline transformation. The depth maps resulting from the multiple stereo views can then be integrated into a geo-referenced map. While the approach presented in [123] is interesting for reactive navigation and control of a UAV, the framework proposed in this paper is designed for the use of model predictive controllers and local replanning algorithms able to operate on a 3 D or 2.5 D map.

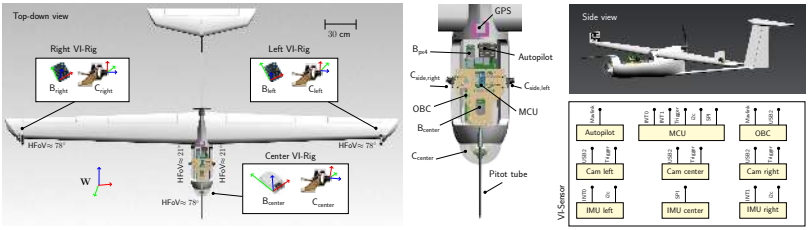


Figure 8.3: Overview showing the UAV platform, sensors, and coordinate frames. An EKF estimates the relative pose between left, right, and center camera. The side cameras are used to identify a wing model in form of a Gaussian prior. The horizontal field of view (HFoV) is denoted for each camera. The schematic on the right bottom depicts the developed VI-sensor that time-synchronizes the sensors on a hardware level.

2 Related Work

Several state-of-the-art monocular visual-inertial odometry (VIO) and simultaneous localization and mapping (SLAM) frameworks have been adopted to stereo or multi-camera operation [26, 129, 162, 190, 254]. Other multi-camera systems are specifically designed for their individual application such as for increased environmental awareness to navigate in cluttered environments using a multi-copter [106, 189] or for obstacle detection and avoidance on-board an agile delta wing [18]. All of these approaches have in common that they assume a fixed and precisely calibrated transformation between the individual cameras. However, for a fixed-wing UAV as shown in Fig. 10.1, a *rigid* stereo system can, due to aerodynamic effects and payload constraints, only be mounted inside the fuselage, limiting the effective baseline in this case to approximately 12 cm. Fig. 8.2 presents the theoretical depth uncertainty and expected disparity for such a rigid small-baseline stereo setup and motivates the necessity for a setup with a wider baseline.

On the other hand, approaches have been developed that are able to cope with different degrees of deviations from the calibrated baseline transformation: Warren et al. [268] continuously estimate thermally induced slow changes in the baseline. In [269], a down-looking stereo pair with a wide-baseline (0.7 m) is employed and the relative transform between the cameras, together with the poses of the stereo rig itself is estimated in an offline bundle adjustment problem. Lanier et al. [155, 156, 241] use a set of accelerometers distributed along the wing to estimate vibrational disturbances without any vision update. For airborne applications, Yang et al. fuse the measurements from a master and slave IMU in an EKF to estimate the relative states, a process referred to as transfer alignment [279]. Achtelik et al. [6] proposed an efficient EKF formulation that fuses relative IMU and vision measurements for stereo vision with two quadrocopters that have an overlapping field of view. Recently, this EKF formulation has been used in [123, 143, 257] for various applications. In [123], the initial concept of *Flexible Stereo* was introduced, further constraining the EKF formulation with a probabilistic wing model. The approach allows obstacle avoidance with a reactive controller as the depth map is computed with respect to the left (or right) wing tip camera.

In contrast to [123], this paper suggests to use a trinocular setup to enable the generation of a geo-referenced and globally consistent map. Furthermore, we suggest a vision update that utilizes photometric feature tracking in combination with relative inertial data to obtain an improved initial position. Whilst [123] is only tested in simulation, this publication demonstrates the real-world application.

3 Contributions And Assumptions

This paper incorporates recent advances in relative visual-inertial state estimation [5, 82, 123] and contains the following contributions:

- A procedure to obtain a probabilistic wing model that considers the aerodynamic forces acting on the wing during the flight.

- Implementation and validation of an efficient framework to estimate the time-varying relative transformation between multiple stereo setups.

- Real world experiments, including platform design and hardware considerations.

To focus on the description of the above-mentioned contributions we assume that the center camera is aligned with the map throughout the paper. For flying in GPS-denied environments this can be realized for instance with [119] or [82]. As the state estimator implemented on our autopilot already provides pose priors in real-time we instead employ the approach suggested in [118] but augment it with IMU edges. In summary, the proposed framework assumes the following input: 1) features with corresponding depth estimates tracked in the center camera for relative camera-camera alignment, 2) pose estimates of the center camera in the map frame for depth map registration.

4 The Approach

In Sec. 4.1 we present our approach to align the wing camera's pose to the center camera. All steps in this section follow the paradigm of staying close to the optimal relative pose to increase the efficiency of the photometric image alignment and relative pose estimation. Sec. 4.2 shortly summarizes the generation of the depth maps while Sec. 4.3 describes our procedure to obtain a probabilistic wing model.

4.1 Inertial-photometric alignment of wing to center camera

The pose of a wing camera is aligned with the center camera using an EKF in relative formulation [5], efficiently fusing relative IMU measurements, photometric visual cues, and a probabilistic wing model [123]. The state vector is defined as follows

$$\mathbf{x} = \left[\bar{q}_c^j,\ \omega_c^c,\ \omega_j^j,\ \mathbf{p}_c^j,\ {}_w\mathbf{v}_c^j,\ {}_w\mathbf{a}_c^c,\ {}_w\mathbf{a}_j^j,\ \mathbf{b}_{\mathbf{a}}^j,\ \mathbf{b}_{\omega}^j\right]^\top \tag{8.1}$$

with $j \in \{\text{left}, \text{right}\}$. In comparison to [5, 123], the state is augmented to compensate for the IMU bias. Formulating the relative pose alignment in a relative EKF is efficient and modular, allowing to easily add additional visual-inertial sensor rigs to, e.g., further increase the overall field of view.

Based on the last iteration's estimate of $\mathbf{T}_{C_{c,k-1}}^{C_{j,k-1}}$, the relative transformation is propagated to the time stamp of the current stereo frame using the IMU measurements of the center and wing camera (as outlined in [5, 123]), resulting in an initial prior for $\mathbf{T}_{C_{c,k}}^{C_{j,k}}$ at time step k. This relative pose prior is then used to project

features observed in the center image into the wing camera via

$$d = \|\mathbf{p}_{\text{lm}}^{W} - \mathbf{p}_{C_{c,k}}^{W}\|_2$$
$$\mathbf{u}_j = \Pi(\mathbf{T}_{C_{c,k}}^{C_{j,k}} \cdot \mathbf{C}_c \cdot d) \tag{8.2}$$

where $\mathbf{p}_{\text{lm}}^{W}$ is the estimated landmark position, d is the observed sparse depth estimate corresponding to this landmark, $\Pi(\cdot)$ is the projection operator, and $\mathbf{C}_c$ is the bearing vector from the center camera to the landmark.

Patches around the expected set of feature positions $\mathbf{u}_j$ are extracted from the wing camera image and the relative transformation aligning the wing to the center camera is further refined using sparse image alignment. The photometric error is minimized as proposed in [80, 82], using a constrained Gauss-Newton solver with pyramidal layers:

$$\mathbf{T}_{C_{c,k}}^{C_{j,k}} = \arg\min_{\mathbf{T}_{C_{c,k}}^{C_{j,k}}} \frac{1}{2}\sum_i \|\delta\mathbf{I}(\mathbf{T}_{C_{c,k}}^{C_{j,k}}, \mathbf{u}_{j,i})\|_{\Sigma_I}^2$$
$$+ \frac{1}{2}\|\mathbf{p}_{C_{c,k}}^{C_{j,k}} - \bar{\mathbf{p}}_{C_c}^{C_j}\|_{\Sigma_\mathbf{p}} + \frac{1}{2}\|\log(\bar{\mathbf{R}}_{C_c}^{C_j\top}\mathbf{R}_{C_{c,k}}^{C_{j,k}})^\vee\|_{\Sigma_\mathbf{R}}. \tag{8.3}$$

For the derivation and notation details we refer to [80, 82]. As motion constraint we choose the transformation prior $\bar{\mathbf{T}}_{C_c}^{C_j} = (\bar{\mathbf{R}}_{C_c}^{C_j}, \bar{\mathbf{p}}_{C_c}^{C_j})$ obtained from the wing model. Alternatively, the current mean and covariance from the EKF could be used. While the translation computed in the vision update presented in [5, 123] is only up to scale, the approach described in this publication computes a full-scale translation. The relative pose from the vision step is then fused in the EKF with the Gaussian wing model as in [123], for the case that the vision step fails, and the framework continues with the next iteration.

Discussion In our experiments, above approach has shown to be a robust way to transfer and track features from one to another camera under challenging conditions (e.g. different illumination or exposure time) as the feature locations are constrained by the relative transformation and are not tracked individually. On the other hand, tracking each feature patch individually, for instance with a Lucas-Kanade (LK) tracker, was not robust and resulted in many outlier matches. Note that this LK tracker implementation showed a good performance for establishing time-sequential frame-to-frame correspondences for the *same* camera.

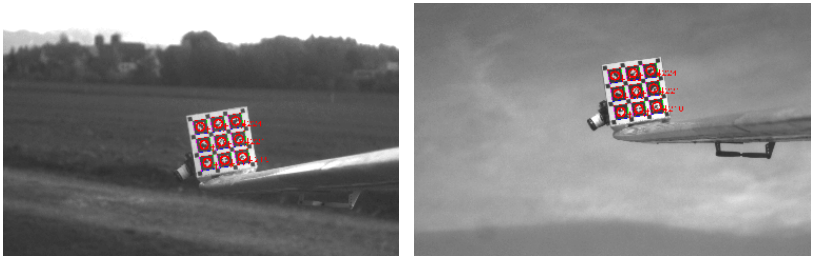
4.2 Dense reconstruction

The optimized camera poses are used to rectify the images using the algorithm proposed in [91] with subsequent stereo block-matching (BM). Based on the three cameras, the following combinations of depth maps can be computed: $\mathbf{d}_{c,l}^{c}$, $\mathbf{d}_{c,r}^{c}$, and $\mathbf{d}_{l,r}^{l}$ with their different characteristics visualized in Fig. 8.2. For instance $\mathbf{d}_{c,l}^{c}$

denotes the depth map computed from the left and center image, expressed in the center camera.

4.3 Obtaining a wing model

The probabilistic wing model used in this paper is represented by a Gaussian mean and covariance for the relative transformation between wing tip (left, right) and center, fuselage IMU. In order to capture the aerodynamic forces acting on the wing the relative transformation needs to be measured *in-air* as shown by Fig. 8.4: The pictures are recorded by the right side camera mounted rigidly inside the fuselage looking onto the wing towards the right wing camera (with attached April tag) and illustrate the aerodynamic lift force acting on the wing.



(a) On-ground **(b)** In-air

Figure 8.4: Side view from camera rigidly mounted inside the fuselage, looking onto the right wing tip camera.

Our proposed calibration process consists of estimating

1. the rigid transformation $\mathbf{T}_{\text{tag}_j}^{C_j}$ (in the lab under static conditions) and
2. the time-varying transformation $\mathbf{T}_{\text{tag}_j}^{C_{\text{side},j}}$ (using data recorded during a flight).

Firstly, we describe how to obtain the rigid transformation $\mathbf{T}_{\text{tag}_j}^{C_j}$ between the wing camera and the wing tag: The experiment setup and involved coordinate frames are shown in Fig. 8.5. The rigid relative transformation between the wing camera and wing tag is given by

$$\mathbf{T}_{\text{tag}_j}^{C_j} = \mathbf{T}_{C_c}^{C_j}\mathbf{T}_{C_{\text{side,j}}}^{C_c}\mathbf{T}_{\text{tag}_j}^{C_{\text{side,j}}} \tag{8.4}$$

where the rigid transformations $\mathbf{T}_{C_c}^{C_j}$ and $\mathbf{T}_{C_{\text{side,j}}}^{C_c} = \mathbf{T}_{B_c}^{C_c}\mathbf{T}_{C_{\text{side,j}}}^{B_c}$ are calibrated with *Kalibr* [220]. Note that $\mathbf{T}_{\text{tag}_j}^{C_j}$ could also be obtained by solving the hand-eye

problem (e.g. with [87]) but requires access to an accurate motion capture system (e.g. *Vicon*). Instead, our procedure relies only on the synchronized images and inertial measurements from the VI-sensor.

Secondly, during the flight, the April tags on the wing cameras are detected by the left respectively right side camera. Based on the April tag detection, the relative transformation $\mathbf{T}_{\text{tag}_j}^{C_{\text{side,j}}}$ is obtained from an absolute pose estimator (PNP). That is, for every frame, the camera pose of the wing camera with respect to the center camera can be computed as

$$\mathbf{T}_{C_c}^{C_j} = \mathbf{T}_{\text{tag}_j}^{C_j} \mathbf{T}_{C_{\text{side,j}}}^{\text{tag}_j} \mathbf{T}_{C_c}^{C_{\text{side,j}}}. \tag{8.5}$$

Note that the side cameras and April tags need to be mounted only once for every UAV type in order to establish the wing model.

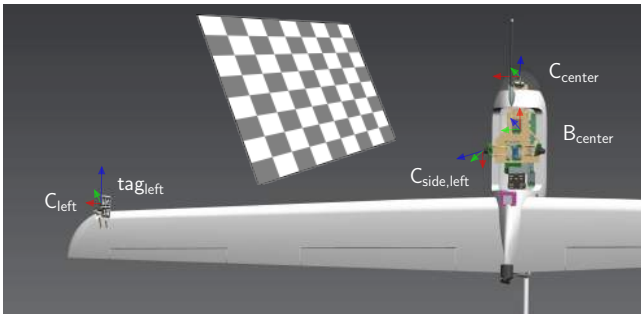


Figure 8.5: Coordinate frames involved in the calibration procedure to obtain a probabilistic wing model (left side).

5 Platform And Hardware

Fig. 11.1 gives an overview of the employed UAV platform *Techpod* and of the VI-sensor system integration. Due to specific hardware requirements, such as the long baseline between the wing and center camera, commercially available VI-sensors [195] could not be used and a customized VI-sensor based on the *Atmega328P* microcontroller unit (MCU) was developed: The Camera-IMU rig for the left and right wing each consist of a 0.3 MP *Aptina MT9V034* camera and an IMU of type *MPU6050*, rigidly mounted behind the camera. The IMU measurements are triggered via interrupts and transferred to the MCU. Due to the required length of the data lines, three i^2C repeaters are required: one next to the MCU and two next to the Camera-IMU rigs on the wing just before the signal enters the *MPU6050*.

The spatially closer and more accurate center IMU (*ADIS16448*) is connected to the MCU via SPI. The three IMUs and the five cameras of same type (three in normal mode, two additional for the wing calibration) are time-synchronized via hardware lines and run at 100 Hz and up to 20 Hz respectively. The rigid, non-time-varying transformations $\mathbf{T}_{B_{\text{center}}}^{B_{\text{px4}}}$ and $\mathbf{T}_{C_j}^{B_j}$ with $j \in \{\text{left}, \text{right}, \text{center}\}$ are calibrated offline with *Kalibr* [220], where B_{px4} refers to the IMU (*ADIS16448*) used by the *pixhawk* Autopilot. An *UP Squared* with *Intel Atom* CPU at 1.6 GHz was used as on-board computer (OBC).

6 Experiments

6.1 Inertial measurements

In particular for the inertial measurement units mounted on the wing tips strong vibrations are to be expected. To quantify this effect, we compare *measured* to *expected* IMU readings. The camera poses $\mathbf{T}_{C_j}^{W}$ are optimized offline (using GPS and vision measurements only) and transformed to the IMU poses $\mathbf{T}_{B_j}^{W}$ via the rigid transformation $\mathbf{T}_{B_j}^{C_j}$. Using our implementation of [246] the expected linear accelerations and angular velocities are computed. The expected and measured (raw, not de-biased) inertial measurements for the center fuselage camera are shown in Fig. 8.6a, 8.6b. The IMU *ADIS16448* is used with a measurement sensitivity of $\pm 500°/s$, FIR filter, and four filter taps. Fig. 8.6c shows the measured (raw, not de-biased) accelerometer readings for the left wing IMU using *MPU6050*'s on-board digital low-pass filter (DLPF) with a cut-off frequency $f_c = 94$ Hz. Fig. 8.6d plots the solution from a low-pass filter with $f_c = 3$ Hz and moving-average filter with window size of 50 on the right. Based on these results, we conclude to use *MPU6050*'s on-board DLPF at $f_c = 5$ Hz (minimal available cut-off frequency) and compensate for the introduced constant, known delays.

6.2 Probabilistic wing model

Fig. 8.7 plots the measured relative transformation between the April tag attached to the right wing camera and the right side camera $\mathbf{T}_{C_{\text{side,right}}}^{\text{tag}_{\text{right}}}$. The experiment begins on the ground with take-off approximately at the 162.5 s mark. The relative translation and orientation vary within 5 cm and 6° respectively during the duration of the experiment (including on-ground detections). The take-off is particularly well observable in the rotation around the z-axis. The observed transformation $\mathbf{T}_{C\text{side,j}}^{\text{tag}j}$ is then transformed to $\mathbf{T}_{Bc}^{Bj}$ via the rigid transformation chain given in Eq. 8.5 which is stored in the EKF in form of a mean and variance for the left-center and center-right stereo pair.

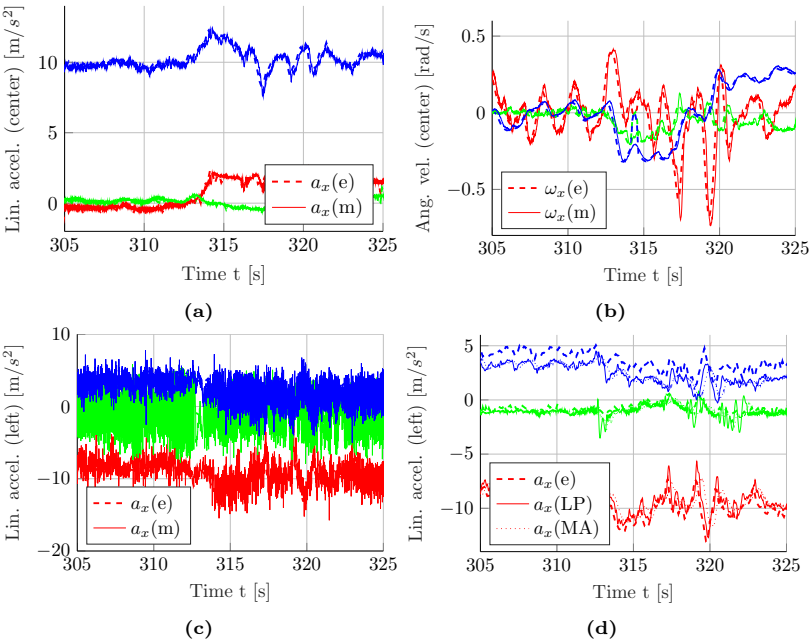


Figure 8.6: Expected (e), measured (m) and filtered (low-pass: LP, moving average: MA) inertial measurements, recorded by the center and left wing tip IMU. The color coding is x (red), y (green), z (blue) throughout the paper.

6.3 Vision update

Fig. 8.8 visualizes the quality difference between the relative camera-to-camera transformation priors, as well as the photometric alignment of the left to the center camera. By *Calib-ground* we denote the transformation estimate $\mathbf{T}_{C_c}^{C_l}$ obtained using *Kalibr* in a static setup and with no load applied to the wings. In contrast, *Calib-air* represents the mean of our wing model (cf. Sec. 4.3). Features tracked in the center camera are projected into the left camera according to Eq. 8.2: The features re-projected with the rigid relative transformation $\mathbf{T}_{C_c}^{C_l}$ obtained from *Calib-ground* are shown in green, the features re-projected using the mean of the wing model (*Calib-air*) are shown in red. One can observe that the features obtained from *Calib-ground* are far off from the correct feature position, while *Calib-air* returns a relative good initial position in particular during level flight. The feature positions refined by the photometric sparse image alignment are marked in blue

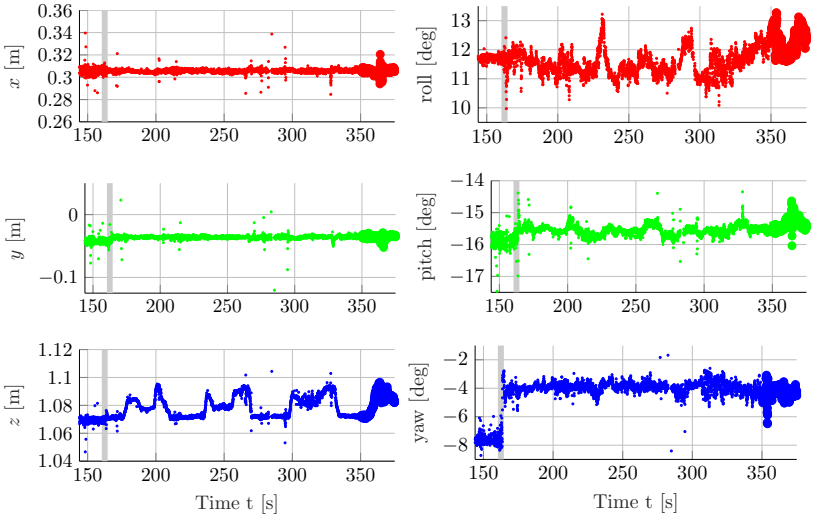


Figure 8.7: Observations of the April Tag attached to the right wing camera, as seen from the right side camera. The grey vertical line marks the take-off (landing is not shown).

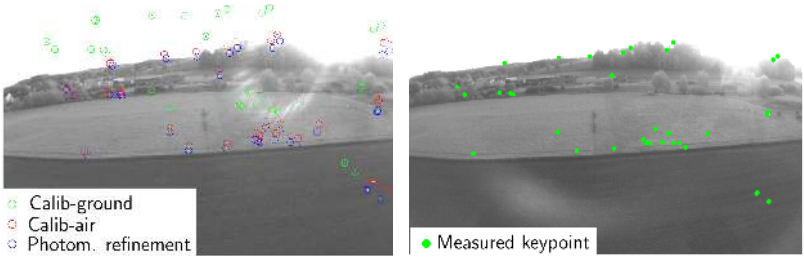
(Eq. 8.3). The photometric formulation shows a robust performance even under challenging conditions such as the lense flare.

6.4 Depth map

Fig. 8.9 depicts the depth maps for the left-center stereo pair, computed using efficient block-matching (BM) [33], with $\mathbf{T}_{C_c}^{C_l}$ from b) *Calib-ground*, c) *Calib-air*, and d) our proposed framework. Areas with low texture variation can potentially be filled in by using different stereo pairs and by observing the area from different viewpoints.

6.5 Runtime

The runtime results of the main modules are given in Table 8.1, measured on an *Intel i7-4800MQ* CPU at 2.70 GHz for comparison. The combination of relative IMU integration, good relative pose prior for the sparse photometric image alignment, and the employed EKF formulation makes the framework extremely efficient. With the tested frame rate of 10 Hz the image stream can be processed with some margin.



(a) Cam left **(b)** Cam center

Figure 8.8: Wing model prior and photometric refinement step for the alignment of the wing and center camera image.

Module	mean [ms]	Remark
Relative IMU integration (0-order)	0.03	Frame to frame
Photometric refinement	3.2	Per frame
Rectification, block-matching	38.1	Per stereo pair

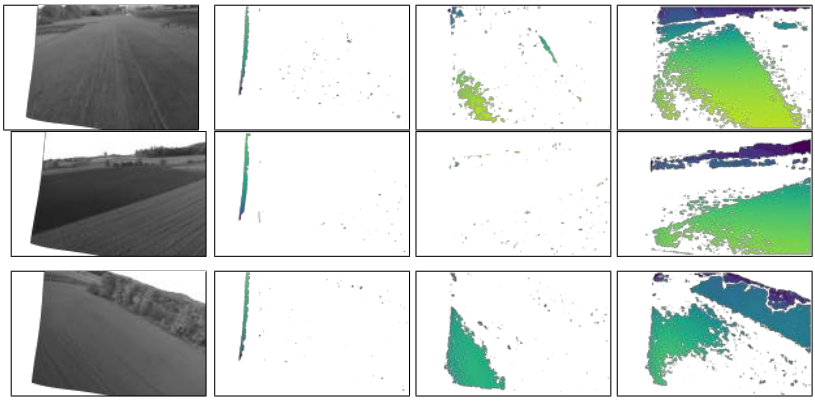
Table 8.1: Runtime results

Furthermore, the modular formulation makes it possible to run several visual-inertial stereo pairs in different threads. At this point, the employed rectification and block matching algorithm appears to become the first bottleneck if the image rate is to be set to a higher value.

7 Conclusions And Future Work

In this paper, we further developed the idea of using visual-inertial sensor systems on-board a UAV for improved environmental awareness and demonstrated the effectiveness in real-world experiments. We investigated the challenges encountered in the three modules of the framework: The problem of measuring strong high-frequency vibrations on the wing tip IMUs was solved in software in form of a low-pass filter. In future work, we aim at a solution that minimizes the vibrations already on the hardware level. Since the relative transformation is estimated, a flexible mount or damping material could be used to isolate the visual-inertial system from the motor and wind gust induced vibrations encountered on the wing.

Furthermore, we described our procedure to obtain a probabilistic wing model, formulated as a Gaussian prior, and showed the discrepancy to the on-ground calibration. In future work, a more sophisticated wing model could be identified. For instance, the wing could be modeled as a cantilever beam with additional inputs



(a) Rectified center camera **(b)** Calib-ground **(c)** Calib-air **(d)** Proposed

Figure 8.9: Single shot depth maps obtained from the left and center image, expressed in the rectified center camera.

such as air speed. As the deformation of the wing influences the lift distribution our model and accelerometer readings could also be incorporated into the on-board controller, e.g. for wind gust rejection.

Finally, in contrast to [123], we used a photometric sparse image alignment formulation to compute the relative transformation between wing and center camera. This approach has shown to be a robust way to transfer and track features from one to another camera under challenging conditions. In a next step, we intend to test our framework on a platform with stronger wing flapping behavior such as (manned) gliders.

Acknowledgment

We wish to thank: T. Stastny (experiments and discussion), T. Steiger and Y. Allenspach (safety pilots), A. Ruckli (robustification of April tag detection), T. Mantel and A. Melzer (help with VI-Sensor).

Paper VI

Sparse And Deep Inverse Compositional Lucas-Kanade Algorithm on SE(3)

Timo Hinzmann and Roland Siegwart

Abstract

This paper introduces SD-6DoF-ICLK, a learning-based Inverse Compositional Lucas-Kanade (ICLK) pipeline that uses sparse depth information to optimize the relative pose that best aligns two images on SE(3). To compute this six Degrees-of-Freedom (DoF) relative transformation, the proposed formulation requires only sparse depth information in one of the images, which is often the only available depth source in visual-inertial odometry or Simultaneous Localization and Mapping (SLAM) pipelines. In an optional subsequent step, the framework further refines feature locations and the relative pose using individual feature alignment and bundle adjustment for pose and structure re-alignment. The resulting sparse point correspondences with subpixel-accuracy and refined relative pose can be used for depth map generation, or the image alignment module can be embedded in an odometry or mapping framework. Experiments with rendered imagery show that the forward SD-6DoF-ICLK runs at 145 ms per image pair with a resolution of 752×480 pixels each, and vastly outperforms the classical, sparse 6DoF-ICLK algorithm, making it the ideal framework for robust image alignment under severe conditions.

Published in:
CoRR, 2020

1 Introduction

Robust image alignment under challenging conditions is an important core capability towards safe autonomous navigation of robots in unknown environments. This paper focuses primarily on the ICLK [16, 178] algorithm that optimizes the alignment between two images on SE(3) by utilizing dense or sparse depth information. One advantage of this approach in contrast to feature-based alignment is that a costly outlier rejection step can be avoided. Applications of six-DoF-ICLK include rigid stereo extrinsics refinement after shocks, visual-inertial odometry systems [82], or non-rigid stereo pair tracking [125]. In this paper, the 6DoF-ICLK is leveraged with the help of DL to make parameters in the framework trainable and the image alignment robust against e.g., challenging light conditions.

2 Related Work

Image alignment approaches can be divided into feature-based and direct methods. Feature-based methods have come a long way from computationally expensive hand-crafted feature detectors and descriptors (SIFT [176], SURF [19]) to faster binary variants (BRISK [158]) and recently trainable approaches (SuperPoint [60], D2-Net [69]). Likewise, direct methods have seen many variants and may rely on dense (DTAM [192]) or sparse depth information (SVO [82]), Mutual Information based Lucas-Kanade tracking [68], and more recently DL-variants of ICLK [181]. In this context, we propose SD-6DoF-ICLK, a learning-based sparse Inverse Compositional Lucas-Kanade (ICLK) algorithm for image alignment on SE(3) using sparse depth estimates. The implemented SD-6DoF-ICLK algorithm is optimized for Graphics Processing Unit (GPU) operations to speed up batch-wise training. The output of the framework consists of sparse feature pairs in both images and the relative pose connecting the two cameras. The feature locations and the estimated relative pose can be further refined using an individual feature alignment step with a subsequent pose, and optional structure refinement, as proposed in [82]. For depth map generation, the estimated relative pose can be fed to classical rectification and depth estimation modules, or to DL-based depth from multi-view stereo algorithms.

3 Methodology

Our proposed SD-6DoF-ICLK framework is depicted in Fig. 9.1: The input is a grayscale or colored reference image $\mathbf{I}_0$ with sparse features. For every sparse feature, a depth estimate is assumed to be known. The objective is to find the relative 6DoF pose $\mathbf{T}_{C_0}^{C_1}$ that best aligns the reference image $\mathbf{I}_0$ to $\mathbf{I}_1$, where $\mathbf{I}_1$ may also be grayscale or colored but contains no depth information or extracted features. To align the images on SE(3) with the described input data, we adopt [181] to sparse depth information as follows: The input images $\mathbf{I}_0$ and $\mathbf{I}_1$ are color normalized and fed as single views to the feature encoder described in [181]. This operation returns four pyramidal images with resolution 752×480 (level 0, input

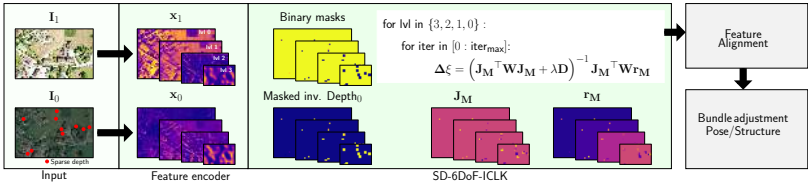


Figure 9.1: Overview of SD-6DoF-ICLK with feature, pose, and structure refinement.

image resolution), 376×240, 188×120 and 94×60 (level 3). The sparse image alignment algorithm described in [82] is designed for CPU operations and explicitly iterates over the pixels of every patch. Instead, we formulate the problem with binary masks to exploit the full advantage of indexing with GPUs and to allow fast batch-wise training and inference. To achieve this, binary masks are created at every level with a patch size of 8×8 pixels surrounding the sparse feature. Similarly, the sparse inverse depth image for image $\mathbf{I}_0$ is generated by setting every pixel within the patch to the inverse depth value.

Fig. 9.1 shows the pseudo-code of the inverse compositional algorithm that, starting from the highest level, iterates over all pyramidal images. For every pyramidal image, the warp parameters are optimized using the Levenberg-Marquardt update step [181, 182]:

$$\mathbf{\Delta}\xi = \left(\mathbf{J_M}^\top \mathbf{W} \mathbf{J_M} + \lambda \mathbf{D}\right)^{-1} \mathbf{J_M}^\top \mathbf{W} \mathbf{r_M} \tag{9.1}$$

where $\mathbf{J_M}$ and $\mathbf{r_M}$ denote the Jacobian $\mathbf{J}$ and residual $\mathbf{r}$ after applying the binary mask of the corresponding pyramidal layer. The convolutional M-Estimator proposed in [181] is used to learn the weights in $\mathbf{W}$. The damping term λ is set to $1e^{-6}$ throughout the experiments.

As shown in Fig. 9.1, the framework continues with a feature alignment step and bundle adjustment step to achieve subpixel accuracy [82].

4 Experiments & Results

Implementation The SD-6DoF-ICLK algorithm and feature alignment step is implemented in pytorch [203] and adopted from [82, 181]. The pose is then refined with GTSAM [57] using a Cauchy loss, to reduce the influence of outliers, and Levenberg-Marquardt for optimization.

Datasets To generate a large amount of training, validation, and test data we implemented a shader program in OpenGL [273] that takes an orthomosaic and elevation map as input and renders an RGB and depth image given a geo-referenced

pose $\mathbf{T}_C^W$, camera intrinsics matrix $\mathbf{K}$, and distortion parameters [116]. No distortion parameters are set in this paper as the ICLK algorithm expects undistorted images as input. Camera positions and orientations of pose pairs are uniformly sampled from locations above the orthomosaic and rejected if the camera's field of view is facing areas outside the map. Training and test data are rendered from two different satellite orthomosaics selected from nearby but non-overlapping locations. The validation set[1] is drawn from the shuffled training data. The OpenGL renderer's task is to augment the training and test data geometrically. Appearance variations are generated using pytorch's build-in color-jitter functionality that randomly sets brightness, contrast, saturation, and hue. This is to emulate challenging light conditions like over-exposure or lens flares. This augmentation creates 10 new, color-altered images for every inserted original image. Sparse features are drawn randomly from a uniform distribution such that $\mathbf{p} = [\mathcal{U}(b, W-b); \mathcal{U}(b, H-b)])$ with border $b = 2 \cdot P + 1$, patch width $P = 8$, image width W and image height H. The number of sparse features for all training, validation, and test set is set to 50.

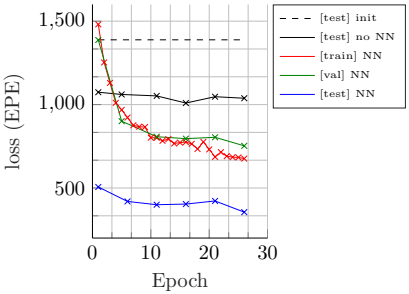


Figure 9.2: Training, validation and test loss (3D End-Point-Error EPE) over 26 epochs.

Param	Value
Batch size	4 (max. memory)
Epochs	26
Num. images train/val	20 000
Validation split	0.2
Num. images test	800
Input image resolution	752×480
Max. iterations	10
Optimizer	SGD
Initial learning rate	$1.0e^{-5}$
Momentum	0.9
Nesterov	False
Weight decay	$4e^{-4}$
Learning rate decay epochs	$[5, 10, 20]$
Learning rate decay ratio	0.5
Validation frequency	5
Test frequency	5

Table 9.1: Training, validation, and test parameters used for session visualized in Fig. 9.2.

Results Fig. 9.2 present the results from a training, validation, and test session over 26 epochs. Training parameters are listed in Tab. 9.1. Analogue to [181], the utilized training loss is the 3D End-Point-Error (EPE) using the rendered dense depth map. After epoch 10, the training loss (red) is below the validation loss (green) and continues to decrease. As the appearance of the images is randomly changed, it may occur that the validation is below the training loss as seen here in the initial set of epochs. On every fifth epoch, the test set is evaluated. The loss of the test set given the initial, incorrect relative transformation is illustrated by the black dashed line for reference. The solid black line is the test set evaluated

[1] The validation split is set to 20% of the total set dedicated for training and validation.

	Initial	SD-6DoF-ICLK	Feature Alignment	Pose Optimization
e_{pixel}	33.986	1.286	0.413	0.121
$e_{\text{transl.}}$	4.934	3.257	3.257	0.089
$e_{\text{rot.}}$	0.075	0.020	0.020	0.000

Table 9.2: Pixel, translational, and rotational error for the subsequent image alignment steps.

for 6DoF-ICLK without learning (also max. 10 iterations) and returns roughly the same result independent of the epoch as expected. Evaluating the test set with SD-6DoF-ICLK demonstrates that the classical 6DoF-ICLK is clearly outperformed. Tab. 9.2 shows the pixel (euclidean distance), translational, and rotational errors that continuously decrease for the subsequent image alignment steps. These results are visualized in Fig. 9.3: The first row shows $\mathbf{I}_1$ with the features projected from $\mathbf{I}_0$ based on the current estimate for the relative transformation $\mathbf{T}_{C_0}^{C_1}$. The red lines illustrate the error with respect to the ground truth feature location. Given the estimate for $\mathbf{T}_{C_0}^{C_1}$, $\mathbf{I}_1$ can be overlaid over $\mathbf{I}_0$, which is shown in the second row.

Figure 9.3: Qualitative results of the image alignment framework.

Runtime The inference time of the SD-6DoF-ICLK is 145 ms on a GeForce RTX 2080 Ti (12GB) for an image resolution of 752×480 pixels, four pyramidal layers, and a maximum of 10 iterations of the incremental optimization. Note that the image resolution of 752×480 was selected for training and forward inference, as it represents the target camera resolution on our UAV. A smaller resolution, however, could be set to speed up the training process and to increase the batch size if desired.

5 Conclusion and Outlook

In this work, we introduced SD-6DoF-ICLK, a learning-based sparse Inverse Compositional Lucas-Kanade (ICLK) algorithm, enabling robust image alignment on SE(3) with sparse depth information as input, and optimized for GPU operations. A synthetic dataset rendered with OpenGL shows that SD-6DoF-ICLK outperforms the classical sparse 6DoF-ICLK algorithm by a large margin, making the proposed algorithm an ideal choice for robust image alignment. The proposed SD-6Dof-ICLK is able to perform inference in 145 ms on a GeForce RTX 2080 Ti with input images at a resolution of 752×480 pixels. To further refine the feature locations and relative pose, individual feature alignment with subsequent pose and, if required, structure refinement is applied as proposed in [82]. In future work, the framework could be embedded into an odometry or mapping framework, or used for depth image generation.

Part C

UAV MISSIONS

Paper VII

Free LSD: Prior-Free Visual Landing Site Detection for Autonomous Planes

Timo Hinzmann, Thomas Stastny, Cesar Cadena, Roland Siegwart, Igor Gilitschenski

Abstract

Full autonomy for fixed-wing unmanned aerial vehicles (UAVs) requires the capability to autonomously detect potential landing sites in unknown and unstructured terrain, allowing for self-governed mission completion or handling of emergency situations. In this work, we propose a perception system addressing this challenge by detecting landing sites based on their texture and geometric shape without using any prior knowledge about the environment. The proposed method considers hazards within the landing region such as terrain roughness and slope, surrounding obstacles that obscure the landing approach path, and the local wind field that is estimated by the on-board EKF. The latter enables applicability of the proposed method on small-scale autonomous planes without landing gear. A safe approach path is computed based on the UAV dynamics, expected state estimation and actuator uncertainty, and the on-board computed elevation map. The proposed framework has been successfully tested in an offline process, operating in a batch scheme using photo-realistic synthetic and in challenging real-world datasets.

Published in:
IEEE Robotics and Automation Letters (RA-L), 2018

1 Introduction

Small-scale autonomous planes promise to become a ubiquitous tool in the commercial, industrial, and scientific sectors due to reduced operational costs and ever increasing robustness. Especially the ability to map large areas and to carry out perpetual surveillance tasks, e.g. by using a solar-powered platform, makes this type of unmanned aerial vehicles (UAVs) interesting for various applications. While mission operation can already be completely automated [200], appropriate landing site detection (LSD) and the actual landing procedure still requires an experienced safety pilot. Furthermore, in future fully autonomous beyond visual line-of-sight (BVLOS) operation, finding an appropriate landing spot in unstructured terrain is essential for handling emergency scenarios.

Existing LSD systems focus on the cases of vertical takeoff and landing (VTOL) platforms, or large-scale planes, may rely on offline-computed data, or require prior knowledge about the environment. These approaches are not suited for small-scale autonomous planes operating in unknown environments which are constrained by potentially limited energy supply and computational power. Furthermore, their size and speed requires taking the wind into consideration, and due to their potential absence of landing gear, preferably landing in flat grass to not damage wings or fuselage.

The present work proposes a real-time visual landing site detection and approach path computation algorithm for autonomous fixed-wing UAVs. To keep the problem complexity manageable, potential landing sites are tracked and ranked over multiple frames. Only the most promising landing sites are forwarded for finer grained, 3 D processing. No a priori data such as markers, pre-classified Digital Surface Maps (DSM), or orthomosaics are utilized which allows the framework to be operated in completely unknown terrain as is exemplary shown in Fig. 10.1. To the best of the authors' knowledge, this paper presents the first such system, which is also suitable for application on small-scale UAVs. The work incorporates wind field and nearby obstacle consideration during approach path generation and decision making. Performance of the full framework is evaluated in unknown terrain using various synthetic datasets and real-world test flights.

2 Related Work

Automated landing of VTOL UAVs has been considered in a broad body of works. For instance, Desaraju et al. [59] propose a vision-based landing site evaluation framework to land on rooftops employing a Gaussian process to estimate the landing site confidence. Forster et al. [83] present an efficient way to compute a vision-based elevation map on-board of a quadrocopter. Johnson et al. [138] use a LIDAR-based elevation map to compute terrain smoothness, roughness, and incidence angles to determine safe landing spots for spacecrafts. Garcia-Padro et al. [95] introduce a contrast descriptor to land an autonomous helicopter far away from obstacles under the assumption that the terrain is flat. Brockers [37], Cheng [46], and Bosch

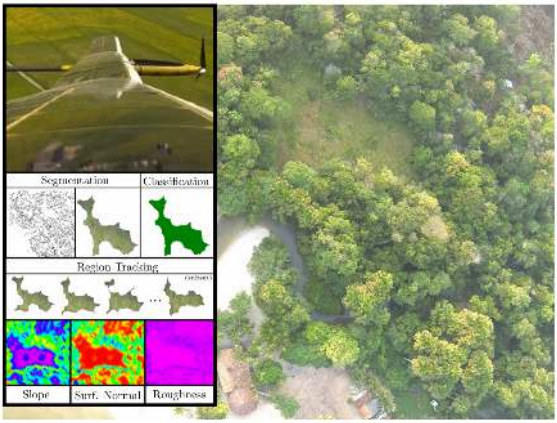


Figure 10.1: The goal is to find the optimal landing spot while considering terrain shape, terrain texture, terrain roughness, terrain slope, surrounding obstacles, estimated local wind field, and UAV dynamics and their uncertainties.

et al. [31] make use of homography decomposition for identifying planar landing spots. Theodore et al. [259] employ a stereo vision rig mounted on an unmanned helicopter to compute a range map and infer safe landing spots based on roughness, slope and distance to closest obstacle. The above approaches have in common that the main criterion for VTOL UAVs is flatness of the landing spot. However, our application requires taking the plane dynamics and additional space requirements during landing into consideration.

For fixed-wing platforms, most research focuses on cases where the system recognizes modified environments or man-made structures, or where the landing site is pre-defined. Visual servoing is employed by Huh et al. [132] to steer a small fixed-wing UAS into a red dome-shaped airbag located in an obstacle-free area. Similarly, the framework proposed by Laiacker et al. [154] recognises a runway from the UAV and compares it to a known model. Given a designated, obstacle-free landing site, the height above the ground plane can be estimated using monocular visual-inertial [17] or biologically inspired stereo vision [260].

In contrast to the aforementioned works, this paper aims at actively selecting appropriate landing spots in an *unknown* environment. This requires generation and assessment of potential candidate areas which has, to the best of our knowledge, only been discussed in two publications. Fitzgerald et al. [78] seek to find suitable areas for crash-landing an airplane in case of emergency. This is achieved by detecting areas without edges on a low-quality image from a defined height of 2500 ft, before classifying them in order to retrieve large grass fields. However, relying on a fixed

height makes this approach disadvantageous in case of emergencies. The closest approach to ours is presented by Warren et al. [270]. However, we see the following caveats that we address with the present work. Firstly, the terrain classification is derived from stored data.Secondly, the approach trajectory and height of nearby obstacles is only considered indirectly by the Principal Component Analysis (PCA). Thirdly, wind is not considered which has a large effect on smaller and light-weight planes. Finally, the approach by Warren et al. [270] does not run in real-time.

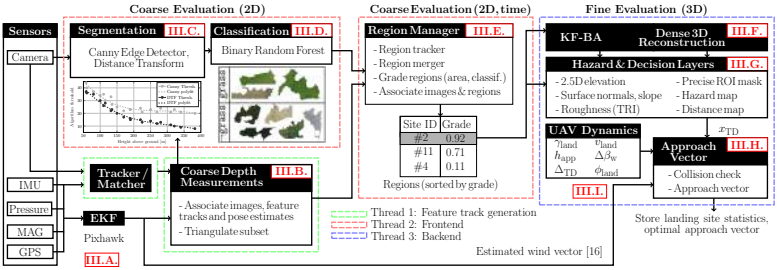


Figure 10.2: Proposed framework for prior-free landing site detection. The frontend segments, classifies, and manages the potential region of interests (ROIs). The most promising site is forwarded to the backend for finer 3D analysis and computation of approach procedure.

3 The Approach

An overview of our proposed algorithm for the detection of landing sites is shown in Fig. 11.1: The raw image is segmented into homogeneous regions (Sec. 3.3) and classified into `grass` or ¬`grass` using a binary Random Forest (RF) classifier (Sec. 3.4). In parallel, the on-board EKF of the *Pixhawk* autopilot estimates UAV pose and local wind field (Sec. 3.1), and depending on the provided image rate and overlap of subsequent frames, a tracker or matcher is employed to connect consecutive camera frames via feature tracks. Resulting coarse depth measurements (Sec. 3.2) are used in the region manager to track region of interests (ROI) based on geometry. The region manager (Sec. 3.5) accumulates all information about the regions and ensures consistency and uniqueness by merging regions. Based on these metrics, a coarse grade determines which region is passed on as a candidate to the fine, 3D evaluation backend. The fine evaluation backend is periodically updated by the frontend with the n most promising ROIs. All observations of a ROI, UAV pose estimates, and previously generated feature tracks are used to perform key-frame based bundle adjustment (BA) and dense 3D reconstruction (Sec. 3.6). Metrics such as terrain slope and roughness (Sec. 3.7) are derived from the classification results and 3D model. A distance-to-hazard map determines the

landing spot with maximum distance to the next hazard. Based on this touch down point, the estimated local wind field (Sec. 3.8, 3.9), and the 2.5D elevation map, a collision-free approach path is computed. The final decision module outputs the landing site location, optimal approach vector, and statistics about the final landing site. The actual tracking of the final approach path is described in [200]. The metrics of the best landing sites are stored to be able to land quickly in the case of an emergency.

3.1 State Estimation

The state estimator on the *Pixhawk* autopilot estimates body poses, velocities, IMU biases, and the wind field using GNSS, IMU, magnetometer, and pressure measurements [160]. The camera pose estimates are forwarded to the on-board computer which associates camera poses to the corresponding images based on the pre-calibrated camera-IMU transformation [86]. These camera pose estimates are used as priors in the bundle adjustment if an area was marked as potential landing spot. Additionally, feature tracks are generated using, depending on the provided framerate, a Kanade-Lucas-Tomasi (KLT) [177] feature tracker or matcher. These feature tracks are used to generate coarse depth measurements (cf. Sec. 3.2 for region tracking in unknown terrain and in the bundle adjustment of the backend thread (cf. Sec. 3.6).

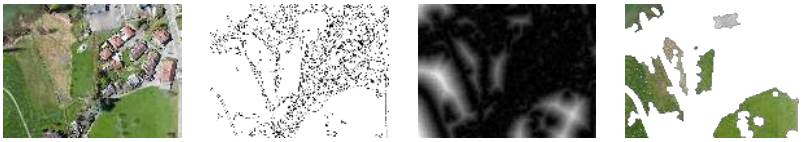
3.2 Coarse Depth Measurements

To obtain a segmentation that is robust to height changes as well as for geometric region tracking, coarse depth measurements are required in the frontend (cf. Sec. 3.5). Since our system is designed to operate in unknown terrain without a priori data, the depth measurements need to be retrieved at runtime[1]. One possibility would be to triangulate a few features at every step and build up a mesh by using, for instance, Delaunay triangulation [272]. However, to obtain depth measurements at a given pixel, computationally expensive ray-casting queries would be required. Furthermore, a depth image obtained from two views from a virtual stereo rig based on unoptimized camera pose estimates is prone to errors since we assume a noisy low-level state estimator. Instead, we take advantage of the feature tracking thread that is running in parallel: To map from 2D to 3D coordinates, the N feature tracks closest to the queried keypoint location are determined. The final height of the requested keypoint location is obtained by performing multi-view triangulation of the N nearby tracks and inversely weighing the resulting triangulated landmark heights by their distance to this keypoint.

[1] Depending on the application and flight altitude, the coarse depth measurements could be obtained from a ground plane approximation.

3.3 Region Segmentation

The Canny edge detector [41] is applied to the grayscale spectrum of the raw image (cf. Fig. 10.3a). The result is shown in Fig. 10.3b. Next, the distance transform [76] is applied to compute for every pixel the distance to the closest non-zero pixel or Canny edge. The distance map, as shown in Fig. 10.3c, is then thresholded to obtain homogeneous regions (cf. Fig. 10.3d). Note that high contrast obstacles, such as the trees in the lower right section of the images, are often already identified at this early stage. The threshold in the Canny edge detector and the distance transform is computed from a function of height, to ensure that the same areas are segmented independently of the UAV's altitude above ground[2] The thresholds are derived from Google Earth imagery and span an altitude range of 58-382 m above ground. For reference, the nominal flight altitude of the deployed UAVs in this publication is between 50 and 250 m .



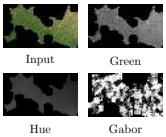
(a) **(b)** **(c)** **(d)**

Figure 10.3: Region segmentation: (a) Original input image, (b) Canny edges, (c) Distance transformation, (d) Segmented regions. Regions with a small area are rejected directly at this step.

3.4 Region Classification

The segmentation module presented in the previous section only ensures that the extracted area is homogeneous. In the classification step, the texture and color properties of the homogeneous area are extracted to classify the regions into `grass` or ¬`grass` as illustrated in Fig. 11.1. For this purpose, we employ a binary Random Forest (RF) [34] classifier which takes the segment from Fig. 10.3d and predicts the binary label. The classifier is trained based on a set of features extracted from the homogeneous regions. The parameters of the classifier, that is, the maximal tree depth and the number of samples needed per branch, are optimized on the training data by 10-fold cross-validation. The ground truth for the classification is established as follows: Homogeneous regions are obtained by the described segmentation algorithm. After visual inspection, the region is manually labeled as "grass" or "not grass".

[2] The height-dependent thresholds were approximated by $p_{\text{canny}}(h) = -1.72e-06h^3 + 0.00148h^2 - 0.43h + 62.97$, and $p_{\text{dtf}}(h) = -1.23e-06h^3 + 0.0011h^2 - 0.39h + 56.82$ as shown in Fig. 11.1.



Color	Texture
$\mu_{\mathrm{B}}, \mu_{\mathrm{G}}, \mu_{\mathrm{R}}, \sigma_{\mathrm{B}}, \sigma_{\mathrm{G}}, \sigma_{\mathrm{R}}$	$\mu_{\mathrm{G0.5}}, \mu_{\mathrm{G1.0}}, \mu_{\mathrm{G5.0}}$
$\mu_{\mathrm{H}}, \mu_{\mathrm{S}}, \mu_{\mathrm{V}}, \sigma_{\mathrm{H}}, \sigma_{\mathrm{S}}, \sigma_{\mathrm{V}}$	$\sigma_{\mathrm{G0.5}}, \sigma_{\mathrm{G1.0}}, \sigma_{\mathrm{G5.0}}$

Figure 10.4: Features used for binary classification of homogeneous regions. Note that the Gabor feature applied in form of a convolutional filter expands the area, but only the information within the ROI mask is used to compute mean and standard deviation.

Feature Space

For each segmented ROI, twelve color and six texture features, are extracted as summarized in Fig. 10.4.

Color For each subimage, the mean and standard deviation for all three color channels are computed across the complete segmented ROI. This is performed not only in the standard RGB color space, but also in the HSV space. In many computer vision applications, the HSV space has proven to be less sensitive to lighting conditions, when comparing to RGB [9]. While the classifier performs better using the HSV color space than RGB only, it performs even slightly better when using the features extracted from both: The classification error for only using RGB is 15.36 %, 14.26 % for HSV, and 14.12 % for RGB and HSV. We hence get a total of (2 color spaces) $\times$ (3 colors) $\times$ (2 features per color) $=$ 12 color features which are computed for every subimage. While more advanced color features can be extracted, e.g. using various combinations of color histograms [249], we here only rely on these very simple features for low computational costs.

Texture Color features are sensitive to illumination and viewing angle. To better assess the spatial arrangement of intensities in an image patch we additionally compute texture descriptors. For this task, we employ Gabor filters [93], linear filters related to the Gabor Wavelet that extract texture features from gray-scale imagery [136] more efficiently than alternatives, such as Local Binary Patterns (LBP) [101, 207]. The following parameters for phase offset φ, standard deviation of the Gaussian function σ, and spatial aspect ratio γ are used: $\varphi = 0$, $\sigma = 4$ and $\gamma = 0.02$. The orientation θ in which the edges are detected is not important in our case, since we try to detect rotation-independent descriptors. The Gabor filtered images are computed by applying a convolutional filter in four directions $\theta \in \{0, \pi/4, \pi/2, 3\pi/4\}$ and taking the mean of the extracted values. This approach yields three Gabor filtered images for the wavelengths $\lambda \in \{0.5, 1, 5\}$. The final descriptors used in the classifier correspond to the mean and standard deviation of each of these Gabor filtered images, hence a total of six texture descriptors.

Figure 10.5: ROI initialization

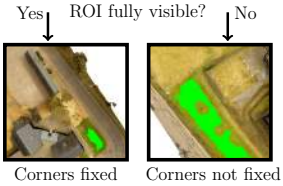


Figure 10.6: ROI tracking

3.5 Region Manager: Tracking, Merging and Updating

Tracking and Updating of ROIs

As illustrated in Fig. 10.5, the classification and segmentation module forwards the contours of a fine classification mask, defined by a set of 2D points, to the region manager. To simplify tracking and to increase the robustness with respect to impairing factors,[3] the fine classification mask is approximated by the minimum-area enclosing rectangle using the rotating caliper method [134, 261]. Next, the 2D positions of the four corners and centroid of the rectangle are projected into 3D based on the available coarse depth measurements (cf. Sec. 3.2). As depicted in Fig. 10.6, two cases are distinguished for initializing and updating ROIs: In the first case, the ROI is fully visible, i.e. all 2D corners are within the current image. If so, the corresponding 3D corners are fixed and ROI statistics (n_{grass}, n_{obs}, grade, cf. Sec. 3.5) are set. In a subsequent frame, a re-detection is triggered if the centroid of the ROI in the current frame is within the corners of an existing ROI, which is determined by the winding number method [271]. In this event, only the tracked ROI statistics are updated. In a second case, if the ROI is not fully visible, i.e. one or more corners are on the border of the image, the 3D corners are set but not fixed. If, in a subsequent re-detection, the ROI is again not fully visible the corners are updated by the vertices of the rectangle that incorporates all 8 corners [134, 261] until the ROI is fully visible and the first case applies.

Merging of tracked ROIs

It can occur that two tracked regions of interest correspond in fact to the same landing area. An example for this would be if only half of the area is detected in a series of subsequent frames, while the other half is detected later on in other frames. However, in following images, the UAV could detect the complete area. Without any merging, the pipeline would update values such as the corner position or the area size for one of these two areas because the projection of the newly detected center point is placed inside it, while the other half would remain unchanged. To

[3] E.g. the depth approximation introduced by the coarse depth measurements utilized for the 2D-3D projection.

avoid this duplication, every time the corner positions of a tracked ROI get updated, we verify for each ROI in the tracker if it belongs to that area, i.e. if the center of the newly updated area is in-between the four corners of the tracked ROI. If that is the case, the two ROIs are merged: the corner positions are set to the ones of the largest area and the grade is updated accordingly.

Grading of tracked ROIs

The tracked ROIs are ranked according to a cost function assigning a grade to each landing spot. The grading function makes use of metrics computed for each tracked region: The area A spanned by the four projected corners, the number of images in which the ROI has been classified as grass n_{grass}, and the total number of images in which it has been observed n_{obs}. The grade is zero if $\text{A} < \text{A}_{\text{min}}$ or $n_{\text{obs}} < n_{\text{obs,min}}$ and $n_{\text{grass}} n_{\text{obs}}^{-1}$ otherwise. In order to reduce the computational load, only the 20 ROIs with the highest grade are retained. This is implemented in form of a FIFO buffer in order to first remove regions which have not been detected or recognized recently.

3.6 Dense 3D Reconstruction

To reduce the computational burden, the subset of frames is iteratively selected for pose refinement and dense reconstruction as follows: The first pose is set as key-frame (KF). Then the next frame for which the feature track connection count first drops below 30 is determined. The predecessor to this frame is the next KF if the baseline is larger than a minimal baseline. Next, for every KF, the best suited stereo-pair is selected based on baseline, epipolar and viewing cost [135]. The selected set of poses is refined by incorporating pre-computed pose priors and feature tracks (cf. Section 3.1). Finally, the optimized poses are used for planar rectification [91, 135] in combination with Semi-Global Block-Matching (SGBM [134]). As described in Section 3.7, inverse distance weighting (IDW) is used to convert from 3D point cloud to 2.5D elevation map which smoothes the depth estimates.

3.7 Hazard and Decision Layers

This module uses the dense point cloud as input in order to evaluate the landing spot with respect to potential hazards such as terrain slope and terrain roughness. The data flow is presented in Fig. 10.7, for a sample visualization we refer to Fig. 10.10. To avoid the high memory load introduced by calculations involving the dense 3D point cloud, we convert to a 2.5 D grid-based elevation map [74] (Fig. 10.10c). The elevation of each cell is computed using KD-tree based [25] IDW in a radius around the cell. From the 2.5 D elevation layer, the surface normal in z-direction n_z of cell c_{ij} is computed based on the current cell and the 8-nearest neighbor cells using PCA. From the surface normal layer, the cell's slope α_{ij} with respect to the ground plane is obtained from $\alpha_{ij} = \arccos(n_{z,ij})$ (Fig. 10.10d). Terrain

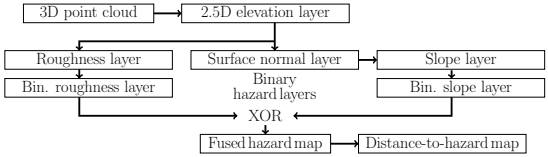


Figure 10.7: Hazard and Decision Layers.

roughness is identified as a second hazard. The terrain ruggedness index (TRI) [270] is computed based on the elevation difference to the 8 adjacent cells and allows, for instance, to differentiate between flat grass, crops, or forest regions (Fig. 10.10e). A fine classification mask is computed as XOR operation of all fine grass classification masks associated with this ROI. All hazard layers are only evaluated in the cells that have been classified as grass in at least one observation. Next, the hazard layers are transformed into binary layers using thresholds that are acceptable for the UAV (Fig. 10.10f). The binary hazard layers are then fused using a XOR operation. In order to find safe and contiguous landing paths, we then apply the distance transform (Fig. 10.10g) to the fused binary hazard map. This yields, for every cell, the distance to the closest hazard. Further decision layers, such as a probabilistic point cloud or classification uncertainty layer, could easily be incorporated.

3.8 Landing Approach Vector

The question remains from which direction the landing spot is to be approached while circumventing the surrounding hazard(s). The local wind field, which is

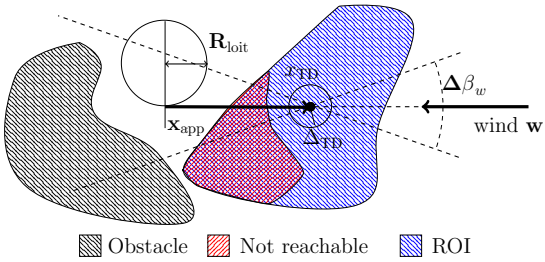


Figure 10.8: Computation of the landing approach vector while considering the local wind field as well as hazards surrounding and within the landing region (ROI).

estimated in real-time by the on-board EKF, constrains the approach vector as illustrated in Fig. 10.8. Small-size fixed wing UAVs need to land against the wind

direction in order to minimize the distance required for landing and to remain in a safe ground velocity region. Furthermore, we consider nearby obstacles obscuring the landing field based on the maximum descent rate of the UAV as well as obstacles in the landing region which are encoded in the distance map. Based on these considerations, the landing approach path is computed [200]:

$$x_{\text{app}} = \frac{v_{\text{land}} \cos(\gamma_{\text{land}}) - w \cos(\Delta\beta_w)}{v_{\text{land}} \sin(\gamma_{\text{land}})} h_{\text{app}}$$
$$R_{\text{loit}} = \frac{(v_{\text{land}} \cos(\gamma_{\text{land}}) + w)^2}{g \tan(\phi_{\text{land}})} \tag{10.1}$$

with

x_{TD} : touch down point	Δ_{TD} : touch down uncertainty (10 m)
w : wind magnitude (estimated)	v_{land} : airspeed ref. (13 m/s)
γ_{land} : flight path angle ref. (4 deg)	$\Delta\beta_w$: crosswind uncertainty (30 deg)
h_{app} : altitude approach (12 m)	ϕ_{land} : maximum bank angle ref. (11 deg),

and approach vector $\mathbf{x}_{\text{app}} = x_{\text{TD}} + \left[x_{\text{app}}\tilde{w}_x,\ x_{\text{app}}\tilde{w}_y,\ h_{\text{app}}\right]^\top$, where $\tilde{\mathbf{w}} = \left[\tilde{w}_x,\ \tilde{w}_y\right]^\top$ is the normalized 2D wind vector. The listed numeric values are used for the research UAV Techpod as described in Section 4.4. Based on the distance-to-hazard map (cf. Sec. 3.8) we efficiently take the touch down uncertainty into consideration. In particular, starting from the safest touch down point, we check all cells traversed by the linear approach path and loiter-down circle for collision to obstacles based on the elevation map and incorporating a safety margin. Note that approach path optimization, in this context, is only used for a more meaningful scoring of potential landing sites and the provision of an informed approach vector. It should not be seen as a replacement for local re-planners which are still necessary for real-time corrections upon the actual landing attempt.

3.9 Wind Vector Estimation

The UAV's state estimator provides online estimates of the local wind field [160]. All measurements taken within a certain distance to the center of a ROI are associated with a landing spot as shown in Fig. 10.12 and denoted by the black circle. To counteract slowly changing wind fields, we compute the final wind vector using the exponentially weighted moving average.

4 Results

4.1 Classification

Fig. 10.9 shows the binary classification of homogenous regions into *grass* and *not grass*. The results from random forest [134] and the multi-layer perceptron

(MLP) based artificial neural network (ANN) [134] are plotted in form of the true positive rate (TPR) vs. the false positive rate (FPR) on the left (ROC chart) and the precision recall curve on the right. While ANN performs slightly better in the validation set, the measured mean and standard deviation to predict a binary label is $2.4e-3 \pm 7.8e-4$ ms for RF and $1.7e-2 \pm 9.0e-3$ ms for ANN. Since ROIs are tracked over several frames (cf. Sec. 3.5) the influence of a single false prediction is mitigated by the probabilistic score as seen in Fig. 10.10 and Fig. 10.11. Hence RF was employed in all further experiments for computational speed-up.

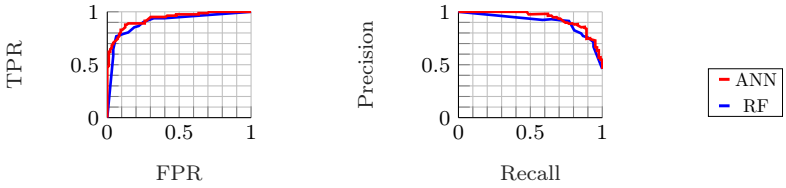


Figure 10.9: Binary classification of homogenous regions into *grass* and *not grass* for artificial neural network (ANN) and random forest (RF)

4.2 Runtime

The runtime evaluated on the real-world experiment “Switzerland" is shown in Table 10.1. The largest amount of the time in the frontend is spent on classification, in particular, to compute Gabor features. The frontend can run at 12.49 Hz with Gabor features and at 21.82 Hz when only relying on color cues. One could speed up the Gabor filter by only retrieving few samples from the image patch instead of using a convolution over the whole patch. However, since the incoming image rate is 4 Hz, the frontend (including Gabor features) is more than three times faster than real-time. Note the efficiency of the geometric region managing. BA, dense reconstruction and terrain analysis introduce, depending on the grid resolution, a certain delay and are available in *near real-time*.

4.3 Semi-Synthetic Dataset

The results obtained from a synthetic dataset are shown in Fig. 10.10. The images are rendered using *Blender* from poses generated by a simple lawnmower scan pattern generator. The underlying mesh was obtained from photogrammetry with images taken from a real camera, hence denoted as *semi*-synthetic. The overview mesh in Fig. 10.10 shows the scan pattern, corresponding frame indices on the left, and the three landing spots with the highest score. The right side of Fig. 10.10 shows the output of segmentation and classification for a sample frame. The second

[3] Evaluated on Intel(R) Core(TM) i7-4800MQ CPU @ 2.70GHz.

	samples	mean ± stdev
Segmentation	250	7.20 ± 0.64
Class. w. Gabor	250	70.30 ± 22.76
- feature vector	2217	3.56 ± 4.95
- predict	2217	$2.4e{-3} \pm 7.8e{-4}$
Class. w/o Gabor	250	36.89 ± 20.28
- feature vector	2217	0.21 ± 0.27
- predict	2217	$2.1e{-3} \pm 8.4e{-4}$
Region Manager	250	1.57 ± 0.39
Bundle Adjustm.	23	20.1 ± 3.2
Dense Reconstr.	23	16.7 ± 2.3
Decision Layers	10	1083 ± 21.09
Approach Vector	10	527.48 ± 20.65

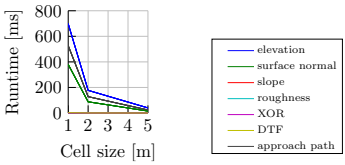


Table 10.1: Runtime in *ms* for the "Switzerland" dataset. Note that the frontend (light gray) and backend (dark gray) run in separate threads. Runtime of frontend, BA, and dense reconstruction is measured per frame. Runtime of decision layers and approach vectors is computed for a ROI point cloud consisting of 4.2×10^6 points (300×300 cells, 1.0 m resolution).

row of Fig. 10.10 plots the number of observations, classification certainty, score, and estimated area over time. Although there are some wrong classifications, the final classification certainty for all three regions is above 95 %. Furthermore, Fig. 10.10 shows the backend, i.e. the dense reconstruction, decision, and hazard layers, and the final touch down point and approach path. Note that touch down points on the right side of the distance map are rejected due to obstacles (house) within the approach path.

4.4 Experiments with Real-World Datasets

The real-world experiments are analysed using datasets recorded onboard of *AtlantikSolar* and *Techpod*. Details about the hardware setup and the employed platforms can be found in [200] and [119], respectively. The first experiment was conducted with the research platform Techpod in snowy scenery in Switzerland (cf. Fig. 10.11). The Fig. shows the segmentation and classification of the landing region that received the highest score. The ROI is then forwarded to the backend thread which generates the decision layers based on a dense point cloud. The terrain slope and terrain roughness are used to compute the distance map. The distance map encodes the distance to the next hazard in form of a memory-friendly grid map. From Fig. 10.10 and 10.11 one can see that already the coarse grading can achieve a large separation between desired and undesired landing spots. Depending on the UAV characteristics and desired landing spot, the final ROIs can be compared based on the output of the fine landing site evaluation. In the next experiment, the distance to terrain elevation, computed by the backend, is depicted as exemplary fine ROI output statistic.

This mentioned second real-world experiment was conducted with AtlantikSolar at the beach of Rio Pará, Brazil. Fig. 10.12 shows the overview mesh and camera poses for visualization, generated with *Pix4D*. On the right-hand side, one can see

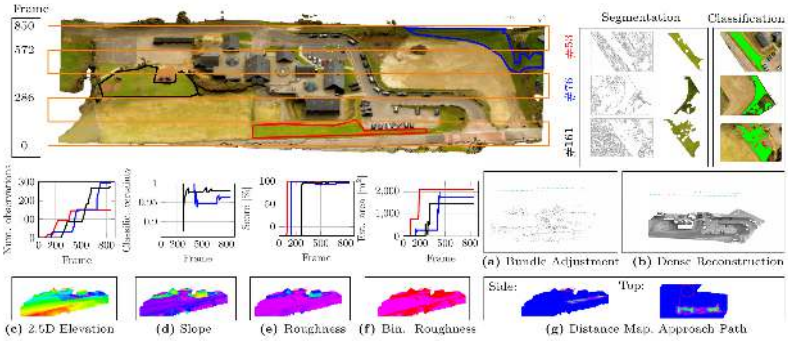


Figure 10.10: Semi-synthetic dataset illustrating the output of the segmentation, classification, tracking over time, and fine 3 D terrain evaluation. The simulated camera is a down-looking *Aptina MT9v034* (0.3 MP).

the 2.5 D elevation map colored by height, the estimated wind vector, and approach path to the selected landing site. The spot in the landing site is selected based on the maximum distance to nearby hazards at touchdown point. The plots below show the path of the UAV with marked landing spot, wind speed measurements, and altitude profile during the approach path. For instance, the margin between UAV altitude and terrain elevation is predicted to drop to ca. 2 m, 35 m before touch down. As discussed in Section 3.8, the algorithm is designed to land against the wind vector. This has the effect of reducing the aircraft's forward ground velocity, thus increasing the perceived descent angle with respect to *ground*, yet still maintaining the chosen *airmass-relative* flight path angle γ_{land} (cf. Equation 10.1).

5 Conclusion

In this paper, we present a vision-based prior-free landing site detection algorithm which is designed for small UAVs, taking into account terrain texture, shape, roughness, and slope. The wind field, which is estimated online, in combination with obscuring obstacles, are taken into consideration when computing a suitable landing spot while regarding UAV dynamics and safety margins. To keep the problem complexity manageable, we segment the environment into regions and use a layered 2.5 D grid map for decision making. The implemented multi-threaded framework combines a light-weight, real-time frontend with a backend which is

[4] Semi-automated: Take-off and landing are performed by a safety pilot. The actual mission is conducted by automated GPS waypoint following.

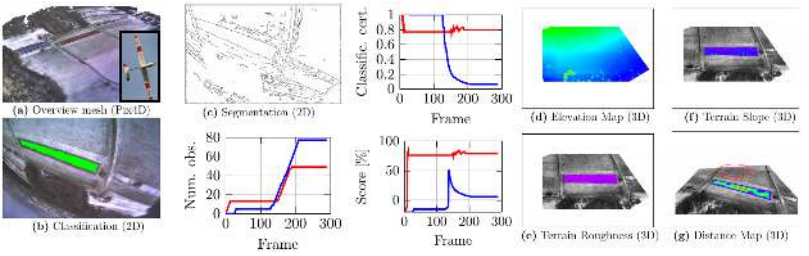


Figure 10.11: Manual flight with Techpod in snowy scenery in Zurich, Switzerland. The employed camera is an *IDS UI-3241LE* (1.3 MP). The experiment underlines the performance in a challenging environment and with an obliquely mounted camera.

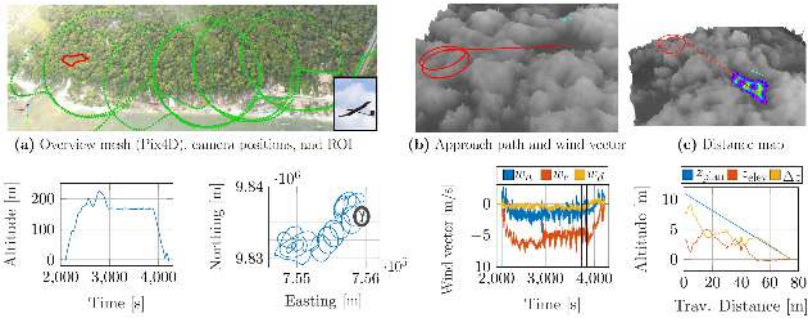


Figure 10.12: Semi-automated[4] flight at the beach of Rio Pará, Brazil, at sunrise using AtlantikSolar and a down-looking *GoPro Hero 3* (12 MP). The figures illustrate the incorporation of wind estimates into the LSD framework. Subfig. (b) and (c) depict the 2.5D elevation map on gray scale. The darker the color, the lower the height or z-value of the elevation map's cell.

periodically updated based on the host's resources. The linear approach path, which is one output of our method, can be tracked as demonstrated in [200]. The actual landing attempt should furthermore be supported by a perception system, local re-planners and low-level autopilot logic to avoid previously unmapped or moving obstacles. In this paper, a simplistic key-frame selection algorithm was employed. In a next step, an algorithm should be designed that guarantees complete coverage while minimizing the reconstruction uncertainty, utilized number of poses, and hence the computational costs. Inter-matches and free-space carving [128] could

be incorporated into the reconstruction process. In future work, the classification and reconstruction uncertainty around a promising landing spot should be actively reduced by optimizing the flight path and, in particular, by low-terrain flights to increase the ground resolution.

Acknowledgment

The research leading to these results has received funding from the Federal office armasuisse Science and Technology under project number n°050-45. Furthermore, the authors wish to thank Philipp Oettershagen, Felix Renaut for an initial implementation of the presented frontend, and Lucas Teixeira (Vision for Robotics Lab, ETH Zurich) for sharing scripts that bridge the gap between *Blender* and *Gazebo*.

Paper VIII

Deep Learning-based Human Detection for UAVs with Optical and Infrared Cameras: System and Experiments

Timo Hinzmann, Tobias Stegemann, Cesar Cadena, Roland Siegwart

Abstract

In this paper, we present our deep learning-based human detection system that uses optical (RGB) and long-wave infrared (LWIR) cameras to detect, track, localize, and re-identify humans from UAVs flying at high altitude. In each spectrum, a customized RetinaNet network with ResNet backbone provides human detections which are subsequently fused to minimize the overall false detection rate. We show that by optimizing the bounding box anchors and augmenting the image resolution the number of missed detections from high altitudes can be decreased by over 20 percent. Our proposed network is compared to different RetinaNet and YOLO variants, and to a classical optical-infrared human detection framework that uses hand-crafted features. Furthermore, along with the publication of this paper, we release a collection of annotated optical-infrared datasets recorded with different UAVs during search-and-rescue field tests and the source code of the implemented annotation tool.

Published in:
CoRR, 2020

1 Introduction

The need for robust human detection algorithms is tremendous and has massively increased over the past years due to the vast amount of emerging applications in the field [67, 193]. With UAV technology blooming, research in the field of human detection from aerial views also steadily evolved and experienced much interest for real-world SaR missions [10, 20, 27, 152, 227]. While the majority of the earlier work on human detection incorporated hand-crafted features, more recent publications started to make use of DL-based detectors, mostly in the form of CNNs [20, 55, 113]. In the superordinate field of object detection, deep CNNs have been already established for several years [150, 245], and current state-of-the-art detectors produce impressive results with possible real-time performance [171, 217]. There exist a few publications known to date that try to use these insights from the field of object detection for human detection in aerial images [29, 44, 174, 206, 267, 283]. Making use of either only optical or only LWIR (also referred to as Thermal-Infrared (TI)) images, still no work so far has considered the use of deep CNNs for combining information from both TI and optical images. As CNNs need a vast amount of data to outperform hand-crafted detectors, the lack of publicly available data might still limit the ubiquity of deep CNNs for human detection in both optical and TI aerial imagery. Available data in the field either only consists of optical [188, 224], or only TI images [53, 173, 211], and no publicly available dataset provides real-world data collected in the field for an expressive evaluation of human detection algorithms in search and rescue scenarios. Our publication provides such a collection of optical-TI field datasets for extensive training and testing of human detection algorithms.

Furthermore, besides optimizing the raw detections, we propose a complete framework to detect, track, localize, and re-identify humans as shown in Fig. 11.1.

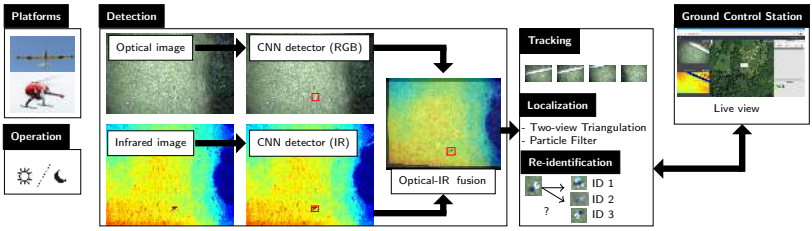


Figure 11.1: Proposed deep learning-based human detection system that uses optical (RGB) and long-wave infrared (LWIR) cameras to detect, track, localize, and re-identify humans from UAVs flying at high altitudes.

Year	Publication	RGB	IR	CNN	Detector
2008	[227]	✓	✓	–	temperature thresholding (thermal) shape matching (thermal) Haar, cascade of boosted classifiers (optical)
2010	[221]	✓	–	–	shadow casting, wavelets & SVM
2011	[139]	–	✓	–	SIFT features & voting scheme
2011	[97]	✓	✓	–	Haar cascade (thermal) Gaussian shape models (optical)
2013	[79]	✓	✓	–	temperature thresholding (thermal), Felzenszwalb & HOG detector (optical)
2014	[210]	–	✓	–	background subtraction, HOG, LBP & BPD (Haar)
2014	[27]	✓	✓	–	saliency maps & HOG
2014	[28]	✓	✓	–	saliency maps & segmentation (thermal), PRD based on integral channel features
2015	[263]	✓	✓	–	background subtraction (thermal), HOG & SVM (thermal)
2016	[152]	✓	✓	–	blob detection (thermal), HOG & SVM (optical & thermal)
2016	[55]	✓	✓	✓	temperature thresholding (thermal), saliency maps (optical), CNN, Haar- & LBP cascade (optical)
2016	[231]	–	✓	✓	temperature thresholding & CNN
2016	[113]	–	✓	✓	thresholding (MSER) & CNN
2017	[20]	✓	–	✓	color thresholding, CNN features & linear SVM
2018	[29]	–	✓	✓	Faster R-CNN
2018	[267]	✓	–	✓	RetinaNet vs. Faster R-CNN & SSD

Table 11.1: Publications on human detection (sorted by publication date)

2 Related Work

Early work in the field of human detection from UAVs [27, 79, 97, 152, 227, 263] strongly resembled the one from object detection and applied classical methods such as Haar-like features introduced by [265], the Felzenszwalb detector [75], or HOG features together with linear Support Vector Machine (SVM) classifiers as introduced by [52]. A lot of this early work [28, 79, 97, 152, 227] concluded that the combination of TI and optical images is highly beneficial for the task of human detection from high aerial views.

With the rise of deep learning, the application of CNNs started to become well-established for the task of human detection from aerial views [20, 55, 113, 231]. By comparing against more classical feature extractors and detectors such as Haar, HOG or SVM, an improvement in detection performance as well as a better generalization when using CNNs, was stated throughout. Research in object detection progressed quickly and gradually yielded deeper, better, and faster-performing object detection networks. On the side of two-stage detectors, the R-CNN network and its successors [102, 103, 223], as well as Feature Pyramid Networkss (FPNs) by [170] gained a

lot of attention. On the other hand, first one-stage detectors such as YOLO and its successors [216–218] or the Single Shot Detector (SSD) by [175], impressed with high frame-rates while still achieving competitive detection performance. Some recent work in the field already started to include these state-of-the-art object detectors such as adaptations of the R-CNN network [174, 206] or one-stage detectors such as YOLO9000 or SSD [44, 283] into their pipelines. The use of either one-stage or two-stage object detectors, however, results in a speed-accuracy trade-off, as shown by [29]. This is in accordance with recent findings by [171]. Their one-stage object detector framework called RetinaNet tries to address this discrepancy between inference speed and detection performance and close the gap between state-of-the-art one-stage and two-stage object detection frameworks. Wang et al. [267], for instance, used RetinaNet [171] as a proposed solution for object detection in aerial views. RetinaNet was evaluated against other one-stage and two-stage detectors, namely SSD and Faster R-CNN on the Stanford drone dataset [224]. The evaluation resulted in a similar statement to the one by [171], showing state-of-the-art performance of RetinaNet compared to two-stage detectors, while running at speeds comparable to the ones obtained from one-stage detectors. This motivates the use of these state-of-the-art one stage detectors also for our task, since their fast inference speeds are crucial when running on-board a UAV with limited computing power.

The vast majority of deep learning architectures is trained and evaluated on optical imagery. In fact, to the best of our knowledge, the only work to date which uses a DL-based human detector on thermal imagery from a UAV was conducted by [29] in 2018, where a Faster R-CNN framework was applied to detect poachers in thermal images. More so, none of the publications make use of *both* the optical and thermal domain together with recent state-of-the-art deep object detection networks. Likely, this is due to the lack of publicly available data: Publicly available datasets either only consist of optical images [188, 224] or only thermal images [53, 173, 211]. In this paper, we approach these research gaps as follows: Firstly, we present a comprehensive performance comparison of state-of-the-art human detection architectures, trained with optical and thermal imagery. In particular, the evaluation comprises YOLO and RetinaNet with different ResNet backbones. For reference, we additionally provide the results from a hand-crafted thermal-optical human detection pipeline [152]. Secondly, to the best of our knowledge, this paper constitutes the first publication on deep learning-based human detection from UAVs which combines optical-thermal imagery. The employed merging strategy is explained in detail in Sec. 3.3. Thirdly, the evaluation is conducted on datasets recorded during SaR field tests. The collection of annotated datasets and the implemented annotation tool is released along with this paper. Despite the great success of recent findings in the field of object detection, very small sample sizes as well as strong-view point variations in aerial images, make the task challenging. Recent work like the one by [174] or [44], clearly shows that adaptations to current object detection networks are crucial for a decent performance on aerial images from UAVs. In this paper, we propose to use optimized custom anchors and up-scaled images to boost the detection from high altitudes and reveal this performance

gain in a detailed evaluation. Finally, we describe in detail our approach to track, localize, and re-identify humans to improve the overall performance beyond the raw detection.

3 Human Detection

3.1 State-of-the-Art Object Detectors Revisited

Our proposed pipeline uses a customized RetinaNet network with a ResNet50 backbone as introduced by [171] as a state-of-the-art one-stage object detector. As already outlined, this is mainly due to limited computing resources on-board of UAVs, as well as similar detection performance of RetinaNet compared to two-stage detectors in recent publications [171, 267]. Our implementation follows the original implementation and introduces crucial customizations, as outlined in Sec 3.2. This proposed framework is then compared against YOLOv3 using a darknet-53 backbone, another state of the art one-stage object detector. Both networks follow a similar basic architecture collecting features at different scales using their respective backbone networks and subsequently regressing and classifying the output bounding boxes. As one of their main contributions, RetinaNet introduces a novel focal loss, based on the well-known cross-entropy loss. Since YOLOv3 still relies on the standard cross-entropy loss, the focal loss further constitutes the main difference between these two object detection frameworks. In [171], Lin et al. state the significant imbalance of image background and foreground in many object detection tasks as the main reason for lacking the performance of conventional one-stage detectors. They were able to show that a conventional cross-entropy loss is easily overwhelmed by the vast amount of background samples seen during training, when running regression and classification directly on top of a dense feature map. By adding a modulating factor $(1 - p_t)^{\gamma}$ to the cross-entropy loss, they define the novel focal loss using a tunable focusing parameter $\gamma > 0$, able to counteract this large imbalance.

3.2 Network Customization

Optimal Anchor Selection

Both RetinaNet as well as YOLOv3 use nine anchor bounding boxes for final bounding box regression. While RetinaNet uses fixed hand-picked anchors, YOLOv3 uses k-means dimension clustering on the COCO dataset [169] to find optimal anchors. Furthermore, during training, the selection strategies, deciding on which anchor bounding boxes are assigned to the ground-truth bounding boxes, differ quite significantly between the two networks. Lin et al. [171] use an adjusted assignment rule originating the region proposal network of Faster R-CNN [223]. Anchors are assigned to ground-truth boxes for an Intersection over Union (IoU) value of 0.5 or higher and to background for IoU values in $[0, 0.4)$. Each anchor is assigned to at most one ground-truth box, and all remaining unassigned anchors

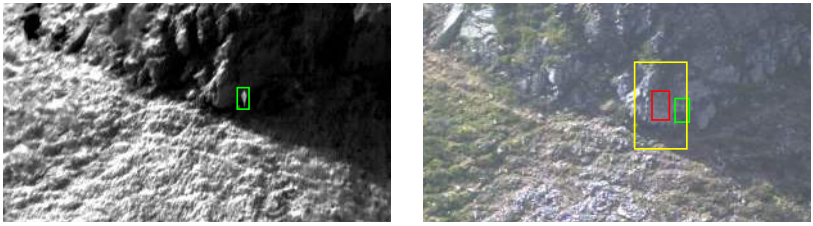
are ignored during training. Redmon et al. [217], on the other hand, do not use such a dual IoU threshold. Instead, they simply assign one anchor per ground-truth bounding box by using the anchor with the largest IoU value. If an anchor is not the best but overlaps a ground-truth box with more than 0.5 IoU, the prediction is ignored. All other anchors, not assigned to any ground-truth boxes, do only incur a loss for the object prediction. The need for customized anchors in our pipeline is crucial: Using the standard anchors for RetinaNet on our training dataset showed that a lot of samples in both optical and thermal images did not contribute to the training process because of their really small size and not reaching an IoU value of 0.4 or higher with any of the standard anchor bounding boxes. Resolving this by changing the added anchor sizes at each level of the original publication by [171] to custom sizes of $\{2^{-2}, 2^{-1}, 2^{0}\}$, the number of bounding boxes contributing to the training was sufficiently increased. More specifically, the amount of optical samples contributing to the final stage of training was increased from 57.7% to 97% on the optical side and from 33.2% to 88.1% on the thermal side of our training dataset.

Image Resolution Augmentation

A second natural way of improving the detection of really small bounding boxes is to simply augment their sizes. Final detection performance can be improved significantly, by either using augmented- or higher-resolution images, as also stated by Bejiga et al. [20]. This can be achieved by increasing the input image size during inference since both networks being fully-convolutional. Doubling the standard image sizes of the RetinaNet variants was able to bring a considerable performance improvement as shown in Sec. 6.

3.3 Optical-Infrared Merging Strategy

The proposed pipeline utilizes optical and thermal imagery. This is especially useful to further reduce false positives during day flights, for detecting humans during night flights, or to find humans in aggravated optical conditions, for instance, when obscured by shadows due to large rocks or trees. The camera intrinsic and extrinsic parameters are used to map feature positions from the optical to the thermal image and vice versa. However, small inaccuracies in the calibration, the image triggering, or exposure time may result in pixel offsets in both spectra. Especially in applications with fast-moving and fast-turning UAVs this may become an issue. To account for these inaccuracies, we propose to use a sliding window to match bounding boxes between the spectra, as illustrated in Fig. 11.2. More specifically, a rectangle nine times the size of the mapped bounding box is searched for a target bounding box using 36 sliding window steps in total. Matching of the sliding window and target bounding boxes is done similarly to the standard procedure proposed by [67], using an IoU threshold of 0.5. After a completed matching step, a logical OR merging scenario is used, considering all bounding boxes from both domains while averaging the respective prediction scores.



(a) (b)

Figure 11.2: Exemplary matching process. Original detection in the thermal image on the left in green. Raw mapped bounding box in the right optical image in red, sliding window area in yellow and final matched bounding box (detection) in green.

3.4 Network Training

To train the network, publicly available, external datasets, as well as internal data, recorded by one of our UAVs, are used. Using the implemented annotation tool, a total of 11 optical sequences consisting of 40 445 images, and 14 thermal sequences containing 1 029 images are available for training. The human bounding boxes in theses sequences are annotated with a distinct ID for each individual human, attributes for the pose (upright, sitting or lying), and an occlusion attribute. Altogether, the newly collected datasets add up to a total of 34 022 human bounding boxes in the optical images, and 37 228 human bounding boxes in the thermal images. In addition to our internal datasets, all available public datasets containing humans from both optical [30, 188, 224] and thermal [53, 149, 173, 211, 274] aerial views have been gathered. Together with these external datasets, the final amount of available data at hand consisted of around 1 000 000 annotations in over 170 000 optical images, and around 200 000 annotations in over 100 000 thermal images.

The final training of the two object detectors was conducted using a two-stage training procedure: Starting with pre-trained Imagenet [228] weights, an initial pre-training step was carried out using all the data at hand, including our newly collected dataset and all available external datasets. The best results in pre-training were achieved by freezing all the backbone weights for both RetinaNet and YOLOv3 and only training and adapting all other, randomly initialized weights to the novel domain. Subsequently, the resulting weights of the pre-training step were then used to initialize a final fine-tuning step. In this step, only the newly collected, domain-specific data was used to train all the layers of both the networks. Training of the two object detectors was carried out according to the original publications of RetinaNet [171] and YOLOv3 [217]. Both networks were trained on a cluster using Nvidia GeForce GTX 1080 Ti GPUs. While YOLOv3 originally uses a smaller sized input image together with a multi-scale training strategy [217], RetinaNet uses a

larger single size input image during training. To be able to train both networks using a similar batch size of eight images, RetinaNet was trained on a total of eight GPUs while YOLOv3 was trainable on a single GPU. Both networks include data augmentation and other training features mentioned in the original publications. Randomly selected sequences out of the external datasets were used as validation sets in pre-training and a fixed sequence of our own dataset was used as validation set for the final fine-tuning training scenario.

4 Human Localization and Re-Detection

This section describes our approach to track, localize, and re-detect humans in the optical spectrum based on qualitative results.

Human Tracking: For every observation classified as human the detector described in Sec. 3 creates a new victim ID. However, the final goal of the proposed framework is to associate every victim with a unique ID and to compute its position or path in 3D. In a first step towards this goal, an object tracker bundles all detections of a victim as long as the human is within the field of view of the camera (cf. Fig. 11.3). The human tracking is tested on the optical image stream with a frame-rate of 4 Hz, resulting in potentially large pixel displacements of an observation between two subsequent frames. To reduce the pixel displacement and simultaneously increase the speed of the tracker the image is half-sampled. Object trackers implemented in [33] were tested, including MIL [15], KCF [112], GOTURN [110], among which CSRT [179] performed best and was selected.

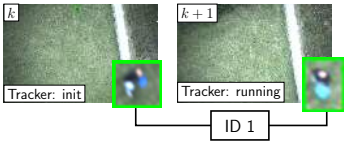


Figure 11.3: Human tracking: Humans that are tracked across subsequent frames $(k,\ k+1)$ are assigned the same human ID.

Human Localization: Given a track of observations in the form of bounding boxes and the corresponding camera poses, the 3D path of the object can be estimated. The camera poses are assumed to be given as input to the framework. As the human may be non-static, two-view triangulation of consecutive observations are used. The center of the bounding box is selected for triangulation, as illustrated in Fig. 11.4.

Figure 11.4: Human localization: Based on two subsequent detections, the georeferenced human position can be triangulated, as visualized in the satellite orthoimage.

Metric Outlier Rejection: The metric bounding box area is used to reject false detections as follows: Given two consecutive observations of an object and the triangulated 3D object position $\mathbf{p}_h^W$, the depth d can be inferred via $d = \|\mathbf{p}_{\text{UAV}}^W - \mathbf{p}_h^W\|_2$. The depth is then used to transfer the bounding box from pixel coordinates to meters, resulting in the points $\mathbf{p}_i^W, i = (1, ..., 4)$ (clockwise). The metric area of the bounding box is then $A = 0.5(\overline{\mathbf{p}_1^W \mathbf{p}_2^W} \times \overline{\mathbf{p}_1^W \mathbf{p}_3^W} + \overline{\mathbf{p}_1^W \mathbf{p}_3^W} \times \overline{\mathbf{p}_1^W \mathbf{p}_4^W})$. Objects with bounding box areas $A > T_{\text{area}}$ are classified as outliers and rejected as visualized in Fig. 11.5.

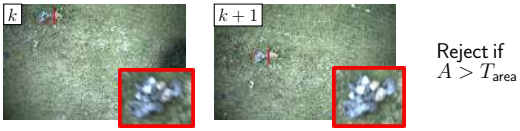


Figure 11.5: Metric outlier rejection: The detection is rejected if the estimated metric area A of the bounding box is above a threshold T_{area}.

Particle Filter: To handle occlusions, re-detections, and to incorporate a probabilistic motion model, a Particle Filter (PF) is initialized for every human. For computational reasons the PF is restricted to two dimensions x and y in UTM coordinates. If necessary, the altitude of the victim can be queried from the existing map. The PF implementation and notation closely follows [244] (cf. Fig. 11.6):

- **Initialization:** Given the first triangulated position, denoted as $\mathbf{z}_0 = (x, y)$ with $\sigma_{\mathbf{z}} = 3$, randomly draw $N = 100$ initial particles $\mathbf{x}_{0,i}^+$ with $i = \{1, ..., N\}$ from $\mathcal{N} \sim (\mu = \mathbf{z}_0, \sigma = \sigma_{\mathbf{z}})$.
- **Propagation:** In the propagation step, compute the a priori particles $\mathbf{x}_{k,i}^-$ given a random walk motion model, assuming a maximum human velocity $\mathbf{v}_{\max} = 1.2\,\frac{\text{m}}{\text{s}}$ in both, the x- and y-direction:
  $\mathbf{x}_{k,i}^- = f_{k-1}(\mathbf{x}_{k-1,i}^+, \mathbf{w}_{k-1}^i) = \mathbf{x}_{k-1,i}^+ + \mathbf{w}_{k-1}^i \Delta t$ where $\mathbf{w}_{k-1}^i$ is a random

velocity $\mathbf{v}_{\text{rand}}$ drawn from the uniform distribution $\mathcal{U} \sim (-v_{\max}, v_{\max})$ and Δt is the time difference between two consecutive frames in seconds.

- **Measurement:** Given a new observation $\mathbf{z}_k$ associated with the victim ID, compute the relative likelihood q_i of this observation for every particle $\mathbf{x}_{k,i}^-$ by evaluating the conditional Probability Density Function (PDF) $p(\mathbf{z}_k|\mathbf{x}_{k,i}^-)$ based on the measurement equation:
  $q_i \sim \alpha^{-1} \exp(-0.5(\mathbf{z}_k - \mathbf{x}_{k,i}^-)^\top \mathbf{R}^{-1}(\mathbf{z}_k - \mathbf{x}_{k,i}^-))$ with $\alpha = 2\pi \det(\mathbf{R})^{\frac{1}{2}}$ and measurement noise $\mathbf{R} = \sigma_{\mathbf{z}} \cdot \mathbf{I}_{2\times 2}$. Normalize via $q_i = \beta^{-1} q_i$ with $\beta = \sum_{j=1}^{N} q_j$.
- **Resampling:** Draw posteriori particles $\mathbf{x}_{k,i}^+$ based on normalized likelihoods q_i using Systematic Resampling (SR) [148].

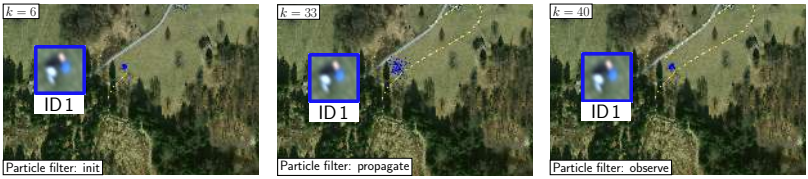
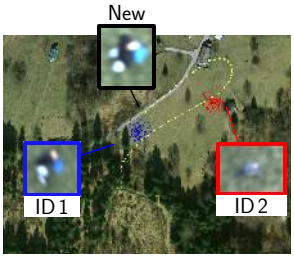


Figure 11.6: Particle filter: The particles for human ID 1 are visualized as blue points, the trajectory flown by the Rega drone is shown in yellow.

Human re-detection: If the UAV flies over a previously visited area and the detector returns an object classified as human, the algorithm needs to decide if the new detection z can be associated to an already observed human h_i or if it is, in fact, a new victim (cf. Fig. 11.7).

Applying the Bayes theorem the probability $p(h_i|z)$ that the new observation z belongs to human h_i can be computed with $p(h_i|z) = \gamma^{-1} p(z|h_i)p(h_i)$, $\gamma = \sum p(z|h_i)p(h_i)$. Appearance-based and spatial information are used to compute $p(h_i|z)$. Firstly, the conditional probability $p(z|h_i)$ is computed based on the triangulated location of the new observation and all existing humans currently tracked by PFs. If $p(z|h_i)p(h_i) < T_{\text{redetect}} \forall i$ a new victim with corresponding PF is initialized. Otherwise the new detection is associated with $\arg\max_{h_i} p(h_i|z)$, i.e., with that human that maximizes the detection probability.

Secondly, the prior $p(h_i)$ which is the probability of observing human h_i is inferred from the similarity between a patch from human h_i and the newly detected human patch using a binary classifier. Since the humans are detected from a high distance we base our decision if two patches contain the same person solely on the color information. Tab. 11.2 presents the similarity between patches computed by comparing their color histograms using the metrics [32] Correlation, Chi-square, Intersection, and Bhattacharyya [22]. The patch numbers 1 to 4 correspond to the detections shown in Fig. 11.7. Note that for Correlation and Intersection, higher



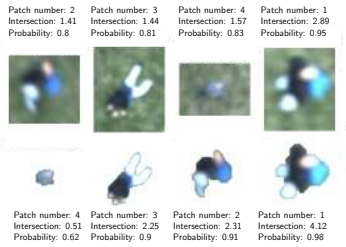


Figure 11.7: Human re-detection: The algorithm needs to decide if the new detection can be associated to an already observed human or if it is, in fact, a new victim. On the right, the histogram similarity is evaluated based on the Intersection metric. The query image is patch number 1.

values correspond to a higher similarity. In contrast, for the metric Chi-square and Bhattacharyya, the lower the value, the higher the similarity. Patch numbers 1, 2, and 3 are observations of the same human. However, this is not reflected by the results in Tab. 11.2 as the background has an influence on the color histogram. Therefore, using the GrabCut algorithm [226], the background is automatically subtracted (cf. Fig. 11.7) before computing the histogram which improves the re-identification results, as shown in Tab. 11.3. Using the sigmoid function, for instance, the results from the Intersection metric are mapped to values between 0 and 1, as shown in Fig. 11.7.

	Patch			
Method	1	2	3	4
Correlation	1.0	0.71	0.83	0.83
Chi-Square	0.0	8.85	2.39	2.52
Intersection	2.89	1.41	1.44	1.57
Bhattacharyya	0.0	0.53	0.37	0.44

Table 11.2: Quantitative results for Fig. 11.7: Patch similarity **without** background subtraction and histogram comparison.

	Patch			
Method	1	2	3	4
Correlation	1.0	0.57	0.83	0.09
Chi-Square	0.0	12.33	1.9	44.92
Intersection	4.12	2.31	2.25	0.51
Bhattacharyya	0.0	0.52	0.41	0.83

Table 11.3: Quantitative results for Fig. 11.7: Patch similarity **with** background subtraction and histogram comparison.

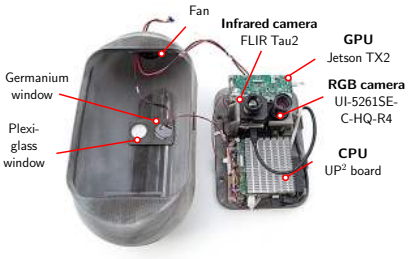
5 Hardware & Experiment Preparation

5.1 Platform and Sensors

The sensorpod used for our experiments is shown in Fig. 11.8c. It consists of an IDS UI-5261SE-C-HQ-R4 1.92 MP RGB camera with a C-mount lens and a

(a) Techpod

(b) Rega drone



(c) Sensorpod with optical-infrared stereo rig.

Figure 11.8: The sensorpod with optical-infrared stereo rig carried by the Rega drone.

horizontal field of view of 32 ° and an infrared camera FLIR Tau2 with a resolution of 640 × 480 pixel. The FLIR Tau2 is mounted on a Teax image grabber with USB2 interface. The GPU is a Jetson TX2 mounted on an Auvidea J120 carrier board and is used for network inference at run-time. It is connected to the CPU, an UP^2 board with Intel Atom Processor, via Ethernet. The CPU has a MSATA storage of 1 TB and handles the triggering of RGB camera, infrared camera, and ADIS16448 IMU. The casing features a plexiglass and Germanium window for the optical and infrared camera, respectively, and a fan for additional air circulation. The two UAV platforms that are used for the various test flights are shown in Fig. 11.8a and 11.8b.

5.2 Geometric Optical-Infrared Camera and Camera-IMU Calibration

The camera intrinsics (focal length, principal point), distortion parameters, and optical-infrared extrinsics (relative pose between the cameras and IMU) are required for optical-infrared image fusion and metric human localization.

For this purpose, different optical-infrared, i.e., dual-modal calibration targets have been developed and improved over time. A dual-modal calibration target allows to directly calibrate the relative pose between the thermal and optical camera without the need for a camera-IMU calibration as an intermediate step. That is the optical-infrared camera can be calibrated as a stereo rig with one single calibration dataset using Kalibr [219]. The evolution of our developed calibration targets is represented by Tab. 11.4 and was inspired by the related publications listed in Tab. 11.5. The different calibration targets shown in Tab. 11.4 and 11.5 are classified based on the taxonomy proposed in [215]: Initial works utilized classical optical

	Picture IR	Picture RGB	RGB	IR	Target type	Working Principle	Features
No.1			✓	✓	**Checkerboard** Paper print-out glued on wood	Color emissivity difference (black, white)	Corners 6×8 squares 0.036 m
No.2			✓	✓	**Checkerboard** Black colored wooden squares, Aluminum squares	Color emissivity difference (black, white) Material emissivity difference (wood, aluminum)	Corners 5×6 squares 0.036 m
No.3			✓	✓	**Square grid** Aluminum target with inserted black wooden circles	Color emissivity difference (black, white) Material emissivity difference (wood, aluminum)	Circles 7×6 circles 0.08 m

Table 11.4: Evolution of our dual-modal calibration targets (sorted by date).

checkerboard calibration targets and heated the target with a flood lamp (cf. target No.1 or [212]). Based on the emissivity difference of black and white squares, the pattern can be made visible also in the TI spectrum. However, fuzzy transitions between black and white edges lead to missed or inaccurate corner detections and unsatisfying calibration results. Based on this finding, we attempted to increase the sharpness of the edges by using materials with contrary emissivity properties. For this we designed target No.2 which is made out of black colored wood (emissivity ≈ 0.8 [167]) and aluminum (emissivity ≈ 0.095 [167]). Likewise, the authors of [229, 247, 264] focused on improving the contrast of checkerboard targets using different materials and masks. However, as for instance pointed out by [282], the corner detection remains inherently error-prone in the TI spectrum, and instead, the usage of circular features is advised. Our final dual-modal calibration target No.3 consists of 6×7 circular features, laser-cut into an aluminum calibration target, and filled with machine-cut black wooden plates. Based on the detections, the camera intrinsics and extrinsics are calibrated with Kalibr [219]. To be able to use Kalibr, the infrared images are inverted and thresholded. We obtained the best calibration results for datasets that are recorded outside where the calibration target is facing the clear sky. For this target, Kalibr reports the following errors: For a single camera calibration $\sigma = 0.09$ (infrared) and $\sigma = 0.09$ (optical); for a camera-IMU calibration $e_{\text{reproj}} = 0.60$ (infrared) and $e_{\text{reproj}} = 0.25$ (optical).

5.3 Optical-Infrared Dataset Collection and Annotation

To further increase the amount of training, validation, and test data we collected additional optical-infrared datasets. The datasets are recorded with sensors mounted on the platforms listed in Fig. 11.8a and 11.8b, at different locations and resembling realistic search-and-rescue missions. A complete list of available datasets is shown in Tab. 11.6. Along with all datasets we release the C++ based annotation tool that takes as input a dataset in the form of a rosbag [214] or video. With the help of this tool, all humans appearing in the collected data have been annotated

Publication	Picture	RGB	IR	Target type	Working Principle	Features
[212]		✓[a]	✓	Checkerboard	Flood lamp Color emissivity difference (black, white)	Corners
[247]		✓	✓	Hermann grid	Material emissivity difference (Styrofoam, air)	Corners
[264]		✓	✓	Hermann grid	Heated backdrop, e.g. monitor Material emissivity difference (cardboard, monitor) difference in temperature and/or difference in thermal emissivity	Corners
[282]		✓	✓	Cross	Difference in temperature (Thermostatic heaters) Color emissivity difference (black, white)	Circles
[229]		✓	✓	Checkerboard	Flood light Color emissivity difference (black, white) Material emissivity difference (ceramic, paper)	Corners
[77]		✓	✓	Wall	Temperature difference (flood lights, fan) Color emissivity difference(black, white) Material emissivity difference (foam, aluminum)	Arcs

Table 11.5: Optical and infrared calibration methods (sorted by publication date)

[a] Not shown in the paper but in principle possible

using upright rectangular bounding boxes. During the labeling process, humans are assigned a unique ID. Every annotation contains an additional attribute for the human posture (*upright*, *sitting* or *lying*) and one for occlusion (*occluded*, *not occluded*). Examples of the available annotated frames are illustrated in Fig. 11.15.

6 Experiments and Results

6.1 Experiment 1: Comparison of RetinaNet, YOLOv3 and Hand-crafted Detector

In a first step, we compare the vanilla deep learning-based detectors to a hand-crafted human detection framework that uses HOG features together with a SVM classifier [152] and that serves as a low baseline solution. This reference pipeline, that was introduced in [152], detects humans in thermal imagery and then uses the corresponding optical image solely to reduce false positives. The performance on the collected `roof_test` dataset, similar to the one by [152], is illustrated in Fig. 11.9 and 11.10, and shows a significant improvement when using deep learning-based object detectors. Furthermore, all of the RetinaNet variants vastly outperform the YOLOv3 framework.

6.2 Experiment 2: RetinaNet evaluation in SaR scenario

In this section, the performance of the proposed human detection pipeline is thoroughly evaluated based on a dataset resembling a search-and-rescue scenario. The `field_test` datasets proved to be very challenging with a lot of very small human samples at different poses recorded from very high aerial views. Occasional

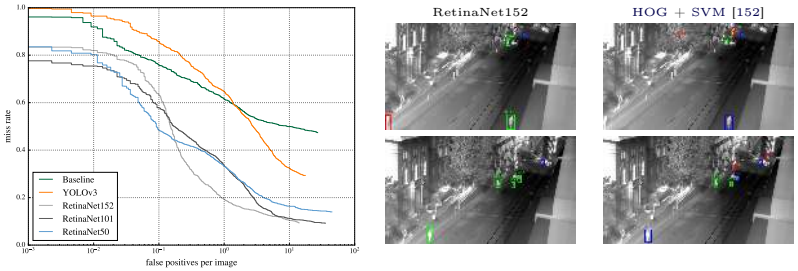


Figure 11.9: Thermal fppi/missrate curves on the `roof_test` set illustrating the large performance improve of deep learning detectors.

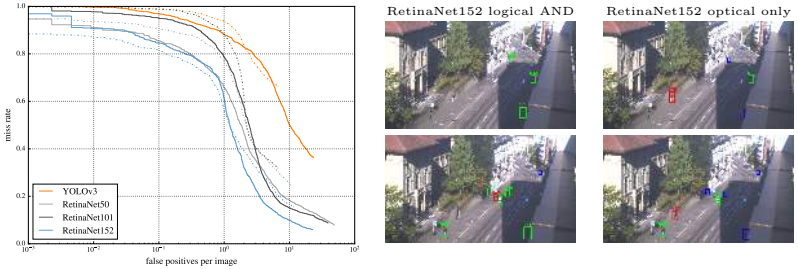


Figure 11.10: Optical fppi/missrate curves on the `roof_test` set. Performance on pure optical information plotted as dashed lines and the improved performance using the logical OR merging scenario in bold. By combining optical and thermal information, both false positives and overall missrates are reduced.

motion-blur due to the flight maneuvers and a lot of heated up rocks make the thermal imagery even more challenging. The performance of our vanilla detectors on the total optical and thermal field evaluation datasets clearly emphasizes this: Only a few percents of the total amount of human bounding boxes are detected. Still, RetinaNet variants and especially the proposed RetinaNet50 network vastly outperforms YOLOv3 in the number of total detectable humans, as depicted in Fig. 11.11. Finally, modifications to the best performing RetinaNet50 variant, as described in Sec. 3.2, show vast improvements for the `field_test` dataset as illustrated in Fig. 11.12. While the logical OR merging scenario of optical and thermal detections already brings a performance improvement, the customized anchor bounding boxes vastly increase detection performance and boost up the total amount of detectable bounding boxes by over 7%. Finally, doubling the

image size during inference further increases this amount to a total of 70% detected bounding boxes while making less than one false positive per image. This is an improvement of over 20% when compared to the plain version of the original RetinaNet50 variant.

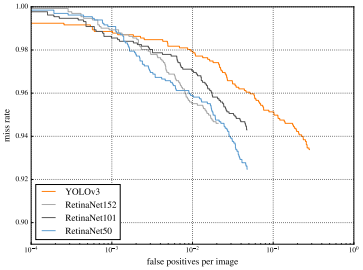


Figure 11.11: Performance of all networks on the total `field_test` sets using standard networks without any changes on anchors, image resolution or any merging of IR-RGB information.

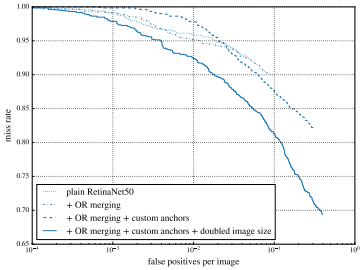


Figure 11.12: Performance on the total `field_test` showing the gradual improvements when adding our modifications to the plain RetinaNet50 variants, as described in Sec. 3.2.

Individual Human Detection A final evaluation investigates how well the pipeline detects human individuals: Every human in our novel dataset is labeled with a unique ID. Re-detections of the same individual can therefore be conveniently recognized. For search and rescue scenarios, one person does not necessarily need to be detected in every single frame. Instead, it is more important that an individual is detected at least once during the mission. It is therefore informative to investigate how many of all the distinct individual human IDs are detected by the pipeline as illustrated in Fig. 11.13 and 11.14. By calculating the miss-rate of actual human IDs instead of single ground-truth bounding boxes, these final results show final human ID miss-rates of the best performing RetinaNet50 of less than 30% while still making less than one false positive per image. This is a successful detection of over 70% of all individual humans at least once.

Qualitative samples Fig. 11.16 presents qualitative examples, including an example for the merging of the optical and infrared image and detected humans during a night flight.

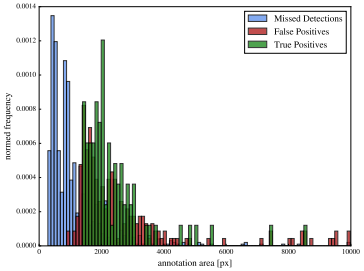


Figure 11.13: Bounding box size analysis of the best performing RetinaNet50 on the optical part of the `davos_rega` test sets. RetinaNet is able to generate true positives at a bounding box size of around 2000 px and fails to make any predictions with sizes below approximately 1500 px.

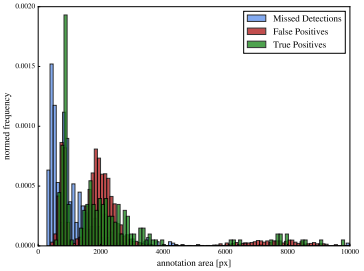


Figure 11.14: Bounding box size analysis of the best performing RetinaNet50 on the optical part of the `davos_rega` test sets using custom anchors. RetinaNet now generates most of its true positive predictions at a bounding box size of around 1000 px and is further able to make predictions across a wider range of sizes.

7 Conclusion

This work presented our human detection framework that is able to detect, track, localize, and re-identify humans from UAVs with the help of an infrared and optical camera. Based on a detailed evaluation, it can be concluded that the RetinaNet variants, in particular RetinaNet50, are superior to YOLOv3 and to a classical human detection pipeline that served as a lower baseline. The major advantage of the RetinaNet architecture appears to be the focal loss that is capable of coping with drastic imbalances between the number of foreground and background samples. Moreover, customizing the anchors is crucial for the detection of humans seen from high altitudes. Enlarged or higher resolution images further improve the detection performance. Finally, the evaluation demonstrated that the logical OR merging scenario of both optical and thermal images helps to improve detection performance, especially for more challenging datasets like the introduced `field_test` sets. Both the quantitative and the qualitative evaluation emphasize the conclusion that our novel pipeline indeed also works on a challenging real-world dataset successfully exploiting and combining information from both optical and thermal images. Given the performance of detecting individual human IDs and the final qualitative examples, it could even be questioned whether the human detection performance might already be surpassed by RetinaNet50 under certain circumstances.

Acknowledgments

The research leading to these results has received funding from the Swiss air rescue organization [3]. Furthermore, the authors wish to thank the following persons for their help: Thomas Mantel (Sensor triggering and calibration), Yves Allenspach (camera mounts), and Daniel Hentzen (circular calibration target for thermal camera and graphical user interface).

Sequence	Frames		Annotations				
	total	annotated	total	upright	sitting	lying	occluded
roof_new	1 900	1 815	4 748	4 748	0	0	584
roof_old	2 898	2 758	23 165	23 165	0	0	2 396
roof_test	447	447	1 462	1 462	0	0	172
roof_val	231	231	991	991	0	0	258
davos_rega_01	2 553	86	159	116	12	31	21
davos_rega_02	4 806	360	543	374	0	169	37
davos_rega_03	5 853	35	66	31	18	17	20
davos_rega_04	2 962	203	322	166	3	153	29
hinwil_01	2 069	210	417	417	0	0	59
hinwil_02	9 028	1 221	1 994	1 360	530	104	262
solair	7 698	97	155	121	0	34	38
Total	**40 445**	**7 463**	**34 022**	**32 951**	**563**	**508**	**3 876**

(a) Internal optical image sequences.

Sequence	Frames		Annotations				
	total	annotated	total	upright	sitting	lying	occluded
roof_new	1 834	1 701	5 752	5 752	0	0	1 825
roof_old	1 480	1 316	6 141	6 141	0	0	477
roof_test	447	447	1 421	1 421	0	0	205
roof_val	231	231	627	627	0	0	138
davos_old	2 204	664	1 004	1 004	0	0	53
davos_rega_01	2 585	11	14	7	0	7	2
davos_rega_02	4 842	66	71	71	0	0	4
davos_rega_03	5 920	16	16	0	16	0	0
davos_rega_04	2 996	115	144	52	0	92	21
hinwil_01	2 012	114	194	194	0	0	17
hinwil_02	730	15	15	0	15	0	2
rothenturm	37 413	9 852	21 785	21 785	0	0	1 120
solair	27 324	0	0	0	0	0	0
tessin	1 011	43	44	27	0	17	1
Total	**91 029**	**14 591**	**37 228**	**37 081**	**31**	**116**	**3 856**

(b) Internal infrared image sequences.

Table 11.6: Recorded and labeled internal datasets.

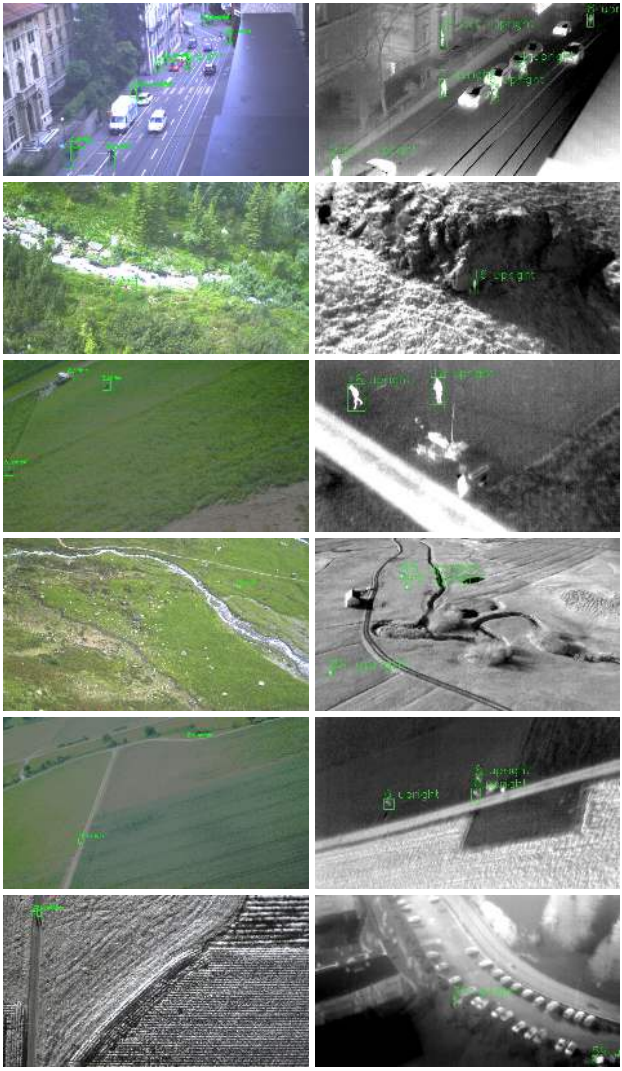


Figure 11.15: Annotated frames of the collected datasets containing a total of over 70 000 humans in different poses, in both optical and thermal imagery, and in a variety of different environments.

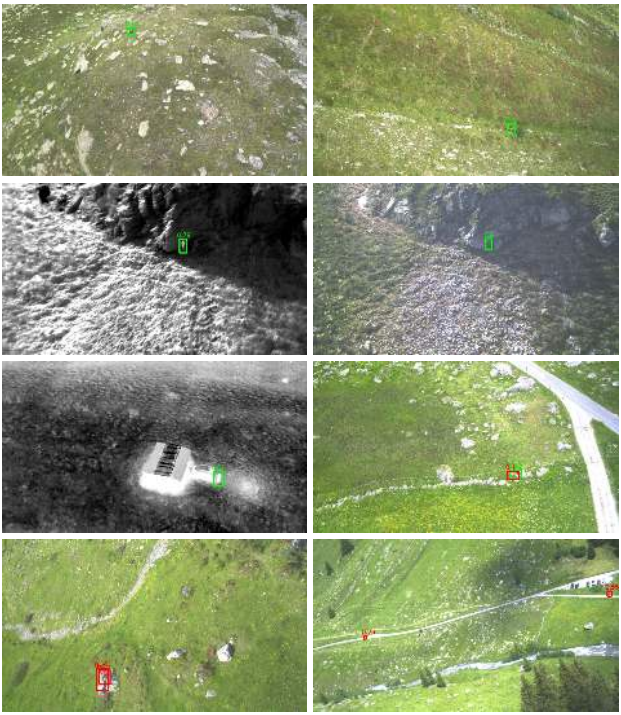


Figure 11.16: Qualitative samples. First row: Successfully detected humans, sitting and occluded. Second row: Working merging of optical and infrared image. Here, the human is easier to detect in the infrared spectrum. Third row: Night flight and detected mannequin that was placed before the flight. Fourth row: Correct detections by the network that have been missed during the manual labeling process.

Sequence	Frames		Annotations	
	total	annotated	total	occluded
Mini-drone [30]				
set10 [a]	543	542	1 207	0
set13 [b]	570	569	569	0
Stanford drone [224]				
bookstore00	13 335	13 335	246 158	1 494
bookstore06	14 558	14 305	64 944	7 090
coupa01	11 966	11 966	71 136	3 726
coupa02	11 966	11 474	66 867	3 082
gates07	2 202	2 202	14 982	365
hyang02	12 272	12 272	172 880	4 855
hyang05	10 648	10 648	123 521	123
hyang07	574	574	14 637	221
hyang09	574	574	1 930	592
hyang10	9 928	9 928	64 228	7 845
hyang12	9 928	9 619	39 030	2 810
little00	1 518	1 518	24 517	508
little01	14 070	13 828	52 399	477
quad03	509	509	2448	0
UAV123 [188]				
bike01	553	553	553	0
person01	799	799	799	0
person02	2 514	2 514	2 514	0
person03	643	643	643	0
person04	254	254	254	0
person05	2 101	2 101	2 101	0
person06	658	658	658	0
person07	1 943	1 873	1 873	0
person08	126	126	126	0
person10	582	514	514	0
person12	1 621	1 548	1 548	0
person13	155	155	155	0
person14	2 034	2 034	2 034	0
person15	712	712	712	0
person16	1 147	1 038	1 038	0
person17	1 852	1 820	1 820	0
person22	24	24	24	0
person23	153	153	153	0
wakeboard02	733	733	733	0
wakeboard03	748	748	748	0
wakeboard04	586	586	586	0
wakeboard05	758	758	758	0
wakeboard06	401	401	401	0
wakeboard07	59	59	59	0
wakeboard08	321	321	321	0
41	**136 638**	**134 988**	**982 578**	**33 188**

(a) Utilized external optical image sequences.

[a] Original name: Normal_Static_Night_Empty_1_3_1 (test)

[b] Original name: Normal_Static_Day_Half_0_1_1 (training)

Sequence	Frames		Annotations	
	total	annotated	total	occluded
ETH TIR [211]				
asl	659	659	1 021	191
sempach06	600	413	413	0
sempach07	370	359	1 391	140
sempach08	634	634	1 359	233
sempach09	576	576	982	59
sempach10	261	215	321	19
sempach11	197	192	707	113
sempach12	775	724	724	50
OTCVBS 1 [53]				
set01	31	31	91	0
set02	28	28	100	0
set03	23	23	101	0
set04	18	18	109	0
set05	23	23	101	0
set06	18	18	97	0
set07	22	22	94	0
set08	24	24	99	0
set09	73	73	95	0
set10	24	24	97	0
OTCVBS 11 [274]				
set02 [a]	1 273	1 273	69 841	0
set03 [b]	1 131	1 131	72 686	0
PTB TIR [173]				
stranger01	95	95	95	0
stranger02	280	280	280	0
stranger03	100	100	100	0
walking	315	315	315	0
VOT TIR [149]				
jacket	1 451	1 451	1 451	178
25	**9 001**	**8 701**	**152 670**	**983**

(b) Utilized external infrared image sequences.

[a] Original name: set2/seq3/nuc.

[b] Original name: set2/set4/nuc.

Table 11.7: All external datasets used for training

Paper IX

Collaborative 3D Reconstruction using Heterogeneous UAVs: System and Experiments

Timo Hinzmann, Thomas Stastny, Gianpaolo Conte, Patrick Doherty, Piotr Rudol, Marius Wzorek, Igor Gilitschenski, Enric Galceran, Roland Siegwart

Abstract

This paper demonstrates how a heterogeneous fleet of unmanned aerial vehicles (UAVs) can support human operators in search and rescue (SaR) scenarios. We describe a fully autonomous delegation framework that interprets the top-level commands of the rescue team and converts them into actions of the UAVs. In particular, the UAVs are requested to autonomously scan a search area and to provide the operator with a consistent georeferenced 3D reconstruction of the environment to increase the environmental awareness and to support critical decision-making. The mission is executed based on the individual platform and sensor capabilities of rotary- and fixed-wing UAVs (RW-UAV and FW-UAV respectively): With the aid of an optical camera, the FW-UAV can generate a sparse point-cloud of a large area in a short amount of time. A LiDAR mounted on the autonomous helicopter is used to refine the visual point-cloud by generating denser point-clouds of specific areas of interest. In this context, we evaluate the performance of point-cloud registration methods to align two maps that were obtained by different sensors. In our validation, we compare classical point-cloud alignment methods to a novel probabilistic data association approach that specifically takes the individual point-cloud densities into consideration.

Published in:
International Symposium on Experimental Robotics (ISER), 2016

1 Introduction

Field robotics has seen great gains in recent years owing both to robustified robotic platforms and increasing autonomous behaviors and capabilities. In particular, autonomous unmanned aerial vehicles (UAVs) of various classes, utilizing state-of-the-art perceptive sensors and sensing techniques, have proven worth in both large- and small-scale mapping applications, providing a wide array of sensor data. In large-scale mapping scenarios, recent developments in solar-powered, fixed-wing UAV (FW-UAV) technology have enabled extreme long-endurance for low-altitude coverage of vast areas in a compact and hand-launchable form [196]. Finer-resolution mapping on a smaller scale has also been demonstrated using aerial laser scans from autonomous helicopters [50]. Fast and fully autonomous generation of up-to-date maps could potentially be a great advantage for rescue workers looking for missing persons, or in disaster management scenarios, like floods, forest fires, and earthquakes. However, a crucial element to the utility of such operations is the ease of use for, possibly, non-technical operators. Further, no single UAV is a one-fits-all solution for the wide array of sensing data that may be required by end users. In these cases, a robotic team of various actors with mixed, but complementing, capabilities, working together within the context of a collaborative, cognitive framework, on a higher-abstraction, would be particularly impactful.

2 Problem statement: Collaborative 3D reconstruction

Point-cloud generation from optical cameras on a large scale from small FW-UAVs is sparse, due to, relatively high flying altitudes and limited image resolution. Contrarily, laser point-clouds generated from low-flying autonomous helicopters are dense, but only cover a small area. Merging these two data types together into a single global map has obvious benefits in the sense of real-world search and rescue or disaster management operations, where a large scale (sparse) map could provide operators with coarse information and a means to select “areas of interest" to send agents for a “closer look". This closer look would provide dense maps of smaller areas which, when merged with the global map, results in a more accurate representation of the environment for both, human operators and collaborating robotic actors. Leveraging the various capabilities of each participating agent in an autonomous manner also requires a higher abstraction of task delegation. In sum, we show a real-world demonstration of distributed, autonomous map making and vision-to-laser point-cloud registration from differing aerial views and mixed sensor data.

In this context, we employ a novel probabilistic data association method [7] that robustly aligns two maps that were generated by different sensors. Compared to [7], we see the following major contribution: Our data was recorded on different agents, sensor units and flight paths and hence represents a real-world scenario. In contrast, the data presented in [7] was recorded by the same agent and/or even

Figure 12.1: Two UAV platforms during their cooperative scanning mission.

Figure 12.2: Aerial image of test site near Motala, Sweden.

with the same sensor which simplifies the registration process.[1]

3 Technical Approach

This section opens with an overview of the delegation framework. For more details we refer to the companion paper [66]. The section proceeds with a description of the state estimation, point-cloud generation and concludes with a focus on the point-cloud registration methods.

3.1 Mission Process and Delegation framework

A high-level depiction of the mission process is provided in Fig. 12.3. The delegation framework [64] which includes delegation modules from each of the participating agents, provides both a formal and software infrastructure for specifying and generating collaborative multi-agent plans to achieve complex goals such as multi-UAV 3D reconstruction of selected regions. Delegation is based on a recursive algorithm that sends requests of the following type, Delegate(*Agent1, Agent2, Task, Context*), where *agent1* makes an attempt to delegate *Task* to *Agent2*, given a specific *Context* specified as a set of constraints. Examples of constraints would be temporal constraints or restrictions on flight altitudes. Agents can be humans or robots. Tasks are represented using Task Specification Trees (TSTs). TSTs have both declarative and procedural descriptions. Internal nodes in a TST represent control modes such as sequence and concurrency, while leaf nodes represent domain dependent elementary tasks executable by different participating platforms. The delegation process, as illustrated in Fig. 12.4, itself begins with a goal request TST often provided by a human operator and if successful, results in an

[1] Due to space constraints in this publication, only a subset of the data can be presented. However, the datasets can be requested from the authors.

Figure 12.3: Mission Process: A human operator broadcasts a goal request for a data acquisition mission via its delegation module. Platforms with available capabilities reply and a delegation process ensues among each of the platforms' delegation modules. If successful, the net result is a joint plan to execute. Upon execution, raw/processed data can be stored locally or globally. During the mission or upon mission completion the human operator can access the results via specialized interfaces.

expanded TST where all constraints are satisfied. Sub-trees in the final TST are also appropriately allocated to those platforms with the proper capabilities. TSTs can be generated dynamically using automated planning techniques, or by using generic TST templates that can be instantiated appropriately. In this example (Fig. 12.4), a concurrent scanning plan is generated for one region where two separate sub-regions are covered by each of the UAVs involved, respectively. The scan-map task in the TST calls a region partitioning algorithm to determine appropriate sub-regions for platforms to scan based on their capabilities. The delegation process itself is quite complex and involves auctions, constraint solving and dynamic TST expansion. The scan-map task involves use of partitioning algorithms and the scan-map-single tasks involve internal path planning by the respective platforms. During the mission execution phase, each system executes its part of the mission TST relative to timing and other constraints.

3.2 State estimation and point-cloud generation

RW-UAV

The state estimation is used both for autonomous navigation and for point-cloud generation by incorporating laser scanner measurements in form of a direct georefer-

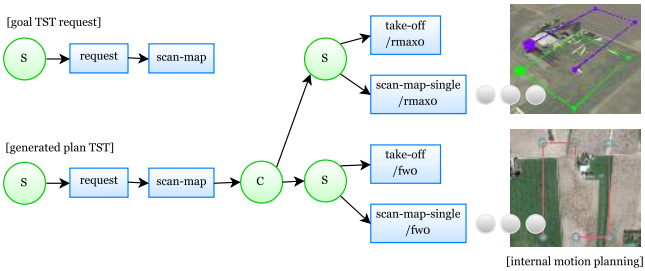


Figure 12.4: Goal TST request from operator and generated plan TST involving both RMAX (/rmax0) and FW-UAV (/fw0). Internal nodes: (C) concurrent, (S) sequence.

encing technique [248]. It is based on a Kalman filter algorithm which fuses inertial and GNSS position data. The deployed Kalman filter uses a linear state-space error dynamic model derived from a perturbation analysis of the equations of motion [36]. The Kalman filter produces state estimation at 50 Hz rate and performs the update step using GNSS measurement at 20 Hz rate.

FW-UAVs

The Pixhawk PX4 auto-pilot performs an indirect EKF-based state estimation as presented in [160]. Within the Kalman filter linear acceleration and angular rates measurements are used for propagation of the system state. Pressure, GPS velocity and position, as well as magnetometer measurements are used for the state update [160]. The estimated states involve the IMU's attitude and position in WGS84 coordinates[2]. The vision point-cloud was acquired with an optical camera using a classical photogrammetric approach[3].

3.3 Point-cloud registration

The alignment of point-clouds generated from two unmanned aerial vehicles with different sensors involves consideration of the following challenging aspects: Firstly, one point in the visual source point-cloud does in general not correspond to a point in the laser target cloud and vice versa. Secondly, the sensor noise models are different: Peaky for laser but more spread for visual points due to camera noise and triangulation errors. Thirdly, the laser point-cloud is in general denser than the vision point-cloud. Furthermore, the robots fly at high altitudes. Consequently, a dominant ground plane and few depth discontinuities are common for most datasets.

[2]For more details about the state estimation framework we refer to [160].
[3]The vision point-clouds are generated using the commercial software pix4d.

Lastly, a rough initial alignment is given by global positioning systems such as GNSS or fused from the state estimator. To register a visual sparse point-cloud to a dense laser point-cloud the following registration algorithms are evaluated with respect to the problem specifications described above: Iterative Closest Point (ICP), Iterative Probabilistic Data Association (IPDA), Generalized Iterative Closest Point (GICP), and Normal Distribution Transform (NDT).

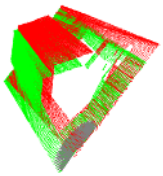


Figure 12.5: Correspondences (grey) for one source point obtained by kd-tree.

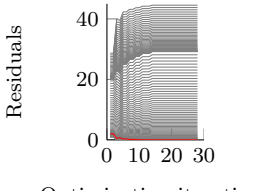


Figure 12.6: Residuals for one source point after one iteration.

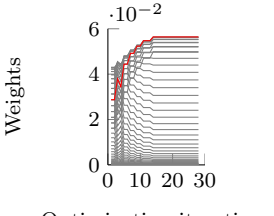


Figure 12.7: Weights for one source point after one iteration.

One iteration of the Probabilistic Data Association [7] approach consists of the following steps: For every point of the source cloud a kd-tree search is performed with maximal radius $r_{kd,}$ and maximal number of returned neighbours n_{kd} as shown in Fig. 12.5. For every source-target correspondence, the residuals and weights are calculated as illustrated in Fig. 12.6 and 12.7 respectively employing expectation-maximization (EM) in combination with e.g. a Gaussian or t-distribution. The red line indicates the evolution of the true correspondence residual and weight for 28 Levenberg-Marquardt optimization steps. Note that the data associations do not change during one iteration but only the residuals and weights update based on the iterative solution of the Levenberg-Marquardt algorithm. These steps can be performed iteratively to increase the area of convergence (IPDA). The advantages of this approach relevant to the problem specifications are the following: (1) a sensor model can be intuitively inserted in the EM-algorithm based on the expected noise, (2) a point of the sparse source cloud can hold correspondences to many points in the target cloud. Iterative Closest Point (ICP) [21, 45, 209, 284] is a widely used registration algorithm that has inspired many variants. For the evaluation we use the classic point-to-point approach implemented in the Point Cloud Library (PCL). It performs the following steps until convergence: (1) For every point in the source cloud find *the* closest point in the target cloud. (2) Estimate and apply the transformation $\mathbf{T}$ that best transforms the source cloud to the target cloud in the sense of a mean squared error. Naturally, ICP's assumption that one point in the source cloud has an exact correspondence in the target cloud is not fulfilled in

	Parameter	Description
IPDA	r_{kd}	Kd-tree radius
	n_{kd}	Kd-tree max. neighbours
	$iter_{max}$	Max. number of iterations
	ls	Leaf size of voxel grid
ICP	d_c	Max. correspondence distance
	ϵ_T	Transformation conv. criteria
	$iter_{max}$	Max. number of iterations
	ls	Leaf size of voxel grid

	Parameter	Description
GICP	d_c	Max. correspondence distance
	ϵ_T	Transformation conv. criteria
	$iter_{max}$	Max. number of iterations
	ls	Leaf size of voxel grid
NDT	Δx	Step size
	Δr	Resolution
	ϵ_T	Transformation conv. criteria
	$iter_{max}$	Max. number of iterations
	ls	Leaf size of voxel grid

Table 12.1: Parameter notations for IPDA, ICP, GICP and NDT.

the sparse-dense registration problem. Generalized ICP (GICP) [237] levers the classic ICP and point-to-plane ICP into a probabilistic framework. Applied to the aerial registration problem, GICP may profit from dominant ground planes due to the high flying altitudes. In the Normal Distribution Transform [24] the points of the cloud are represented in form of a probability distribution and hence no explicit point correspondences between source and target cloud are established. With regard to the sparse-to-dense point-cloud registration, NDT may fail if the visual cloud is too sparse.

4 Platform description

The platforms used for the experiments include a rotary-wing Yamaha RMAX and the two fixed-wing UAVs named Techpod and senseSoar.

4.1 RW-UAV

The Yamaha RMAX helicopter [65], shown in Fig. 12.1 and 12.3, has a rotor diameter of 3.1 m, a maximum take-off weight of 94 Kg and a payload capability of about 30 kg. The platform is capable of fully autonomous navigation, including take-off and landing. The basic sensor suite used for autonomous navigation includes a fiber optic tri-axial gyro system and a tri-axial accelerometer system, a RTK GNSS positioning system and an infrared altimeter used for automatic landing. Onboard sensors used for mapping missions include color and thermal video cameras, as well as a class 1 SICK LMS511 PRO 2D laser scanner. The laser scanner's maximum range is 80 meters with a maximum scanning FoV of 190°.

4.2 FW-UAVs

Techpod

The small unmanned research plane Techpod is shown in Fig. 12.1. It has a classic T-tail configuration, is equipped with one propeller, has a wingspan of 2.60 m and a nominal speed of around 12 m/s. The sensor and processing unit [196] as well as the PX4 auto-pilot are located inside the modified fuselage and allow autonomous mission execution such as GPS waypoint following.

senseSoar

The highly versatile solar-UAV senseSoar was developed at the Autonomous Systems Lab for search & rescue missions and has a wingspan of 3.1 m. With its solar panels it is able to generate an electric power of around 140 W and has shown long-endurance capabilities. Likewise as Techpod, senseSoar is hand-launchable and carries the sensor pod inside the fuselage.

5 Experimental results

The datasets were collected at two locations: (1) in Motala, Sweden which includes a flight field with several houses and trees. The resulting experiments are presented in Sections 5.1 & 5.2; (2) at a mountainside in Isollaz, Italy as presented in Experiment III in Section 5.3.

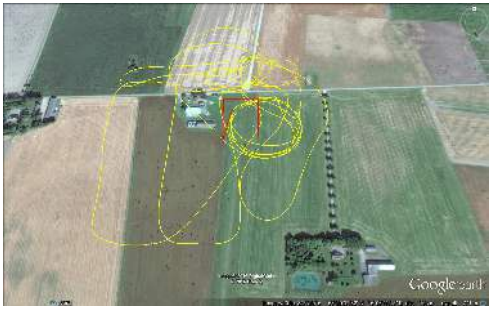


Figure 12.8: Sample flight path of the FW-UAV (yellow) for Exp. I and II: The FW-UAV is loitering in-air until it receives the command for scanning the area from the delegation framework. Based on this request, the path-planner located on the ground station generates a scanning pattern which is transmitted to the FW-UAV via telemetry. After execution, the imagery is sent to the ground station via WiFi where the point-cloud is generated. The path of the RW-UAV is plotted in red. The nominal altitude of the FW- and RW-UAV is 100 m and 48 m.

5.1 Exp. I: Complementary factor of vision-laser pc-alignment

In a first experiment, the vision point-cloud generated with images recorded by a Sony ActionCam HDR-AS100V mounted on Techpod is aligned to the laser point-cloud generated with a SICK laser scanner onboard of the RMAX. Fig. 12.9-12.11 show a satellite image, colored vision and laser point-cloud of the region of interest. Note that the laser did not receive response for parts of the roof and barn due to a steep observation angle, relatively low altitude of the RMAX, and

due to non-reflective surfaces. On the other hand, the laser point-cloud contains

Figure 12.9: Satellite image as reference.

Figure 12.10: Vision point-cloud.

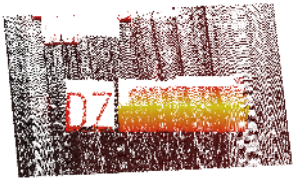


Figure 12.11: Laser point-cloud colored by height

less measurement noise and a higher level of detail as can be best seen in Fig. 12.12 which e.g. depicts a wind vane in the top right corner not observed by the visible light camera. These observations underline the complementary factor of the two point-clouds which, when aligned, result in a more complete model of the environment. Fig. 12.12 and 12.13 illustrate the initial misalignment from

Figure 12.12: Side-view: Vision point-cloud colored by pixel intensity and laser point-cloud in green.

Figure 12.13: Top-view: Vision point-cloud colored by pixel intensity and laser point-cloud in green.

side and top view respectively. This georeferencing error is given in Table 12.2 and consists of a translational offset of several meters and a small rotation. The transformation was obtained by careful manual alignment of the point-clouds and used to evaluate point-cloud registration methods quantitatively. Note that due to the noisy character of the data, this manual alignment should not be considered perfect as slightly varying alignments seem still visually satisfying. Nevertheless, this method allows to reason about convergence and general trends. Fig. 12.14 and 12.15 show the transformation error for IPDA and ICP plotted over the number of iterations. Both ICP and IPDA converge to almost the same transformation. Furthermore, from the given plots it can be seen that the altitude offset converges first in very few iterations due to the dominant ground planes. The translational offset in x and y usually needs more iterations to converge. Fig. 12.17 shows the

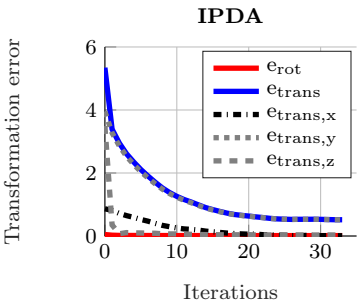


Figure 12.14: IPDA: Rotational (red) and translational (blue) misalignment error.

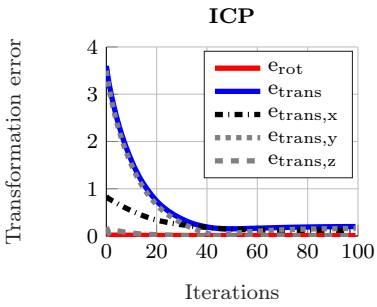


Figure 12.15: ICP: Rotational (red) and translational (blue) misalignment error.

t_x	-1.05 m
t_y	3.52 m
t_z	6.46 m
φ	-0.0162 rad
θ	0.0152 rad
ψ	-0.0206 rad

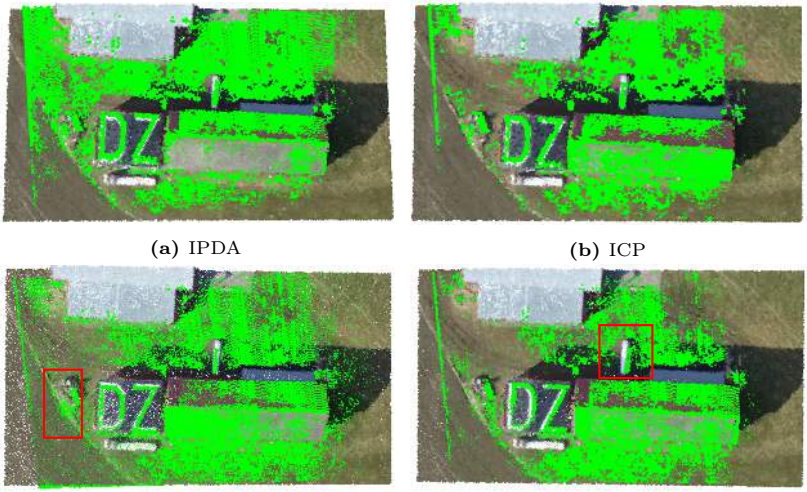
	Parameters	e_{trans}	e_{rot}	iter.
IPDA	$r_{kd}: 5.0, n_{kd}: 50, iter_{max}: 1000, ls: 0$, student-t	0.5160	0.0129	34
ICP	$d_c: 10, \epsilon_T: 10^{-16}, iter_{max}: 500, ls: 0$	0.1992	0.0091	100
GICP	$d_c: 10, \epsilon_T: 10^{-16}, iter_{max}: 500, ls: 0$	4.075	0.0473	6
NDT	$\Delta x: 0.1, \Delta r: 0.1, \epsilon_T: 10^{-16}, iter_{max}: 1000, ls: 0$	1.8213	0.025	1000

Table 12.2: Initial misalignment transformation error and final translational and rotational offset for IPDA, ICP, GICP and NDT. The translation error e_{trans} and rotation error e_{rot} are computed as proposed in [209].

aligned vision and laser point-cloud using IPDA. Fig. 12.16(a)-(d) show the final alignments computed by the individual registration methods. The figures and Table 12.2 illustrate that all methods show convergence, however, small misalignment errors are visible for GICP and NDT.

5.2 Exp. II: Changes in the environment

This experiment evaluates if agents that possess only poor absolute position sensing capabilities can register to an a-priori obtained and well georeferenced point-cloud. This evaluation gives an idea of how well the different methods can deal with changes in the environment as well as about their region of convergence. For this purpose, we align the vision point-cloud shown in Fig. 12.19 to the previously generated laser point-cloud given in Fig. 12.18. Several changes in the environment can be spotted, in particular, the vegetation, location of cars and of a small house. Furthermore,



(a) IPDA **(b)** ICP

(c) GICP **(d)** NDT

Figure 12.16: Fig. (a)-(d) illustrate that all registration methods converged, however, small misalignment errors are observable for GICP and NDT depicted by the red rectangles.

we generate a random large initial misalignment error between both point-clouds as shown in Table 12.3. Fig. 12.20-12.22 as well as Table 12.3 demonstrate that IPDA, in particular when employing t-distribution, results in the lowest final misalignment error, followed by GICP and ICP, whereas NDT diverged for this scenario.

5.3 Exp. III: Point-cloud sparsity

In Experiment III, we present a very challenging dataset consisting of a tree region with few man-made structures. The mission procedure is illustrated in Fig. 12.23: The FW-UAV, equipped with a Sony ActionCam and a grayscale Aptina MT9v034 camera, generates a rough initial point-cloud as soon as it receives the command from the delegation framework initiated by the human operator. Subsequently, the RW-UAV scans the region of interest with the aid of the SICK laser for refinement. The generated laser and vision point-clouds are shown in Fig. 12.24 and 12.25 respectively. We deliberatively use the point-cloud generated by the low-resolution grayscale camera for point-cloud registration and employ IPDA to underline its performance when dealing with dense and sparse point-clouds. In contrast to

Figure 12.17: Aligned vision and laser point-cloud (green) using IPDA. The experiment underlines the complementary factor of employing laser and vision to obtain a more complete model of the environment.

Figure 12.18: Point-cloud generated by the laser scanner mounted on the RW-UAV. The point-cloud consists of 568'839 points and is colored by height.

Figure 12.19: Point-cloud generated by the RGB camera mounted on the FW-UAV. The house area shown in the top consists of 163'595 points.

Experiment I and II, the images are geo-registered by the onboard EKF instead of using the raw GPS measurements. As expected, the initial misalignment error is limited to $3.34, -2.51, -0.26$ m for the translational and -0.016, 0.0082, 0.0089 rad for the rotational offset. The initial misalignment and final registration are shown in Fig. 12.26.

6 Conclusion

In this paper, we presented an automated delegation framework that translates top-level commands of the human operator into low-level commands of the employed agents. We validated the framework based on realistic scenarios in two locations including more than 20 flights using a RW- and FW-UAV representing an arbitrary fleet of heterogenous agents. Furthermore, we chose the task of scanning a common area as *one* exemplary mission of the delegation framework. The point-clouds acquired during this scanning process are automatically registered and transferred

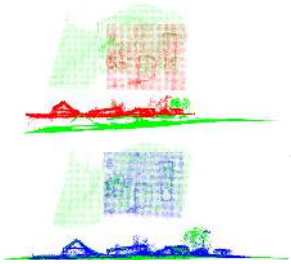


Figure 12.20: Initial misalignment and final registration using IPDA with t-distribution.

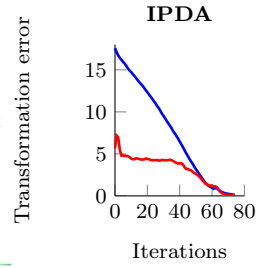


Figure 12.21: IPDA: Translation (blue) and rotation error (red; multiplied by $1e2$ for better visualization).

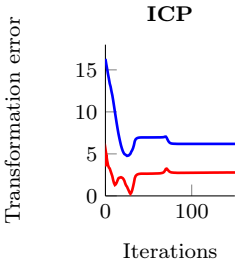


Figure 12.22: ICP: Translation (blue) and rotation error (red; multiplied by $1e2$ for better visualization).

t_x	11.44 m
t_y	12.97 m
t_z	−8.32 m
φ	0.025 rad
θ	−0.0013 rad
ψ	−0.0411 rad

	Parameters	e_{trans}	e_{rot}	iter.
IPDA	$r_{kd}: 5.0, n_{kd}: 200, iter_{max}: 200, ls: 1.5$, student-t	0.1152	0.011	74
	$r_{kd}: 5.0, n_{kd}: 200, iter_{max}: 200, ls: 1.5$, Gaussian	3.3278	0.0243	80
ICP	$d_c: 10, \epsilon_T: 10^{-16}, iter_{max}: 500, ls: 1.5$	6.1891	0.0278	190
GICP	$d_c: 10, \epsilon_T: 10^{-16}, iter_{max}: 500, ls: 0.1$	2.6757	0.0201	33
NDT	$\Delta x: 0.1, \Delta r: 1.0, \epsilon_T: 10^{-16}, iter_{max}: 500, ls: 0$	22.2777	0.1126	500

Table 12.3: Initial misalignment transformation error and final translational and rotational offset for IPDA, ICP, GICP and NDT.

back to the human operator and visualized in the dynamic cognitive map. Our experiments show the complementary factor of vision-laser point-cloud registration from aerial views and demonstrate the successful deployment of the Probabilistic Data Association algorithm. The final goal of this project will be to allow accurate path planning of unmanned ground vehicles (UGV) or smaller multicopter UAVs based on the aligned map and, for instance, to delegate them inside the buildings' interior. Future work will also include the integration of a previously presented human detection algorithm [263] into the delegation framework. The algorithm returns the UTM location of possible victims along with their detection uncertainties. Other agents may verify these possible human detections to decrease the false alarm rate.

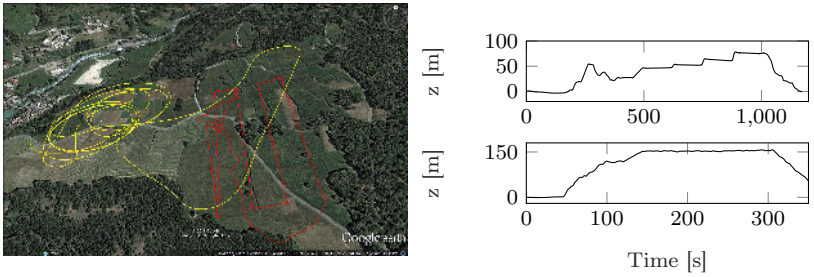


Figure 12.23: The satellite image shows the flight path of the fixed-wing UAV in yellow and of the RMAX helicopter in red. The plots on the right show the altitude in form of height above ground with respect to the individual starting positions.

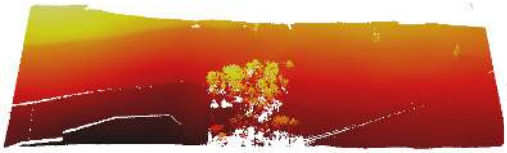


Figure 12.24: One of the laser strips to be aligned to the vision point-cloud.

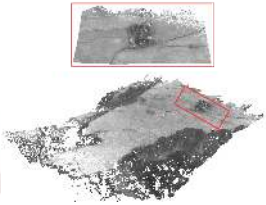
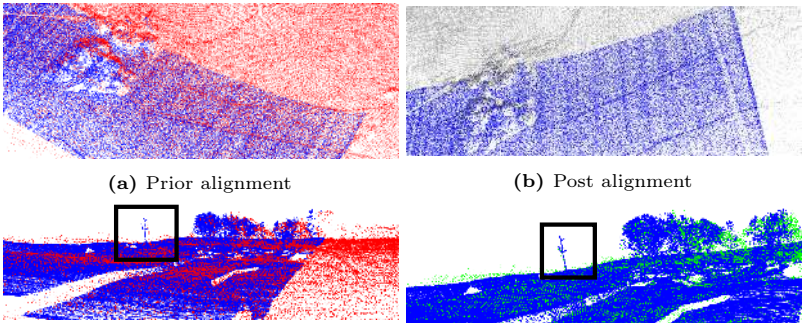


Figure 12.25: Point-cloud generated by the grayscale camera mounted on the FW-UAV.

Acknowledgment

The research leading to these results has received funding from the European Commission's Sevenths Framework Programme (FP7/2007-2013) under grant agreement n°285417 (ICARUS) and n°600958 (SHERPA). Furthermore, the authors want to thank Gabriel Agamennoni and Simone Fontana for providing an initial implementation of the probabilistic data association algorithm.



(a) Prior alignment **(b)** Post alignment

(c) Prior alignment **(d)** Post alignment

Figure 12.26: The dense laser point-cloud is shown in blue. The prior misalignment is especially visible at the hill gradients. The tree region was only partially captured by the FW-UAV and could be densified by the laser scan.

Bibliography

[1] Aurora flight sciences. https://www.aurora.aero/. Accessed: 2020-08-02.

[2] Pix4D dataset Cadastre. https://support.pix4d.com/hc/en-us/articles/202561399-Example-Datasets-Available-for-Download-Cadastre-.

[3] Rega – the swiss air-rescue organization. www.rega.ch/en/. Accessed: 2020-08-02.

[4] Yamaha rmax. https://www.yamahamotorsports.com/motorsports/pages/precision-agriculture-rmax. Accessed: 2020-08-02.

[5] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart. Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments. In *ICRA*, pages 3056–3063. IEEE, 2011.

[6] M. W. Achtelik, S. Weiss, M. Chli, F. Dellaert, and R. Siegwart. Collaborative stereo. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2242–2248. IEEE, 2011.

[7] G. Agamennoni, S. Fontana, R. Y. Siegwart, and D. G. Sorrenti. Point Clouds Registration with Probabilistic Data Association. In *Proc. of The International Conference on Intelligent Robots and Systems*. IEEE, 2016.

[8] A. Agarwala et al. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, July 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141966. URL http://doi.acm.org/10.1145/1141911.1141966.

[9] M. Agoston. *Computer Graphics and Geometric Modelling*. Computer Graphics and Geometric Modeling. Springer, 2005. ISBN 9781852338183. URL https://books.google.ch/books?id=fGX8yC-4vXUC.

[10] M. Andriluka, P. Schnitzspan, J. Meyer, et al. Vision Based Victim Detection from Unmanned Aerial Vehicles. In *International Conf. on Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ*, pages 1740–1747. IEEE, 2010. ISBN 9781424466757. doi: 10.1109/IROS.2010.5649223.

[11] I. Anokhin, P. Solovev, D. Korzhenkov, A. Kharlamov, T. Khakhulin, A. Silvestrov, S. Nikolenko, V. Lempitsky, and G. Sterkin. High-resolution daytime translation without domain labels, 2020.

[12] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[13] F. Aznar, M. Pujol, and R. Rizo. Visual navigation for uav with map references using convnets. In *Conference of the Spanish Association for Artificial Intelligence*, pages 13–22. Springer, 2016.

[14] P. Azzari et al. *An Evaluation Methodology for Image Mosaicing Algorithms*, pages 89–100. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-88458-3. doi: 10.1007/978-3-540-88458-3_9. URL http://dx.doi.org/10.1007/978-3-540-88458-3_9.

[15] B. Babenko, M. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 983–990, 2009.

[16] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[17] D. B. Barber et al. Autonomous landing of miniature aerial vehicles. *JACIC*, 4(5):770–784, 2007. doi: 10.2514/1.26502. URL https://doi.org/10.2514/1.26502.

[18] A. J. Barry and R. Tedrake. Pushbroom stereo for high-speed navigation in cluttered environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3046–3052. IEEE, 2015.

[19] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, pages 404–417, 2006.

[20] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani. A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery. *Remote Sensing*, 9(2):100, 2017. ISSN 20724292. doi: 10.3390/rs9020100.

[21] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, Feb. 1992. ISSN 0162-8828. doi: 10.1109/34.121791. URL http://dx.doi.org/10.1109/34.121791.

[22] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[23] Y. Biadgie and K. A. Sohn. Feature detector using adaptive accelerated segment test. In *2014 Int. Conf. on Information Science Applications (ICISA)*, pages 1–4, May 2014. doi: 10.1109/ICISA.2014.6847403.

[24] P. Biber. The Normal Distribution Transform: A New Approach to Laser Scan Matching, 2003.

[25] J. L. Blanco et al. nanoflann: a C++ header-only fork of FLANN, a library for Nearest Neighbor (NN) wih KD-trees. https://github.com/jlblancoc/nanoflann, 2014.

[26] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 298–304. IEEE, 2015.

[27] P. Blondel, A. Potelle, C. Pegard, and R. Lozano. Human Detection in Uncluttered Environments: from Ground to UAV View. In IEEE, editor, *13th Int. Conf. on Control Automation Robotics & Vision (ICARCV), 2014*, pages 76–81, 2014. ISBN 9781479951994. doi: 10.1109/ICARCV.2014.7064283.

[28] P. Blondel, A. Potelle, C. Pegard, and R. Lozano. Fast and viewpoint robust human detection for SAR operations. In *International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE*, pages 1–6. IEEE, 2014. ISBN 9781479961399. doi: 10.1109/SSRR.2014.7017675.

[29] E. Bondi, F. Fang, M. Hamilton, et al. SPOT Poachers in Action: Augmenting Conservation Drones With Automatic Detection in Near Real Time. *The Thirtieth AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI-18)*, pages 7741–7746, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16282.

[30] M. Bonetto, P. Korshunov, G. Ramponi, and T. Ebrahimi. Privacy in Mini-drone Based Video Surveillance. *Proceedings - International Conference on Image Processing, ICIP*, 2015-Decem:2464–2469, 2015. ISSN 15224880. doi: 10.1109/ICIP.2015.7351245.

[31] S. Bosch et al. Autonomous detection of safe landing areas for an UAV from monocular images. In *Intern. Conf. on Intelligent Robots and Systems, Beijing (China)*, 2006.

[32] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[33] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[34] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

[35] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Syst. J.*, 4(1):25–30, Mar. 1965. ISSN 0018-8670. doi: 10.1147/sj.41.0025. URL http://dx.doi.org/10.1147/sj.41.0025.

[36] K. R. Britting. *Inertial navigation systems analysis*. John Wiley & Sons Canada, Limited, 1971. ISBN 9780471104858.

[37] R. Brockers et al. Autonomous landing and ingress of micro-air-vehicles in urban environments based on monocular vision. In *SPIE Defense, Security, and Sensing*, 2011.

[38] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision*, 74(1):59–73, Aug. 2007. ISSN 0920-5691. doi: 10.1007/s11263-006-0002-3. URL http://dx.doi.org/10.1007/s11263-006-0002-3.

[39] M. Bürki, M. Dymczyk, I. Gilitschenski, C. Cadena, R. Siegwart, and J. Nieto. Map management for efficient long-term visual localization in outdoor environments. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 682–688. IEEE, 2018.

[40] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Computer Vision (ECCV), 2010 IEEE European Conference on*, pages 778–792.

[41] J. Canny. Finding Edges and Lines in Images. Technical report, Cambridge, MA, USA, 1983.

[42] J. F. Canny. Finding edges and lines in images. Master's thesis, MIT, 1983.

[43] L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert. Eliminating conditionally independent sets in factor graphs: a unifying perspective based on smart factors. In *ICRA*, 2014.

[44] Y. C. Chang, H. T. Chen, J. H. Chuang, and I. C. Liao. Pedestrian Detection in Aerial Images Using Vanishing Point Transformation and Deep Learning. *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1917–1921, 2018. ISSN 15224880. doi: 10.1109/ICIP.2018.8451144.

[45] Y. Chen and G. Medioni. Object Modelling by Registration of Multiple Range Images. *Image Vision Comput.*, 10(3):145–155, Apr. 1992. ISSN 0262-8856. doi: 10.1016/0262-8856(92)90066-C. URL http://dx.doi.org/10.1016/0262-8856(92)90066-C.

[46] Y. Cheng. Real-time surface slope estimation by homography alignment for spacecraft safe landing. In *Robotics and Automation, 2010 IEEE Intern. Conf. on*, pages 2280–2286. IEEE, 2010.

[47] H. Chiu, A. Das, P. Miller, S. Samarasekera, and R. Kumar. Precise vision-aided aerial navigation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 688–695, Sept 2014. doi: 10.1109/IROS.2014.6942633.

[48] H.-P. Chiu, S. Williams, F. Dellaert, S. Samarasekera, and R. Kumar. Robust vision-aided navigation using Sliding-Window Factor graphs. In *ICRA*, 2013.

[49] G. Conte and P. Doherty. A visual navigation system for uas based on geo-referenced imagery. XXXVIII-1/C22, 09 2012.

[50] G. Conte, P. Rudol, and P. Doherty. Evaluation of a Light-weight Lidar and a Photogrammetric System for Unmanned Airborne Mapping Applications. *PFG Photogrammetrie, Fernerkundung, Geoinformation*, 2014(4):287–298, 08 2014. doi: 10.1127/1432-8364/2014/0223. URL http://dx.doi.org/10.1127/1432-8364/2014/0223.

[51] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 2008.

[52] N. Dalal and W. Triggs. Histograms of Oriented Gradients for Human Detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 886–893. Elsevier, 2005. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.177. URL http://eprints.pascal-network.org/archive/00000802/.

[53] J. W. Davis and M. A. Keck. A Two-Stage Template Approach to Person Detection in Thermal Imagery. In *Workshops on Application of Computer Vision, 2005 IEEE*, pages 364–369. IEEE, 2005.

[54] T. A. Davis, J. R. Gilbert, S. I. Larimore, and E. G. Ng. A column approximate minimum degree ordering algorithm. *ACM*, 2004.

[55] D. C. De Oliveira and M. A. Wehrmeister. Towards Real-Time People Recognition on Aerial Imagery Using Convolutional Neural Networks. In *19th Int. Symp. on Real-Time Distributed Computing (ISORC), 2016 IEEE*, pages 27–34. IEEE, 2016. ISBN 978-1-4673-9032-3. doi: 10.1109/ISORC.2016.14. URL http://ieeexplore.ieee.org/document/7515608/.

[56] F. Dellaert. Factor Graphs and GTSAM: A Hands-on Introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sept 2012. URL https://research.cc.gatech.edu/borg/sites/edu.borg/files/downloads/gtsam.pdf.

[57] F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. 2012.

[58] F. Dellaert. Factor Graphs and GTSAM: A Hands-on Introduction. Technical Report GT-RIM-CP&R-2012-002, GT RIM, Sept 2012. URL https://research.cc.gatech.edu/borg/sites/edu.borg/files/downloads/gtsam.pdf.

[59] V. R. Desaraju et al. Vision-based landing site evaluation and informed optimal trajectory generation toward autonomous rooftop landing. *Auton. Robots*, 39(3):445–463, 2015.

[60] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description, 2017.

[61] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.

[62] K. Doherty, D. Fourie, and J. Leonard. Multimodal semantic slam with probabilistic data association. In *2019 international conference on robotics and automation (ICRA)*, pages 2419–2425. IEEE, 2019.

[63] P. Doherty and F. Heintz. A Delegation-Based Cooperative Robotic Framework. In *ROBIO*, pages 2955–2962. IEEE, 2011.

[64] P. Doherty, F. Heintz, and J. Kvarnström. High-level Mission Specification and Planning for Collaborative Unmanned Aircraft Systems using Delegation. *Unmanned Systems*, 1(1):75–119, 2013. ISSN 2301-3850, EISSN 2301-3869.

[65] P. Doherty, J. Kvarnström, M. Wzorek, P. Rudol, F. Heintz, and G. Conte. HDRC3 - A Distributed Hybrid Deliberative/Reactive Architecture for Unmanned Aircraft Systems. In K. P. Valavanis and G. J. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*, pages 849–952. 2014.

[66] P. Doherty, J. Kvarnstroem, P. Rudol, M. Wzorek, G. Conte, C. Berger, T. Hinzmann, and T. Stastny. A Collaborative Framework for 3D Mapping using Unmanned Aerial Vehicles. *Int. J. Comput. Vision*, 2016.

[67] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian Detection: A Benchmark. *2009 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 304–311, 2009. ISSN 1063-6919. doi: 10.1109/CVPRW.2009.5206631.

[68] N. D. H. Dowson and R. Bowden. Mutual information for lucas-kanade tracking (milk): An inverse compositional formulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):180–185, 2008.

[69] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[70] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-net: A trainable cnn for joint detection and description of local features. *arXiv preprint arXiv:1905.03561*, 2019.

[71] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[72] H. Eisenbeiß. *UAV Photogrammetry*. ETH Zurich, Switzerland:, 2009.

[73] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Computer Vision–ECCV 2014*, pages 834–849. Springer International Publishing, 2014.

[74] P. Fankhauser et al. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. In *Robot Operating System (ROS)*, volume 1, chapter 5. Springer, 2016. ISBN 978-3-319-26052-5. doi: 10.1007/978-3-319-26054-9{_}5. URL http://www.springer.com/de/book/9783319260525.

[75] P. F. Felzenszwalb, I. C. Society, R. B. Girshick, S. Member, D. Mcallester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE transactions on pattern analysis and machine intelligence*, 32 (9):1627–1645, 2010. ISSN 0162-8828.

[76] P. F. Felzenszwalb et al. Distance Transforms of Sampled Functions. *Theory of Computing*, 8(1):415–428, 2012. doi: 10.4086/toc.2012.v008a019. URL https://doi.org/10.4086/toc.2012.v008a019.

[77] A. Filippov and O. Dzhimiev. Long range 3d with quadocular thermal (lwir) camera. *arXiv preprint arXiv:1911.06975*, 2019.

[78] D. Fitzgerald et al. A vision based forced landing site selection system for an autonomous UAV. In *Intelligent Sensors, Sensor Networks and Information Processing Conf.. Proc. of the 2005 Int. Conf. on*, pages 397–402. IEEE, 2005.

[79] H. Flynn and S. Cameron. Multi-modal People Detection from Aerial Video. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, pages 815–824. Springer, 2013. ISBN 9783319009698. doi: 10.1007/978-3-319-00969-8.

[80] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE, 2014.

[81] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *RSS*, 2015.

[82] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robotics*, 33(2):249–265, 2017. doi: 10.1109/TRO.2016.2623335. URL https://doi.org/10.1109/TRO.2016.2623335.

[83] C. Forster et al. IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation. In *Proc. of Robotics: Science and Systems*, Rome, Italy, July 2015. doi: 10.15607/RSS.2015.XI.006.

[84] D. Fourie, J. Leonard, and M. Kaess. A nonparametric belief solution to the bayes tree. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2189–2196. IEEE, 2016.

[85] P. Furgale. *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. University of Toronto, 2011. ISBN 9780494781852. URL https://books.google.ch/books?id=2BR5MwEACAAJ.

[86] P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IROS*, 2013.

[87] F. Furrer, M. Fehr, T. Novkovic, H. Sommer, I. Gilitschenski, and R. Siegwart. *Evaluation of Combined Time-Offset Estimation and Hand-Eye Calibration on Robotic Datasets*. Springer International Publishing, Cham, 2017. ISBN 978-3-319-67361-5.

[88] F. Furrer et al. *RotorS—A Modular Gazebo MAV Simulator Framework*, pages 595–625. Springer International Publishing, Cham, 2016. ISBN 978-3-319-26054-9. doi: 10.1007/978-3-319-26054-9_23. URL http://dx.doi.org/10.1007/978-3-319-26054-9_23.

[89] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. *TPAMI*, 2010.

[90] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010.

[91] A. Fusiello et al. A Compact Algorithm for Rectification of Stereo Pairs. *Mach. Vision Appl.*, 12(1):16–22, July 2000. ISSN 0932-8092. doi: 10.1007/s001380050003. URL http://dx.doi.org/10.1007/s001380050003.

[92] A. Fusiello et al. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000. ISSN 1432-1769. doi: 10.1007/s001380050120. URL http://dx.doi.org/10.1007/s001380050120.

[93] D. Gabor. Theory of Communication. *J. IEE*, 93(26):429–457, Nov. 1946.

[94] D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012.

[95] P. J. Garcia-Pardo et al. Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems*, 38(1):19–29, 2002.

[96] S. Garg, N. Suenderhauf, and M. Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. *Proceedings of Robotics: Science and Systems XIV*, 2018.

[97] A. Gąszczak, T. P. Breckon, and J. Han. Real-time People and Vehicle Detection from UAV Imagery. In *Intelligent Robots and Computer Vision XXVIII: Algorithms and Techniques*, volume 7878, page 78780B. International Society for Optics and Photonics, 2011.

[98] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena. X-view: Graph-based semantic multi-view localization. *IEEE Robotics and Automation Letters*, 3(3):1687–1694, 2018.

[99] M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart. Visual place recognition with probabilistic voting. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3192–3199. IEEE, 2017.

[100] M. Gehrig, E. Stumm, T. Hinzmann, and R. Siegwart. Visual place recognition with probabilistic voting. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3192–3199. IEEE, 2017.

[101] M. Ghiasi et al. Fast semantic segmentation of aerial images based on color and texture. In *2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pages 324–327, Sept 2013. doi: 10.1109/IranianMVIP.2013.6780004.

[102] R. Girshick. Fast R-CNN. *CoRR*, pages 1440–1448, 2015.

[103] R. Girshick, J. Donahue, T. Darrell, U. C. Berkeley, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2–9, 2014.

[104] H. Goforth and S. Lucey. Aligning across large gaps in time, 2018.

[105] H. Goforth and S. Lucey. Gps-denied uav localization using pre-existing satellite imagery. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2974–2980, 2019.

[106] P. Gohl, D. Honegger, S. Omari, M. Achtelik, M. Pollefeys, and R. Siegwart. Omnidirectional visual obstacle detection using embedded FPGA. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3938–3943. IEEE.

[107] B. Grelsson, M. Felsberg, and F. Isaksson. Efficient 7d aerial pose estimation. In *Robot Vision (WORV), 2013 IEEE Workshop on*, pages 88–95. IEEE, 2013.

[108] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A Tutorial on Graph-Based SLAM. *ITSM*.

[109] K. Han, C. Aeschliman, J. Park, A. C. Kak, H. Kwon, and D. J. Pack. Uav vision: Feature based accurate ground target localization through propagated initializations and interframe homographies. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 944–950. IEEE, 2012.

[110] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016.

[111] E. M. Hemerly. Automatic Georeferencing of Images Acquired by UAV's. *Int. J. of Automation and Computing*, 11(4):347–352, 2014. ISSN 1751-8520. doi: 10.1007/s11633-014-0799-0. URL http://dx.doi.org/10.1007/s11633-014-0799-0.

[112] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.

[113] C. Herrmann, T. Müller, D. Willersinn, and J. Beyerer. Real-time person detection in low- resolution thermal infrared imagery with MSER and CNNs. In *Electro-Optical and Infrared Systems: Technology and Applications XIII*, volume 9987, page 99870I. Int. Society for Optics and Photonics, 2016. doi: 10.1117/12.2240940.

[114] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-IMU-based localization: Observability analysis and consistency improvement. *IJRR*, 2014.

[115] T. Hinzmann and R. Siegwart. Sparse and deep inverse compositional Lucas-Kanade algorithm on SE(3). *CoRR*, 2020.

[116] T. Hinzmann and R. Siegwart. Deep UAV localization with reference view rendering. *CoRR*, 2020.

[117] T. Hinzmann, T. Schneider, M. Dymczyk, A. Melzer, T. Mantel, R. Siegwart, and I. Gilitschenski. Robust map generation for fixed-wing UAVs with low-cost highly-oblique monocular cameras. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3261–3268. IEEE, 2016.

[118] T. Hinzmann, T. Schneider, M. Dymczyk, A. Melzer, T. Mantel, R. Siegwart, and I. Gilitschenski. Robust map generation for fixed-wing UAVs with low-cost highly-oblique monocular cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3261–3268. IEEE, 2016.

[119] T. Hinzmann, T. Schneider, M. Dymczyk, A. Schaffner, S. Lynen, R. Siegwart, and I. Gilitschenski. Monocular visual-inertial SLAM for fixed-wing UAVs using sliding window based nonlinear optimization. In *International Symposium on Visual Computing*, pages 569–581. Springer, 2016.

[120] T. Hinzmann, T. Stastny, G. Conte, P. Doherty, P. Rudol, M. Wzorek, E. Galceran, R. Siegwart, and I. Gilitschenski. Collaborative 3D reconstruction using heterogeneous UAVs: System and experiments. In *International Symposium on Experimental Robotics*, pages 43–56. Springer, 2016.

[121] T. Hinzmann, J. L. Schönberger, M. Pollefeys, and R. Siegwart. Mapping on the fly: Real-time 3d dense reconstruction, digital surface map and incremental orthomosaic generation for unmanned aerial vehicles. In *Field and Service Robotics, Results of the 11th International Conference, FSR 2017, Zurich, Switzerland, 12-15 September 2017*, pages 383–396, 2017. doi: 10.1007/978-3-319-67361-5_25. URL https://doi.org/10.1007/978-3-319-67361-5_25.

[122] T. Hinzmann, J. L. Schönberger, M. Pollefeys, and R. Siegwart. Mapping on the fly: Real-time 3d dense reconstruction, digital surface map and incremental orthomosaic generation for unmanned aerial vehicles. In *Field and Service Robotics*, pages 383–396. Springer, 2018.

[123] T. Hinzmann, T. Stastny, C. Cadena, R. Siegwart, and I. Gilitschenski. Free LSD: Prior-free visual landing site detection for autonomous planes. *IEEE Robotics and Automation Letters*, 3(3):2545–2552, 2018.

[124] T. Hinzmann, T. Taubner, and R. Siegwart. Flexible stereo: Constrained, non-rigid, wide-baseline stereo vision for fixed-wing aerial platforms. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2550–2557. IEEE, 2018.

[125] T. Hinzmann, C. Cadena, and J. Nieto. Flexible trinocular: Non-rigid multi-camera-IMU dense reconstruction for UAV navigation and mapping. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1137–1142, 2019.

[126] T. Hinzmann, T. Stegemann, C. Cadena, and R. Siegwart. Deep learning-based human detection for UAVs with optical and infrared cameras: System and experiments. *CoRR*, 2020.

[127] H. Hirschmuller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, Feb 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1166.

[128] A. Hornung et al. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Auton. Robots*, 34(3):189–206, 2013.

[129] S. Houben, J. Quenzel, N. Krombach, and S. Behnke. Efficient multi-camera visual-inertial SLAM for micro aerial vehicles. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1616–1622. IEEE, 2016.

[130] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. An Observability-Constrained Sliding-Window Filter for SLAM. In *IROS*, 2011.

[131] M. Huber, T. Hinzmann, R. Siegwart, and L. H. Matthies. Cubic range error model for stereo vision with illuminators. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 842–848. IEEE, 2018.

[132] S. Huh et al. *A Vision-Based Automatic Landing Method for Fixed-Wing UAVs*, pages 217–231. Springer Netherlands, Dordrecht, 2010. ISBN 978-90-481-8764-5. doi: 10.1007/978-90-481-8764-5_11. URL http://dx.doi.org/10.1007/978-90-481-8764-5_11.

[133] A. Irschara, C. Hoppe, H. Bischof, and S. Kluckner. Efficient Structure from Motion with Weak Position and Orientation Priors. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 21–28, June 2011. doi: 10.1109/CVPRW.2011.5981775.

[134] Itseez. Open Source Computer Vision Library. https://github.com/itseez/opencv, 2015.

[135] A. Jäger. Real-Time Monocular Dense 3D Reconstruction For Robot Generation. Master's thesis, ETH Zurich, Autonomous Systems Lab, 2015.

[136] A. K. Jain et al. Unsupervised texture segmentation using Gabor filters. In *Intern. Conf. on Systems, Man, and Cybernetics Conference Proc.*, pages 14–19, Nov 1990. doi: 10.1109/ICSMC.1990.142050.

[137] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3304–3311. IEEE, 2010.

[138] A. E. Johnson et al. Lidar-based hazard avoidance for safe landing on Mars. *J. of guidance, control, and dynamics*, 25(6):1091–1099, 2002.

[139] K. Jüngling and M. Arens. Local Feature Based Person Detection and Tracking Beyond the Visible Spectrum. In *Augmented Vision and Reality*, pages 3–32. Springer, 2011. ISBN 978-3-642-11567-7. doi: 10.1007/978-3-642-11568-4. URL http://www.springerlink.com/index/10.1007/978-3-642-11568-4.

[140] M. Kaess, H. Johannsson, R. R. 0001, V. Ila, J. J. Leonard, and F. Dellaert. isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *ICRA*, 2011.

[141] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *IJRR*, 2012.

[142] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Manipulating attributes of natural scenes via hallucination. *ACM Transactions on Graphics*, 39(1), Nov. 2019.

[143] M. Karrer, M. Agarwal, M. Kamel, R. Siegwart, and M. Chli. Collaborative 6DoF Relative Pose Estimation for two UAVs with Overlapping Fields of View. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[144] L. Kneip and P. Furgale. OpenGV: A Unified and Generalized Approach to Real-Time Calibrated Geometric Vision. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1–8, May 2014. doi: 10.1109/ICRA.2014.6906582.

[145] L. Kneip, R. Siegwart, and M. Pollefeys. *Real-time Scalable Structure from Motion: From Fundamental Geometric Vision to Collaborative Mapping*. ETH, 2012. URL https://books.google.ch/books?id=EglulwEACAAJ.

[146] T. Koch, X. Zhuo, P. Reinartz, and F. Fraundorfer. A new paradigm for matching uav-and aerial images. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3), 2016.

[147] K. Konolige. Small Vision Systems: Hardware and Implementation. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, pages 111–116, 1997.

[148] P. Kozierski, M. Lis, and J. Ziętkiewicz. Resampling in particle filtering-comparison. 2013.

[149] M. Kristan, J. Matas, A. Leonardis, et al. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, Nov 2016. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2516982.

[150] A. Krizhevsky and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[151] J. Kümmerle, T. Hinzmann, A. S. Vempati, and R. Siegwart. Real-time detection and tracking of multiple humans from high bird's-eye views in the visual and infrared spectrum. In *International Symposium on Visual Computing*, pages 545–556. Springer, 2016.

[152] J. Kümmerle, T. Hinzmann, A. S. Vempati, and R. Siegwart. Real-Time Detection and Tracking of Multiple Humans from High Bird's-Eye Views in the Visual and Infrared Spectrum. In *International Symposium on Visual Computing*, pages 545–556, 2016.

[153] R. Laganière. Composing a bird's eye view mosaic. *Vision Interface*, pages pp. 382–386, 2000.

[154] M. Laiacker et al. Vision aided automatic landing system for fixed wing UAV. In *IROS*, pages 2971–2976. IEEE, 2013.

[155] P. Lanier. Stereovision Correction Using Modal Analysis. Master's thesis, Virginia Polytechnic Institute and State University, 2010.

[156] P. Lanier, N. Short, K. Kochersberger, and L. Abbott. Modal-based camera correction for large pitch stereo imaging. In *Structural Dynamics, Volume 3*, pages 1225–1238. Springer, 2011.

[157] S. Leutenegger. *Unmanned Solar Airplanes*. PhD thesis, Diss., ETH Zürich, Nr. 22113, 2014.

[158] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[159] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization. *IJRR*, 2014.

[160] S. Leutenegger, A. Melzer, K. Alexis, and R. Siegwart. Robust state estimation for small unmanned airplanes. In *CCA*, 2014.

[161] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *IJRR*, 2015.

[162] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[163] S. Leutenegger et al. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. In *Proc. of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[164] M. Li and A. I. Mourikis. Optimization-based estimator design for vision-aided inertial navigation. *RSS*, 2013.

[165] M. Li, B. H. Kim, and A. I. Mourikis. Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera. In *ICRA*, 2013.

[166] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. *International Journal of Computer Vision*, 128(3): 657–678, Nov 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01250-9. URL http://dx.doi.org/10.1007/s11263-019-01250-9.

[167] D. R. Lide. *CRC handbook of chemistry and physics*, volume 85. CRC press, 2004.

[168] H. Lin, C. Lu, H. Tsai, and T. Kung. The analysis of emi noise coupling mechanism for gps reception performance degradation from ssd/usb module. In *2014 International Symposium on Electromagnetic Compatibility, Tokyo*, pages 350–353, May 2014.

[169] T.-Y. Lin, M. Maire, S. Belongie, et al. Microsoft COCO: Common Objects in Context. In *European conference on computer vision*, pages 740–755, 2014.

[170] T.-y. Lin, P. Doll, R. Girshick, K. He, B. Hariharan, S. Belongie, F. Ai, and C. Tech. Feature Pyramid Networks for Object Detection. *Cvpr*, 1(2):3, 2017. ISSN 0006-291X. doi: 10.1109/CVPR.2017.106.

[171] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2999–3007, 2017. ISSN 15505499. doi: 10.1109/ICCV.2017.324.

[172] F. Lindsten, J. Callmer, H. Ohlsson, D. Törnqvist, T. B. Schön, and F. Gustafsson. Geo-referencing for uav navigation using environmental classification. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1420–1425. IEEE, 2010.

[173] Q. Liu and Z. He. PTB-TIR: A Thermal Infrared Pedestrian Tracking Benchmark. *arXiv preprint arXiv:1801.05944*, pages 1–10, 2018. URL http://arxiv.org/abs/1801.05944.

[174] T. Liu, H. Y. Fu, Q. Wen, D. K. Zhang, and L. F. Li. Extended Faster R-CNN For Long Distance Human Detection: Finding Pedestrians in UAV Images. *2018 IEEE International Conference on Consumer Electronics, ICCE 2018*, pages 1–2, 2018. doi: 10.1109/ICCE.2018.8326306.

[175] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD : Single Shot MultiBox Detector. In *European Conf. on computer vision*, volume 1, pages 21–37. Springer, 2016. ISBN 9783319464480. doi: 10.1007/978-3-319-46448-0.

[176] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[177] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th Int. Joint Conf. on Artificial Intelligence*, pages 674–679, 1981. URL http://dl.acm.org/citation.cfm?id=1623264.1623280.

[178] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[179] A. Lukezic, T. Vojir, L. ˇCehovin Zajc, J. Matas, and M. Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6309–6318, 2017.

[180] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger. Taking a deeper look at the inverse compositional algorithm. *CoRR*, abs/1812.06861, 2018. URL http://arxiv.org/abs/1812.06861.

[181] Z. Lv, F. Dellaert, J. Rehg, and A. Geiger. Taking a deeper look at the inverse compositional algorithm. In *CVPR*, 2019.

[182] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963. doi: 10.1137/0111030. URL http://dx.doi.org/10.1137/0111030.

[183] A. Martinelli. Visual-inertial structure from motion: observability vs minimum number of sensors. *ICRA*, 2014.

[184] R. Mascaro, L. Teixeira, T. Hinzmann, R. Siegwart, and M. Chli. GOMSF: Graph-optimization based multi-sensor fusion for robust UAV pose estimation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1421–1428. IEEE, 2018.

[185] S. Medina, Z. Dai, and Y. Gao. Where is this? video geolocation based on neural network features. *arXiv preprint arXiv:1810.09068*, 2018.

[186] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *ICRA*, 2007.

[187] A. I. Mourikis and S. I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *CVPRW*, 2008.

[188] M. Mueller, N. Smith, and B. Ghanem. A Benchmark and Simulator for UAV Tracking. In *Proc. of the European Conf. on Computer Vision (ECCV)*, pages 445–461. Springer, 2016.

[189] M. Müller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, and W. Stürzl. Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3701–3708. IEEE, 2018.

[190] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. doi: 10.1109/TRO.2017.2705103.

[191] E. Nerurkar, K. Wu, and S. Roumeliotis. C-KLAM: Constrained keyframe-based localization and mapping. In *ICRA*, 2014.

[192] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 2320–2327, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-1101-5. doi: 10.1109/ICCV.2011.6126513. URL http://dx.doi.org/10.1109/ICCV.2011.6126513.

[193] D. T. Nguyen, W. Li, and P. O. Ogunbona. Human detection from images and videos: A survey. *Pattern Recognition*, 51:148–175, 2016. ISSN 00313203. doi: 10.1016/j.patcog.2015.08.027. URL http://dx.doi.org/10.1016/j.patcog.2015.08.027.

[194] M. Nielsen. True orthophoto generation. Master's thesis, Technical Univ. of Denmark, 2004.

[195] 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 *ICRA*, 2014.

[196] P. Oettershagen, T. J. Stastny, T. Mantel, A. Melzer, K. Rudin, G. Agamennoni, K. Alexis, and R. Siegwart. Long-Endurance Sensing and Mapping using a Hand-Launchable Solar-Powered UAV. *FSR*, 2015.

[197] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, K. Alexis, and R. Siegwart. Perpetual flight with a small solar-powered UAV: Flight results, performance analysis and model validation. In *2016 IEEE Aerospace Conference*, pages 1–8. IEEE, 2016.

[198] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, S. Leutenegger, K. Alexis, and R. Siegwart. Design of small hand-launched solar-powered UAVs: From concept study to a multi-day world endurance record flight. *Journal of Field Robotics*, 34(7):1352–1377, 2017.

[199] P. Oettershagen, T. Stastny, T. Hinzmann, K. Rudin, T. Mantel, A. Melzer, B. Wawrzacz, G. Hitz, and R. Siegwart. Robotic technologies for solar-powered uavs: Fully autonomous updraft-aware aerial sensing for multiday search-and-rescue missions. *Journal of Field Robotics*, 35(4):612–640, 2018.

[200] P. Oettershagen et al. Robotic technologies for solar-powered UAVs: Fully autonomous updraft-aware aerial sensing for multiday search-and-rescue missions. *Journal of Field Robotics*, 2017.

[201] B. O. Olawale et al. A Four-Step Ortho-Rectification Procedure for Geo-Referencing Video Streams from a Low-Cost UAV. 9(8):1445–1452, 2015.

[202] C. F. Olson, A. I. Ansar, and C. W. Padgett. Robust registration of aerial image sequences. In *International Symposium on Visual Computing*, pages 325–334. Springer, 2009.

[203] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[204] B. Patel, T. Barfoot, and A. Schoellig. Visual localization with google earth images for robust global pose estimation of uavs. *IEEE Robotics and Automation Letters*, 2020.

[205] T. Patterson, S. McClean, P. Morrow, and G. Parr. Utilizing geographic information system data for unmanned aerial vehicle position estimation. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 8–15. IEEE, 2011.

[206] A. G. Perera, A. Al-Naji, Y. W. Law, and J. Chahl. Human Detection and Motion Analysis from a Quadrotor UAV. In *IOP Conf. Series: Materials Science and Engineering*, volume 405, page 12003. IOP Publishing, 2018. doi: 10.1088/1757-899X/405/1/012003.

[207] M. Pietikäinen et al. Color texture classification with color histograms and local binary patterns. In *In: IWTAS*, pages 109–112, 2002.

[208] M. Pollefeys, R. Koch, and L. V. Gool. A simple and efficient rectification method for general motion. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 496–501 vol.1, 1999. doi: 10.1109/ICCV.1999.791262.

[209] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP Variants on Real-world Data Sets. *Auton. Robots*, 34(3):133–148, Apr. 2013. ISSN 0929-5593. doi: 10.1007/s10514-013-9327-2. URL http://dx.doi.org/10.1007/s10514-013-9327-2.

[210] J. Portmann, S. Lynen, M. Chli, and R. Siegwart. People Detection and Tracking from Aerial Thermal Views. In *International Conf. on Robotics and Automation (ICRA), 2014 IEEE*, pages 1794–1800. IEEE, 2014. ISBN 9781479936854.

[211] J. Portmann, S. Lynen, M. Chli, and R. Siegwart. People Detection and Tracking from Aerial Thermal Views. In *International Conf. on Robotics and Automation (ICRA), 2014 IEEE*, pages 1794–1800. IEEE, 2014. ISBN 9781479936854.

[212] S. Prakash, P. Y. Lee, and T. Caelli. 3d mapping of surface temperature using thermal stereo. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–4, Dec 2006. doi: 10.1109/ICARCV.2006.345342.

[213] T. Qin, P. Li, and S. Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

[214] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[215] J. Rangel, S. Soldan, and A. Kroll. 3d thermal imaging: Fusion of thermography and depth cameras. In *International Conference on Quantitative InfraRed Thermography*, 2014.

[216] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. *Proceedings - 30th IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:6517–6525, 2017. ISSN 0146-4833. doi: 10.1109/CVPR.2017.690.

[217] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*, 2018. ISSN 0146-4833. doi: 10.1109/CVPR.2017.690. URL http://arxiv.org/abs/1804.02767.

[218] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conf. on computer vision and pattern recognition*, pages 779–788, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.91. URL http://arxiv.org/abs/1506.02640.

[219] J. Rehder, K. Gupta, S. Nuske, and S. Singh. Global pose estimation with limited gps and long range visual odometry. In *2012 IEEE international conference on robotics and automation*, pages 627–633. IEEE, 2012.

[220] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart. Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016.

[221] V. Reilly, B. Solmaz, and M. Shah. Geometric constraints for human detection in aerial imagery. In *European Conf. on Computer Vision*, pages 252–265. Springer, 2010. ISBN 3-642-15566-9, 978-3-642-15566-6.

[222] F. Remondino, L. Barazzetti, F. Nex, M. Scaioni, and D. Sarazzi. UAV photogrammetry for mapping and 3D modeling – Current status and future perspectives. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1):C22, 2011.

[223] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[224] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. *Lecture Notes in Computer Science*, 9912 LNCS:549–565, 2016. ISSN 16113349. doi: 10.1007/978-3-319-46484-8_33.

[225] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[226] C. Rother, V. Kolmogorov, and A. Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[227] P. Rudol and P. Doherty. Human Body Detection and Geolocalization for UAV Search and Rescue Missions Using Color and Thermal Imagery. In *Aerospace Conf., 2008 IEEE*, pages 1–8. IEEE, 2008. ISBN 1424414881. doi: 10.1109/AERO.2008.4526559.

[228] O. Russakovsky, J. Deng, H. Su, et al. ImageNet Large Scale Visual Recognition Challenge. In *Int. J. of Computer Vision*, volume 115, pages 211–252. Springer US, 2015. ISBN 0920-5691. doi: 10.1007/s11263-015-0816-y. URL http://dx.doi.org/10.1007/s11263-015-0816-y.

[229] P. Saponaro, S. Sorensen, S. Rhein, and C. Kambhamettu. Improving calibration of thermal stereo cameras using heated calibration board. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 4718–4722, Sep. 2015. doi: 10.1109/ICIP.2015.7351702.

[230] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12716–12725, 2019.

[231] A. Sathyan, J. Cohen, and M. Kumar. Deep Convolutional Neural Network For Human Detection And Tracking In FLIR Videos. In *AIAA Infotech @ Aerospace*, page 1412. American Institute of Aeronautics and Astronautics, 2016. ISBN 978-1-62410-388-9. doi: 10.2514/6.2016-1412. URL http://arc.aiaa.org/doi/10.2514/6.2016-1412.

[232] T. Sattler, B. Leibe, and L. Kobbelt. Improving Image-Based Localization by Active Correspondence Search. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 2012.

[233] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart. maplab: An open framework for research in visual-inertial mapping and localization. *IEEE Robotics and Automation Letters*, 2018. doi: 10.1109/LRA.2018.2800113.

[234] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[235] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys. A vote-and-verify strategy for fast spatial verification in image retrieval. In *Asian Conference on Computer Vision*, pages 321–337. Springer, 2016.

[236] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler. Semantic visual localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6896–6906, 2018.

[237] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. doi: 10.15607/RSS.2009.V.021.

[238] M. Shan, F. Wang, F. Lin, Z. Gao, Y. Z. Tang, and B. M. Chen. Google map aided visual navigation for uavs in gps-denied environment. *CoRR*, abs/1703.10125, 2017. URL http://arxiv.org/abs/1703.10125.

[239] A. Shetty and G. X. Gao. Uav pose estimation using cross-view geolocalization with satellite imagery. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1827–1833. IEEE, 2019.

[240] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR94)*, 1994.

[241] N. J. Short. 3-D Point Cloud Generation from Rigid and Flexible Stereo Vision Systems. Master's thesis, Virginia Tech, 2009.

[242] G. Sibley, L. Matthies, and G. S. Sukhatme. Sliding window filter with application to planetary landing. *JFR*, 2010.

[243] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim. Integrated position estimation using aerial image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 24(1):1–18, 2002.

[244] D. Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[245] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, pages 1–14, 2014. ISSN 09505849. doi: 10.1016/j.infsof.2008.09.005. URL http://arxiv.org/abs/1409.1556.

[246] F. Sittel, J. Müller, and W. Burgard. Computing velocities and accelerations from a pose time sequence in three-dimensional space. Technical Report 272, University of Freiburg, Department of Computer Science, 2013. URL http://www.muellerj.de/publications/papers/sittel13tr.pdf.

[247] K. Skala, T. Lipić, I. Sović, L. Gjenero, and I. Grubišić. 4d thermal imaging system for medical applications. *Periodicum biologorum*, 113(4):407–416, 2011.

[248] J. Skaloud. *Optimizing Georeferencing of Airborne Survey Systems by INS/DGPS*. PhD thesis, University of Calgary, Alberta, 1999.

[249] J. R. Smith. *Color for Image Retrieval*, pages 285–311. John Wiley & Sons, Inc., 2002. ISBN 9780471224631. doi: 10.1002/0471224634.ch11. URL http://dx.doi.org/10.1002/0471224634.ch11.

[250] N. Smolyanskiy, A. Kamenev, and S. Birchfield. On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach, 2018.

[251] K.-H. Son, Y. Hwang, and I.-S. Kweon. Uav global pose estimation by matching forward-looking aerial images with satellite images. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3880–3887. IEEE, 2009.

[252] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. URL https://www.ros.org.

[253] D. Steedly, C. Pal, and R. Szeliski. Efficiently registering video into panoramic mosaics. In *ICCV*, pages 1300–1307. IEEE Computer Society, 2005.

[254] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar. Robust stereo visual inertial odometry for fast autonomous flight. *IEEE Robotics and Automation Letters*, 3(2):965–972, 2018.

[255] J. Surber, L. Teixeira, and M. Chli. Robust visual-inertial localization with weak gps priors for repetitive uav flights. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 6300–6306. IEEE, 2017.

[256] C. Tang and P. Tan. Ba-net: Dense bundle adjustment network, 2018.

[257] L. Teixeira, F. Maffra, M. Moos, and M. Chli. VI-RPE: visual-inertial relative pose estimation for aerial vehicles. *IEEE Robotics and Automation Letters*, 3(4):2770–2777, 2018. doi: 10.1109/LRA.2018.2837687. URL https://doi.org/10.1109/LRA.2018.2837687.

[258] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli. Aerial single-view depth completion with image-guided uncertainty estimation. *IEEE Robotics and Automation Letters*, 5(2):1055–1062, 2020.

[259] C. Theodore et al. Flight trials of a rotorcraft unmanned aerial vehicle landing autonomously at unprepared sites. In *Annual Forum Proceedings-American Helicopter Society*, 2006.

[260] S. Thurrowgood et al. A biologically inspired, vision-based guidance system for automatic landing of a fixed-wing aircraft. *J. of Field Robotics*, 31 (4):699–727, 2014. ISSN 1556-4967. doi: 10.1002/rob.21527. URL http://dx.doi.org/10.1002/rob.21527.

[261] G. Toussaint. Solving geometric problems with the rotating calipers, 1983.

[262] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. URL http://lmb.informatik.uni-freiburg.de//Publications/2017/UZUMIDB17.

[263] A. S. Vempati, G. Agamennoni, T. Stastny, and R. Siegwart. Victim Detection from a Fixed-Wing UAV: Experimental Results. In *Advances in Visual Computing*, pages 432–443. Springer, 2015.

[264] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan, and T. Wark. A mask-based approach for the geometric calibration of thermal-infrared cameras. *IEEE Transactions on Instrumentation and Measurement*, 61(6): 1625–1635, June 2012. ISSN 1557-9662. doi: 10.1109/TIM.2012.2182851.

[265] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, volume 1. IEEE, 2001. ISBN 0769512720.

[266] K. Wang and S. Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020.

[267] X. Wang, P. Cheng, X. Liu, and B. Uzochukwu. Fast and Accurate, Convolutional Neural Network Based Approach for Object Detection from UAV. *arXiv preprint arXiv:1808.05756*, 2018. URL http://arxiv.org/abs/1808.05756.

[268] M. Warren, D. McKinnon, and B. Upcroft. Online calibration of stereo rigs for long-term autonomy. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3692–3698. IEEE, 2013.

[269] M. Warren, P. Corke, and B. Upcroft. Long-range stereo visual odometry for extended altitude flight of unmanned aerial vehicles. *The Intern. J. of Robotics Research*, 35(4):381–403, 2016.

[270] M. Warren et al. *Enabling Aircraft Emergency Landings Using Active Visual Site Detection*, pages 167–181. Springer Intern. Publishing, 2015. ISBN 978-3-319-07488-7. doi: 10.1007/978-3-319-07488-7_12. URL http://dx.doi.org/10.1007/978-3-319-07488-7_12.

[271] K. Weiler. An incremental angle point in polygon test. In P. S. Heckbert, editor, *Graphics Gems IV*, pages 16–23. Morgan Kaufmann, 1994.

[272] S. Weiss and R. Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[273] M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[274] Z. Wu, N. Fuller, D. Theriault, and M. Betke. A Thermal Infrared Video Benchmark for Visual Analysis. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, pages 201–208, 2014. ISSN 21607516. doi: 10.1109/CVPRW.2014.39.

[275] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A Probabilistic, Flexible, and Compact 3D map Representation for Robotic Systems. In *In Proc. of the ICRA 2010 workshop*, 2010.

[276] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[277] S. Yahyanejad. *Orthorectified Mosacking of Images from Small-scale Unmanned Aerial Vehicles*. PhD thesis, Alpen-Adria Universität Klagenfurt, 2013.

[278] S. Yahyanejad and B. Rinner. A fast and mobile system for registration of low-altitude visual and thermal aerial images using multiple small-scale UAVs. *ISPRS J. of Photogrammetry and Remote Sensing*, 104:189–202, 2015. ISSN 09242716. doi: 10.1016/j.isprsjprs.2014.07.015. URL http://dx.doi.org/10.1016/j.isprsjprs.2014.07.015.

[279] P. Yang, X. Chen, and J. Wang. Decoupling of airborne dynamic bending deformation angle and its application in the high-accuracy transfer alignment process. volume 19, page 214. Multidisciplinary Digital Publishing Institute, 2019.

[280] A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand. Vision-based absolute localization for unmanned aerial vehicles. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3429–3434. IEEE, 2014.

[281] X. Yu, S. Chaturvedi, C. Feng, Y. Taguchi, T.-Y. Lee, C. Fernandes, and S. Ramalingam. Vlase: Vehicle localization by aggregating semantic edges. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3196–3203. IEEE, 2018.

[282] Z. Yu, S. Lincheng, Z. Dianle, Z. Daibing, and Y. Chengping. Camera calibration of thermal-infrared stereo vision system. In *2013 Fourth International Conference on Intelligent Systems Design and Engineering Applications*, pages 197–201. IEEE, 2013.

[283] K. Yun, L. Nguyen, T. Nguyen, et al. Small target detection for search and rescue operations using distributed deep learning and synthetic data generation. *CoRR*, abs/1904.11619, 2019. URL http://arxiv.org/abs/1904.11619.

[284] Z. Zhang. Iterative Point Matching for Registration of Free-form Curves and Surfaces. *Int. J. Comput. Vision*, 13(2):119–152, Oct. 1994. ISSN 0920-5691. doi: 10.1007/BF01427149. URL http://dx.doi.org/10.1007/BF01427149.

Curriculum Vitae

Timo Hinzmann
born September 25, 1988
citizen of Germany

Education

2015–2020	*Doctoral studies (Dr.Sc.)* Autonomous Systems Lab (ASL), ETH Zurich, Switzerland Supervised by Prof. Roland Siegwart
2012–2014	*Master of Science (M.Sc.)* Robotics, Systems and Control (RSC) ETH Zurich, Switzerland
2008–2011	*Bachelor of Science (B.Sc.)* Electrical Engineering and Information Technology Karlsruhe Institute of Technology, Germany
1999–2008	*High School* Kepler Gymnasium Pforzheim, Germany.

Previous Dissertations

2014	*Master Thesis* Jet Propulsion Laboratory (JPL), USA Title: "Robust Vision-based Navigation for Micro Air Vehicles" Supervised by Stephan Weiss and Roland Brockers.
2014	*Semester Project* Autonomous Systems Lab (ASL), ETH Zurich, Switzerland Title: "Adaptive control of multirotor aerial vehicles" Supervised by Michael Burri, Sammy Omari and Markus Achtelik.
2011	*Bachelor Thesis* Institute of Systems Optimization (ITE), Karlsruhe Institute of Technology (KIT), Germany Title: "Path planning of a differential drive ground robot" Supervised by Justus Seibold.

Work Experience (Selection)

2012	*Internship* BMW Group Research & Development, Munich, Germany Design, integration and evaluation of advanced driver assistance systems
2010-2011	*Research assistant* Institute of Optimization (ITE), Karlsruhe Institute of Technology (KIT) Design of PCBs for indoor pedestrian navigation
2009-2011	*Research assistant* Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB) Karlsruhe Model predictive control for fuel consumption reduction of heavy trucks

Research Projects

Project involvement during doctoral studies:

- *ICARUS*
  Integrated Components for Assisted Rescue and Unmanned Search Operations
- *SHERPA*
  Smart collaboration between Humans and ground-aerial Robots for improving rescuing activities in Alpine environments
- *AtlantikSolar*
  A UAV for the first-ever autonomous solar-powered crossing of the Atlantic Ocean
- *SolAIR*
  Solar-powered Automated Aerial Imaging and Reconnaissance Using Infrared Cameras
- *armasuisse Science & Technology*
  Vision-based landing, obstacle avoidance, SLAM, and low-altitude flights
- Swiss air rescue organization *Rega*
  Human detection and tracking using an optical-infrared camera rig