

University of Nevada, Reno

**Multi-Modal Landmark Detection and Tracking for Odometry
Estimation in Degraded Visual Environments**

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in
Computer Science & Engineering

by

Shehryar Masaud Khan Khattak

Dr. Kostas Alexis - Thesis Advisor
December 2017

© by Shehryar Masaud Khan Khattak 2017
All Rights Reserved.



THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

SHEHRYAR MASAUD KHAN KHATTAK

Entitled

**Multi-Modal Landmark Detection And Tracking For Odometry Estimation In
Degraded Visual Environments**

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

Dr. Kostas Alexis, Advisor

Dr. Monica Nicolescu, Committee Member

Hao Xu, Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

December, 2017

Abstract

This thesis focuses on the development of a multi-modal odometry estimation framework for the purpose of robot localization and mapping in visually degraded environments. Visual and depth information are encoded at the feature detection and descriptor extraction level, providing robustness to the landmark selection process in low illumination and texture-less conditions. An extended Kalman filter framework is used to predict landmark positions using inertial measurements in successive frames and the matching pixel error is used as an innovation term. For localization performance evaluation the proposed approach is compared to ground-truth provided by Vicon system and a state-of-the-art visual-inertial odometry estimation framework. The mapping performance is demonstrated by mapping a large room in dark conditions.

Dedication

Dedicated to my parents, sisters and teachers.

Acknowledgments

This material is based upon work supported by the Department of Energy under Award Number [DE-EM0004478].

I would like to thank my advisor, Dr. Kostas Alexis without whose research guidance and support this work would not have been possible. In title he maybe only my advisor, but the multiple roles played by him as a mentor, a colleague, a lab mate and a friend have given me a holistic understanding and appreciation for the work I do.

I would like to express my deepest gratitude to Dr. Christos Papachristos, who has been a mentor and a friend since the beginning of our lab.

I would like to thank my lab mates and close friends, Tung Dang and Frank Mascarich, who I have had the pleasure of having numerous discussions on research topics and life in general. They have always provided me valuable input and I have learned a lot from them.

Finally, I would like to appreciate my family and loved ones without whose support and understanding I could not have done this work.

Table of Contents

1	Thesis Introduction, Contribution, and Content	1
1.1	Introduction	1
1.2	Contribution	5
1.3	Content	5
2	Related Work and Motivation	7
2.1	Related Work	7
2.2	Motivation	10
3	Methodology	11
3.1	Overview	11
3.2	Feature Generation	12
3.2.1	Image Pre-Processing	13
3.2.2	Image Registration	15
3.2.3	Common Image Generation	16
3.2.4	Feature Selection	21
3.3	Descriptor Extraction	23
3.3.1	Descriptor Selection	23
3.3.2	Descriptor Details	24
3.4	Filter Framework	29

3.4.1	Overview	29
3.4.2	State Propagation	31
3.4.3	State Update	33
4	Experimental Results	34
4.1	Sensor Setup and Calibration	34
4.1.1	IMU characterization	35
4.1.2	Camera-IMU Extrinsic Calibration	37
4.2	Handheld Localization Test	38
4.3	On-board Robot Localization Test	41
4.4	Mapping Test	43
5	Conclusions and Future Work	48
5.1	Conclusions	48
5.2	Future Work	49

List of Tables

4.1	Noise and Bias parameters for the accelerometers	37
4.2	Noise and Bias parameters for the gyroscopes.	37
4.3	RMSE Localization errors in hand-held tests	41
4.4	RMSE Localization errors in aerial tests	43

List of Figures

1.1	Example of good texture information in good illumination conditions	3
1.2	Example of good structural information in low illumination conditions	4
3.1	An overview of the proposed multi-modal odometry framework. . . .	12
3.2	Depth image before filtering	14
3.3	Depth image after filtering	15
3.4	Registered RGB point-cloud	17
3.5	ORB Keypoints on Visual Image	18
3.6	Normailzed ORB Score Image	18
3.7	Normailzed Depth Score Image	20
3.8	Normailzed Common Score Image	21
3.9	Example of Selected Features	22
3.10	Overview of Descriptor Generation	24
3.11	Descriptor Sampling Pattern	25
4.1	Allan Standard Deviation Plot for accelerometer	36
4.2	Allan Standard Deviation Plot for gyroscope	36
4.3	Localization Plot(Top-Down View)	39
4.4	Localization Plot for individual axes	40
4.5	Localization Absolute Error Plot for individual axes	41

4.6	Robot Platform	42
4.7	Robot Trajectory Plots	43
4.8	RGB map created in visually-degraded conditions	44
4.9	Height colored map created in visually-degraded conditions	45
4.10	Front view of top right portion of the map	46
4.11	Front view of bottom right portion of the map	47

Chapter 1

Thesis Introduction, Contribution, and Content

1.1 Introduction

During the past decade robots have seen an unprecedented expansion in their utility as they take on more tasks typically reserved to be performed by humans. With an increasing area of robotic applications including many mission critical tasks such as infrastructure inspection [1–9], disaster response [10–12] and security monitoring [13, 14], the need for robots to operate in a variety of environments reliably becomes crucial. To accomplish their tasks robots rely on a variety of sensors to localize and navigate in different environmental conditions with GPS and vision sensors being the more typical and traditionally used sensors of choice for outdoors and indoors environments respectively. In indoor environments, where GPS is not available or other GPS-denied environments, a variety of methods are available that utilize vision sensors, alone or in conjunction with inertial sensors, to estimate robot

pose and odometry for localization and tasks. Although vision based methods are very popular with both feature based and dense methods showing reliable results, they are nonetheless prone to failure in poor illumination conditions and low texture environments.

As an alternative, and due to recent miniaturization and cost competitiveness, range/depth sensors are another popular choice to perform indoor robot navigation tasks. Compared to visual data, depth measurements from a depth sensor are not affected by scene illumination and texture changes. Methods relying on depth data although are able to reliably perform in visually degraded environments, i.e. low-illumination and texture-less conditions, but tend to suffer in structureless and symmetric environments, e.g. long corridors, as they rely on the structure and geometry of their environment to estimate robot pose. It can be noted that although both of these sensing modalities suffer individually in respective degraded conditions, they can be utilized in complementary manners. Figure: 1.1 shows an example where complementary nature of visual and depth sensing modalities can be very useful. The wall poster in the scene (highlighted by the red box) is an example of a texture rich object in visual domain, however the same area provides no geometric or structural information in the depth domain as the depth difference between the wall and the poster is smaller than the sensing resolution of the sensor.



Figure 1.1: Visual and Depth images of a scene taken in good illumination conditions are shown side-by-side. The poster on the wall provides rich texture information in visual domain but does not provide any structural information in depth domain.

Similarly, Figure: 1.2 provides an example of the usefulness of depth sensing. It can be easily noted that in low illumination conditions the same scene cannot provide good visual information, e.g. the wall poster doesn't provide good texture information in these conditions, however the depth information remains unaffected and can still provide useful geometric or structural information as shown by the chair in the scene (highlighted by the red box).

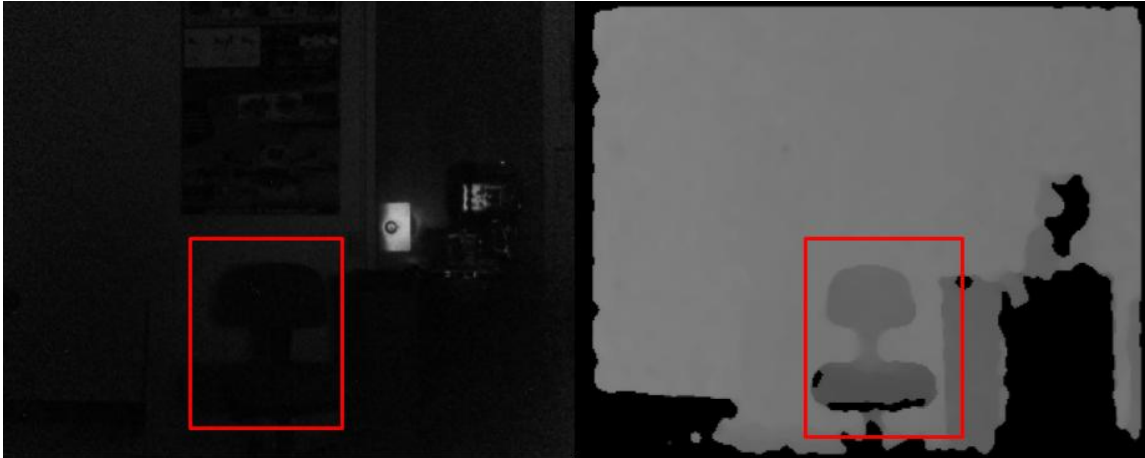


Figure 1.2: Visual and Depth images of a scene taken in low illumination conditions are shown side-by-side. Due to low illumination conditions reliable texture information is very low, however depth image remains unaffected and provides good structural and geometric information.

Noting the complementary nature of information in both sensing modalities, a case can be made that using them together can improve odometry estimation in terms of accuracy and robustness. However, most methods that utilize both sensing modalities either estimate the robot pose using each sensing modality separately and then combining the results or choose a primary sensing modality for estimation and utilize the second sensing modality for improving on the estimates. Both approaches have shown consistent results for certain types of environments. However, approaches that work by combining only the final estimates can be prone to scaling issues if the underlying estimation processes for each sensing modality are different. Similarly, if a primary sensing modality is chosen the overall estimation process is prone to be less robust when the primary sensing modality suffers significantly. However, such problems can be alleviated if sensing fusion occurs during the earlier stages of the estimation process, i.e.e the feature detection and descriptor extraction level, as it makes the final odometry estimation process less prone to ill-conditioning of one or the other sensing modality.

In this work we present a multi-modal odometry estimation framework that uses fused visual and depth information at the feature level to provide a robust odometry estimate. By using features salient in both sensing modalities, we improve the robustness of features when information in both sensing domains is present as well as augment detection and description of features in degraded environments. In our approach odometry estimates are generated by tracking these features in an extended Kalman filter framework where new feature locations are propagated scene to scene using inertial information and error in feature location is used as an innovation term. The advantage of utilizing multi-modal features, in addition to the ability to navigate in degraded environments, is that it keeps the filter state small allowing for odometry calculation to be done in real time on-board a robot.

1.2 Contribution

In this work we present an odometry estimation framework for localization and mapping applications in visually degraded conditions. Our framework fuses multi-modal features, generated using visual and depth data, with inertial data in an extended Kalman filter framework. To the best of our knowledge our approach is unique in taking advantage of fusion of visual and depth data at feature detector and descriptor level for odometry estimation for the purpose of navigating in visually degraded environments.

1.3 Content

This thesis is organized as follows: Chapter 2 presents a literature review of the related techniques with a discussion of their strengths and areas of potential im-

provements followed by the derived motivation for this work. Chapter 3 describes in the theoretical background and the implementation details about the three main components of the proposed framework namely a) Feature Generation (section: 3.2), b) Descriptor Extraction (section: 3.3) and c) Filter Framework (section: 3.4). In Chapter 4 we introduce the utilized sensor along with the detailed notes about the methods and toolboxes used for the calibration and characterization of this sensor (section: 4.1). Utilizing this sensor, preliminary results for localization and mapping are shown. Finally, Chapter 5 discusses ideas for improvement as part of future work and concludes this thesis.

Chapter 2

Related Work and Motivation

2.1 Related Work

Visual odometry for robot navigation tasks is a well established field and has been utilized in a number of different purposeful applications [15, 16]. Broadly, visual odometry methods can be categorized in two categories namely: feature based methods and direct methods. Feature based methods depend on the tracking of a sparse set of detected salient points between successive frames. These methods rely on the repeatability of detection of same feature points and the robustness of matching extracted descriptors across multiple frames to extract camera pose and structure of the scene. These techniques have shown very reliable results on very large motions between successive scene, as shown in [17, 18]. However, these methods rely on the threshold set for feature detection and descriptor matching. Furthermore, such techniques also necessitate the requirement of some sort of outlier rejection method such as [19] to deal with wrong correspondences. Thus in case of feature based method descriptor extraction, matching and outlier detection can become computationally

burdensome. Direct methods on the other hand operate by directly comparing local intensity gradients in an image and can be significantly faster than feature based methods as they save time by avoiding features detection, description and matching operations. Direct methods such as [20] have shown to perform at very high frame rates. These approaches track image patches instead of points and are dependent on correct frame to frame warping estimation to perform reliably and need to track a large number of patches for robustness.

A popular approach to reduce the number of features to be tracked and to reduce the search space for feature matching from frame to frame is to combine Inertial Measurement Unit (IMU) data with visual data. These inertial measurements can be integrated in visual odometry frameworks by utilizing an extended Kalman filter framework, as shown in [21], and can be utilized with both feature based methods and direct methods such as those mentioned above. In [22] authors employ an IMU driven filtering framework where they use the re-projection error of 3D landmarks for filter updates. They use feature matching to perform filter updates and have shown consistent results over long trajectories. Similarly, in [23], authors propose a robot-centric extended Kalman filter framework which keeps track of a small number of intensity patches position along with the pose of the robot. Position of patches are propagated by using the IMU measurements during the prediction step of the filter and a filter update is performed by using intensity patch alignment error as an innovation term. By using IMU measurements the authors are able to reduce the search space for patch alignment between two frames and thus reduce the number of patches to be tracked making the whole process computationally tractable, as well as showing particularly robust performance during very fast motions.

Although visual odometry techniques have seen a lot of growth in variety and robustness in the past years, yet the fact remains that all these techniques rely on proper scene illumination and availability of texture for their operation. In recent years due to increased miniaturization and affordability, depth sensors have also become popular for robot navigation applications as the provided direct depth measurements are not prone to illumination changes or lack of texture. These sensors can be broadly categorized in two categories based on their sensing range and density of data produced. Light Detection and Ranging (LIDAR) units can produce depth measurements at long ranges and return data in the form of sparse point clouds. These sensors are especially popular in autonomous cars research [24] given their ability to provide depth measurements over long distances. Techniques using LIDAR data along with IMU integration to generate odometry estimates have shown very robust results over long ranges [25] but tend to suffer at short ranges and in structure-less environments where geometric constraints are not enough to constrain the underlying optimization process [26, 27]. On the other hand, dense depth sensors produce dense depth data at short ranges and can be easily combined with RGB images on pixel-to-pixel basis given extrinsic calibration. These RGB-D cameras have become very popular since the first release of Microsoft Kinect in 2010 and have enabled the development of RGB-D odometry estimation techniques such as [28–31].

Although these approaches take advantage of the availability of direct depth estimates, on closer inspection it can be noted that handling of depth and visual data is done separately. In [28–30] feature detection and matching is done solely on the visual image for odometry estimation and depth data is utilized for correct scale estimation and mapping purposes. Similarly, in [31] depth data is used only to validate the visual odometry estimation. Due to this separate handling of data in both sensing modalities, the overall odometry estimation remains prone to illumination changes and lack

of texture. To remedy these problems some recent approaches [32–34] propose to encode visual and depth information on feature detection and descriptor level. These approaches although do improve the robustness in large illumination changes they are sensitive to the quality of depth data and can become computationally burdensome for real-time operations [35, 36].

2.2 Motivation

Motivated by the discussion above in this thesis we present an extended Kalman filtering framework that fuses inertial, visual and depth information for odometry estimation. Inspired by [23] we use a robot-centric formulation and use inertial measurements to predict feature pixel positions between frames and use re-projection error as an innovation term for the update step. We encode visual and depth information at feature detector and descriptor level making them more robust in visually degraded conditions, i.e. low illumination and texture-less conditions. By combining visual and depth information at feature level we only need to track a small number of multi-modal features in the filter state making the whole odometry computationally tractable. To the best of our knowledge this tightly integrated multi-modal framework has no precedent in robot odometry estimation literature.

Chapter 3

Methodology

3.1 Overview

In this chapter we detail our multi-modal framework. Broadly our approach can be divided into three main components namely: a) Feature Generation, b) Descriptor Extraction and c) the extended Kalman filter framework. Visual and depth images are used in the first two components and hence encode information from both sensing modalities. The main idea behind fusion of vision and depth at feature detection and description level is to augment both these steps in case one of the sensing modalities encounters degraded conditions as detailed in section: 2.1. This multi-modal approach also enables us to track visual and depth features in a combined manner rather than tracking them separately as part of the filter state and hence in keeping the filter state small. By tracking features as part of the state we are able to calculate the uncertainty of every feature individually and hence in a sense control the contribution each feature makes to the odometry calculation process. Figure: 3.1 shows a diagrammatic overview of the whole framework.

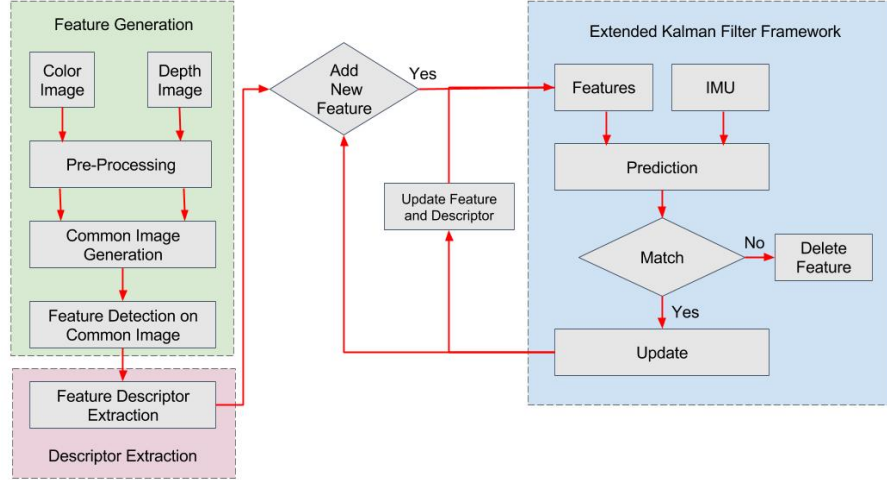


Figure 3.1: An overview of the proposed multi-modal odometry framework. Visual and Depth images are utilized for generation and description of features. The features are tracked in an extended Kalman filter framework where IMU data is used to predict feature positions in successive frames. In the update step of the filter framework if feature matching is successful the difference of predicted feature location and matched feature location is used as the innovation term. If feature matching fails, the feature is marked invalid and it’s corresponding uncertainty statistics are reset. When the tracked number of features falls below a certain threshold new features are added, for successfully tracked features their pixel location and descriptor are updated.

3.2 Feature Generation

In the field of computer vision, a lot of research has been done on detection of invariant and repeatable features in images [37, 38]. On the other hand in the depth domain most of the feature detection methods developed operate on 3D pointclouds rather than directly on depth images. Similarly, a small body of work exists which combines depth and visual information for the detection of multi-modal features. Although such approaches [33, 34, 39, 40] take advantage of both sensing modalities they, are usually very slow in operation as they are designed with the motivation to perform registration tasks rather than generating real-time odometry estimates. In this section we describe an approach for the detection of multi-modal features suitable for real-time odometry estimation tasks. We accomplish this by generating

a combined score image for visual and depth images which identifies salient points in both modalities.

3.2.1 Image Pre-Processing

Upon receiving a pair of visual and depth images we start by pre-processing them in preparation for feature detection and descriptor extraction.

First, for visual image, we convert the RGB image into grayscale and undistort it using the intrinsic calibration matrix obtained by the camera calibration package provided in Robot Operating System (ROS). Although we know the obtained visual image has some noise at the initial phase we do not perform any image filtering at this stage in order to not over-smooth the image and deal with it later in the feature detection and descriptor extraction phases.

As compared to visual images, depth images are very noisy as they are obtained using an active sensor. Such active sensors typically use structured light, IR stereo or Time-of-Flight technology which necessitates that the sensor carries its own source of light in the respective spectrum. Figure: 3.2 shows an image obtained from our sensor with black spots showing missing depth information. As depth sensors are active in nature, the emitted light which is infra-red(IR) in our case, might not return to the sensor due to high reflectance angle of the surface it falls upon or due to absorption of IR light by objects of dark color. Similarly, due to noise as mentioned in [41], very bright spots show incorrect depth measurements with values outside the sensor measurement range .

Dependent on the depth sensor, range depth measurements lying outside the bounds must be filtered out and marked as invalid. For our sensor all points outside the range



Figure 3.2: Depth image obtained from our sensor before any filtering operations are performed. Black spots show areas of missing depth. Very bright spots show incorrect depth measurements.

of $0.75m \geq depth \leq 6.0m$ are marked as invalid and their value is set to zero. Our depth image is obtained as an unsigned 16-bit image and reports depth measurements up to a resolution of millimeters. As mentioned in [42] and [41] depth sensors lose sensitivity as a function of depth and hence it is advised to lower the precision of measurements. Doing this is helpful as it aids computation of consistent Normals of surfaces, which are used in descriptor extraction. Keeping this in mind we round depth measurements to the nearest centimeter value. Subsequently, to smooth out the depth image and fill in some of the gaps in missing depth data we apply a median filter. Median filter works by selecting a neighborhood around a pixel and replacing the value of the pixel with the median value of the neighborhood. Hence, median filter can fill out missing data while preserving edges. Figure: 3.3 shows a filtered depth image after mentioned pre-processing operations are applied to Figure: 3.2.



Figure 3.3: Depth image obtained after filtering operations have been applied.

3.2.2 Image Registration

In order to establish pixel-to-pixel correspondence between visual and depth images they need to be registered using the intrinsic and extrinsic calibration parameters of the camera. Using intrinsic calibration parameters of the depth camera we project all pixels having valid depth values into the $3D$ space.

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_{dep \rightarrow 3D} = K_{dep}^{-1} d_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}_{dep}$$

where X_i, Y_i, Z_i represent projected $3D$ coordinates of pixel i w.r.t the depth camera frame. K_{dep}^{-1} represents the intrinsic calibration matrix of the depth camera, d_i is the depth value of the pixel and u_i, v_i are the depth pixel coordinates. Using extrinsic calibration parameters we then transform the projected $3D$ points from the depth

camera frame to the visual camera frame.

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_{vis \rightarrow 3D} = R_{dep}^{vis} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_{dep \rightarrow 3D} + T_{dep}^{vis}$$

where R_{dep}^{vis} and T_{dep}^{vis} represent extrinsic rotation and translation matrices between visual and depth cameras. We then re-project transformed 3D points into the 2D image space using intrinsic camera parameters of the visual camera.

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix}_{vis} = K_{vis} \frac{1}{Z_i} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}_{vis \rightarrow 3D}$$

where u_i, v_i represent the pixel location in visual image corresponding to pixel i in the depth image. K_{vis} represents the intrinsic calibration matrix of visual camera. When projecting 3D points to 2D space we perform an occlusion check if two depth pixels transform to the same pixel position in visual image, in case this happens we keep the smallest depth. Performing registration between two images allows us to generate pixel-to-pixel maps from the visual image to the depth image and vice versa. This also allows to generate an RGB pointcloud that we later use for mapping purposes. Figure: 3.4 shows the generated RGB point-cloud of the same scene.

3.2.3 Common Image Generation

For the detection of feature points from visual and depth images we create a common image with each pixel annotated with a score in the 0-255 range describing its saliency in either one or both domains. For detection of salient points in visual im-



Figure 3.4: RGB pointcloud generated during the registration process.

ages we make use of the ORB feature detector [43]. We choose ORB primarily for two reasons. First, ORB detects FAST keypoints [44] across multi-scale image pyramids making the keypoints less sensitive to noise and scale variation. Secondly, ORB returns Harris corner score for each detected keypoint providing us with a metric for each keypoint. We normalize these Harris scores in 0-255 range and mark them accordingly in the common image. Figure: 3.5 shows the detected ORB keypoints on the visual image. Similarly, Figure: 3.6 shows the normalized score image using the same keypoints.



Figure 3.5: Red dots show the detected ORB keypoints on the visual image.

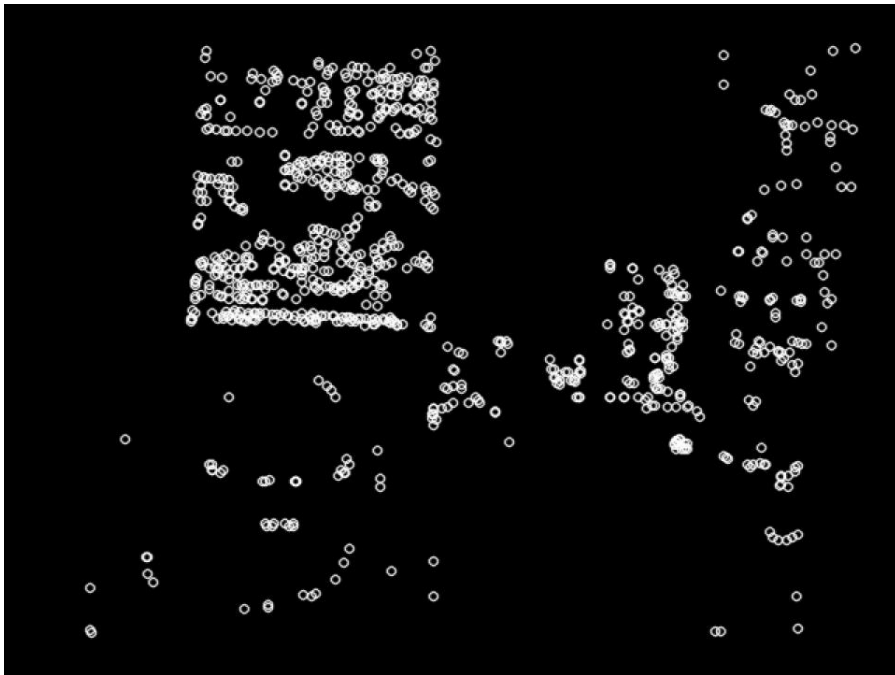


Figure 3.6: Image shows pixels marked with the normalized ORB score.(Binary image shown here for ease of viewing purposes).

For detection of keypoints in depth image, we take an initial inspiration from the work of [45] related to calibration of camera and lasers. We make use of the intuition that edges in depth images and visual images would lie along the same coordinates. Using this intuition we first normalize the depth image into the 0-255 range and use an edge detector to get an initial estimate of edge locations. This is primarily done for data reduction purposes in order to keep the process computationally tractable for real-time operations. We make use of Canny edge detector primarily because of its highly optimized code available as part of the OpenCV library and built-in non-maxima suppression operation. As Canny edge detector makes use of Sobel derivative operator, pixels that lie near invalid depth points create a very strong edge response. To eliminate these incorrect edges we create an inverse invalid depth mask with invalid depth points marked as 1. We dilate this mask and multiply with the detected edge points image to suppress incorrect edges. Next to generate a score metric we choose a neighborhood around each remaining edge point in the filtered depth image. We calculate the number of points in the neighborhood of each edge point which have depth difference higher than a defined threshold with respect to the corresponding edge point. This operation allows us to differentiate between corner and edge points. We normalize the score for each point in the 0-255 range. Furthermore, to filter out unstable points during this operation, we reject any edge point if an invalid depth point lies in its neighborhood. As depth sensors are active sensors object boundaries are not constant in their pixel locations in the depth images as mentioned in [41]. To improve the repeatability of detection of depth points we apply a small dilation kernel on the detected depth points. Figure: 3.7 shows the detected depth edges with normalized score.

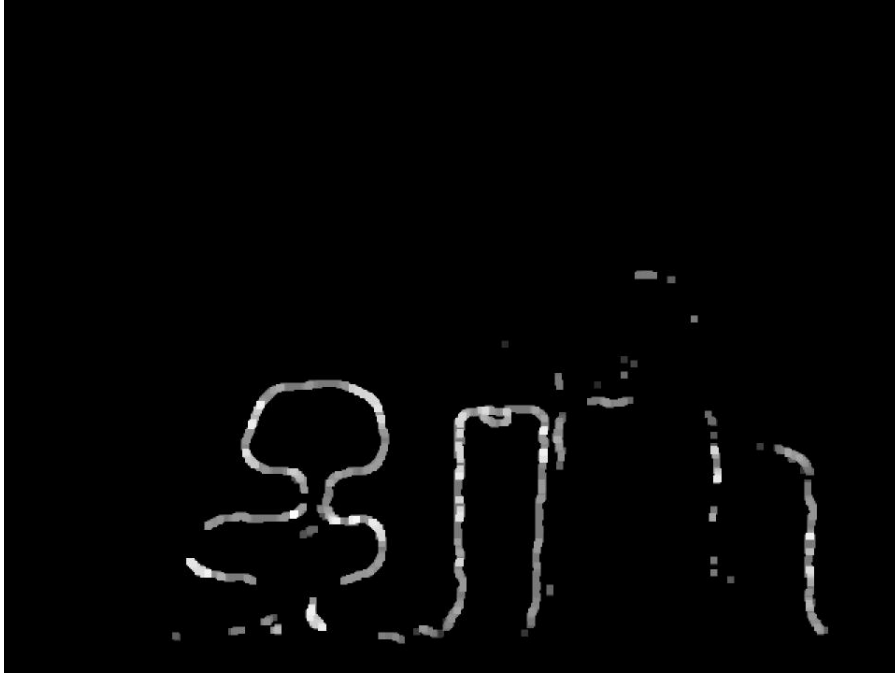


Figure 3.7: Image shows detected depth edges with score normalized.

We combine the scores for both visual and depth corner images with the idea that pixels salient in both domains should score higher. We add the scores, with maximum score not exceeding 255, in our final common image. Figure: 3.8 shows the combination of visual and depth scored images for selection of features.



Figure 3.8: Final combined visual and depth score image to be used for final feature selection.

3.2.4 Feature Selection

For odometry calculation purposes it is important to have a good distribution of feature points across an image as in the case of feature points clustered in one area of the image a large amount of features can be lost during any significant motion of the camera. This sudden loss of a large number of features can cause odometry estimates to diverge.

For this purpose we divide our calculated common image into a 5×5 grid. From each grid patch we select 10 best features giving us a total of 250 possible candidate points. We filter out points which fall very near to the edge of the image. The distance to the edge is selected according to the size of the descriptor extraction neighborhood size. We sort the remaining feature points according to their score and apply a minimum Euclidean distance constraint. We first select the best feature point in our candidate

and add it to our selected list. Next, we select the second best feature in the candidate list and check if it satisfies the minimum Euclidean distance constraint against the features in the selected feature list. If it does, we add it to the list otherwise we move down the list. The process is terminated when a predefined number of features are selected or we exhaust our search in the candidate list. Whenever a feature is to be added to the state it is first checked against all features already tracked in the state to check if minimum Euclidean distance constraint is satisfied. Figure: 3.9 shows an instance of 25 best selected features from the combined score image. From the location of features selected it can be seen that they are selected from both visual and depth domains.



Figure 3.9: An instance of 25 best selected features from the combined score image that satisfy the selection criteria. Comparing the location of selected features to the Visual and Depth Score images it can be observed that the selected features originate from both domains.

3.3 Descriptor Extraction

3.3.1 Descriptor Selection

Given a set of selected feature points originating from both visual and depth domains we next extract descriptors in order to track them in successive frames. As mentioned in section: 2.1 very few approaches have been proposed to create descriptors encoding information from both sensing modalities with CSHOT [46], RISAS [34] and BRAND [32] being some examples. For real-time odometry calculations in addition to good recall and precision performance, it is of utmost importance that the descriptor extraction and matching processes are fast. CSHOT and RISAS are categorized as histogram based descriptors as they work by creating and concatenating multiple histograms. For matching purposes histogram based descriptors utilize Euclidean distance as a metric. Both of these techniques are computationally expensive [34, 46] hence rendering histogram based descriptors, although very good for registration purposes, not suitable for our application. BRAND on the other hand belongs to the category of binary descriptors which are very popular in the field of computer vision because of their computational and memory efficiency. Binary descriptors work by sampling the neighborhood of a keypoint and performing lightweight pixel-wise comparisons to generate a bit string representation of the neighborhood. Matching of binary descriptors is very fast as this is done by calculating Hamming distance by performing *XOR* operation between two bit strings. For this reason in our application we choose the BRAND descriptor, with some modifications, as it is the only binary descriptor in our knowledge that encodes both visual and depth information. Below we present a short overview of the BRAND descriptor along with our modification and the motivation behind the changes. We encourage the reader to read the original publication [32].

3.3.2 Descriptor Details

Following similar approaches in the field of computer vision, BRAND generates a descriptor by performing pair-wise pixel tests in the neighborhood of a keypoint by selecting pixel pair locations according to a pre-determined sampling pattern. To encode the information from both sensing modalities, BRAND first generates two separate visual and depth descriptors of the same length, by performing intensity and geometric tests respectively, and then combines them by performing bit-wise *OR* operations. Figure: 3.10 shows a visual representation of the descriptor generation process.

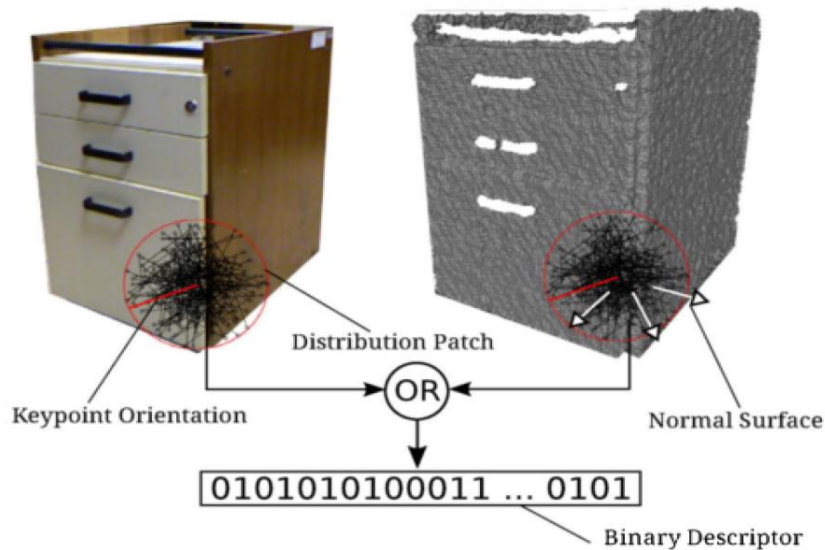


Figure 3.10: This figure shows two descriptors of equal size are created using data in each sensing modality and then combined together by performing bit-wise *OR* operations. (Image courtesy [32])

To create a sampling pattern BRAND takes its inspiration from [47]. However, instead of using the whole pattern of size 48×48 , only pairs that lie within a circle of radius of 24 pixels are used, making the sampling pattern less sensitive to in-plane rotation. To reduce sensitivity to image noise, this sampling pattern is also

pre-smoothed by applying a Gaussian kernel of size 9×9 . Figure: 3.11 shows the sampling pattern used in this work.

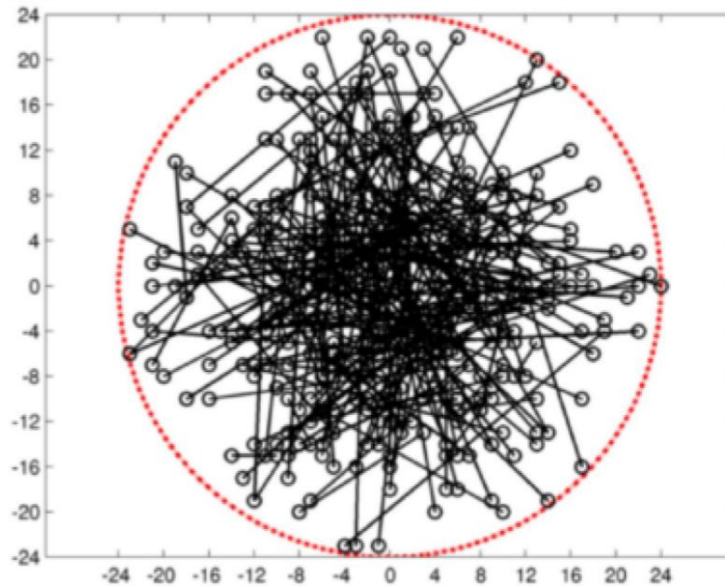


Figure 3.11: Descriptor sampling pattern of size 48×48 showing 256 sampled pairs of pixel locations to be used to perform intensity and geometric tests. (Image courtesy [32])

Furthermore to make the descriptor less sensitive to scale changes depth data at keypoint locations is used to estimate scale using the following equation:

$$\text{scale} = \max \left(0.2, \frac{3.8 - 0.4 \max(2, \text{depth})}{3} \right)$$

The estimated scale is used to scale the size of the sampling pattern which can vary from a size of 9×9 to 48×48 depending on the depth of the keypoint.

For generation of the visual part of the descriptor, similar to other approaches in computer vision, pair-wise pixel intensity comparison is performed. For a pair of pixel locations P_1 and P_2 with intensities I_1 and I_2 the visual comparison function V can

be represented as:

$$V(P_1, P_2) = \begin{cases} 1 & \text{if } I_1 < I_2 \\ 0 & \text{otherwise} \end{cases}$$

However, different from the original work in [32], we choose to modify the above comparison for two reasons. First, comparing pixel intensity directly is susceptible to pixel noise and small intensity changes due to noise can cause erroneous bit switching in the the descriptor. Secondly, in low illumination conditions, vision camera sensors are susceptible to dark noise, which is given as:

$$\text{dark noise} = \sqrt{\text{dark current} \times \text{integration time}}$$

Dark noise can generate false intensity values which although are insignificant in good illumination conditions, they can become significant in low illumination conditions when intensity values are very small, hence causing visual descriptor, to be very noisy and leading to false positive and negative matches. To alleviate these two problems we make the following changes. First, to reduce the sensitivity of the descriptor to pixel noise instead of comparing pixel intensity values we compare mean intensity values using patches of size 9×9 created around sampled pair locations. Secondly, to reduce the effect of dark noise we subtract a small intensity value, representing intensity changes due to dark noise, from the mean intensity values of the patches before performing the pair-wise comparison. This intensity representation of dark noise ($I_{\text{dark noise}}$) was calculated by collecting images in a very dark environment where we expect the intensity value to be zero and calculating the mean intensity value across these images. For our sensor we use an intensity value of 5 as a representation of

maximum dark noise of the sensor. The modified intensity comparison function for a pair of pixel locations P_1 and P_2 with mean patch intensities \bar{I}_1 and \bar{I}_2 can be written as:

$$V(P_1, P_2) = \begin{cases} 1, & \text{if } \max(0, \bar{I}_1 - I_{\text{dark noise}}) < \max(0, \bar{I}_2 - I_{\text{dark noise}}) \\ 0, & \text{otherwise} \end{cases}$$

The max function in the above equation is used to ensure that the minimum allowed intensity value is 0.

For the generation of the depth part of the descriptor two pair-wise geometric tests are performed in order to encode underlying surface properties. To perform these geometric tests, first the depth image is projected into 3D space using the intrinsic camera parameters of the depth camera and stored in pointcloud format using the Point Cloud Library (PCL). At each 3D point in the pointcloud, surface normals are estimated using neighborhood points. As we have a dense and organized pointcloud we take advantage of normals estimation using integral images [42], available as part of PCL for faster calculations. We then carry out the following two geometric tests:

Normal Displacement Test: This test compares the angle between two normal vectors to check if it is greater than a pre-defined threshold angle, by computing their dot product. For a pair of pixel locations P_1 and P_2 with normals n_1 and n_2 this geometric test function G_1 can be written as:

$$G_1(P_1, P_2) = \begin{cases} 1, & \text{if } (n_1 \cdot n_2) < \textit{threshold} \\ 0, & \text{otherwise} \end{cases}$$

In our work the threshold is chosen to be $\cos(45^\circ)$.

Convexity Test: Although the Normal Displacement Test captures the curvature of the surface yet it does not tell us anything about the nature of the curved surface, i.e. if it is convex or concave. To disambiguate we take the dot product of the vector difference of the normals vectors at pixel pair locations and the vector pointing from the first 3D point to the second 3D point and mark the surface as concave if the dot product of these two difference vectors is less than zero. For a pair of pixel locations P_1 and P_2 with normals n_1 and n_2 and 3D point locations p_1 and p_2 , this geometric test function G_2 can be written as:

$$G_2(P_1, P_2) = \begin{cases} 1, & \text{if } (n_1 - n_2) \cdot (p_1 - p_2) < 0 \\ 0, & \text{otherwise} \end{cases}$$

The depth descriptor is then created by performing bit-wise *AND* operation between the results of Normal Displacement Test and Convexity Test for every pair wise comparison and combined depth comparison D function can be written as:

$$D(P_1, P_2) = G_1 \wedge G_2$$

Finally, visual and depth information is combined by performing bit-wise *OR* operation on the calculated visual and depth descriptors represented as V and D respectively. The final bit string at keypoint location k can be written as:

$$BRAND(k) = \sum_{i=1}^{256} 2^{i-1} (V_i \vee D_i)$$

3.4 Filter Framework

3.4.1 Overview

In the field of computer vision, the problem of pose estimation and 3D reconstruction has been extensively studied which has resulted in the development of a number of approaches some of which we have mentioned in section: 2.1. However, if only visual cues are used for motion estimation the resulting approach may lack robustness. For example, feature-based approaches are sensitive to feature mismatches and hence require some method like RANSAC [19] for pruning of mismatches. Similarly, direct intensity approaches require some constraints to be applied on the optimization process in order for the solution to not diverge. When tracking points or regions of interest between successive images, a search has to be performed over the entire images making the search space larger and hence making them computationally expensive and more sensitive to fast motions. Employing an IMU as an additional sensing modality can *significantly* improve both robustness and accuracy of these approaches because of the complementary nature of IMU data. An IMU can provide reliable information for short motions at a very high update rate which can not only be used as a good motion prior for pose estimation processes. IMU integration can also be used to reduce the search space for feature matching as well as aid in pruning outlier matches. Furthermore, for monocular systems an IMU provides observability of scale hence making motion estimates more robust and useful. A number of proposed approaches have successfully combined visual and inertial information utilizing both visual features [48–50] and direct image intensity information [23,51,52] for pose estimation.

Our approach falls under the category of feature-based approaches as we aim to track multi-modal features across successive frames. Similar to the above-mentioned approaches, we employ an extended Kalman filter framework for the integration of inertial data with our multi-modal features. Our overall filter structure is similar to the one proposed in [23]. The proposed filter framework follows a fully robot-centric formulation as it allows for the decoupling of unobservable states, namely position and yaw, from the rest of the filter states. In this formulation three coordinate frames namely, the Inertial Measurement Unit (IMU) fixed coordinate frame \mathcal{I} , the camera fixed frame \mathcal{V} , and the world inertial frame \mathcal{W} , are used. As described in Section: 3.2.2 we register our depth image with respect to the visual image hence all the depth data is expressed in camera fixed frame \mathcal{V} and does not require a separate coordinate frame. For the parametrization of multi-modal features, a landmark approach is followed which models 3D feature locations by a using 2D bearing vector (parametrized with azimuth and elevation angles) and a depth parameter. In this work inverse depth parametrization similar to [21] was used. The advantage of using this landmark parametrization is that it allows for integrating new features into the filter state without a delay as we can initialize a feature with a random depth, in case of missing depth data, with a large uncertainty without affecting the bearing vector estimate. This compact representation also allows to carry feature points as part of the state and to estimate their joint uncertainty. Hence the filter state can be written as:

$$\mathbf{x} = \left[\overbrace{\begin{bmatrix} \mathbf{r} & \mathbf{q} & \mathbf{v} & \mathbf{b}_f & \mathbf{b}_\omega & \mathbf{c} & \mathbf{z} \end{bmatrix}}^{\text{pose, } l_p} \mid \underbrace{\begin{bmatrix} \boldsymbol{\mu}_0, & \cdots & \boldsymbol{\mu}_J & \rho_0 & \cdots & \rho_J \end{bmatrix}}_{\text{features states, } l_f} \right]^T$$

robot states, l_s
features states, l_f

where l_p, l_s, l_f are dimensions, \mathbf{r} is the robot-centric position of the IMU expressed in \mathcal{I} , \mathbf{v} represents the robot-centric velocity of the IMU expressed in \mathcal{I} , \mathbf{q} is the IMU attitude represented as a map from $\mathcal{I} \rightarrow \mathcal{W}$, \mathbf{b}_f represents the additive accelerometer

bias expressed in \mathcal{I} , \mathbf{b}_ω stands for the additive gyroscope bias expressed in \mathcal{I} , \mathbf{c} is the translational part of the IMU–cameras extrinsics expressed in \mathcal{I} , \mathbf{z} represents the rotational part of the IMU–cameras extrinsics and is a map from $\mathcal{I} \rightarrow \mathcal{V}$, while $\boldsymbol{\mu}_j$ is the bearing vector to feature j expressed in \mathcal{V} and ρ_j is the depth parameter of the j^{th} feature such that the feature distance d_j is $d(\rho_j) = 1/\rho_j$.

3.4.2 State Propagation

In our framework state propagation is driven by using proper acceleration $\hat{\mathbf{f}}$ and rotational rate measurements $\hat{\boldsymbol{\omega}}$ provided by the IMU. Both measurements are affected by bias and noise, as we track bias as part of our filter state we can write the bias-corrected but noise-affected inertial measurements as:

$$\begin{aligned}\hat{\mathbf{f}} &= \tilde{\mathbf{f}} - \mathbf{b}_f - \mathbf{w}_f \\ \hat{\boldsymbol{\omega}} &= \tilde{\boldsymbol{\omega}} - \mathbf{b}_\omega - \mathbf{w}_\omega\end{aligned}$$

where \mathbf{b}_f and \mathbf{b}_ω represent bias parameters and \mathbf{w}_f and \mathbf{w}_ω represent noise parameters for accelerometer and gyroscope measurements respectively. Using bias corrected IMU measurements and given an extrinsic calibration between IMU and camera coordinate frame(extrinsic calibration process in described in Chapter: 4) estimated camera linear velocity $\hat{\mathbf{v}}_\mathcal{V}$ and camera rotational velocity $\hat{\boldsymbol{\omega}}_\mathcal{V}$ can be written as:

$$\begin{aligned}\hat{\mathbf{v}}_\mathcal{V} &= \mathbf{z}(\mathbf{v} + \hat{\boldsymbol{\omega}}^\times \mathbf{c}) \\ \hat{\boldsymbol{\omega}}_\mathcal{V} &= \mathbf{z}(\hat{\boldsymbol{\omega}})\end{aligned}$$

Using these equations, the following set of continuous differential equations for state propagation can be written as:

$$\begin{aligned}
\dot{\mathbf{r}} &= -\hat{\boldsymbol{\omega}}^\times \mathbf{r} + \mathbf{v} + \mathbf{w}_r \\
\dot{\mathbf{v}} &= -\hat{\boldsymbol{\omega}}^\times \mathbf{v} + \hat{\mathbf{f}} + \mathbf{q}^{-1}(\mathbf{g}) \\
\dot{\mathbf{q}} &= -\mathbf{q}(\hat{\boldsymbol{\omega}}) \\
\dot{\mathbf{b}}_f &= \mathbf{w}_{bf} \\
\dot{\mathbf{b}}_\omega &= \mathbf{w}_{b\omega} \\
\dot{\mathbf{c}} &= \mathbf{w}_c \\
\dot{\mathbf{z}} &= \mathbf{w}_z \\
\dot{\boldsymbol{\mu}}_j &= \mathbf{N}^T(\boldsymbol{\mu}_j) \hat{\boldsymbol{\omega}}_\nu - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{N}^T(\boldsymbol{\mu}_j) \frac{\hat{\mathbf{v}}_\nu}{d(\rho_j)} + \mathbf{w}_{\mu,j} \\
\dot{\rho}_j &= -\boldsymbol{\mu}_j^T \hat{\mathbf{v}}_\nu / d'(\rho_j) + w_{\rho,j}
\end{aligned}$$

where $^\times$ represents the skew symmetric matrix of a vector, term $\mathbf{N}^T(\boldsymbol{\mu})$ is projection of a 3D vector onto the 2D tangent space around the bearing vector, \mathbf{g} is the gravity vector and \mathbf{w}_\star represent all white Gaussian noise processes.

A forward Euler integration scheme is used for the discretization of the continuous differential equations. As IMU measurements are received at a high update rate and the time between two update steps is small, IMU measurements are pre-integrated by simply calculating their mean before applied for the calculation of Jacobians.

3.4.3 State Update

Given camera intrinsic calibration parameters ($\boldsymbol{\pi}$) we can compute the pixel coordinates of a bearing vector as $\mathbf{p}_i = \boldsymbol{\pi}(\boldsymbol{\mu}_i)$, where \mathbf{p}_i and $\boldsymbol{\mu}_i$ denote the pixel coordinates and the bearing vector of a feature i respectively. As mentioned, before using the IMU measurements the bearing vectors of each feature are propagated frame to frame predicting the new pixel location of the feature in the new image. By using the predicted uncertainty of the feature we create a window centered around predicted feature location from which we sample feature points. In this work we sample 3 feature points from the window based on our feature detection approach detailed in section: 3.2 and extract the descriptors for matching. The difference between the pixel coordinates of the best matched feature location and the predicted feature location is used as an innovation term.

When features are introduced in the filter state we check if depth measurement is available for that feature, if available we initialize the feature with depth measurement with a smaller associated uncertainty whereas for features without any depth measurement available we initialize them with a fixed depth value and a large associated uncertainty. Allowing for depth measurements to be directly integrated allows for the faster convergence of depth estimates for features without depth measurements. Similarly allowing for features without depth measurements to be integrated ensures that very good visual features are utilized even though they do not have any depth measurement. If a feature is successfully tracked over a certain time period we re-extract and update its descriptor in order to minimize the warping effects, as well as update the depth if a depth measurement is available by taking the median depth of a small patch around the feature's current location and calculating its mean with the feature's estimated depth at the time.

Chapter 4

Experimental Results

4.1 Sensor Setup and Calibration

For the testing of the proposed framework, an Intel Realsense ZR300 sensor was used as it provides visual, depth and inertial data in a single sensor package which is compact, lightweight and low cost. This sensor calculates depth by employing infrared (IR) stereo cameras and carries an on-board IR projector hence eliminating the need of external illumination for range measurements. Disparity calculations for depth estimation using stereo cameras are done on-board the sensor chip hence eliminating the need for host side computation of depth. The visual camera has a maximum resolution of 1920×1080 pixels with a diagonal field-of-view(FOV) of 75° and can operate at a frame rate of 30Hz. In our localization framework images of resolution 640×480 pixels were used. The depth camera has a maximum resolution of 628×468 pixels with a diagonal field-of-view(FOV) of 70° and operates at a frame rate of 30Hz. To get data from the sensor, a Robot Operating System (ROS) driver package was used. The default IMU data from the ROS driver is provided at an update rate

of 400Hz, however upon investigation it was found that internally the accelerometer data is sampled at 250Hz and the gyroscope data is sampled at 200Hz, hence the IMU sensor messages had repeated data or invalid data. As the driver package is open-sourced it was corrected to publish valid data at an update rate of 200Hz. For the synchronization of visual and depth images, a software synchronization is performed in our framework upon receiving images. However, IMU measurements are neither hardware nor software synced with the images in our implementation.

The intrinsic and extrinsic calibration for the visual and depth cameras are stored on-board the sensor and can be accessed via the provided ROS driver package. Although the intrinsic calibration for the visual camera was provided, for verification purposes a calibration using a checker-board pattern and ROS camera calibration package, which implements the work of [53], was performed. Both calibration results were very similar with a re-projection error of less than 0.1 pixels. The intrinsic calibrations of visual and depth cameras as well as the extrinsic calibration from depth camera frame to visual camera frame were used for the registration of depth image to visual image for pixel-to-pixel correspondence and generation of point-cloud as mentioned in section: 3.2.2.

4.1.1 IMU characterization

Although intrinsic and extrinsic calibrations are provided for the cameras, no noise or bias parameters for the IMU or the IMU-Camera extrinsics are provided, which are essential for the functioning of our framework. For IMU the noise is modeled as Additive White Noise or Noise Density whereas bias can be modeled as Random Walk. These parameters can be usually found in the manufacturers provided data sheet but as Intel neither provides this data sheet nor the model or manufacturer

information of the IMU sensor we have to identify them ourselves. There are many parametric and non-parametric methods available for determining IMU model parameters. Determining them using Allan Standard Deviation plots is the most common approach as listed in the IEEE standard [54]. An IMU data set in stationary conditions in a minimal vibration environment was recorded. Allan variance is computed by calculating the variance by taking M-samples at a time from the data set. These M-samples are taken according to an increasing range of pre-defined sampling times. Figure: 4.1 and 4.2 show the Allan Standard Deviation plots for the accelerometer and gyroscope data respectively.

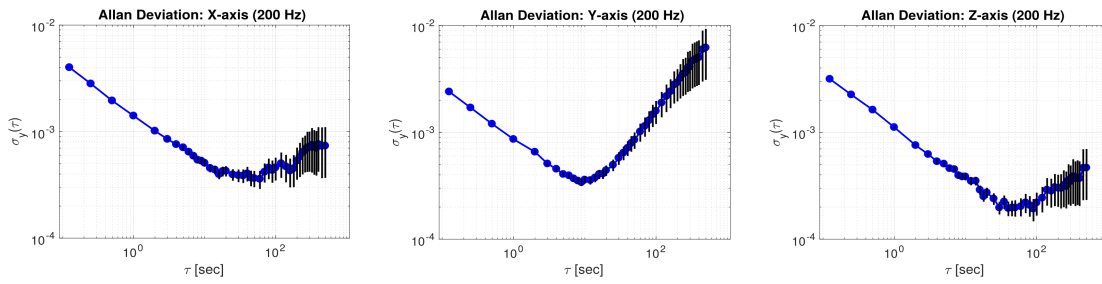


Figure 4.1: Allan Standard Deviation plots for X,Y and Z axis of the accelerometer.

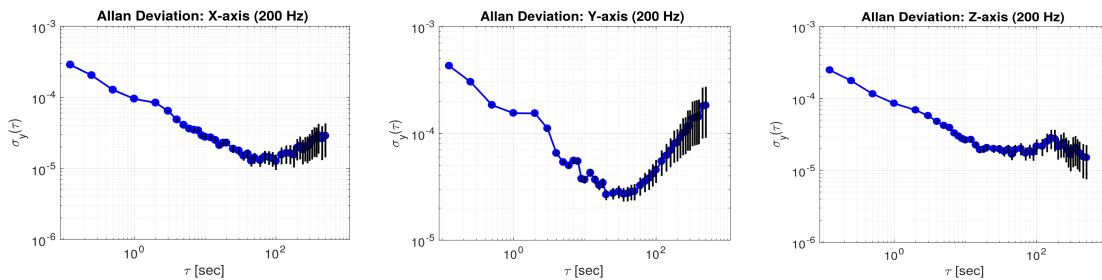


Figure 4.2: Allan Standard Deviation plots for X,Y and Z axis of the gyroscope.

The Noise Density can be directly calculated from the plot by looking up the value at a sampling time of 1 second ($\tau = 1 \text{ sec}$) whereas for Random Walk calculation we need to fit a line $y = 0.5x + c$ to the curve and find where it intersects with axis at sampling time of 3 second ($\tau = 3 \text{ sec}$). The calculated IMU parameters are given in

Table: 4.1 and 4.2:

Accelerometer	
Noise Density ($\frac{m}{s^2} \frac{1}{\sqrt{Hz}}$)	
X-axis	$1.406e - 03$
Y-axis	$8.624e - 03$
Z-axis	$1.115e - 03$
Random Walk ($\frac{m}{s^3} \frac{1}{\sqrt{Hz}}$)	
X-axis	$8e - 05$
Y-axis	$4e - 04$
Z-axis	$4e - 05$

Table 4.1: Noise and Bias parameters for the accelerometers

Gyroscope	
Noise Density ($\frac{rad}{s} \frac{1}{\sqrt{Hz}}$)	
X-axis	$9.508e - 05$
Y-axis	$1.544e - 04$
Z-axis	$8.529e - 05$
Random Walk ($\frac{rad}{s^2} \frac{1}{\sqrt{Hz}}$)	
X-axis	$3 - 06$
Y-axis	$1e - 05$
Z-axis	$4e - 06$

Table 4.2: Noise and Bias parameters for the gyroscopes.

4.1.2 Camera-IMU Extrinsic Calibration

Given the calculated IMU parameters an extrinsic calibration to identify the transformation from camera coordinate frame to IMU coordinate frame was performed using the work of [55] via their open-source package available at [56]. Although sensor manufacturing process can change the extrinsic calibration between IMU-Camera and calibration should be performed for every sensor individually we provide our result for reference purposes. The extrinsic calibration matrix $\begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$ from the visual

camera coordinate frame to the IMU coordinate frame is given below:

$$\begin{bmatrix} -0.00531084 & -0.00010303 & 0.99998589 & 0.02669322 \\ -0.99982049 & -0.01818722 & -0.00531184 & -0.09188259 \\ 0.01818751 & -0.99983459 & -0.00000642 & 0.01360754 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.2 Handheld Localization Test

To evaluate localization performance of our proposed algorithm we compared it in indoor light conditions against ground truth, provided by a VICON system and a visual-inertial framework ROVIO [23]. The purpose of this test was to validate the accuracy of localization and compare performance against a visual-inertial framework for the selected sensor. A trajectory along a $20m$ long rectangular path with 90° turns was followed. Figure: 4.3 shows a top-down plot of the path followed. It can be noted that although the shape of the path followed by our proposed framework and ROVIO is correct, both approaches are prone to scaling issues. In monocular setup ROVIO relies on accuracy of IMU for the estimation of gravity vector to correctly estimate feature depths. In our proposed framework, unless features are far away, we initialize them with direct depth estimates which can lead to a better estimation of scale. Comparison with a state-of-the-art visual-inertial odometry estimation framework also highlights the effects and limitations imposed by sensor quality on the estimation process. Naturally, besides the algorithmic and methodological improvements, better results can be derived if advanced sensors are used.

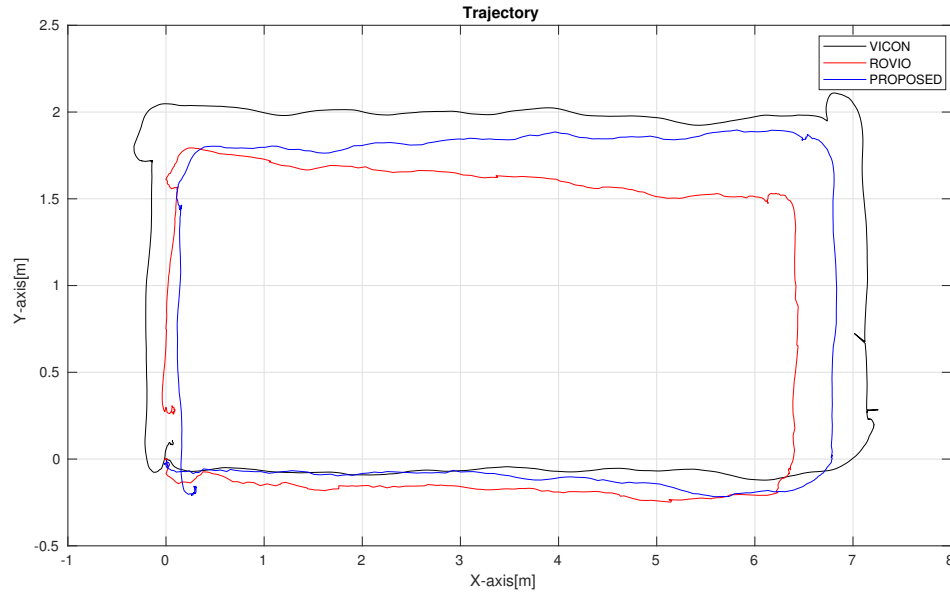


Figure 4.3: In a hand-held test we followed a rectangular path in indoor lighting conditions. Plot shows a top-down view of the path followed and compares our proposed approach against ROVIO and ground-truth provided by VICON.

Similarly, Figure: 4.4 shows localization comparison along each individual axis with Figure: 4.5 showing absolute error along each axis when compared to Vicon system. Along X and Y axes both approaches follow the ground-truth trajectory but in Z-axis we can see a drift which can be caused by the low-excitation of IMU along that axis. We calculate the Mean Squared Error and Root Mean Squared Error for each axis for the followed trajectory in Table: 4.3.

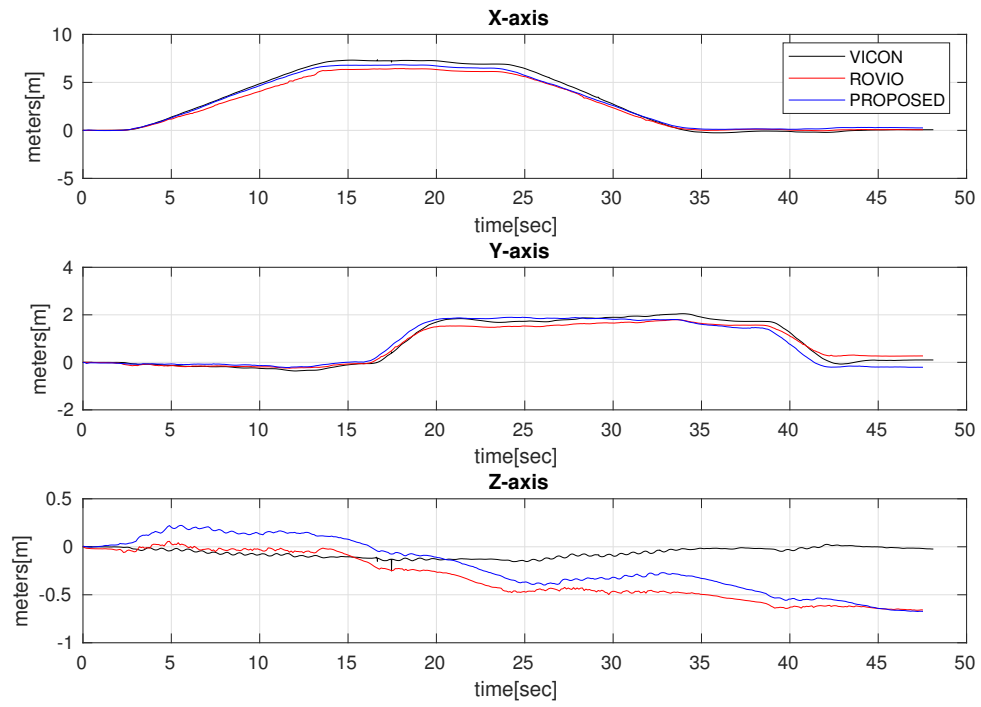


Figure 4.4: Comparison of our proposed approach against ROVIO and VICON system along each axis.

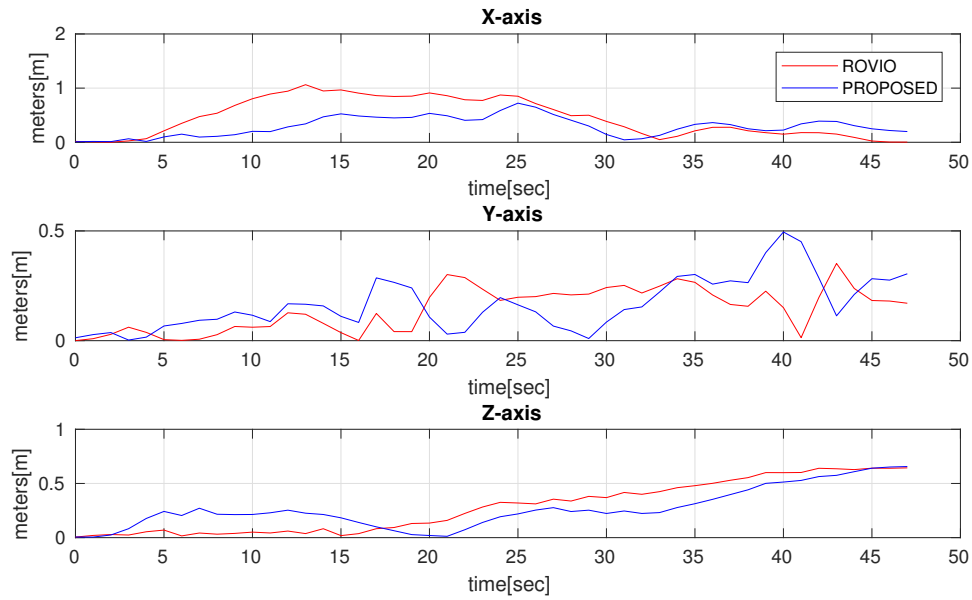


Figure 4.5: Absolute localization error plots showing comparison of our proposed approach and ROVIO with respect to VICON system along each axis.

Localization Error RMSE(meters)	
PROPOSED	
X-axis	0.2854
Y-axis	0.1985
Z-axis	0.3211
ROVIO	
X-axis	0.4965
Y-axis	0.2740
Z-axis	0.3647

Table 4.3: RMSE Localization errors with respect to VICON ground-truth during hand-held tests

4.3 On-board Robot Localization Test

To evaluate real-time performance on a robot, the proposed framework was deployed on a custom build hexa-rotor platform. Figure: 4.6 shows the robot fitted

with the ZR300 sensor facing forward. The robot is equipped with an Intel NUC Core-i7-5557U, for high level processing tasks and a pixhawk autopilot for attitude and position control of the robot. For the control of the robot a linear Model Predictive Controller (MPC) is utilized. The robot was tasked to fly autonomously a

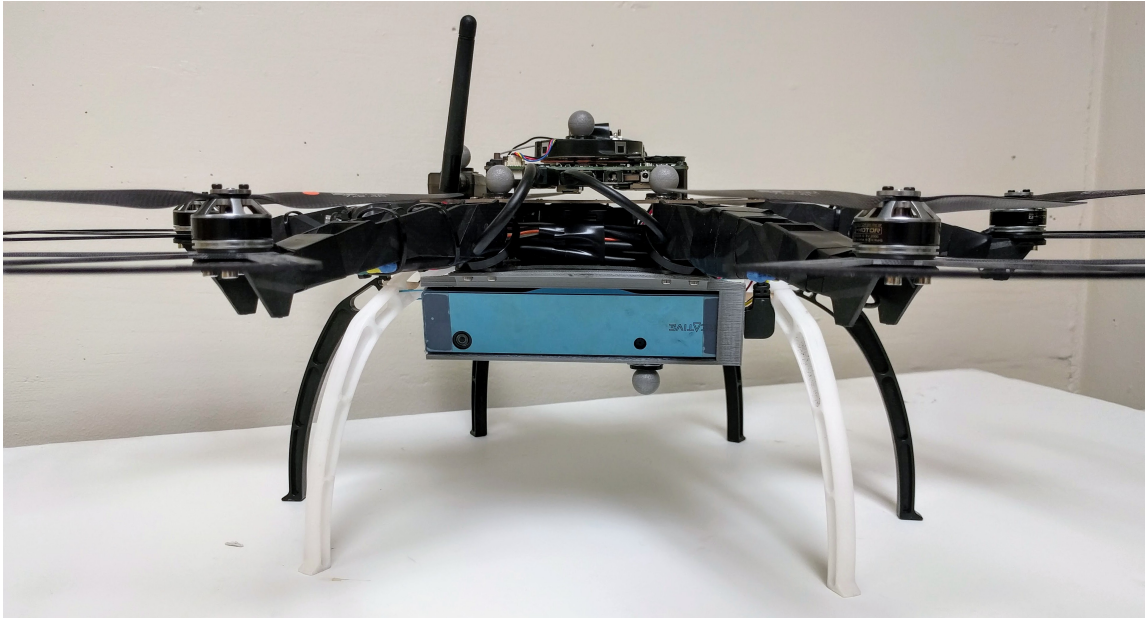


Figure 4.6: A custom built hexa-rotor platform was fitted with the ZR300 sensor and was used to evaluate the performance of the odometry framework. Odometry estimates were used in a Model-Predictive-Controller (MPC) to fly the robot autonomously on a predefined trajectory.

predefined trajectory utilizing odometry generated from our framework. The position of the robot was tracked using Vicon system for comparison purposes.

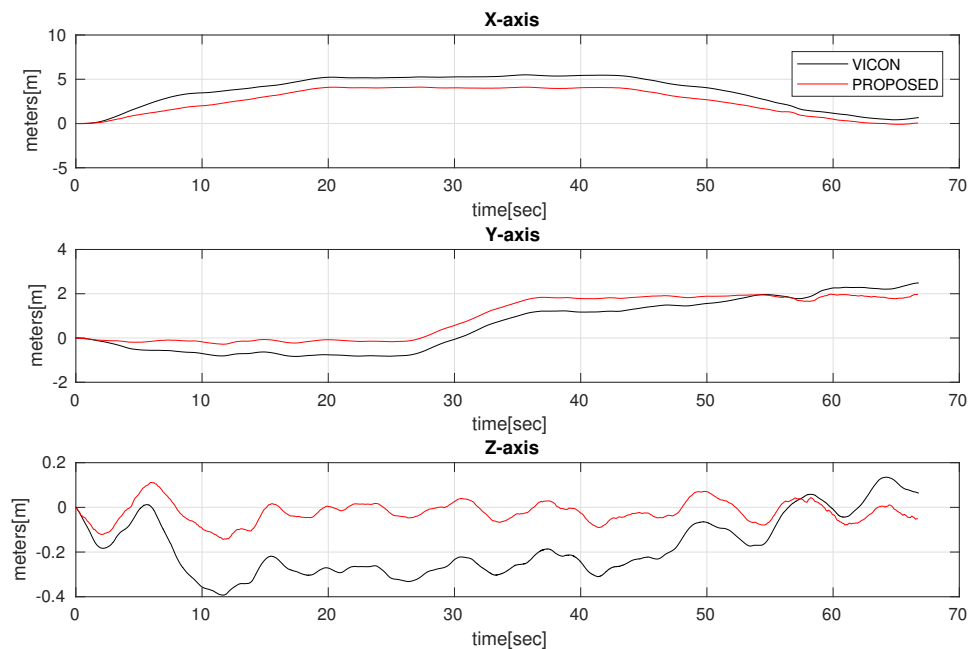


Figure 4.7: Plots show position along each axis as robot followed the predefined path autonomously. Position estimates are compared to ground truth provided by a Vicon system.

Localization Error RMSE(meters)	
PROPOSED	
X-axis	1.0234
Y-axis	0.3217
Z-axis	0.0856

Table 4.4: RMSE Localization errors with respect to VICON ground-truth during autonomous robot flight

4.4 Mapping Test

For testing the performance of our algorithm in visually-degraded conditions we mapped a room with lights turned off. Mapping was performed by annotating registered point-clouds with pose estimates obtained from our algorithm. No pose re-

finement, such as using Iterative Closest Point (ICP) to stitch point-clouds together, was applied on the map as we are interested in pose estimation performance of our algorithm in visually degraded conditions. Figure: 4.8 shows a top-down view of the created map using registered RGB point-clouds arranged according to pose estimates (shown in blue color).

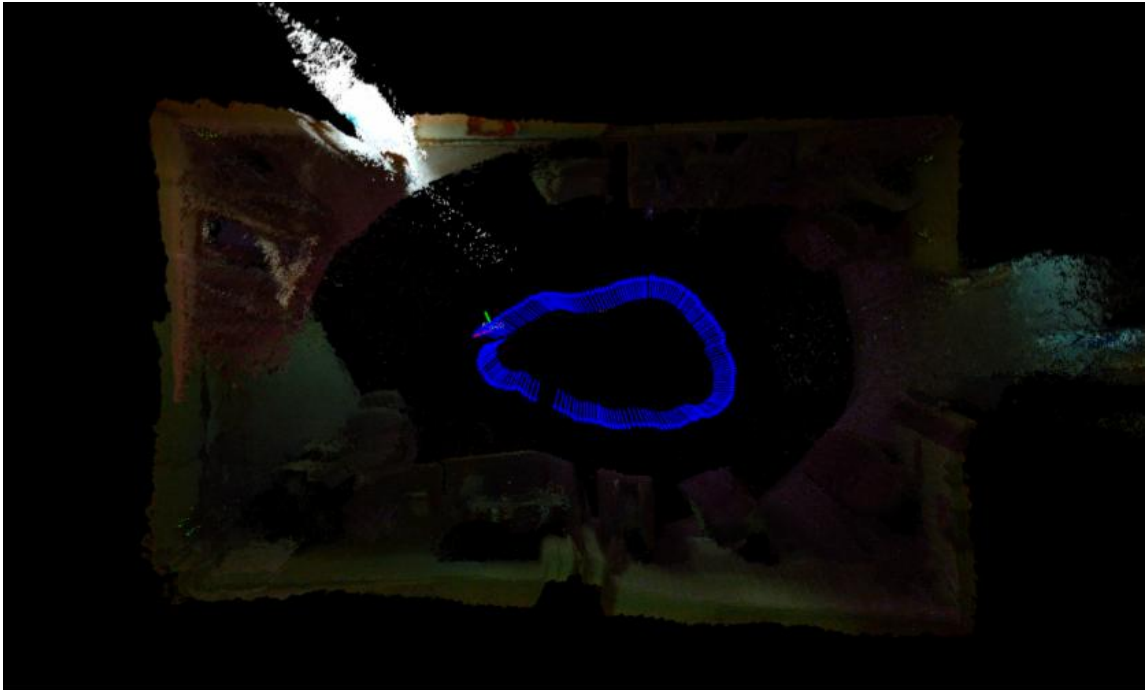


Figure 4.8: A top-down view of the 3D map created in visually degraded conditions using pose estimates (colored blue) generated by our proposed algorithm and RGB colored pointclouds. It can be observed that in such dark conditions the built map is consistent.

The same map but colored according to height values is shown in Figure: 4.9 and provides a better understanding of the objects placed in the room and the performance of the proposed approach. In the RGB colored map it can be seen that the top and bottom right corners have very low illumination and as a result neither our proposed algorithm or the previously compared visual-inertial framework ROVIO [23] could not extract any visual features. However, as seen in the height colored map these

dark corners have some chairs and boxes placed in them which provide geometric information that is utilized during the creation of our multi-modal features enabling us to perform mapping in such visually degraded conditions.

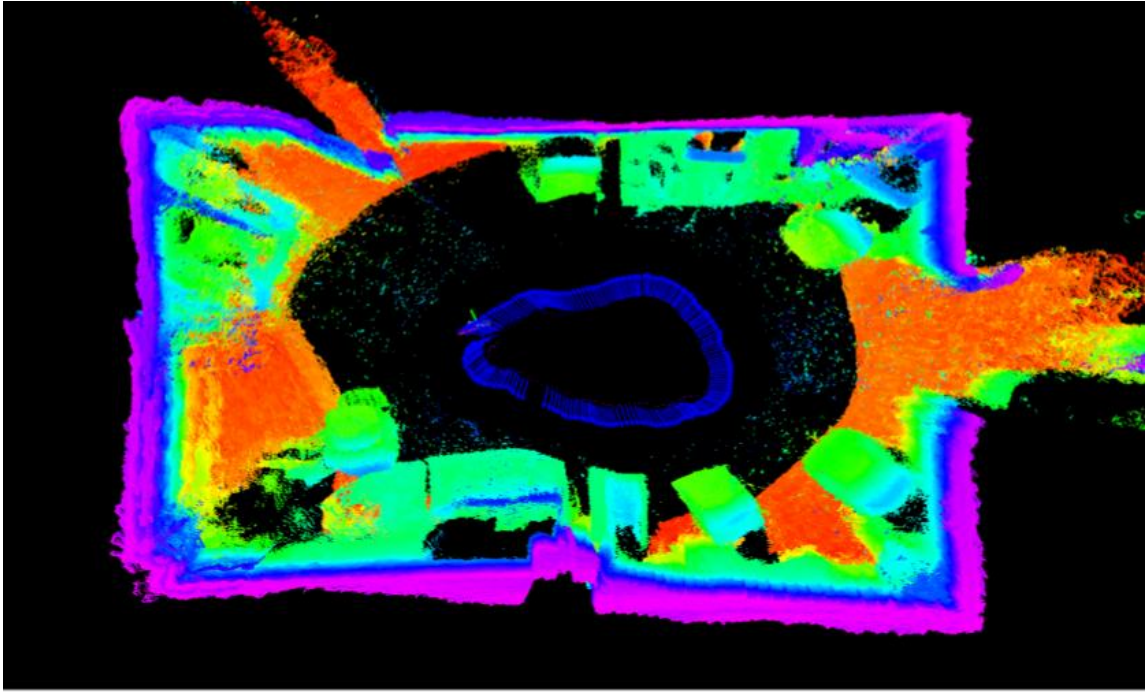


Figure 4.9: A top-down view of the 3D map created in visually degraded conditions using our proposed algorithm. The map is colored according to height values.

Figure: 4.10 and 4.11 show the maps of the top and bottom right corners of the room from another perspective to provide a better understanding of the operating visual conditions and the consistency of the created map.

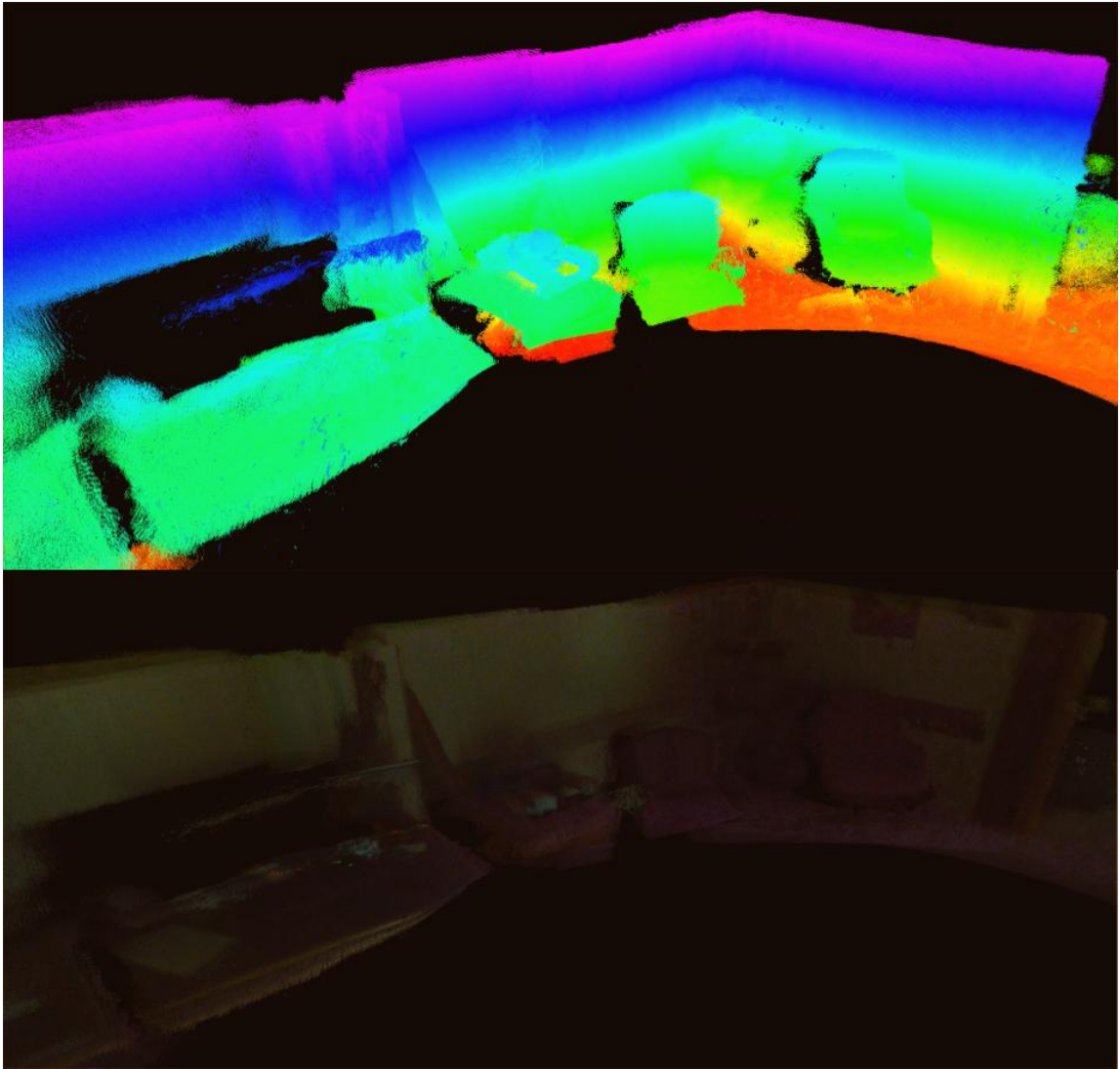


Figure 4.10: Front view of the RGB and height colored map of the top right corner of the room.

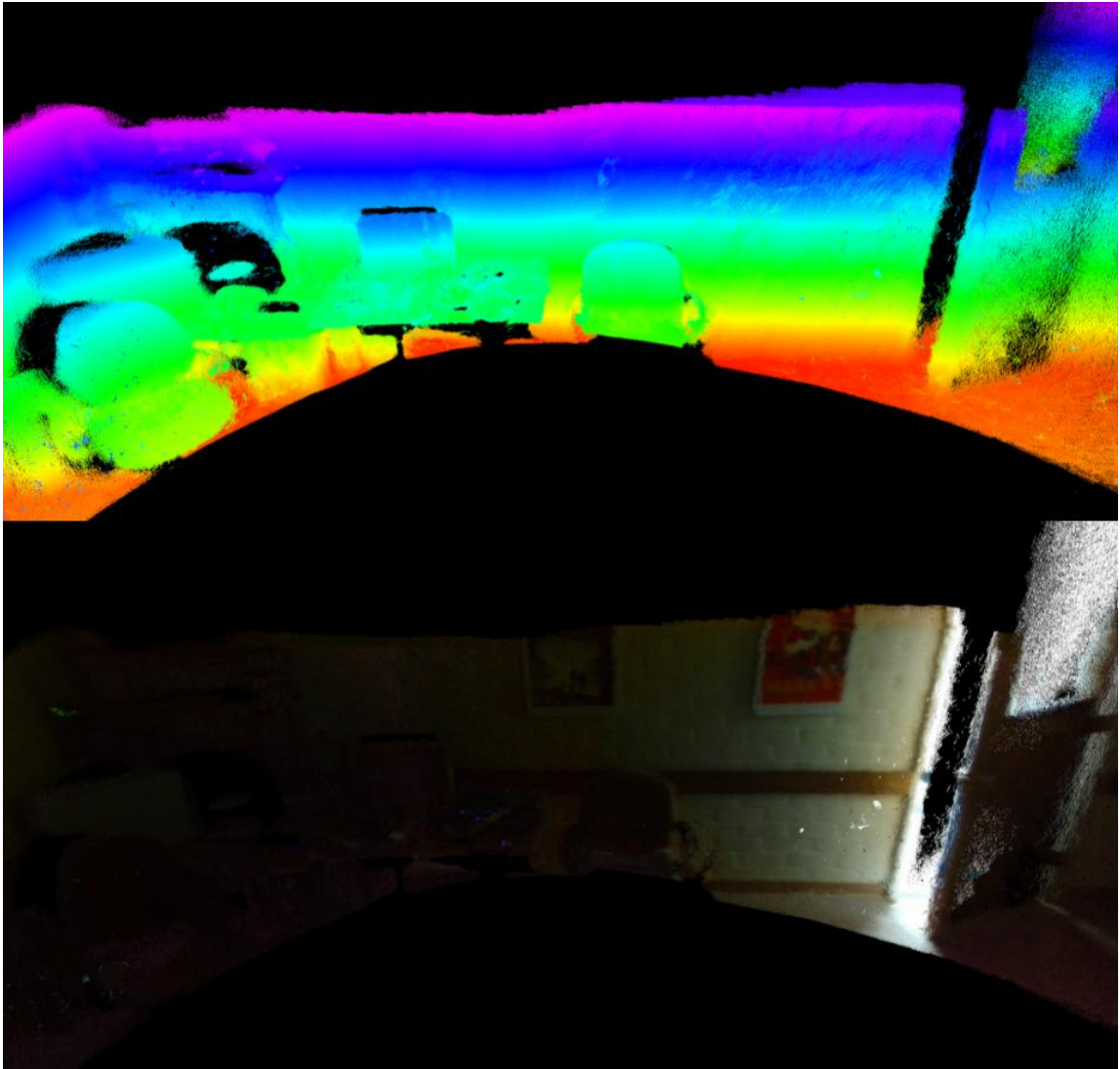


Figure 4.11: Front view of the RGB and height colored map of the bottom right corner of the room..

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis we presented an odometry estimation framework for localization and mapping applications in visually degraded environments with low-illumination and in texture-less conditions. Different from other works, our proposed approach fuses visual and depth information at the feature level enabling odometry estimation in low illumination and texture less environments. Multi-modal information is utilized during the feature generation and descriptor extraction level and is further fused with inertial information in an extended Kalman filter framework. Fusion with inertial information provides a motion prior and helps to reduce the search space for feature matching. This allows us to track features as part of our filter state and enables joint uncertainty estimation of pose and features while remaining computationally tractable. Localization performance was demonstrated by comparing the odometry estimates of our proposed approach to another visual-inertial framework and external ground truth provided by Vicon system. Operation in visually degraded conditions

was demonstrated by performing mapping of a room at night with the lights turned off.

5.2 Future Work

Future work may include reformulation of the filter to incorporate depth measurements as an innovation term in the update step. Currently, direct depth measurements are either incorporated in the filter when a new feature is added or if a feature is tracked over a period of time its depth estimate is updated without directly effecting the filter as measurement.

Comparison with current state of the art visual-inertial odometry framework revealed the impact and importance the sensing quality has on the odometry estimation process. In our tests we saw improper scaling of translation estimates and drift along Z-axis which can be primarily attributed to improper inertial measurements. In future we plan to use an external IMU to improve performance as well as introduce synchronization between between inertial measurements and images (visual and depth).

Better descriptor generation is also a potential area of improvement for future work. Currently, our descriptor simply combines information from both domains using an *OR* operation. This can make the descriptor noisy and less robust if information coming from one sensing modality is degraded. In our work we improved the visual part of the descriptor to be less prone to noise in low illumination conditions. In future we plan to work on the depth part of the descriptor to make it more information rich and repeatable.

A long term goal would be to introduce a long and short term loop closure back-end to improve odometry and mapping consistency.

Bibliography

- [1] C. Papachristos, S. Khattak, and K. Alexis, “Uncertainty-aware receding horizon exploration and mapping using aerial robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017. [Online]. Available: https://github.com/unr-arl/rhem_planner
- [2] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon path planning for 3d exploration and surface inspection,” *Autonomous Robots*, pp. 1–16, 2016.
- [3] A. Bircher, K. Alexis, M. Burri, P. Oettershagen, S. Omari, T. Mantel and R. Siegwart, “Structural inspection path planning via iterative viewpoint resampling with application to aerial robotics,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6423–6430. [Online]. Available: <https://github.com/ethz-asl/StructuralInspectionPlanner>
- [4] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova and R. Siegwart, “Receding horizon ”next-best-view” planner for 3d exploration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016. [Online]. Available: <https://github.com/ethz-asl/nbvplanner>

- [5] L. Yoder and S. Scherer, “Autonomous exploration for infrastructure modeling with a micro aerial vehicle,” in *Field and Service Robotics*. Springer, 2016, pp. 427–440.
- [6] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [7] Z. Fang, S. Yang, S. Jain, G. Dubey, S. Roth, S. Maeta, S. Nuske, Y. Zhang, and S. Scherer, “Robust autonomous flight in constrained and visually degraded shipboard environments,” *Journal of Field Robotics*, vol. 34, no. 1, pp. 25–52, 2017.
- [8] S. Montambault, J. Beaudry, K. Toussaint, and N. Pouliot, “On the application of vtol uavs to the inspection of power utility assets,” in *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on*. IEEE, 2010, pp. 1–7.
- [9] S. Khattak, C. Papachristos, and K. Alexis, “Change detection and object recognition using aerial robots,” in *International Symposium on Visual Computing*. Springer, 2016, pp. 582–592.
- [10] H. Balta, J. Bedkowski, S. Govindaraj, K. Majek, P. Musialik, D. Serrano, K. Alexis, R. Siegwart, and G. Cubber, “Integrated data management for a fleet of search-and-rescue robots,” *Journal of Field Robotics*, 2016.
- [11] S. M. Adams and C. J. Friedland, *A survey of unmanned aerial vehicle (UAV) usage for imagery collection in disaster research and management*. publisher not identified, 2011.
- [12] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grix, F. Ruess, M. Suppa, and D. Burschka, “Toward a fully autonomous uav: Re-

- search platform for indoor and outdoor urban search and rescue,” *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.
- [13] I. Maza, F. Caballero, J. Capitán, J. R. Martínez-de Dios, and A. Ollero, “Experimental results in multi-uav coordination for disaster management and civil security applications,” *Journal of intelligent & robotic systems*, vol. 61, no. 1, pp. 563–585, 2011.
- [14] C. Papachristos, S. Khattak, and K. Alexis, “Autonomous exploration of visually-degraded environments using aerial robots,” in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 775–780.
- [15] D. Scaramuzza and F. Fraundorfer, “Visual odometry: Part i: The first 30 years and fundamentals,” *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.
- [16] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part ii: Matching, robustness, optimization, and applications,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.
- [17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “Orb-slam: a versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [18] B. Kitt, A. Geiger, and H. Lategahn, “Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme,” in *Intelligent Vehicles Symposium (IV)*, 2010.
- [19] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

- [20] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.
- [21] A. J. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *null*. IEEE, 2003, p. 1403.
- [22] M. Li and A. I. Mourikis, “High-precision, consistent ekf-based visual-inertial odometry,” *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.
- [23] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust visual inertial odometry using a direct ekf-based approach,” in *Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 298–304.
- [24] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, “Towards fully autonomous driving: Systems and algorithms,” in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 163–168.
- [25] J. Zhang and S. Singh, “Loam: Lidar odometry and mapping in real-time,” in *Robotics: Science and Systems Conference*, Pittsburgh, PA, July 2014.
- [26] A. Censi, “On achievable accuracy for range-finder localization,” in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 4170–4175.
- [27] J. Zhang, M. Kaess, and S. Singh, “On degeneracy of optimization-based state estimation problems,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 809–816.

- [28] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for rgb-d cameras,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3748–3754.
- [29] M. Labbe and F. Michaud, “Appearance-based loop closure detection for online large-scale and long-term operation,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 734–745, 2013.
- [30] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual odometry and mapping for autonomous flight using an rgb-d camera,” in *Robotics Research*. Springer, 2017, pp. 235–252.
- [31] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, “3-d mapping with an rgb-d camera,” *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.
- [32] E. R. Nascimento, G. L. Oliveira, M. F. Campos, A. W. Vieira, and W. R. Schwartz, “Brand: A robust appearance and depth descriptor for rgb-d images,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1720–1726.
- [33] L. O. Vasconcelos, E. R. Nascimento, and M. F. Campos, “Kvd: Scale invariant keypoints by combining visual and depth data,” *Pattern Recognition Letters*, vol. 86, pp. 83–89, 2017.
- [34] K. Wu, X. Li, R. Ranasinghe, G. Dissanayake, and Y. Liu, “Risas: A novel rotation, illumination, scale invariant appearance and shape feature,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4008–4015.

- [35] Z. Chen, S. Czarnuch, A. Smith, and M. Shehata, “Performance evaluation of 3d keypoints and descriptors,” in *International Symposium on Visual Computing*. Springer, 2016, pp. 410–420.
- [36] L. A. Alexandre, “3d descriptors for object and category recognition: a comparative evaluation,” in *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*, vol. 1, no. 3, 2012, p. 7.
- [37] T. Tuytelaars, K. Mikolajczyk *et al.*, “Local invariant feature detectors: a survey,” *Foundations and trends[®] in computer graphics and vision*, vol. 3, no. 3, pp. 177–280, 2008.
- [38] Y. Li, S. Wang, Q. Tian, and X. Ding, “A survey of recent advances in visual feature detection,” *Neurocomputing*, vol. 149, pp. 736–751, 2015.
- [39] I. Sipiran and B. Bustos, “Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes,” *The Visual Computer*, vol. 27, no. 11, pp. 963–976, 2011.
- [40] M. Karpushin, G. Valenzise, and F. Dufaux, “Good features to track for rgb-d images,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017.
- [41] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, “Intel (r) realsense (tm) stereoscopic depth cameras,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1267–1276.
- [42] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab, “Adaptive neighborhood selection for real-time surface normal estimation from organized point

- cloud data using integral images,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2684–2689.
- [43] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571.
- [44] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” *Computer Vision–ECCV 2006*, pp. 430–443, 2006.
- [45] J. Levinson and S. Thrun, “Automatic online calibration of cameras and lasers.” in *Robotics: Science and Systems*, 2013, pp. 24–28.
- [46] F. Tombari, S. Salti, and L. Di Stefano, “A combined texture-shape descriptor for enhanced 3d feature matching,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 809–812.
- [47] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” *Computer Vision–ECCV 2010*, pp. 778–792, 2010.
- [48] R. Mur-Artal and J. D. Tardós, “Visual-inertial monocular slam with map reuse,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017.
- [49] M. Quan and S. Piao, “Robust visual-inertial slam: combination of ekf and optimization method,” *arXiv preprint arXiv:1706.03648*, 2017.
- [50] A. I. Mourikis and S. I. Roumeliotis, “A multi-state constraint kalman filter for vision-aided inertial navigation,” in *Robotics and automation, 2007 IEEE international conference on*. IEEE, 2007, pp. 3565–3572.

- [51] P. Tanskanen, T. Naegeli, M. Pollefeys, and O. Hilliges, “Semi-direct ekf-based monocular visual-inertial odometry,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 6073–6078.
- [52] X. Zheng, Z. Moratto, M. Li, and A. I. Mourikis, “Photometric patch-based visual-inertial odometry,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3264–3271.
- [53] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [54] I. Board, “Ieee standard specification format guide and test procedure for single-axis interferometric fiber optic gyros,” *IEEE Std*, pp. 952–1997, 1998.
- [55] P. Furgale, J. Rehder, and R. Siegwart, “Unified temporal and spatial calibration for multi-sensor systems,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1280–1286.
- [56] P. Furgale, J. Maye, J. Rehder, T. Schneider, and L. Oth, “Kalibr,” <https://github.com/ethz-asl/kalibr>, 2014.

List of Publications

C. Papachristos, S. Khattak, and K. Alexis, Uncertainty-aware receding horizon exploration and mapping using aerial robots, in IEEE International Conference on Robotics and Automation (ICRA), May 2017. [Online]. Available: https://github.com/unr-arl/rhem_planner

S. Khattak, C. Papachristos, and K. Alexis, Change detection and object recognition using aerial robots, in International Symposium on Visual Computing. Springer, 2016, pp. 582592

C. Papachristos, S. Khattak, and K. Alexis, Autonomous exploration of visually-degraded environments using aerial robots, in Unmanned Aircraft Systems (ICUAS), 2017 International Conference on. IEEE, 2017, pp. 775780.

F. Mascarich, T. Wilson, T. Dang, S. Khattak, C. Papachristos, and K. Alexis, Towards robotically supported decommissioning of nuclear sites, arXiv:1705.06401, 2017.

F. Mascarich, S. Khattak, C. Papachristos, and K. Alexis, A Multi-Modal Mapping Unit for Autonomous Exploration and Mapping of Underground Tunnels, Interna-

tional IEEE Aerospace Conference 2018. (*Accepted*)