2017

HETEROGENEOUS MULTI-SENSOR FUSION FOR 2D AND 3D POSE ESTIMATION

Hanieh Deilamsalehy
Michigan Technological University, hdeilams@mtu.edu

HETEROGENEOUS MULTI-SENSOR FUSION FOR 2D AND 3D POSE ESTIMATION

By

Hanieh Deilamsalehy

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Electrical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2017

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Electrical Engineering.

Department of Electrical and Computer Engineering

Dissertation Advisor:	*Dr. Timothy C. Havens*
Committee Member:	*Dr. Michael C. Roggemann*
Committee Member:	*Dr. Zhaohui Wang*
Committee Member:	*Dr. Thomas Oommen*
Department Chair:	*Prof. Daniel R. Fuhrmann*

Dedication

To my parents, teachers and friends. I would not be where I am, if it was not because of all the people in my life.

Contents

List of Figures

List of Tables

Preface

This dissertation contains my research performed during my Ph.D. program in Electrical Engineering at Michigan Technological University. The focus of the work is sensor fusion and its applications in the real world. In this regard, first a two dimensional image based scenario from railroad industry was considered and then a more complicated three dimensional sensor fusion algorithm was developed for *Unmanned Aerial Vehicle* (UAV) applications.

Chapter 1 of this dissertation gives an introduction to sensor fusion and the problems to be solved using fusion techniques, followed by Chapter 2 which contains my research work for the railroad industry and is published in SAGE Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit as *An Automatic Method for Detecting Sliding Railway Wheels and Hot Bearings Using Thermal Imagery*. In addition, parts of my work in this area were published and presented in Joint Rail Conferences 2015-2017 as *Automatic method for detecting and categorizing train car wheel and bearing defects*, *Detection of sliding wheels and hot bearings using wayside thermal cameras* and *Sensor Fusion of Wayside Visible and Thermal Imagery for Rail Car Wheel and Bearing Damage Detection*. This work was done under supervision of my advisor Dr. Timothy C. Havens, as well as Dr. Pasi Lautala. The algorithm was designed and developed by myself and the wheel thermal flow model

was simulated by Dr. Ezequiel Medici and Mr. James Davis.

Chapter 3 presents my research work for UAV pose estimation, which was previously published in the ASME Journal of Dynamic Systems, Measurement, and Control as *Heterogeneous Multi-Sensor Fusion for Mobile Platform 3D Pose Estimation* and IEEE SENSORS as *Sensor fused three-dimensional localization using IMU, camera and LiDAR*. The original idea of this work was proposed by my advisor Dr. Timothy C. Havens and design and implementation of the algorithm was done by myself. Data collection was done by Mr. Joshua Manela and myself.

Chapter 4 is an extension of the pose estimation method in Chapter 3. This chapter introduces a novel method to optimize and improve the accuracy of the UAV pose estimation. This work is submitted to the Elsevier journal of Robotics and Autonomous System as *Fuzzy Adaptive Extended Kalman Filter for Robot 3D Pose Estimation* and currently is under review. The research topic was identified by me and as the first author I designed and implemented the algorithm under supervision of Dr. Timothy C. Havens.

Chapter 5 summarizes and concludes my Ph.D. program research work and proposes potential future work in this area.

Acknowledgments

First and foremost I have to acknowledge my appreciation for all the guidance, inspiration and support of my advisor Dr. Timothy C. Havens. It has been an honor to have a brilliant mentor like him.

Special thanks to Dr. Pasi Lautala who patiently prepared guidance and suggestions throughout my academic path. I also acknowledge Prof. Daniel R. Fuhrmann for all his support specially at the beginning of my Ph.D. program.

And finally I would like to thank all my friends and family who have helped me learn, understand and grow during all these years.

The work presented in Chapter 2 was supported by a National University Rail (NURail) center, a US DOT-OST tier and University Transportation center [grant number DTRT12-G-UTC18, 2012]. The material presented in Chapters 3 and 4 is based upon work supported by the Michigan Department of Transportation (MDOT), grant #2016-0067. Superior, a high performance computing cluster at Michigan Technological University, was used in obtaining some of the results presented in this publication.

List of Abbreviations

UPRR	Union Pacific Railroad
SLAM	Simultaneous Localization and Mapping
IMU	Inertial Measurement Unit
GPS	Global Positioning System
LiDAR	Light Detection And Ranging
UAV	Unmanned Aerial Vehicle
EKF	Extended Kalman Filter
DOF	Degree Of Freedom
INS	Inertial Navigation System

Abstract

Sensor fusion is a process in which data from different sensors is combined to acquire an output that cannot be obtained from individual sensors. This dissertation first considers a 2D image level real world problem from rail industry and proposes a novel solution using sensor fusion, then proceeds further to the more complicated 3D problem of multi sensor fusion for UAV pose estimation.

One of the most important safety-related tasks in the rail industry is an early detection of defective rolling stock components. Railway wheels and wheel bearings are two components prone to damage due to their interactions with the brakes and railway track, which makes them a high priority when rail industry investigates improvements to current detection processes. The main contribution of this dissertation in this area is development of a computer vision method for automatically detecting the defective wheels that can potentially become a replacement for the current manual inspection procedure. The algorithm fuses images taken by wayside thermal and vision cameras and uses the outcome for the wheel defect detection. As a byproduct, the process will also include a method for detecting hot bearings from the same images. We evaluate our algorithm using simulated and real data images from UPRR in North America and it will be shown in this dissertation that using sensor fusion techniques the accuracy of the malfunction detection can be improved.

After the 2D application, the more complicated 3D application is addressed. Precise, robust and consistent localization is an important subject in many areas of science such as vision-based control, path planning, and SLAM. Each of different sensors employed to estimate the pose have their strengths and weaknesses. Sensor fusion is a known approach that combines the data measured by different sensors to achieve a more accurate or complete pose estimation and to cope with sensor outages. In this dissertation, a new approach to 3D pose estimation for a UAV in an unknown GPS-denied environment is presented. The proposed algorithm fuses the data from an IMU, a camera, and a 2D LiDAR to achieve accurate localization. Among the employed sensors, LiDAR has not received proper attention in the past; mostly because a 2D LiDAR can only provide pose estimation in its scanning plane and thus it cannot obtain full pose estimation in a 3D environment. A novel method is introduced in this research that enables us to employ a 2D LiDAR to improve the full 3D pose estimation accuracy acquired from an IMU and a camera. To the best of our knowledge 2D LiDAR has never been employed for 3D localization without a prior map and it is shown in this dissertation that our method can significantly improve the precision of the localization algorithm. The proposed approach is evaluated and justified by simulation and real world experiments.

Chapter 1

Introduction

In today's world, a variety of data are provided by different sources. Sensors are widely used to collect all kinds of information from the environment. The application of sensors varies from common home appliances utilization, cameras and cell phones to medical, industrial and military applications. Nevertheless, in numerous applications, the information acquired only by one single sensor is not complete, accurate or robust enough for be relied on. Sensitive decisions might be taken based on the sensory data and there is a significant need to introduce methods to overcome their flaws. Sensor manufacturing technology advances every day and sensors become more and more advanced and accurate, however, for variety of applications one single sensor is yet not enough. Hence, a very common solution to overcome this problem and obtain the desired result is to employ several sensors instead of one and use *sensor fusion*

techniques to generate more reliable results.

Sensor fusion is the process of combining the information obtained by different sensors to compute an output better than what could be obtained by every single sensor. The fusion can be done at different levels: raw data, feature level or decision level. The choice of the method depends on the application, computational power, etc.

The complexity of the fusion algorithm depends on the dimensionality of the data. As the dimensionality of the data increases, the level of the complexity of the algorithm increases accordingly. The desired output can be as simple as a scalar such as a voltage or it might be a high dimensional vector. This dissertation first focuses on a two dimensional image fusion case, which then is followed by a complicated challenging three dimensional scenario of UAV pose estimation. In the rest of this chapter, the two cases are briefly introduced and then in the following chapters they will be discussed in detail.

1.1 Two dimensional sensor fusion for image processing

The demand for better utilization of rail network capacity is increasing every day. Faster, more reliable and more efficient inspection of railroad rolling stock can help

increase both its utilization and reliability. Wheels form the contact point between rolling stock and track and as such are critical elements in transferring tons of loads from the train to the track. Any defect in the wheels can rapidly grow to critical size. Especially since many defects such as flat spots worsen gradually, early detection is the best way to prevent potentially disastrous outcomes. These all describe why the industry desire for automatic defect detection is growing so fast.

The trend to employ wayside monitoring systems and automatic defect detection is increasing. Inspection equipment such as WIMs (Wheel Impact Monitors), wheel profile detectors, HBDs (Hot Box Detectors) and acoustic detection systems [7] are installed on or next to the track and as the train passes they collect data and provide warnings if anomalies exceeding tolerances are found. Among these inspection devices are thermal and visible spectrum cameras. In our papers [22, 23] we introduced methods that could automatically detect the wheel and bearing defects from the thermal imagery. The only information existing in the thermal imagery is the pixel temperature and, therefore, defects that have distinctive thermal patterns can be reliably detected. Nevertheless, the lack of any structural or color data results in poor segmentation outcome. On the other hand, visible spectrum cameras can provide texture and structural information as well as color which can be very helpful in image segmentation. Thus, while some types of defects are visible to inspectors in the thermal imagery, having the visible spectrum imagery at the same time can help make the defect defection procedure automatic by contributing to the wheel and

bearing detection. This is discussed later in this dissertation.

Even though vision camera images provide us with a variety of information, they have several limitations. Unlike thermal cameras, visual spectrum cameras do not provide any information about temperature. Also in certain conditions, such as night time when there is poor lighting, limited visual information can be obtained. This concludes that if there is a method to combine thermal and visual spectrum data, more informative images will be acquired that can be used effectively for wheel and bearing defect detection algorithms. Each of the thermal and vision cameras sensors has its strength and weaknesses and image fusion can be beneficial to take advantage of both cameras capabilities and compensate for their deficiencies.

In the work presented in this dissertation an important step of the wheel and bearing defect detection method is to automatically detect and segment the wheel and bearing. The error in this step is a big source of failure in the defect detection [22, 23]. In this work, the goal is to implement a wheel and bearing defect detection from the thermal imagery and improve the method's accuracy by fusing the thermal and vision camera images and employing the fused image as the input of the wheel and bearing detection algorithm.

There are limited examples of fusing thermal and vision camera images in the past for applications such as person detection [83]. However, to the best of our knowledge, this is the first time that train wheel and bearing defects are detected by taking advantage

of fusion methods. We also believe that similar information fusion techniques could be considered for other sensor-based applications in the rail industry.

In Chapter 2 we first discuss our automatic detection and segmentation method solely based on thermal imagery. This algorithm identifies the wheel and bearing portion of the image. Then we develop a method, using *Histogram of Oriented Gradients* to extract features of these regions. These feature descriptors are later employed by a *Support Vector Machine* to build a fast classifier with a good detection rate, which can detect abnormalities in the wheel. At the end, we train the algorithm using simulated images of sliding wheels and test it on several thermal images collected in a revenue service by Union Pacific Railroad (UPRR) in North America. Since the wheel and bearing segmentation is identified as a significant source of error, a sensor fusion based algorithm is proposed and we show that the algorithm can be improved by fusing thermal and visual imagery.

1.2 Three dimensional sensor fusion for UAV pose estimation

Precise, robust and consistent localization of a moving platform such as a robot is a necessary task for many applications, e.g., robot navigation control, environment mapping, SLAM, autonomous vehicle and even medical applications such as robotic

surgery. Pose estimation is usually done by collecting the data obtained from several sensors such as IMU, GPS, and camera mounted on the object/platform and fusing the acquired information. Each of these sensors has their strengths and weaknesses. Sensor fusion is a known approach that combines the data measured by different sensors to achieve a more accurate or complete pose estimation and to cope with sensor outages.

In this dissertation, a 3D pose estimation algorithm is presented for a UAV in an unknown GPS-denied environment. Assuming that the robot is moving in a three-dimensional (3D) world, its location is fully defined by six degrees-of-freedom (6DOF): three Euler angles and three position coordinates. Some 3D sensors, such as IMUs and cameras, have been widely used for 3D localization. Yet, there are other sensors, like 2D LiDARs, which can give a very precise estimation in a 2D plane but they have not been employed for 3D estimation since the sensor is unable to obtain the full 6DOF. In Chapter 3 of this dissertation an algorithm is proposed that fuses the data from an IMU, a camera, and a 2D LiDAR using EKF to achieve accurate localization in the unknown environment. No prior map of the environment is provided and the moving platform will estimate its position only by using the perception obtained from the sensory data.

Among the employed sensors, LiDAR has not received proper attention in the past; this is mostly because a 2D LiDAR can only provide pose estimation in its scanning

plane and thus it cannot obtain a full pose estimation in a 3D environment. In this work, a novel method is introduced that employs a 2D LiDAR to improve the full 3D pose estimation accuracy acquired from an IMU and a camera, and it is shown that this method can significantly improve the precision of the localization algorithm. The proposed approach is evaluated and justified by simulation and real world experiments.

Despite the 3D trajectory of the robot considered in this work, there can be a considerable amount of the time in which the robot almost moves on a plane during the time interval between the two sensor readings; e.g., a ground vehicle moving on a flat surface or a drone flying at high altitude to collect visual data. This was the motivation for Chapter 4 of this dissertation, which proposes a novel method using a *fuzzy inference system* to further improve the estimated pose obtained in Chapter 3. The method determines the trajectory of the robot and the sensor reliability during the time interval between two readings, and based on this information defines the weight of the 2D LiDAR sensor in the final fused pose by adjusting *Extended Kalman Filter* parameters. Simulation and real world experiments show that the pose estimation error can be significantly decreased using this proposed method.

1.3 Publications

The work in this dissertation is based on the following publications.

† Deilamsalehy H, Havens TC, Manela J. *Heterogeneous multi-Sensor fusion for mobile platform 3D pose estimation.* Journal of Dynamic Systems, Measurement, and Control. ASME 2016.

† Deilamsalehy H, Havens TC. *Sensor fused three-dimensional localization using IMU, camera and LiDAR.* In SENSORS, 2016 IEEE 2016 Oct 30 (pp. 1-3). IEEE.

† Deilamsalehy H, Havens TC, Lautala P, Medici E, Davis J. *An automatic method for detecting sliding railway wheels and hot bearings using thermal imagery.* Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit. 2016 Mar 22:0954409716638703.

† Deilamsalehy H, Havens TC, Lautala P. *Sensor fusion of wayside visible and thermal imagery for rail car wheel and bearing damage detection.* In 2017 joint rail conference 2016 Apr 18. American Society of Mechanical Engineers.

† Deilamsalehy H, Havens TC, Lautala P. *Detection of sliding wheels and hot bearings using wayside thermal cameras.* In 2016 joint rail conference 2016

Apr 12 (pp. V001T02A002-V001T02A002). American Society of Mechanical Engineers.

† Deilamsalehy H, Havens TC, Lautala P. *Automatic method for detecting and categorizing train car wheel and bearing defects.* In 2015 joint rail conference 2015 Mar 23 (pp. V001T02A007-V001T02A007). American Society of Mechanical Engineers.

† Deilamsalehy H, Havens TC. *Fuzzy Adaptive Extended Kalman Filter for Robot 3D Pose Estimation.* In review, Journal of Robotics and Autonomous Systems. ELSEVIER 2017.

Chapter 2

Detection of Train Car Sliding Wheels and Hot Bearings Using Wayside Thermal and Visual Cameras

2.1 Introduction

As the demand for rail transportation capacity and speed grows, there is an increasing trend toward higher productivity and efficiency in the rail industry. Both railroad

track and rolling stock have rigorous inspection requirements to maintain the safety of the network. Even then, 84% of all the rolling stock-related accidents in 23 countries are confirmed to be caused by wheel set or bogie defects [74]. According to Liu et al. [49], bearing failure and broken wheels are the third and fourth major cause of freight train derailments on main tracks in North America. The authors also note that these derailment causes are most prevalent at speeds above 40 kmh (25 mph), exposing the rail industry to extensive damages from each occurrence.

Current inspection methods of rolling stock components include both automated and visual systems, but the industry is increasingly moving toward automatic detector and performance based rolling stock maintenance to improve efficiency and to reduce costs and reliance on human interpretation. Wayside monitoring systems are most commonly used for automated rolling stock inspection processes [61]. Inspection equipment are installed at fixed locations in or next to the track, where the train passes over the section. As the train rolls through the inspection station, different inspection sensors collect the information on possible defects like hot bearings, hot wheels, dragging equipment, and high, wide, or shifted loads. After inspecting a train, modern wayside detectors will automatically report their findings by radio or wireless connections [53].

Many defects such as flat spots worsen gradually. Therefore, a fast and early detection of these defects can prevent further and more serious damage. Flat spots occur mainly

as a result of violent braking, causing the wheel to lock up and slide on the rail while the train is still moving, making detection process during such a braking event ideal. There are also other less common reasons for a flat spot to occur, such as poorly adjusted, frozen, or defective brakes and also contamination on the rail such as grease, leaves, snow, and ice [39]. A flat spot induces great forces on both the rolling stock and the track and can lead to fatigue damage and failure of various vehicle and track components such as wheel sets, bearings, and rail ties [74]. In some cases, the damage can even cause later derailment which is a serious concern for the rail industry. The wheel-track interaction has been studied in several papers [3, 37, 43, 52] which describe the importance of the effect of a defective wheel on the track. In addition to the safety considerations, this flat spots cause an unpleasant noise, reducing passenger comfort and disturbing people in adjacent properties. Thus, an online automatic wheel defect detection system, which monitors the wheel condition and detects sliding wheels at an early stage, can help the maintenance to be scheduled more proactively, improving safety and reducing operational disruptions. We will discuss some of the existing methods to detect these type of defects and then move on to our proposed method.

2.2 Hypothesis and Research Methodology

The process of the railway wheel sliding on the rail heats up the steel wheel, which can be observed and potentially detected in thermal images. We hypothesize that

the heat pattern generated from a sliding wheel can be automatically detected in the thermal imagery, making it a noteworthy technology for sliding wheel detection. In this work, we will identify the heat pattern to detect hot spots/sliding wheels and also to detect the hot bearing, another common defect of wheel assembly. The basic procedure for our proposed automatic wheel defect monitoring method is as follows:

1. Acquire a labeled data set with thermal images of defective and normal train wheels.
2. Partition the available data set into a training set to train the proposed algorithm, and a test set to evaluate its operation.
3. Segment the wheel part of all the images in both training and test sets.
4. Extract the wheel features of both training and test sets.
5. Train the wheel classifier using feature descriptors extracted from the training set.
6. Evaluate resulting classifier on the test data set.

Furthermore, the procedure for automatic hot bearing detection method is as follows:

1. Segment the bearing part of the thermal images.
2. Calculate the mean intensity/temperature of the bearings.

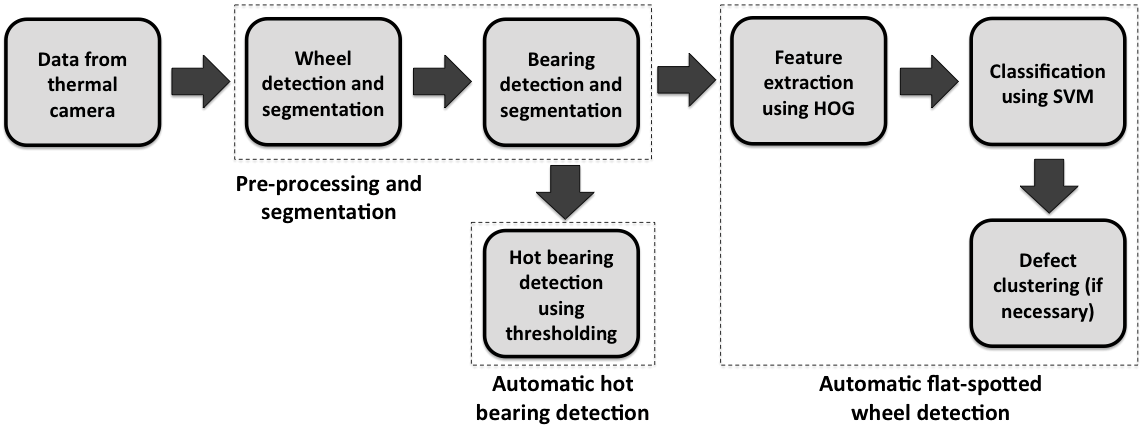


Figure 2.1: Automatic wheel and bearing defect detection block diagram

3. Detect hot bearings based on a temperature threshold.

The rest of this chapter is organized as follows. We first briefly review the previous work in this field and then will go through our proposed algorithm steps in details (Fig. 2.1). Each phase will be described in the following sections. We then complete the chapter by demonstrating our proposed automated monitoring method on a collection of simulated images and real thermal imagery taken on several trains on a Union Pacific Railroad (UPPR)[1].

2.3 Previous Work

Previous work that focuses on automatic wheel defect detection includes different techniques, such as acoustic, optical, thermal, and laser based detection technologies

[1]A North American class 1 freight railroad

[8].

One way to detect flat spots and hot bearings is using sound-based (acoustic) detection. This method is based on the fact that defective wheels and bearings produce vibration [8, 20]. The work of Papaelias et al. [61] is an example of this method, in which the authors implement an integration of acoustic emission and vibration analysis for on-board evaluation of bearings and wheels. The challenge with this method is that the measured acoustic signal might contain the surrounding noise, limiting the method accuracy or increasing implementation cost to remove the noise.

Another approach uses vision cameras that are installed as a wayside monitoring system, but wheel failures are not always visually detectable. This makes visual images by themselves insufficient for detecting certain wheel defects. According to Asplund et al. [6], only half of all wheels with a force peak over 400kN are visually seen as damaged wheels; the American Association of Railroads (AAR) recommends that a wheel with peak impact force of 222.41-226.89kN due to a single flat, should be replaced [74]. Visual cameras can be useful for detecting other types of defects which are visible in the images. Li et al., employed vision cameras for automatic wheel bearing bolt defect detection [47]. In contrast to the wheel surface which is our study in this dissertation, bolts are visible in the vision camera image and thus the necessary information can be acquired using the camera.

Measurement of wheel profile can also be helpful for detecting abnormalities. Wheel

profile detectors employ digital images to determine the profile of the tread while in service and then compare it to the standard profile [8]. Examples of a wheel profile case study using laser scanning are presented by Asplund et al. [5, 6]. In these papers, the authors study the Wheel Profile Monitoring System (WPMS) on a track section in Sweden in order to detect wheel-related failures.

An alternative well studied procedure to detect a wheel flat is to measure the dynamic force or acceleration of the track under the wheel [13]. This method is known as Wild Impact Load Detector (WILD) and can be used as a predictive and proactive maintenance system [58]. In a related work [74], UI Alam et al. went one step further, studying the impact of multiple wheel flats by measuring the acceleration of the wheel set.

One of the most common methods to detect hot bearings is employing hot box detectors [8]. Hot box detectors work based on the principle that an axle bearing will emit a large amount of heat when it is close to failing.

While there is no question that the currently available methods have been successful in identifying sliding wheel/flat spot and bearing defects, the continuing interest by the industry toward alternative technologies reveals that there are still opportunities for development. For example, one of the challenges with acoustic and hot box detectors in identifying sliding wheels is their incapability to reliably detect uneven temperature distributions within the wheel. This has created an interest among

industry to investigate the use of thermal cameras as an alternative solution for detection. Hence, in this dissertation we explore how the generated heat pattern from a sliding wheel can be automatically detected using a thermal camera. We also demonstrate how the same camera can be used simultaneously to identify the hot bearings. Our results show that this method has much promise for effective detection of these type of defects.

2.4 Thermal Image Segmentation (Pre-Processing)

A few samples of the thermal images we are working with are shown in Fig. 2.2. Furthermore, several examples of sliding wheels are demonstrated in Fig. 2.3. Comparing the images of normal wheels and the defective ones show that sliding wheels possess a distinctive heat pattern at the wheel-track contact point. In the following sections, we will explain our proposed algorithm for detecting this heat pattern in the damaged wheels.

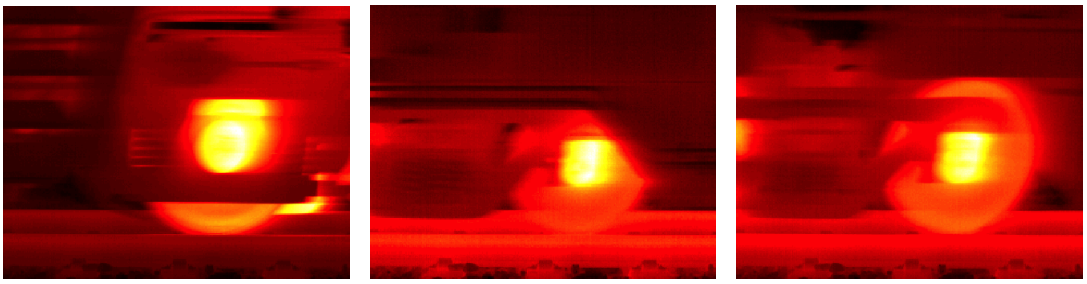


Figure 2.2: Examples of normal train wheel thermal images taken by wayside thermal imaging system

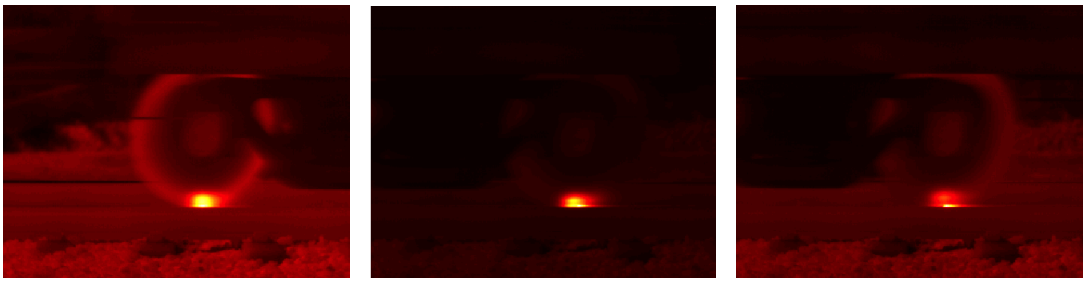


Figure 2.3: Examples of anomalous train wheel thermal images taken by wayside thermal imaging system

2.4.1 Automatic wheel detection and segmentation

The first step of algorithm is to segment the wheel portion of the image from the suspension hardware in the thermal image. As one can see in Fig. 2.2 and 2.3 , in addition to the wheel, the image may contain hardware components of the train as well as the track, which play no role in our investigation process and might interfere with the sliding wheel detection algorithm. As these images show, the wheel can be partially to almost fully occluded by suspension hardware. Hence, our algorithm must

be flexible and effective at detecting the wheel portion of the image automatically. If the thermal image has been captured while the wheel was sliding, this signifies the hot spot is located at the contact point of the wheel and the track and this part is always visible in the images. It is also possible that the wheel has rotated after sliding, and the hot spot is somewhere else along the wheel. It might even be occluded by the suspension hardware in the thermal image. Nevertheless, as long as the hot spot is visible in the image, it can potentially be detected. In order to locate this hot area in the image, first we need to recognize the wheel in the thermal image.

The train wheel is originally in the shape of a circle, but because of the combined effect of the motion of the train and the rolling shutter of the imager, there is skewness in the shape of the wheel and the wheel appears as an ellipse in the image. To automatically detect the elliptical portion of the image associated with the wheel, we employ the Hough Transform (HT) [70]. The HT is a feature extraction method that is widely used in image processing. Originally, this method was used to extract lines in an image, but it can be extended to extract more complicated and arbitrary shapes, e.g., circles or ellipses. In this work, we will use the extended version of the transform to detect the elliptical wheel in thermal imagery. The parametric equation of an ellipse is

$$x^2 + p_1xy + p_2y^2 + p_3x + p_4y + p_5 = 0, \tag{2.1}$$

where p_1, p_2, p_3, p_4 and p_5 determine the shape and the location of the ellipse. Since

the ellipse is defined by five parameters according to (2.1), by identifying five independent points on the ellipse, we can calculate the parameters. Following the HT method, first five independent points in the image are selected, and the ellipse consisting of these points is found. This process is done for all the points/pixels in the thermal image. Then, every ellipse receives votes from all the points located on its perimeter. The summation of the acquired votes for each ellipse determines how strong this ellipse is. In other words, the ellipse in the image that has the most points located on its perimeter will be chosen. In this approach, we are actually mapping the xy coordinates to a five dimensional space, using a five dimensional accumulator for the HT. If we choose to do an exhaustive search, the order of complexity employing this method will be $O(n^5)$. Therefore, finding an ellipse using the HT will be expensive in terms of memory usage and computation time. To overcome the complexity problem, we use a Canny edge detector [14] to reduce the number of pixels investigated, reducing the processing cost. We first apply the Canny edge detector to the wheel thermal image to obtain a binary image consisting of only edges; then, we use the HT to detect the train wheel in the image. To speed up the process even more, we use the Randomized Hough Transform (RHT) [9]. In RHT, Hough Transform is applied to a random sub-sample of all the image pixels, instead of all possible pixels. Assuming that there are enough wheel edge pixels in the binary image, the HT still should be able to consistently detect the wheel part.

There are several ways to mathematically define an ellipse. As mentioned before, five

independent parameters should be determined. In this chapter, we define the ellipse by the following parameters:

† Center of ellipse (x and y coordinates),

† Orientation of ellipse which is defined as its angle with x axis,

† Length of major axis of ellipse,

† Length of minor axis of ellipse.

We follow the method introduced by Xie et al. [77] to detect the wheel. For each pair of image pixels (x_1, y_1) and (x_2, y_2), we assume that these points are two vertices on the major axis of an ellipse, then we can calculate the following parameters for the ellipse:

$$x_0 = (x_i + x_j)/2, \tag{2.2a}$$

$$y_0 = (y_i + y_j)/2, \tag{2.2b}$$

$$a = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}/2, \tag{2.2c}$$

$$\alpha = \text{atan}[(y_j - y_i)/(x_j - x_i)]. \tag{2.2d}$$

In the above equations, (x_0, y_0) is the center of the ellipse, a is half length of the major axis, and α is the orientation. As Fig. 2.4 illustrates, the minor axis can be

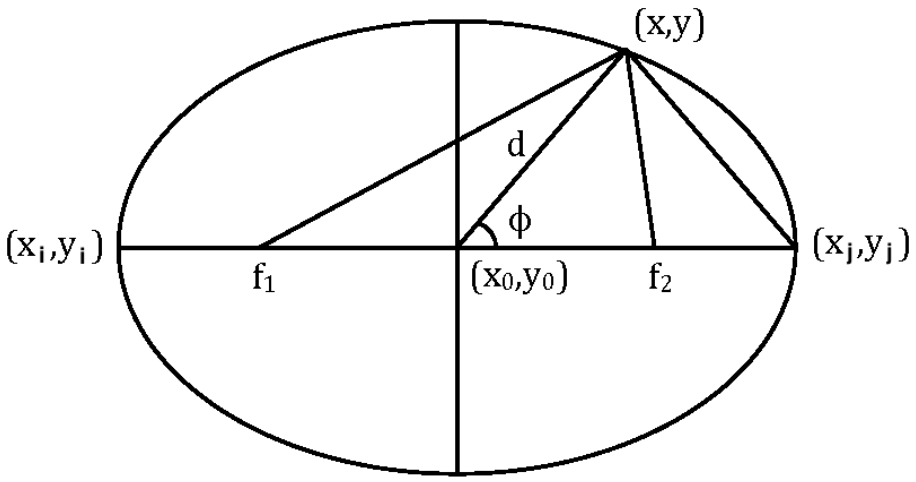
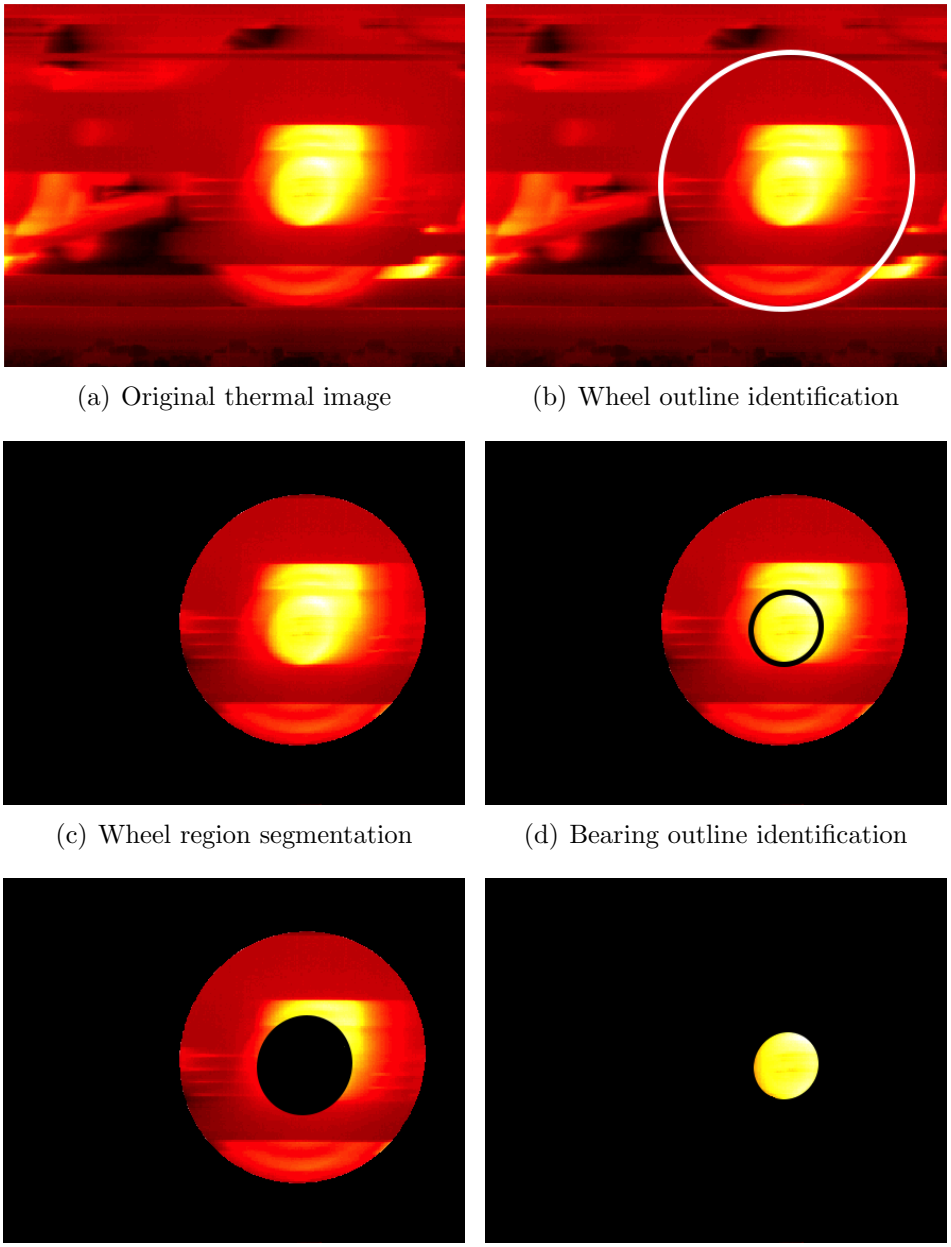


Figure 2.4: Ellipse geometry

calculated as

$$b = \sqrt{(a^2 d^2 (\sin(\phi))^2)/(a^2 - d^2(\cos(\phi))^2)}, \tag{2.3}$$

where b is half length of the minor axis, $\cos(\phi) = (a^2 + d^2 - f^2)/(2ad)$, and d is the distance between the point (x, y) on the ellipse and the center of ellipse. Accordingly, if we have (x_1, y_1) and (x_2, y_2), we can calculate all the ellipse parameters except for the minor axis. Therefore, we use the HT to vote on the half length of the minor axis. This way, a one-dimensional accumulator is enough to detect the ellipse and we output the parameters for the best ellipse found in the thermal image [9], i.e., the ellipse with the maximum number of votes. This procedure of finding the elliptical wheel and subsequent bearing detection and extraction is illustrated in Fig. 2.5.



(a) Original thermal image

(b) Wheel outline identification

(c) Wheel region segmentation

(d) Bearing outline identification

(e) Extracted wheel region

(f) Extracted bearing region

Figure 2.5: Image segmentation of wheel and bearing regions

2.4.2 Automatic Bearing Detection and Segmentation

In addition to abnormal heat pattern generated by the sliding wheel, a hot bearing may also causes elevated heat pattern in the images. To differentiate between these two causes, it is necessary to separate the bearing portion from the overall image, enabling the remainder of the image to be used for sliding wheel identification. Once the bearing portion has been identified and separated, its heat pattern can also be used at an indication of potential faulty (hot) bearing.

Similar to a wheel, because of the motion effect in the image, the bearing is also seen as an ellipse in the image. Therefore, to find the bearing, we apply the HT to the extracted wheel part and detect the bearing as a separate part. The procedure is analogous to wheel detection, explained in the previous section. As a consequence of train movement, there is a motion blur effect which can be observed primarily at the edges. Because of this blurriness, for the purpose of wheel heat pattern analysis, the bearing should be removed from the wheel image with a safety margin, which is bigger than the detected bearing (see Fig. 2.5). On the other hand, for the hot bearing detection algorithm, this safety margin is unnecessary and in fact the bearing should not include any wheel part. Therefore, in order to detect hot bearing, we have to segment the detected bearing without any margin.

2.5 Automatic sliding wheel detection

2.5.1 Feature extraction using Histogram of Oriented Gradients (HOG)

After wheel and bearing extraction, the corresponding segmented parts of the thermal image are ready for further analysis. The first step after pre-processing is to extract image features with which we can train our algorithm. Image features must be chosen in a way that a defective wheel can be distinguished from a normal wheel. Our image set is a collection of thermal camera imagery, which does not contain any distinguishable texture. This means the only information can be obtained from the images is pixels temperature. In thermal imagery, the temperature is represented as pixel intensities. Hence, we need to identify local heat pattern of the sliding wheels, which in our data set can be observed at the wheel-track contact point. Fig.2.3 illustrates this pattern on the sliding wheel.

We have to employ feature descriptors that can capture the hot spot pattern. In order to do this task, we use the Histogram of Oriented Gradients (HOG) feature. HOG is a feature descriptor which is widely used in computer vision and image processing to detect object in imagery. HOG decomposes an image into square cells of a given

size, then it counts occurrences of intensity gradient orientation in localized portions of the image [68]. It has been widely accepted as one of the best features to capture local shape information about objects in imagery.

The essential thought behind the HOG descriptors is that local pattern within an image can be described by the distribution of intensity gradients or edge directions. The implementation of these descriptors can be achieved by dividing the image into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within each cell. The HOG descriptor operates on localized cells; therefore is has the advantage that upholds invariance to geometric and photometric transformations, except for object orientation which is not an issue in our work. To obtain HOG feature descriptor, there are four main steps which should be taken:

1. Gradient computation: Calculate the gradient values.

2. Orientation binning: Create the cell histograms.

3. Descriptor blocks: Group the cells together into larger blocks.

4. Block normalization: Normalize the gradient strengths.

The implementation of HOG is illustrated in Fig. 2.6. As it can be seen in the figure, by applying HOG to an image, we will obtain a histogram of its intensity gradient

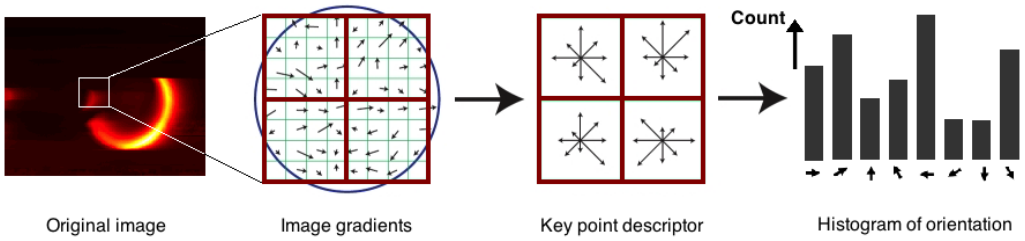


Figure 2.6: Histogram of Oriented Gradients (HOG) algorithm

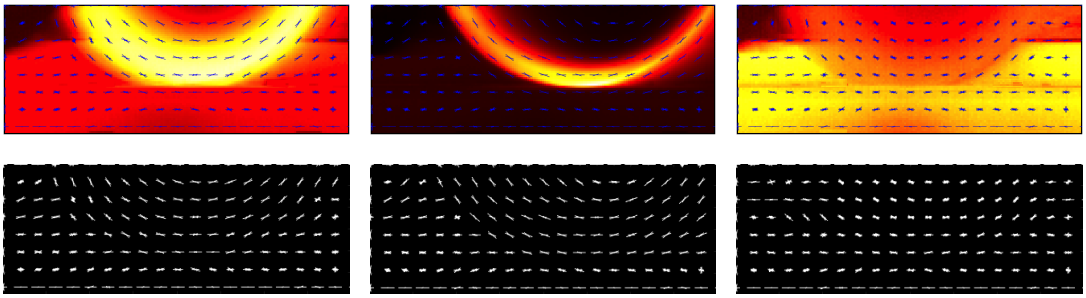


Figure 2.7: Visualization of HOG feature descriptors of normal train wheels. The top row visualizes HOG features on the wheel thermal images and the bottom row is HOG visualizations of the same thermal images without the wheel on the background

orientations.

In this research work, first we segment a window in the thermal image, which includes the wheel. Since we detected the wheel portion in the pre-processing stage, we know all the parameters needed to locate the wheel in the image. Thus, the position of the window can be chosen in a way that contains the wheel. The original image size is 240×320 pixels and the area of interest is divided into 8×8 pixels regions subsequently. Each of these pixel regions is called a cell. The smaller the cell size is, the more details it captures. We examined different cell sizes and for the purpose of this research work, 8×8 pixels is the appropriate cell size to capture the desired

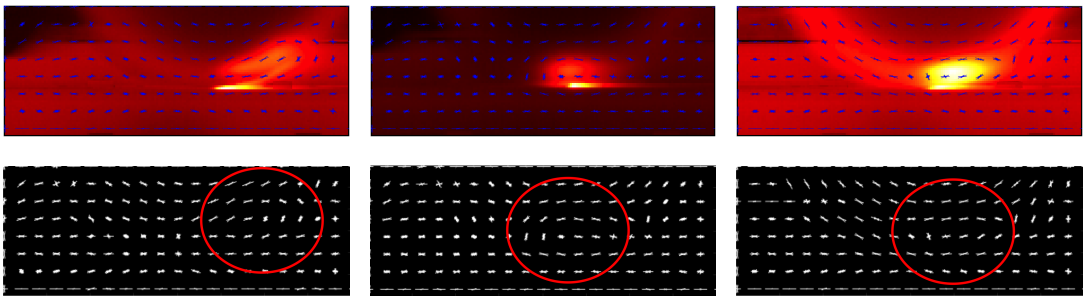


Figure 2.8: Visualization of HOG feature descriptors of train wheels. The top rows visualizes HOG features on the wheel thermal images and the bottom row is HOG visualizations of the same thermal images without the wheel on the background. In the bottom row, the areas inside the red circles demonstrate the feature descriptors of the contact point of the sliding wheels.

features. Next, for each cell we calculate a one-dimensional histogram of gradient orientations over pixels in the cell. These histograms capture the local heat pattern properties. The gradient at each pixel is discretized, and then each pixel votes for the orientation with a weight which depends on the magnitude of its gradient. Finally, the histogram of each cell is normalized with respect to the gradient energy in a neighborhood around it. The HOG features for three normal and three sliding wheels are illustrated in Fig. 2.7 and 2.8 accordingly. The features are shown on the thermal image and without the image at the background. It can be visually seen that the orientation pattern of HOG descriptors around the hot spot, which is shown with a circle, is different from the same part in a normal wheel.

2.5.2 Classification and sliding wheel detection with Support Vector Machine (SVM)

The HOG descriptors extracted from the thermal images provide a feature set by which we can potentially distinguish defective wheels from normal wheels. After acquiring the feature descriptors, we train a Support Vector Machine (SVM) classifier to detect the sliding wheel. SVM is a supervised learning method, which can find a decision boundary between two classes based on their feature data. Assume that we have a training set D which is defined as

$$D = \{(\mathbf{x}_i, y_i))|\mathbf{x}_i \in R^p, y_i \in \{-1, 1\}\}_{i=1}^{N}, \tag{2.4}$$

where y_i is the label of feature vector $\mathbf{x}_i$ (for example $y_i = +1$ indicates a normal wheel and $y_i = -1$ indicates a defective wheel), p is the number of features in the vector $\mathbf{x}_i$ and n is the number of data points in the training set. We want to find the optimal hyper plane which separates the data labeled as $y_i = +1$ from the ones labeled as $y_i = -1$. SVM is a method by which an optimal decision function can be learned from training data D. This is illustrated in Fig. 2.9.

In our work, the two classes are normal and sliding wheels and our data set consists of a set of simulated thermal images and a set of real images taken by a wayside thermal

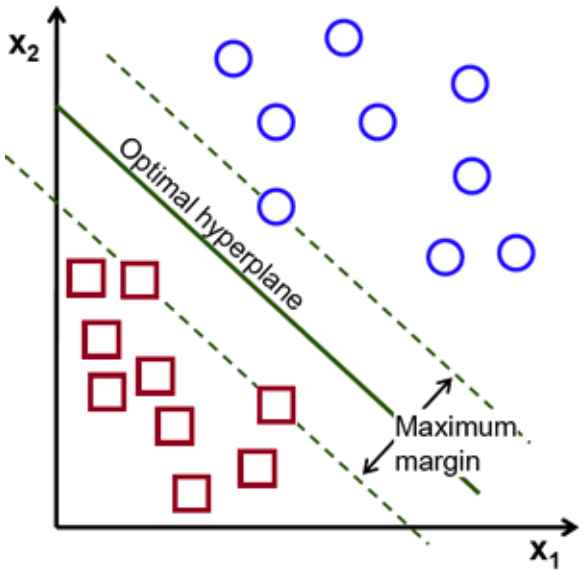


Figure 2.9: Support Vector Machine (SVM) for linearly separable data points

camera on the UPRR. The inputs to SVM are the HOG feature descriptors. The SVM uses the feature descriptors in the training data to learn a detection algorithm that can be applied to classify incoming wheels as defective or not. Depending on the wheel damage level, different approaches should be taken after detecting a sliding wheel. Hence, it can be helpful if the damage level is also determined. After detecting the defective wheel, it can be further categorized into different classes, which determine the level and severity of the damage. We call this procedure *defect clustering* and it is done based on the size and number of flat spots. In our previous work [25], we developed and introduced such algorithm that can be applied to the detected damaged wheel images to estimate the damage level.

2.6 Automatic Hot Bearing Detection

As mentioned before, the bearing region of the image is automatically detected by applying the HT to the detected wheel. To calculate the bearing temperature, we do not use any safety margin while segmenting the bearing. Instead, we use exactly the bearing pixels (see Fig. 2.5). Next, we calculate the mean intensity of the bearing, which has a direct relation with the bearing mean temperature. By having the mean intensity of the bearing, its mean temperature can be easily calculated. A threshold, based on the maximum normal bearing temperature/intensity should be set and those bearings that have a mean temperature/intensity above this threshold are labeled as *hot bearings*. The intensity distribution of the bearings of the UPRR data set and a hypothetical threshold for hot bearing are illustrated in Fig. 2.10. As the actual temperatures were not provided by the UPRR and the primary objective of the research was on sliding wheel identification, we didn't test the accuracy of bearing detection procedure, but rather the intensity values of the image data set to demonstrate the idea.

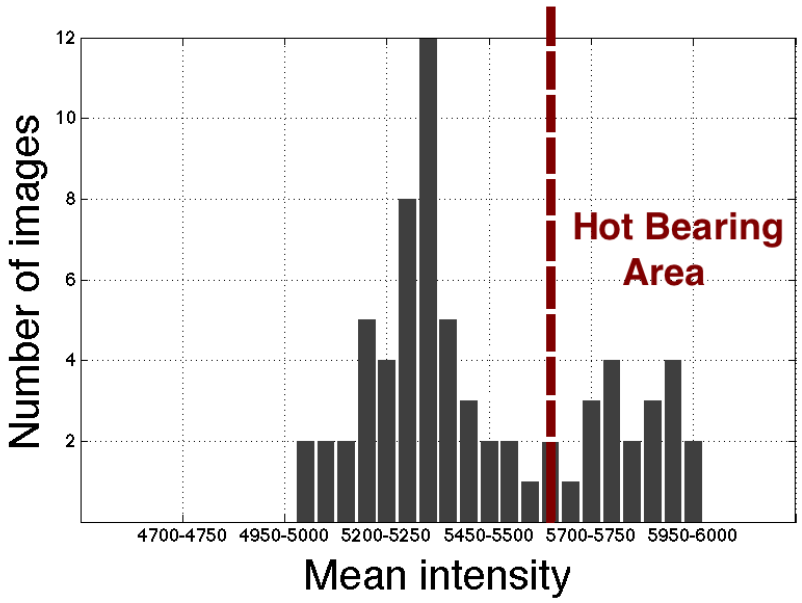


Figure 2.10: Bearing intensity distribution for the UPRR data set

2.7 Implementation and Results

To evaluate our proposed algorithm, we applied it to a set of simulated and real thermal images. Creation of simulated images was necessary as our defective wheel sample data set was not large enough by itself to train the model. To simulate train wheel thermal images we used ANSYS [25] which employs Finite Element Method (FEM) to generate the model. Our model simulates the wheel and bearing temperatures and identifies the temperature gradient at the wheel-track contact point. The simulation parameters are explained in details in the next section.

2.7.1 Simulating the wheel temperature profile using Finite Element Method (FEM)

Temperature distribution on the railway wheel can be obtained through Finite Element Method (FEM) simulations [64]. The advantage of using a numerical model to generate the wheel temperature profile is the possibility to test the versatility of our proposed technique under different heat flux scenarios [54]. A two dimensional, static, steady state FEM model of the wheel and rail was developed using ANSYS FEM software, where a set of heat sources and sinks are set to mimic the heat fluxes of the wheel while in motion. For instance, the convection heat transfer coefficient around a wheel moving at 80 km/h (50 mph) is approximately $h = 56W/m^2k$ [10]. Using this heat transfer coefficient, the resulting Biot number becomes $Bi = 1.14$. A Biot number larger than one, indicates that the temperature gradient inside the wheel is larger than between the wheel and the air. This is also indicative of a very small thermal boundary layer [10]. To mimic this condition, the wheel surface is modeled as a constant temperature source term with no convection to the air. The outside temperature of the wheel is then set at 300 $^{\circ}C$. This temperature is based on experimental measurements of a train wheel overheating due to the contact friction between the rotating wheel and the rail under normal wheel-rail interaction [21]. The

axle bearing is also set at the same temperature. Once the wheel experiences an abnormal behavior, inducing the wheel sliding on the rail instead of rotating, an extra localized heat source will appear between the wheel and the rail. This overheating due to sliding is modeled as an area on the wheel with a higher temperature. The temperature of the hot spot is varied from 550 to 800 $^{\circ}C$ in our study and its size is between 15 to 40 pixels. Additionally, ambient air temperature is modeled as 25 $^{\circ}C$ and the rail temperature is modeled as 80 $^{\circ}C$ (Fig. 2.11). The wheel and the rail are modeled with material properties of steel. The temperature profiles obtained from the FEM simulations are then imported into MATLAB for post-processing. The post-processing consists of uploading the FEM simulations files and converting the temperature profiles into an image with similar image size and properties as those obtained from thermal camera. Furthermore, the motion blur effect is simulated using MATLAB build in Wiener filter. In order to have more realistic images, Gaussian noise is added to the them. This noise resembles the overall noise of the thermal camera and environment. An example of the simulated thermal camera images is shown in Fig. 2.12.

2.7.2 Training and testing the algorithm

We applied our proposed algorithm to a set of simulated data as well as a real data set from the UPRR. The image size for both data sets is 240×320 pixels. We divided the

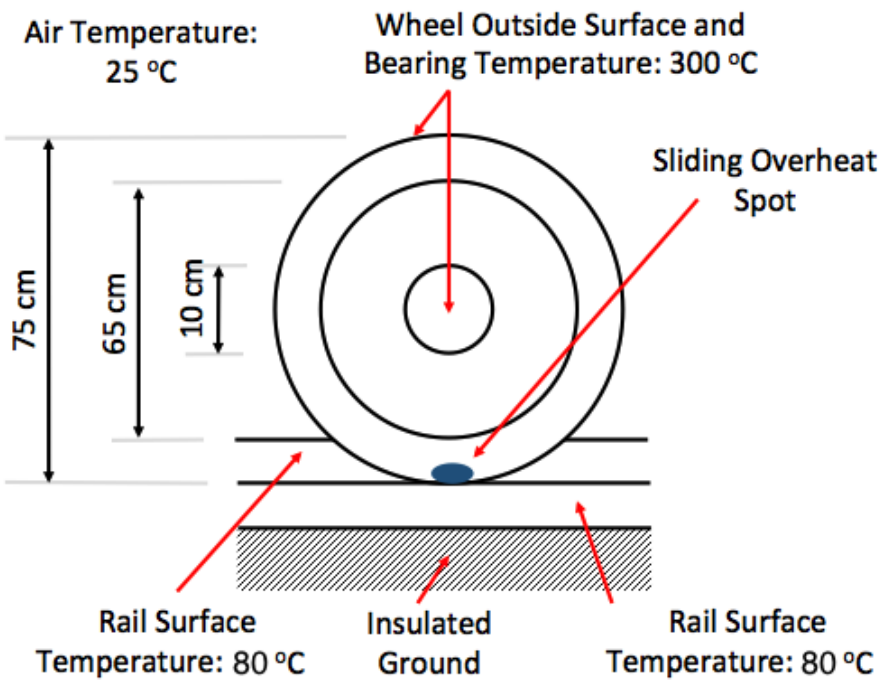


Figure 2.11: Schematic diagram of the thermal FEM wheel-rail system

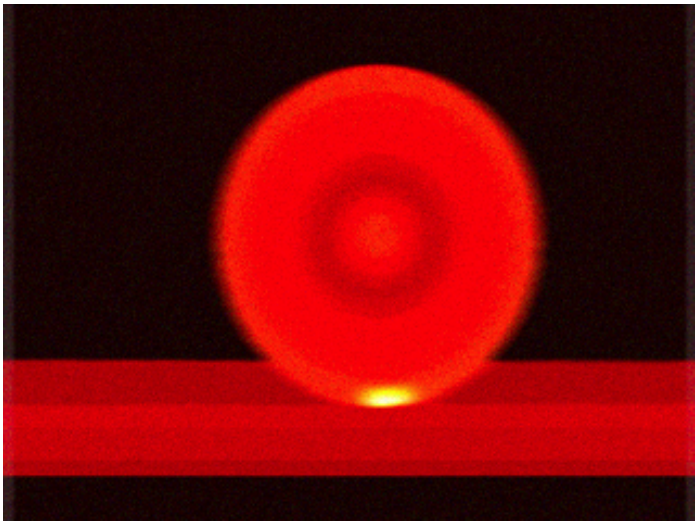


Figure 2.12: Simulated thermal wheel image using FEM

available simulated images into two sets of training and test and used the real data set for the evaluation purpose only. The training data set consists of 200 simulated images in which 100 of the images are normal train wheels and 100 of them are

sliding wheels. The test data sets include a set of simulated images with 50 normal wheels and 50 defective wheels and a set of real images with 400 and 8 normal and sliding wheels, respectively. We first train our algorithm on the training set. After the training phase, we apply the trained algorithm to the simulated test data set to evaluate its performance on the simulated images. Then we apply it to the test set consisting of real camera images, in order to evaluate the accuracy of the algorithm for real world data. Table 3.4 outlines the application results of our proposed algorithm to each of the test data sets separately, as well as the overall performance. As the results show, the accuracy of our proposed algorithm for the simulated data set was 100% which means our algorithm was able to detect all the sliding wheels without any false alarms. For the real data set, the algorithm was able to detect 88% of all the defective wheels and it identified all the normal wheels correctly. The major algorithm failure reason was wheel segmentation inaccuracy that can be mitigated in the future through improvements in the wheel detection process. Despite the fact that our algorithm was trained on a simple simulated model (which was built based on only two parameters; size and temperature of the hot spot), it still resulted in good accuracy for the real world data.

Table 2.1
Wheel defect results on simulated and Union Pacific data set.

Data set	**Number of normal wheels in the data set**	**Number of defective wheels in the data set**	**Algorithm precision for normal wheels (%)**	**Algorithm precision for defective wheels (%)**
Training set	100	100	-	-
Simulated test set	50	50	100	100
Real test set	400	8	100	88
Total test set	450	58	100	98

2.8 Fusion of Visible and Thermal Imagery

Up to this point, all the information was extracted from the thermal imagery. However, every sensor has its strengths and weaknesses. There are some types of defects that are not detectable at an early stage in the images taken by a vision camera while these defects generate a distinctive heat pattern on the wheel or bearing that is clearly visible in the thermal imagery. On the other hand, other damages might be detectable from the visible spectrum image, but not necessarily have a distinguishable heat pattern in the thermal imagery. Since a thermal image is basically built of solely temperature data, it excludes other critical information, such as texture or color. This makes thermal and visible spectrum imagery complementary and suggests that if the images are fused the result can benefit from the strengths of both sensors.

During the sliding wheel detection algorithm based on thermal imagery we noticed that a significant source of error was the inaccurate wheel and bearing segmentation.

Hence, if the accuracy of wheel detection can be improved, the overall design will operate better. In this section, wavelet decomposition is employed to extract the features of the thermal and vision imagery. Then the two images are merged based on their decompositions and a fused image is composed. The resulting fused image contains more information than each individual image and can be used as an input for image-based wheel and bearing defect detection algorithms. To verify the proposed method and to show an example of this application, it is demonstrated on a real data set from a Union Pacific rail line to identify sliding wheels.

2.8.1 Image fusion approach

Fig. 2.13 shows samples of wheel images taken by a wayside visible spectrum camera. It can be seen that as mentioned, visible spectrum imagery provides visual context while the temperature information can be obtained from thermal imagery.

Figure 2.13: Examples of train wheel images taken by wayside visible spectrum imaging system

A big challenge in the algorithm proposed earlier, is to detect and segment the wheel

and bearing sections from the image. In this section we are trying to improve the accuracy of the wheel and bearing extraction by fusing thermal and visual spectrum imagery.

Image fusion can be classified to three major approaches based on the level that the fusion takes place [83]:

1. Pixel level,

2. Feature level, and

3. Decision level.

The information is collected thorough different sensors from environment, then depending on the fusion method different techniques are applied to obtain the desired output. In the pixel level fusion, the thermal and visual images are directly combined into one single image and then the subsequent processing techniques can be applied to the fused image. If features of the input images are extracted first, and then are combined into a single one, feature level fusion has been applied. In the decision level fusion, first image features are extracted, decisions are made based on the feature descriptions of each image and at the end the decisions are combined to a single one. The block diagram of the three classes of image fusion is shown in Fig. 2.14 to clarify the difference between the fusion levels.

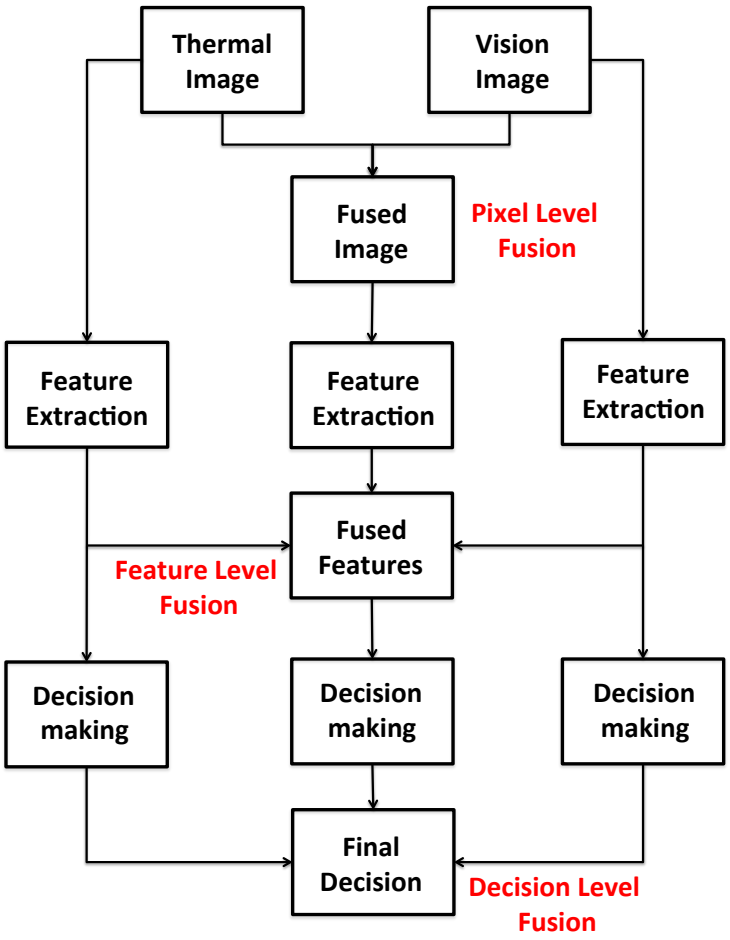


Figure 2.14: General block diagram of different fusion levels

Among the three approaches, pixel fusion level is probably most used in the literature and the fusion happens at the lowest processing level. The method employed in this dissertation is pixel level fusion in which we combine the thermal and vision camera imagery into one image that contains more information than every single input image.

Different techniques can be applied to this final image to detect wheel and bearing defects.

Some of the most common pixel level fusion methods are averaging techniques, *principle component analysis* (PCA) and Wavelet transform. More methods are mentioned and compared in the paper by Aslantas et al. [4]. In this research, we will fuse train car wheel thermal and vision imagery using a Wavelet transform and show that by employing this method more informative images can be obtained. In addition, we will also fuse the two images using a simple minimum pixel value in which the smaller intensity value of the two images is chosen and show that in some applications, such as the one in our work, even employing a simple method that does not need much computation power can result in a fused image with noticeably more information than a single image.

2.8.2 Wavelet transformation

According to Fourier theory, a signal (and an image which is its expansion to 2D) can be represented as the sum of sines and cosines. As the Heisenberg uncertainty principal expresses, time and frequency are complementary variables and the more precisely one is determined, the less precisely the other one can be known. Fourier transform has high frequency resolution and no time resolution. This is because the

Fourier transform represents data as a sum of sine and cosine waves which are not localized in time. Hence, Fourier transform is not the proper choice for the condition where sudden changes are needed to be well localized.

Considering the type of images that are employed in this work and the defects that should be detected, abrupt changes in contrast is a feature that we want to be able to capture. The thermal imagery is basically only made of contrast changes and some defects such as hot spots are detected through capturing sudden changes in contrast. In a similar way, contrast changes in visual spectrum imagery can potentially represent some type of image features. Thus, it is desirable to employ a method that can capture abrupt changes in contrast in the image. This leads us to the Wavelet transform which is similar to the Fourier transform, but it is well localized in time and frequency.

A wavelet is a wave form signal that has zero mean and exists only for a finite time duration. There are different shapes of wavelets and one example is shown in Fig. 2.15.

A Wavelet transform has two important characteristics: scaling and shifting. Scaling stretches or shrinks the signal in time. A stretched wavelet helps in capturing the slowly varying changes in the image while a shrunken wavelet can capture the abrupt changes. Shifting a Wavelet moves its center to the left or right so it can be aligned with the feature that we are looking for. According to the image and specific application, the appropriate wavelet should be chosen. More guidance about wavelet analysis can be found in [72].

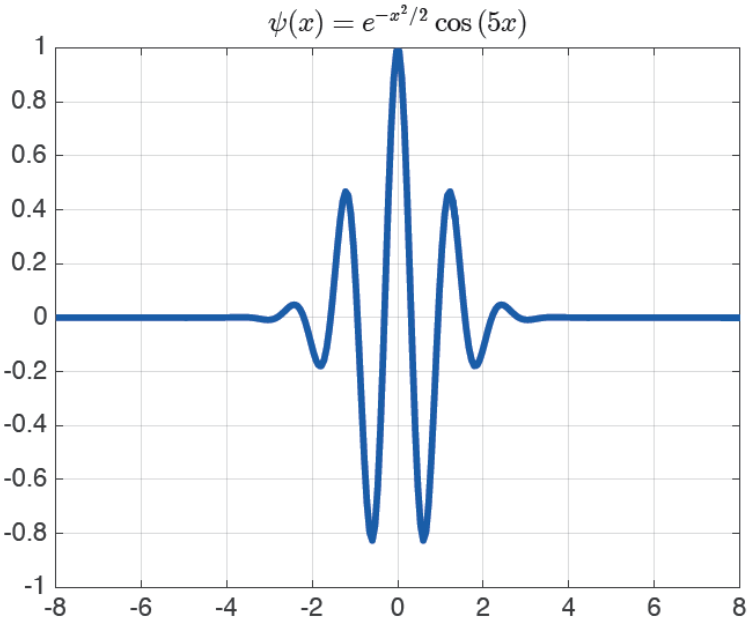
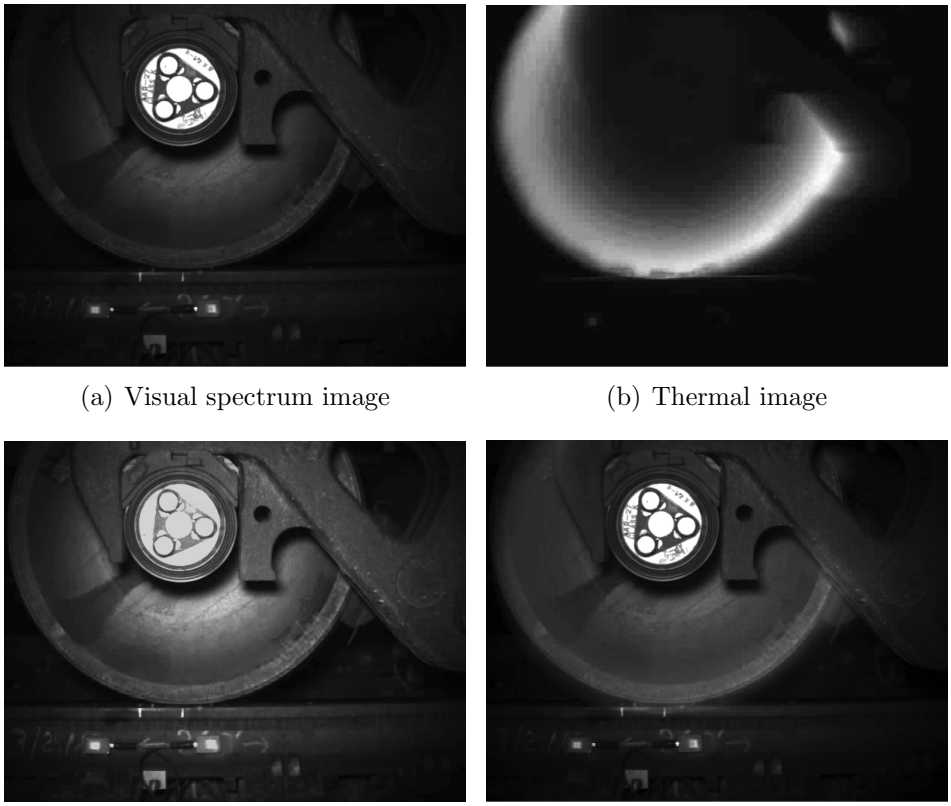


Figure 2.15: Wavelet example

The original input images which are the images taken by visual spectrum and thermal wayside camera are shown in Fig. 2.16(a) and 2.16(b), and the resulting output for the two employed fusion methods of *min* and *Wavelet* are demonstrated in Fig. 2.16(c) and 2.16(d), respectively.

The fused image shown in Fig. 2.16(c) is obtained by overlaying the two input images and choosing the minimum pixel intensity. The second fused image shown in Fig. 2.16(d) is acquired by applying the wavelet fusion method. In this approach, first wavelet decomposition of the two original images are obtained and then based on these decomposition, the two images are merged [81]. In both cases it can be seen that more details can be seen in the fused image than the original images.



(a) Visual spectrum image (b) Thermal image

(c) Fused image using Min pixel value (d) Fused image using Wavelet

Figure 2.16: Image fusion using different methods

At this point the wheel and bearing detection and extraction procedure can be applied to the fused image as illustrated in Fig. 2.17.

We applied the algorithm to a data set from Union Pacific (UP) railroad including 294 thermal and visual images. The data set did not contain any defective wheels, so we evaluated the effect of image fusion on the wheel segmentation accuracy and compared the results to the wheel segmentation of thermal imagery. The outputs showed that the wheel segmentation was improved by over 30% using the fused images. Considering that the wheel segmentation is the first and most challenging step of

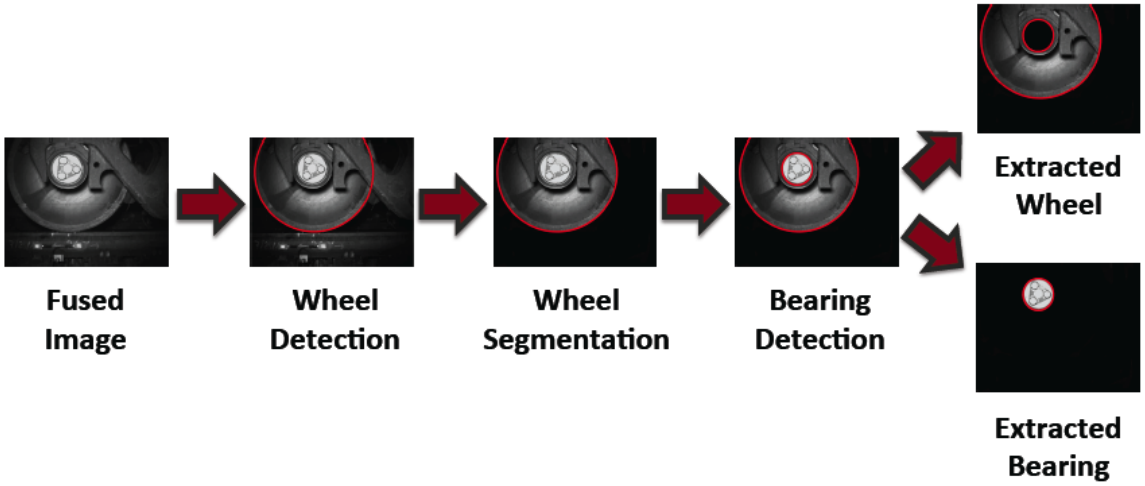


Figure 2.17: Detection and segmentation of wheel and bearing from fused image

sliding wheel detection, we expect that the proposed fusion algorithm should result in noticeable improvement in wheel and defect detection.

A similar conclusion can be made for the case of hot bearing detection. The hot bearing detection only requires calculation of the mean temperature and comparison with a given threshold. The bearings that exceeded the threshold are considered to be hot. The output totally depends on the accuracy of the bearing extraction and is not possible if the bearing section can not be detected accurately. The developed method, improved the bearing segmentation accuracy in the given data set. It should be noted that since bearing detection occurs following the wheel segmentation, precise wheel extraction is important for bearing segmentation.

2.9 Summary and Future Work

The objective of wayside detection systems for rolling stock is to identify failures and inform the operators about the need to remove or repair the parts before they cause more damage or an accident. To achieve this goal, fast and reliable defect detection methods are necessary. This chapter introduced a novel automatic method for detection of sliding wheels and hot bearings from thermal imagery. Our proposed algorithm offers an alternative method for detecting sliding wheels and hot bearings, one that can reliably identify uneven temperature distributions and defective bearings.

To evaluate the accuracy of our sliding wheel detection method, we applied the algorithm to a set of simulated wheel images as well as real data obtained from the UPRR. The results showed that it was able to detect 98% of the total number of simulated and real world defective wheels in addition to identifying all the normal wheels without any false alarm.

In addition to sliding wheel detection, it was realized that thermal imagery can be used for hot bearing detection with little additional effort. Since the majority of our hot bearing detection algorithm takes place in conjunction with the sliding wheel detection procedure, the only additional effort to identify hot bearings in this approach includes comparison of the calculated mean intensity/temperature with a set threshold.

The objective was to find the optimum algorithm which is accurate for detecting patterns indicative of a sliding wheel and at the same time, reasonable in terms of time and memory needed for computational purposes. This was successfully done in the research. Since the current project concentrated on sliding wheels, no emphasis was placed on identifying defects outside the wheel/rail interface. However, our algorithm can detect the flat spots at any other point of the wheel, as long as it is visible in the thermal image. The next research step will apply the same method toward detecting hot spots located throughout the rim and to remove the potential occlusion by the car bogie components, two cameras will be installed in series to secure that a full wheel rotation is visible. An important and difficult part of our algorithm is to identify the wheel and bearing parts in the thermal imagery. Hence, a sensor fusion method was introduced that provided accurate wheel and bearing segmentation in wayside imagery to further improve the wheel and bearing defect detection. The method was applied to a data set obtained on a real railroad and the result of the wheel and segmentation on the fused images were compared with the segmentation on the thermal imagery and it was shown that the accuracy was improved. Since the wheel and bearing segmentation are the most challenging steps and the main source of error in the sliding wheel and hot bearing algorithm, we expect fusing images to help improve the precision of the defect detection algorithm.The specific sliding wheel and hot bearing application were given as an example, but the use of the thermal and visual spectrum image fusion method introduced in this chapter is not limited to

these specific application.

Future process improvements for this part include additional steps of image pre-processing with focus on noise cancellation and debluring to obtain better wheel and bearing segmentation accuracy. In addition, fusing thermal imagery with visible-spectrum imagery potentially can provide the ability to specify the location (car, axle) of the defective wheel or bearing in addition to the benefit of detection. Furthermore, a train wheel former history/profile can be fused with the result of the wheel inspection algorithm for more accurate conclusions and possible wheel damage prediction.

Chapter 3

Heterogeneous Multi-Sensor Fusion for Mobile Platform 3D Pose Estimation

3.1 Introduction

Pose—position and orientation—estimation is a non-separable part of a variety of applications from health condition monitoring [40] to robotics. Depending on the application, the environment, the available platform and budget, etc., different sensors with various capabilities might be chosen to collect the data. It is not always possible

to use the most accurate or expensive sensor. Some sensors might not even be able to operate in certain conditions. For example, GPS signal can be degraded or lost in certain areas. These reasons altogether encourage researchers to employ methods to fuse the information from different sensors with various characteristics to overcome sensor deficiencies and achieve improved results. There are various algorithms to fuse the data obtained from different sensors; an *Extended Kalman filter* (EKF), is among the most common methods employed for this task.

Each of the sensors employed in this work has its strengths and weaknesses. An *inertial measurement units* (IMU) is an accurate sensor; however, it is not reliable over long time periods because it integrates the bias and its estimations drift over the time, suffering from a cumulative error. A relatively good pose estimation can be obtained from a 2D LiDAR. Nevertheless, since this type of LiDAR scans the environment in a plane, it is only suitable for an environment with three *degrees of freedom* (DOF), when two coordinates and an angle fully define the pose. In the 3D world there are 6-DOF and therefore a full pose estimation can not be obtained from a 2D LiDAR. A camera, on the other hand, can give all 6-DOF, but with a lower accuracy especially along its focal axis. In terms of speed, an IMU is a fast sensor with a high refresh rate, while a LiDAR and a camera are relatively slow.

In the rest of this chapter, we first review the previous work and then introduce our proposed modular algorithm and compare its results with the state of the art.

3.2 Previous Work

Recently, robot localization has seen great success and improvement in accuracy according to both sensors employed and advances in algorithm development. There are various works targeting the problem of localizing an object in a 3D environment from different perspectives [11, 15, 16, 27, 33, 34, 55, 67]. While some methods might be based on the information received from a single sensor, most of them employ sensor fusion. Depending on the type of the available sensors and their characteristics, different approaches can be taken for the pose estimation. The previous literature in this area can be discussed from the perspective of the environment of the work, fusion method, and employed sensors.

From the environment of the work point of view, a significant question to be answered is whether the vehicle is operating indoor or outdoor. This makes a noticeable difference, because unlike indoor, in outdoor environment GPS signal is available and more often used. A common sensor fusion approach, which is widely used in the outdoor environment, is IMU-GPS fusion. Despite the fact that IMUs are a very accurate sensor, they cannot be trusted in the long term since they integrate bias and hence accumulate error. GPS, on the other hand, gives an absolute measurement of the position that is not as accurate as IMU measurements. In this area, sensor fusion methods are either dedicated to pose estimation in GPS-available environment [11],

or they operate based on the assumption that the GPS signal is available for the majority of the time, and propose a method for the case that it is lost for a short time [15]. As long as the GPS signal exists, the localization relies on the data coming from the GPS as well as the IMU; if the GPS signal is lost, then the pose will be obtained only from the IMU.

A magnetometer (or compass) can be employed to improve the orientation estimation. However, magnetometers have poor accuracy during fast movement and in addition, they cannot provide measurements for position estimation. Hence, they are often employed along with GPS in outdoor localization allowing correction for both position and orientation. A popular method of indoor localization is to employ IMU along with another sensor such as a camera, as explained by Mirzaei et al. [55]. Their algorithm uses an iterated EKF to fuse the IMU data with camera measurements of known feature points on a target to find the transformation between the IMU and camera. This method is also widely employed to estimate the 3D pose of a mobile vehicle [17, 35]. Hesch et al. [33] also developed an observability analysis and consistency improvement for this technique.

From the fusion method perspective, a large number of fusion algorithms employ EKF. However, *Unscented Kalman Filters* (UKF) have been an alternative for EKF, specifically when a first order linearization does not sufficiently describe the nonlinear system. UKF is employed in [42] to perform a target-free calibration by estimating the

position of a series of static landmarks in the environment. The complete state vector includes the position of landmarks as well as the sensor-related measurements. It should be mentioned that even though UKF is commonly known as an alternative for EKF, it is not necessarily the preferred option. Choosing between the EKF and UKF depends on the accuracy needed, computational overhead, and the noise level [46]. If the sensor sampling rates are short enough (such as in our work), UKF performance may not be superior to the EKF and for the case of limited computational power and resources, EKF is a more appropriate choice. Often, if the the application is aiming for real-time localization, EKF is employed. It has also been shown that noise has a major effect on the performance of UKF, and its accuracy is significantly reduced if SNR is low [73]. While it is true that a poor choice of the initial state of a EKF can cause it to diverge, in a real UAV mission its initial state is usually accurately known. Therefore, choosing EKF or UKF is an application dependent decision [30], and for this work we will focus on the use of an EKF.

If there is no knowledge of the system model, or the system is too complicated such that system modeling is very costly or difficult—e.g., a large network of sensors that communicate with each other—there are fusion methods that rely on the sensors characteristics rather than the system model itself. Fuzzy sensor fusion is an example of this approach. For example, in [2, 31] the authors employ a multi-sensor fusion using fuzzy techniques. Their proposed algorithms employ a weighted mean to obtain the final estimation from different sensor readings and the aggregation weights are

assigned based on the individual sensors' accuracy and bandwidth. In approaches like this, the sensors behavior must be known very well and they are usually employed when the full system cannot be easily modeled.

In terms of the employed sensors, in addition to IMU, GPS and camera are the sensors that are commonly used and LiDAR is also becoming more popular. A precise 3D pose estimation usually can not be obtained from one camera, especially because it is difficult to estimate the distance traveled along its focal axis. Therefore, stereoscopic vision is broadly used when more than one camera is available and multiple images can be obtained from different viewpoints. It has been shown that by using multiple cameras, a better 3D pose estimation can be obtained. In [16], the authors suggest a robust pose estimation method based on the data from an IMU and two cameras embedded on an *unmanned aerial vehicle* (UAV). In [67], the authors employed two stereo configurations using four cameras on a *micro aerial vehicle* (MAV) to implement an autonomous real-time navigation system. The hardware is configured such that one stereo camera is facing forward for the purpose of *simultaneous localization and mapping* (SLAM) and the other is facing downward for ground plane detection. A 3D pose estimation is obtained from each stereo configuration and then they are fused with IMU data using an EKF to improve the results.

Unlike cameras, 2D LiDAR has not been commonly used for 3D localization purposes. The reason is that a 2D LiDAR can not give a full pose estimation in 3D environment.

Nevertheless, having prior knowledge of the environment in the form of environment features or object maps make it possible to employ a 2D LiDAR in a 3D environment to improve pose estimation [34]. For example, the authors of [82] employed an *inertial navigation system* (INS) aided by a 2D LiDAR to localize a vehicle in a 3D environment. In this paper the authors assumed that certain plane features of the environment have been previously mapped and can be compared to the LiDAR measurement and use these mapped features to localize the LiDAR measurements.

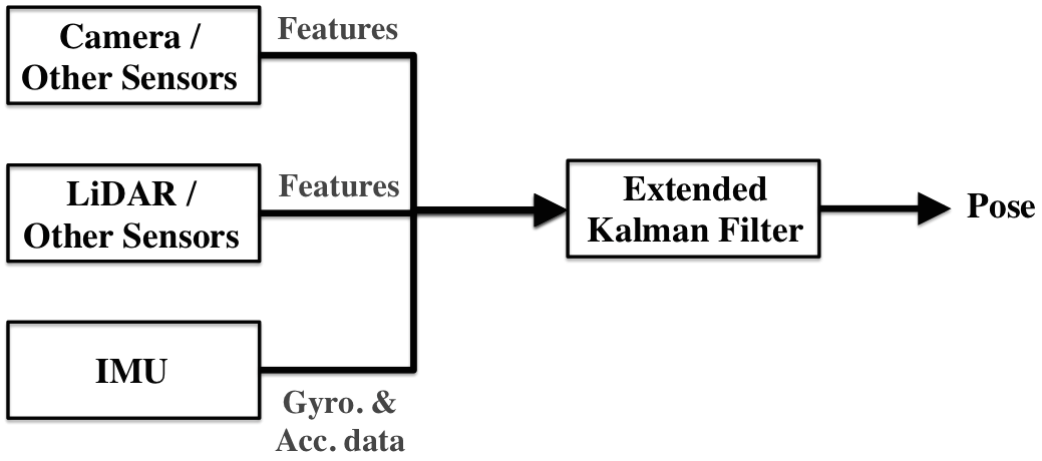
While 3D scanning LiDARs are capable of estimation of a full 3D pose, usually they have the drawback of being too expensive and bulky. However, a precise pose estimation often can be acquired from a 3D LiDAR with the addition of an available high-quality 3D map. As an alternative to a 3D LiDAR, it is also possible to employ two perpendicular 2D LiDARs, one for the vertical axis and the other to cover the horizontal axis to estimate 3D pose [27].

3D pose estimation from the fusion of IMU and camera is a well studied and developed area of research. LiDAR scans have been employed for 2D pose estimation. However, to the best of our knowledge, employing a 2D LiDAR with an IMU and camera in an unknown environment has not received suitable attention. Hence, in this chapter, we propose a method to accomplish this goal. To fuse the IMU and camera estimations, we adapt the method introduced by Weiss and Siegwart [76]. In contrast to many of the previous works, which include the visual features in the fusion algorithm, the

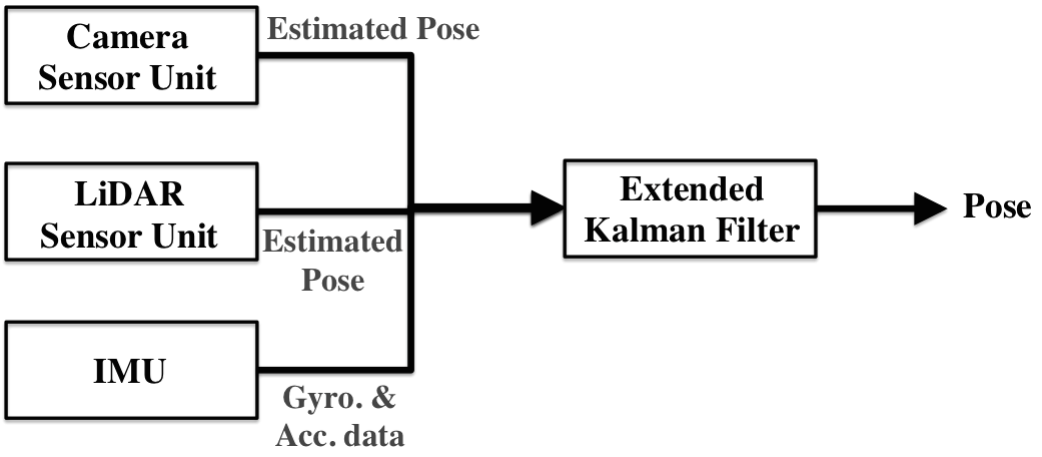
algorithms proposed in [76] do not fuse features and rather treats the visual framework as a black box which is able to output an unscaled pose estimate. The advantage of this approach is that the algorithm becomes modular and can be widely applicable to nearly any existing single sensor solution. Also, as we show, the sensor units can be replaced by any other sensor(s), as long as the whole unit—i.e., sensor(s) + processing—can output a pose estimation and its uncertainty. This makes the algorithm suitable for our purpose as we can add in a 2D LiDAR module. Weiss et al. [76] use the EKF as the fusion tool to combine the results of the vision module with IMU measurements. Similarly, we add in a 2D LiDAR module. Furthermore, since the design of our fusion algorithm is modular, it is relatively easy to add a new sensor unit later using the extra information to (ideally) have a better pose estimation; all that is needed is a way to produce unscaled pose estimates from the sensor. The block diagram of our proposed approach, compared to the commonly used standard approach, is illustrated in Fig. 3.1.

3.3 Localization Algorithm

Before continuing the discussion, the employed axes and angles have to be defined. As illustrated in Fig. 3.2, the z axis points up, the x axis points forward, and the y axis points to the right with respect to the UAV body. The yaw (α) angle is defined as a rotation about the z axis, pitch (β) as a rotation about the y axis, and roll (γ)



(a) Standard approach



(b) Modular approach

Figure 3.1: Different sensor fusion approaches for pose estimation.

is a rotation about the x axis.

The idea is to employ the pose estimation from the camera and LiDAR, whenever they are available, to correct the IMU pose estimation. As demonstrated in Fig. 3.1, the modular approach introduced in [76] is used and each of the LiDAR and camera sensor units along with their localization algorithm are treated as a black box which is

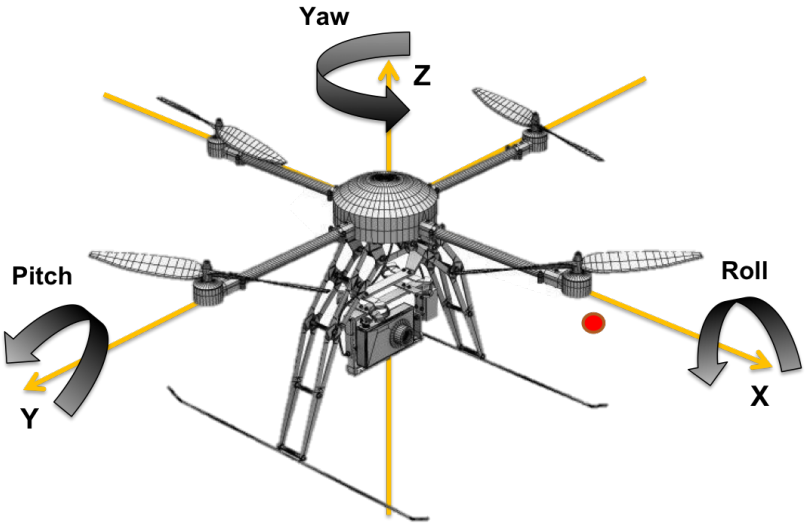


Figure 3.2: UAV principal axes and orientation geometry. Forward is indicated by the red dot.

able to output a pose estimation. The output of the camera sensor unit is a complete 3D pose while the LiDAR unit offers a 2D pose estimation in its local plane.

3.3.1 Sensor units

Camera sensor unit: The most widely used model for the camera is the pinhole model. This model describes the relationship between the coordinates of a 3D point and its projection onto the image plane of the camera. The mathematical equation of the model is [57]

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} k_u & 0 & \frac{u_0}{f} \\ 0 & k_v & \frac{v_0}{f} \\ 0 & 0 & \frac{1}{f} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{3.1}$$

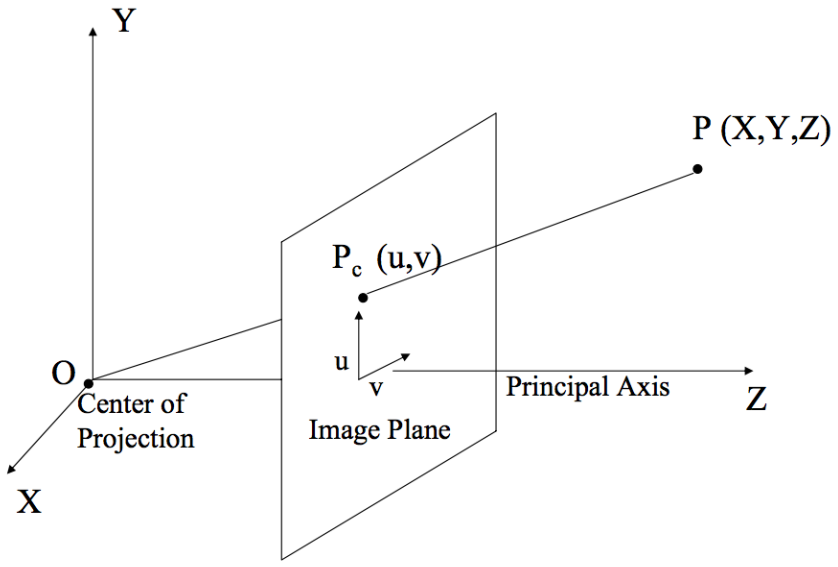


Figure 3.3: Pinhole camera model

in which, u and v are the point of interest coordinates in the image plane, X, Y and Z are the same point coordinates in 3D world frame and k_u and k_v are the scale factors along the axes of pixel coordinates [59]. u_0 and v_0 are the pixel coordinates of the principal point and f is the focal length of the camera lens. The geometry of the pinhole camera model is illustrated in Fig. 3.3.

The camera parameters are usually obtained from the standard calibration process using a checker board [62]. In this paper, in order to achieve the pose estimation based on camera images, a 1-Point RANSAC algorithm for EKF Filtering is employed based on the work done by J. Civera et al. [19].

LiDAR sensor unit:LiDAR is an optical remote-sensing technology which measures

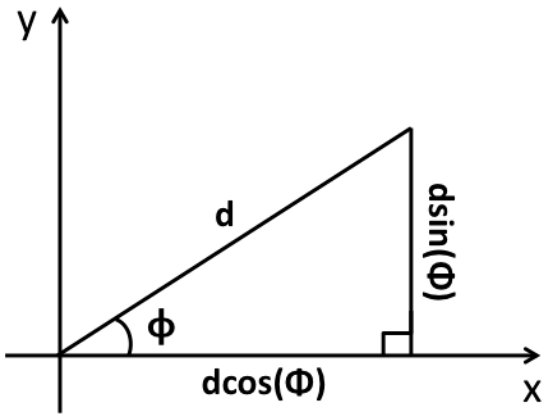


Figure 3.4: Derivation of x and y coordinates from the LiDAR readings

the distance and angle from the sensor to an object, using a laser beam. The distance between the sensor and the object is calculated by measuring the time interval between an emitted laser pulse and reception of the reflected pulse. This technology is used to create high resolution maps for different applications. One of its applications is Simultaneous Localization and Mapping (SLAM) in which the goal is to construct a map of unknown environment while simultaneously keeping track of the robot pose.

A 2D LiDAR, radially scans the environment in a plane. In order to obtain the coordinates of each scanned point, the following equations are employed

$$x = dcos\phi, \quad y = dsin\phi. \tag{3.2}$$

in which d is the distance from LiDAR to the scanned point and ϕ is the beam angle. The geometry for the equations (3.2) is illustrated in Fig. 3.4. It should be noted that x and y are the coordinates of the scanned point in the local frame of the LiDAR and

not with respect to the global reference frame. In order to find the global coordinates, the local coordinates have to be transformed to the global frame.

There are different common methods such as Iterative Closest point (ICP) [65] for estimating the pose, from LiDAR scan matching. In this paper we employ Chamfer distance in order to mach the consecutive scans and calculate the sensor pose [60].

IMU model: An IMU is an electric device that measures the velocity, orientation, and gravitational forces using a combination of data from accelerometers, gyroscopes, and magnetometers. In this work, only the raw data from the accelerometer and gyroscope are employed.

The 3-axis accelerometer measures the acceleration along each of the x, y, and z axes in the IMU local frame, and the 3-axis gyroscope measures the angular velocities related to each axis with respect to the global frame. To model the IMU, its bias and noise should be considered. The noise n is assumed to be a zero-mean white Gaussian random signal and the bias b is modeled as a random process. Therefore, the true and the measured values for angular velocities and linear accelerations can be written as

$$\omega_t = \omega_m - b_\omega - n_\omega, \quad a_t = C(q_t)(a_m - b_a - n_a) + g, \tag{3.3}$$

where the subscripts t and m denote the true and the measured values, respectively. g is the gravity vector, q_t is the quaternion equivalent of the true attitude, and $C(q_t)$

is the rotation matrix relevant to q_t. In equations (3.3), n_ω and n_a represent the noise of gyro and accelerometer and b_ω and b_a denote the relevant biases. As mentioned, these biases are random processes; their dynamics are modeled as

$$\dot{b}_\omega = n_{b_\omega}, \quad \dot{b}_a = n_{b_a}. \tag{3.4}$$

As a result of these biases, IMU measurements tend to drift over time. Mitigating this drift is accomplished by fusing this information with other sensor readings such as camera or LiDAR.

3.3.2 Methodology Overview

An EKF was chosen as the fusion method, which linearizes the state about an estimate of the current mean and covariance and assumes additive process and measurement noises. The general formulation of the EKF is

$$x_k = f(x_{k-1}, u_{k-1}) + \omega_{k-1}, \quad z_k = h(x_k) + \nu_k, \tag{3.5}$$

where subscript k denotes the time step, ω and ν are the zero-mean Gaussian process and observation noises with with covariances Q and R. In the rest of this chapter, we will drop the subscript k for the noise covariances since it is assumed that they are constant. In equation (3.5), u is the input control vector.

In real applications, the functions f and h are usually nonlinear and can not be applied to covariances directly. This problem is solved by taking the Jacobian of these functions. The filter consists of two major steps: *predict* and *update* that can be summarized as following.

Predict:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}), \qquad \text{State estimate}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}, \qquad \text{Covariance estimate}$$

Update:

$$\tilde{y}_k = z_k - h(\hat{x}_{k|k-1}), \qquad \text{Measurement residual}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k, \qquad \text{Innovation}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \qquad \text{Kalman gain}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k, \qquad \text{Update state estimate}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}, \qquad \text{Update covariance estimate}$$

where

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}_{k-1|k-1}, u_{k-1}}, \quad H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{x}_{k|k-1}}.$$

In this work, the state of the system is a 16-length vector which is represented by

$$x = \begin{bmatrix} p_{\{1\times3\}} & v_{\{1\times3\}} & q_{\{1\times4\}} & b_{\omega\{1\times3\}} & b_{a\{1\times3\}} \end{bmatrix}^T.$$

It contains the position of the IMU p, with respect to the global frame, IMU linear velocity v, and the quaternion q that represents the rotation of the IMU from the reference frame. The two remaining parameters in the filter state express the gyroscope and accelerometer biases, b_ω and b_a, along each axis.

3.3.3 Extended Kalman Filter (EKF)

Three sets of equations are discussed in this work: *true*, *nominal*, and *error state* equations, with the true state being a composition of the nominal and error states. The true state is decomposed into small and large signal with the large signal being the nominal state and the small signal representing the error state. The nominal state does not take into account the noise terms and other model imperfections. Therefore, it accumulates error and as time passes, it drifts from the true state. These errors are estimated and collected in the error states to correct the nominal states. The filter correction happens at the arrival of information from the camera and LiDAR, which arrives at a lower frequency than the IMU.

Employing quaternions for the attitude representation has the advantage of both

computational simplicity and also Gimbal lock prevention. The hypothesis is that at each time step, the change in the attitude is small enough that the four quaternion values can be estimated by three numbers, $\theta = [\alpha \ \beta \ \gamma]$, where $\delta q \simeq [1 \ \frac{1}{2}\delta\theta]$. This property is later used in the EKF error calculation.

Considering the state vector, the three sets of EKF equations can be introduced as following.

The true-state equations: The differential equations which govern the state of the system are

$$\dot{p}_t = v_t, \tag{3.6a}$$

$$\dot{v}_t = a_t, \tag{3.6b}$$

$$\dot{q}_t = \frac{1}{2} q_t \otimes \bar{\omega}_t, \tag{3.6c}$$

$$\dot{b}_\omega = n_{b_\omega}, \quad \dot{b}_a = n_{b_a}. \tag{3.6d}$$

Subscript t represents the true values and ω_t and a_t are defined in (3.3). Operator $\otimes$ represents the quaternion multiplication and $\bar{\omega}_t = [0 \quad \omega_t]$. Substituting the values of

a_t and ω_t from (3.3) yields the kinematic system

$$\dot{p}_t = v_t, \tag{3.7a}$$

$$\dot{v}_t = C(q_t)(a_m - b_a - n_a) + g, \tag{3.7b}$$

$$\dot{q}_t = \frac{1}{2} q_t \otimes (\bar{\omega}_m - \bar{b}_\omega - \bar{n}_\omega), \tag{3.7c}$$

$$\dot{b}_{\omega_t} = n_{b_\omega}, \quad \dot{b}_{a_t} = n_{b_a}, \tag{3.7d}$$

where $\bar{b}_\omega = [0 \quad b_\omega]$ and $\bar{n}_\omega = [0 \quad n_\omega]$.

The nominal-state equations: To obtain the nominal state, the expectation of equations (3.7a)–(3.7d) are taken, leading to

$$\dot{p} = v, \tag{3.8a}$$

$$\dot{v} = C(q)(a_m - b_a) + g, \tag{3.8b}$$

$$\dot{q} = \frac{1}{2} q \otimes (\bar{\omega}_m - \bar{b}_\omega), \tag{3.8c}$$

$$\dot{b}_\omega = 0, \quad \dot{b}_a = 0. \tag{3.8d}$$

Since all noises are zero-mean Gaussian, by taking the expectation, these terms disappear.

The error-state equations: Finally the last sets of equations are the linearized

dynamics of the error-state.

$$\delta\dot{p} = \delta v, \tag{3.9a}$$

$$\delta\dot{v} = -C(q)[a_m - b_a]_\times \delta\theta - C(q)\delta b_a - n_a, \tag{3.9b}$$

$$\delta\dot{\theta} = -[\omega_m - b_\omega]_\times \delta\theta - \delta b_\omega - n_\omega, \tag{3.9c}$$

$$\delta\dot{b}_\omega = n_{b_\omega}, \quad \delta\dot{b}_a = n_{b_a}. \tag{3.9d}$$

These equations are derived from equations (3.7a)–(3.7d) based on the small signal assumption [51]. The operator $[.]_\times$ is the skew symmetric matrix

$$[V]_\times \triangleq \begin{bmatrix} 0 & -v_z & -v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}. \tag{3.10}$$

The data obtained from the sensors is not continuous, and thus, the next step is to discretize the equations.

3.3.3.1 Error estate discretization and state matrix initialization

The error states are defined as the difference between the estimated states and the measured states and are shown using the notation

$$\tilde{x} = [\delta p \quad \delta v \quad \delta\theta \quad \delta b_\omega \quad \delta b_a]^T. \tag{3.11}$$

In other words, $\tilde{x} = x_m - \hat{x}$, where x_m is the measured state vector and $\hat{x}$ is the estimated one. In the error state vector, θ is the attitude error which is defined by the error quaternion using the small angle assumption. This handles the quaternion in its minimal representation and allows us to represent the uncertainty of the attitude by a 3×3 covariance matrix.

Since the data obtained from the sensors are not continuous, the equations must be discretized. After discretization, the linearized error state equation can be summarized as

$$\dot{\tilde{x}} = F_d\tilde{x} + G_d n, \tag{3.12}$$

where n is the noise vector $n = [n_a \; n_\omega \; n_{b_a} \; n_{b_\omega}]^T$ and F_d and G_d are

$$F_d = \begin{bmatrix} -\lfloor\hat{\omega}\rfloor_\times & -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ -C(\hat{q})\lfloor\hat{a}\rfloor_\times & 0_{3\times3} & 0_{3\times3} & -C(\hat{q}) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \tag{3.13a}$$

$$G_d = \begin{bmatrix} -I_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & -C(\hat{q}) & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \tag{3.13b}$$

where $0_{3\times3}$ indicates a matrix of zeros and $I_{3\times3}$ indicates the identity matrix. The discrete time covariance matrix can be derived as

$$Q_d = \int_{\Delta t} F_d G_c Q_c G_c^T F_d^T \, dt, \tag{3.14}$$

where Q_c is the continuous time system noise covariance matrix

$$Q_c = \begin{bmatrix} \sigma_{n_a}^2 & 0 & 0 & 0 \\ 0 & \sigma_{n_{b_a}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{n_\omega}^2 & 0 \\ 0 & 0 & 0 & \sigma_{n_{b_\omega}}^2 \end{bmatrix}. \tag{3.15}$$

Therefore, at the prediction step, before any measurement from the LiDAR or camera arrives, the following steps should be taken in order:

1. Propagate the state variables using (3.8a)–(3.8d);
2. Calculate F_d and Q_d according to (3.13a) and (3.14);
3. Compute the propagated state covariance matrix as $P_{k+1|k} = F_d P_{k|k} F_d^T + Q_d$.

To update the state, the measurements from sensors other than the IMU are required, which is discussed in the following section.

3.3.3.2 Measurement model

The measurement model includes models for two sensor units: the camera and the LiDAR. Each sensor unit is treated as a black box that can output a complete or partial

pose estimation and a covariance matrix that indicates the measurement uncertainty.

The camera sensor unit can estimate a complete 3D pose estimation. However, the employed LiDAR is a 2D scanner and is only able to output an estimation in its plane. To fuse the LiDAR measurements with other sensor measurements, a full 3D pose estimation is required. Our proposed solution to this problem is to use the latest EKF estimation for the three state variables that are not given by the LiDAR estimation in order to extend the LiDAR pose estimation and create a complete state. This will result in a zero residual and accordingly no update of the filter state vector for those elements that the LiDAR is unable to measure.

The following equations govern the position and attitude measurements from a sensor unit,

$$\hat{p}_s = \hat{p} + C(\hat{q})P_{si}, \quad \hat{q}_s = q_{si} \otimes \hat{q}, \tag{3.16}$$

where $\hat{p}_s$ and $\hat{q}_s$ are the (camera/LiDAR) sensor unit estimations of the position and orientation and P_{si} and q_{si} define the transformation between the IMU and the sensor unit. The residuals $\tilde{p}_s = p_s - \hat{p}_s$ and $\tilde{q}_s = q_s \otimes \hat{q}_s^{-1}$ are obtained by calculating the difference between the measured and estimated pose. The measurement error is defined as $\tilde{z} = \begin{bmatrix} \tilde{p}_s & \tilde{q}_s \end{bmatrix}^T$, and according to the EKF formulation, we know that

$\tilde{z} = H\tilde{x}$. The state observation matrix [76] can be computed as

$$H = \begin{bmatrix} H_p & H_q \end{bmatrix}^T, \tag{3.17a}$$

$$H_p = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & -C(\hat{q})^T \lfloor p_{si} \rfloor_\times & 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \tag{3.17b}$$

$$H_q = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} & C(q_{si}) & 0_{3\times3} & 0_{3\times3} \end{bmatrix}. \tag{3.17c}$$

Next, the innovation and the Kalman gain are calculated according to

$$S = HPH^T + R, \quad K = PH^T/S, \tag{3.18}$$

and then the state correction vector is obtained and the states can be updated as

$$\hat{\tilde{x}} = K\tilde{z}, \quad \hat{x} = \hat{x} + \hat{\tilde{x}}. \tag{3.19}$$

Last, the error state covariance is updated according to

$$P = (I_{15\times15} - KH)P(I_{15\times15} - KH)^T + KRK^T. \tag{3.20}$$

3.4 Experimental Results

Synthetic and real world experiments were conducted to investigate the performance of our proposed algorithm. The goal was to demonstrate that fusing the 2D pose estimation from the LiDAR can improve the 3D pose estimation. These random flights were designed to evaluate the pose estimation algorithms.

3.4.1 Simulated experiments

Using the mathematical model in 3.3 random UAV flight paths were generated in MATLAB. These trajectories are shown in Fig. 3.5 and 3.6 as well as the true position and orientation of the UAV for each simulated path.

The three sensors (IMU, camera and LiDAR) form a rigid body; therefore, the transformation between them is constant. The variance values of simulated noise and bias of the sensors are listed in Table 3.1 and it can be seen that the LiDAR unit estimations are more accurate than the cameras, which matches the expected real-world sensors. The LiDAR setup is assumed to be horizontal. This means when the UAV body frame and the reference frame axis are aligned (e.g, at the beginning of the experiment) the LiDAR is able to measure the position in the xy plane. Therefore,

Table 3.1
Variances of noise and bias for different sensors

Sensor	**Gyro.**	**Acc.**	**LiDAR**	**Camera**
Bias	0.001	0.01	-	-
Noise	0.01	0.1	Position 0.03 Attitude 0.003	Position 0.05 Attitude 0.005

no estimation in the vertical z direction can be obtained from the LiDAR. Likewise, it is able to measure the yaw angle, but not the pitch and roll.

The pose is continuously predicted according to equations (3.8a)-(3.8d) at the frequency of the IMU. The update stage however occurs at a lower frequency, i.e., when a measurement is obtained from the LiDAR or camera. After obtaining the LiDAR/-camera unit measurements, they are used in the EKF to correct the pose estimation from the IMU. To evaluate our proposed algorithm, the results for two different scenarios are compared. The first scenario is the 3D pose estimation from the fusion of IMU and camera using EKF that is commonly performed in previous works. In the second scenario, which is the method proposed in this dissertation, a 2D LiDAR is also included. Hence, the update happens any time a measurement from the LiDAR or camera arrives. For the evaluation purpose, the mean errors of each of the estimators are calculated. The estimation errors for x, y and z and also yaw (α), pitch (β) and roll (γ) angles are compared for the simulated paths. The error plots are shown in Figs. 3.7 and 3.8 for two paths, accordingly.

As it is evident from the error plots, especially in the position error, the mean error of

the estimator that only uses the camera unit tends to grow until a new measurement from the camera arrives. This divergence (drift) is because the IMU integrates bias and, if its estimation is not corrected using another sensor, it will deviate from the true value and hence the error grows. As soon as the measurement from the camera or LiDAR arrives, the EKF updates the predicted state and the error drops. The faster the camera/LiDAR unit measurements arrive, the more often the state gets updated resulting in error reduction. The camera unit is slower than the LiDAR sensor unit and therefore it updates the filter at a lower rate compared to the LiDAR. This is the first reason that employing the LiDAR unit (which has a higher measurement rate) will help to improve the pose estimation. Moreover, the frequency of the LiDAR is different than the camera, hence it updates the filter at different time stamps than the camera. Thus, even if the LiDAR was not faster than the camera, employing it, would increase the overall update rate of the filter. Finally, the LiDAR produces very accurate pose estimation compared to the camera and the more accurate the pose estimation from a sensor is, the more improvement can be obtained. The mean error results for the two simulated paths are summarized in Tables 3.2 and 3.3. It can be concluded from the observed error values that fusing the LiDAR with other sensors can considerably improve the accuracy of pose estimation.

Table 3.2
Simulated path 1 mean errors using only camera and using camera and LiDAR

Pose		**Errors**	
		IMU & camera	**IMU, camera & LiDAR**
Position	**X (cm)**	0.152992	0.058311
	Y (cm)	0.275828	0.048763
	Z (cm)	0.184305	0.029428
Orientation	**α (rad)**	0.002749	0.001568
	β (rad)	0.001920	0.001170
	γ (rad)	0.003179	0.001321

Table 3.3
Simulated path 2 mean errors using only camera and using camera and LiDAR

Pose		**Errors**	
		IMU & camera	**IMU, camera & LiDAR**
Position	**X (cm)**	0.210673	0.069537
	Y (cm)	0.232446	0.041411
	Z (cm)	0.176245	0.060828
Orientation	**α (rad)**	0.003696	0.001837
	β (rad)	0.003881	0.002547
	γ (rad)	0.002729	0.002069

3.4.2 Real World Experiments

An experiment was conducted in a motion capture room which is equipped with 12 Vicon cameras that collectively track the trajectory of the sensor platform. This system can provide position measurements accurate to the order of a few millimeters and the orientation accurate to the order of a degree. The pose obtained from the motion capture system was used as ground truth data. The employed sensor platform includes an IMU, a camera, and a LiDAR that was mounted on a hexacopter, as shown

in Fig. 3.9. The sensors form a rigid body and are fixed in their position; hence, once the transformation between them is found at the beginning of the experiment, it can be used for all future calculations and there is no need to include the calibration parameters in the filter state. The hexacopter was too large for flying in the motion capture room; therefore, to use the Vicon camera system for obtaining ground truth, we had to dismount the platform and proceed without using the UAV. In this case, the platform was carried by a person to simulate UAV flight and the sensory data was logged to a file to be processed after the collection.

The IMU used in the experiment was a VectorNav VN-200s which has a 3-axis accelerometer and a 3-axis gyroscope and works at the frequency of 100 Hz. The employed camera was a Point Grey Blackfly BFLY-PGE-13E4C-CS, which is a single lens digital CCD. It was calibrated using the standard checkerboard pattern before the experiment and the whole camera module is able to output 2 frames/sec. For the LiDAR sensor unit, a Hokuyo UTM-30LX-EW was chosen. This LiDAR is a 2D scanner with frequency of 40 Hz. It has a measurement range of 30 meters, a field of view of 270°, and an angular resolution of 0.25°. Measurements from all the three sensors (IMU, camera and LiDAR) were time synchronized. The output frequency of the Vicon motion capture system is 100 Hz, which is the same as the IMU. Hence, in order to match and compare the ground truth data with the estimated values, we time-synchronized the data by correlating the yaw angle measurements. The noise variance values for the sensor units are listed in Table 3.4.

Table 3.4
Noise variance for different sensor units

Sensor	**Gyro.**	**Acc.**	**LiDAR**	**Camera**
Noise	0.01	0.1	Position 0.07 Attitude 0.06	Position 0.2 Attitude 0.1

Table 3.5
Motion capture room experiment mean errors using only camera and using camera and LiDAR

Pose		**Errors**	
		IMU & camera	**IMU, camera & LiDAR**
Position	**X (cm)**	0.106457	0.020331
	Y (cm)	0.12587	0.018429
	Z (cm)	0.228594	0.061857
Orientation	$\boldsymbol{\alpha}$ **(rad)**	0.003458	0.001222
	$\boldsymbol{\beta}$ **(rad)**	0.006589	0.005713
	$\boldsymbol{\gamma}$ **(rad)**	0.008344	0.007905

The true trajectory of the mobile platform obtained from the Vicon camera system in the motion capture room is shown in Fig. 3.10, as well as the true position and orientation. To evaluate the results, the same procedure as the simulation was employed. Two scenarios were considered. In the first scenario only the camera was used to correct the IMU estimation, and in the second scenario both the LiDAR and camera were employed. As in the simulated experiment, the mean errors for both cases were calculated and compared to evaluate the proposed algorithm. The obtained error results of the EKF using only the camera and the EKF employing both the camera and LiDAR are summarized in Table 3.5. It can be concluded that using the LiDAR has noticeably decreased the error and hence improved the localization accuracy.

3.5 Summary and Future Work

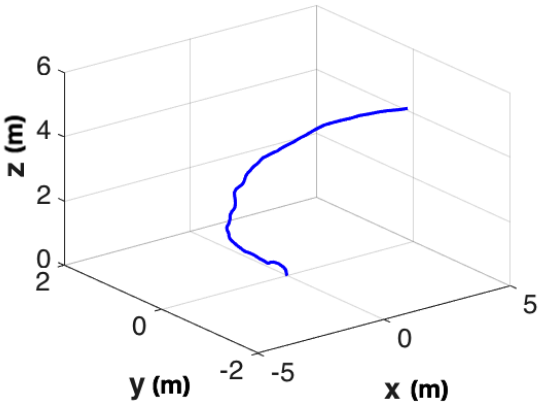
In this chapter, a novel method for full pose estimation of a mobile platform in an unknown GPS-denied environment using a sensor-fused IMU/camera/LiDAR was presented, employing no prior knowledge of the environment map. A 2D LiDAR was used along with an IMU and a camera to estimate the pose of a UAV platform.

The camera and LiDAR modules were treated as black boxes that can be replaced with any other sensor module as long as the sensor module can estimate the pose with a known certainty. Therefore, the proposed algorithm is not limited to any specific type of sensors. Furthermore, this modular approach has the advantage that gives a general solution to the sensor fusion problem and hence, our method can be used with any existing pose estimation algorithm including ready-to-use solutions for the camera and LiDAR. Using simulation and real word experiments, the proposed method was compared against the state of the art for indoor localization that uses a camera to correct the IMU estimations and it was demonstrated that taking advantage of the introduced approach and including the 2D LiDAR, considerably improves the localization accuracy. This improvement was demonstrated both in simulation, where the trajectory was generated randomly and thus there were sudden changes in the UAV motion, and also in a real world experiment, with a more realistic UAV flight path. It should be mentioned that, in general, a path with less sudden orientation

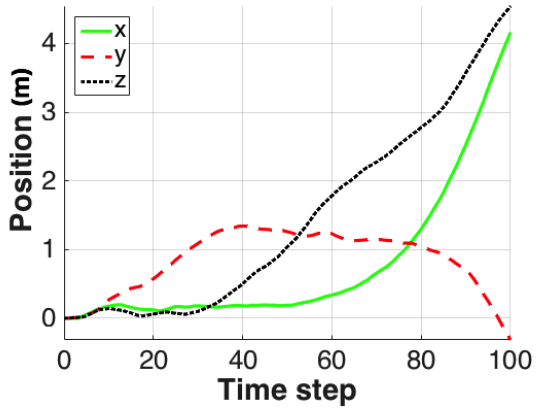
change will result in better pose estimation. This is because the chosen fusion tool was EKF, which linearizes the state between the two sensor readings, and the closer the path is to linear in the update interval, the smaller the linearization error will be; therefore, the obtained results will be more accurate.

From the simulation and experiment it can be concluded that, depending on the platform angles, fusing the LiDAR estimation can have more effect on some dimensions than the others. This is because the LiDAR only can estimate 3-DOF, which means it can improve the estimated pose more along the axes that are better aligned with its body frame than the those that are not. In our experiment, the LiDAR significantly improved the position estimates and the yaw because it was mostly kept in a horizontal plane for the real-world experiment (as it would be on a UAV). Generally, by including the LiDAR, the overall pose estimation will always be improved.

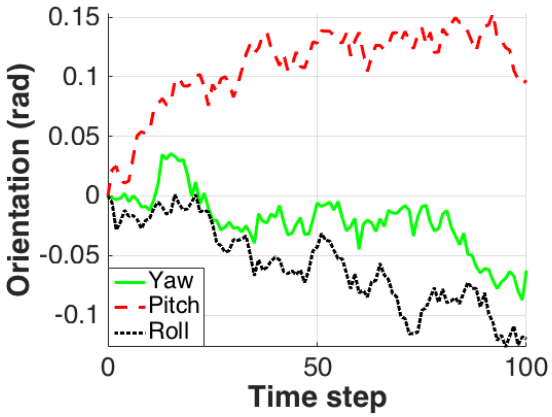
The covariance matrices R and Q are assumed to be constant in this work. In future, the estimation can be improved by dynamically updating these matrices during the estimation process. One idea is to use a fuzzy inference system, to continuously update these matrices as the measurements from sensors arrive which is the future work for our research.



(a) Simulated path 1

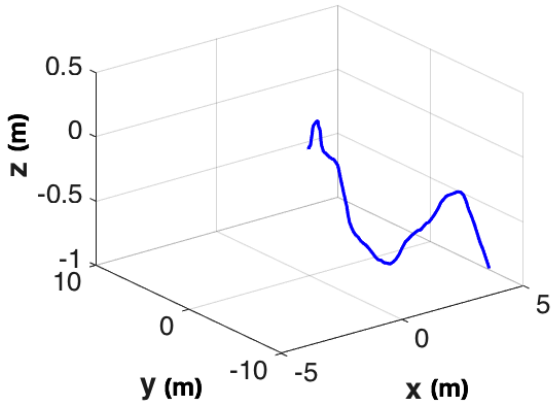


(b) Simulated path 1 true position

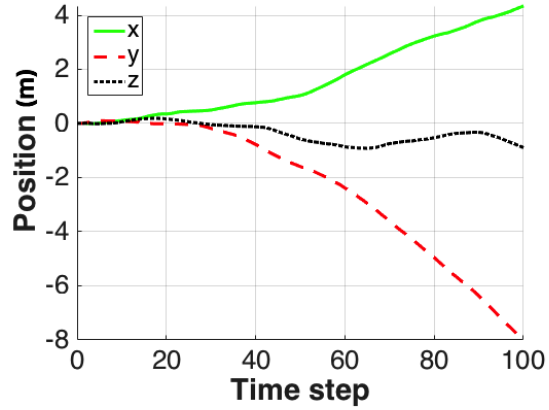


(c) Simulated path 1 true orientation

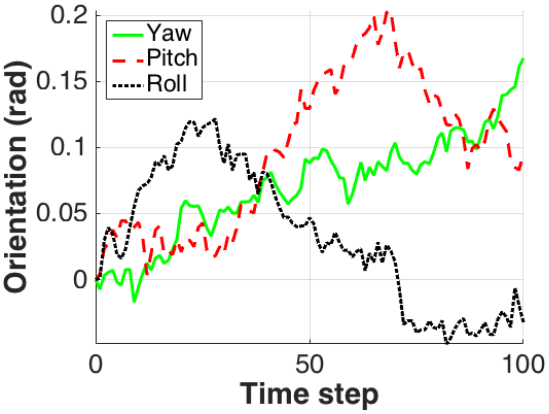
Figure 3.5: UAV simulated path 1 trajectory



(a) Simulated path 2

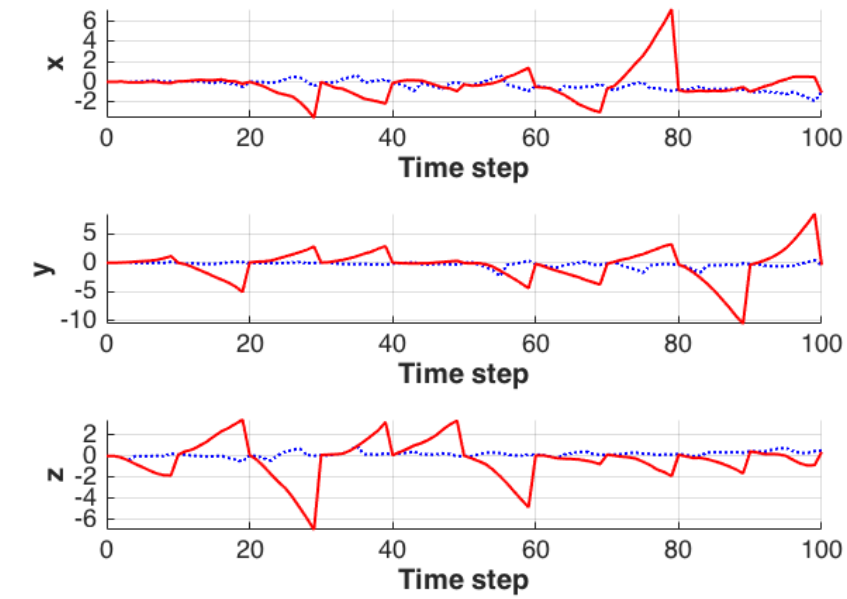


(b) Simulated path 2 true position

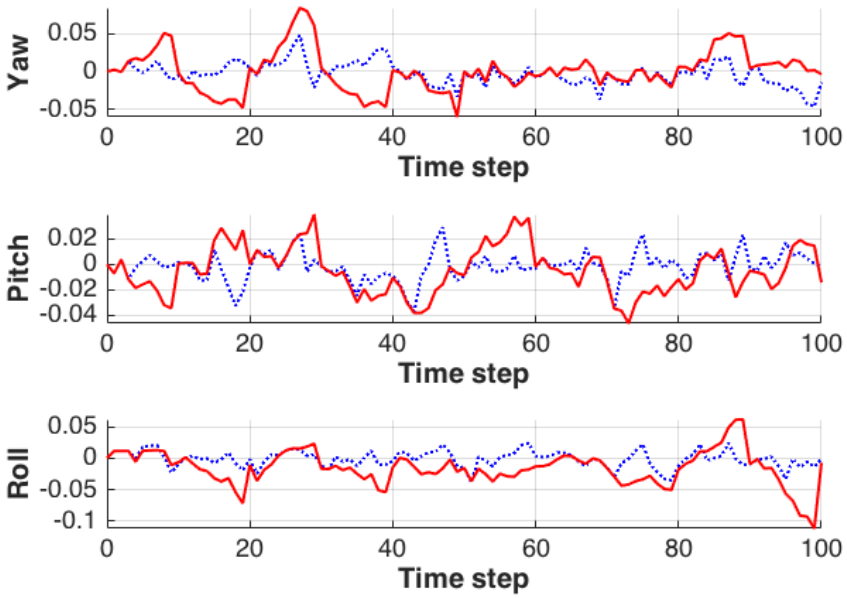


(c) Simulated path 2 true orientation

Figure 3.6: UAV simulated path 2 trajectory

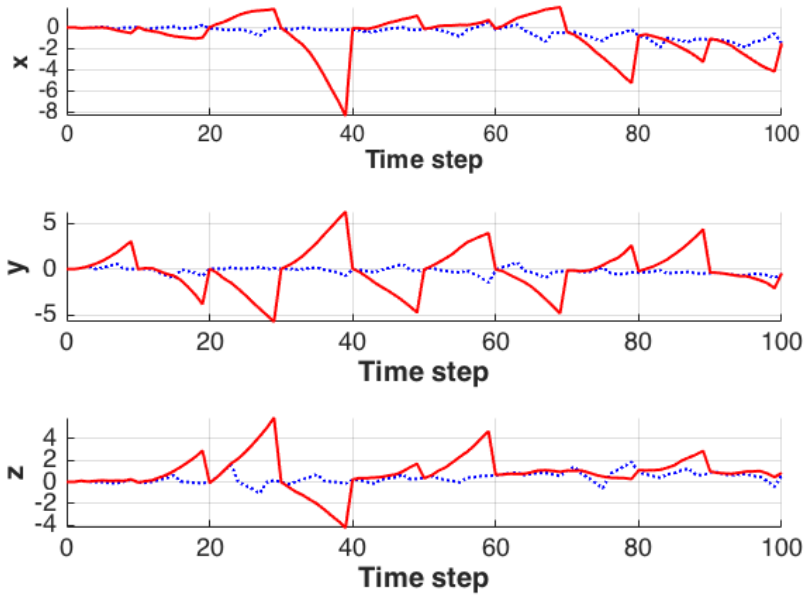


(a) Position error comparison (cm)

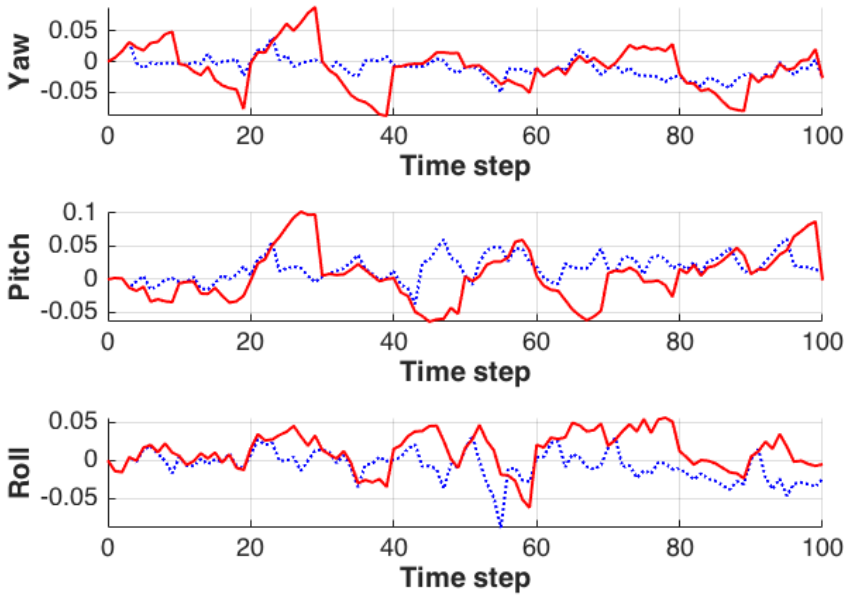


(b) Attitude error comparison (rad)

Figure 3.7: Path 1 errors using only camera (red solid plots) and using camera and LiDAR (blue dotted plots)

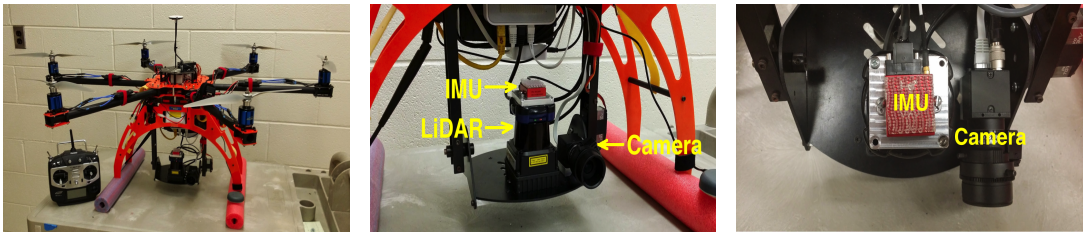


(a) Position error comparison (cm)



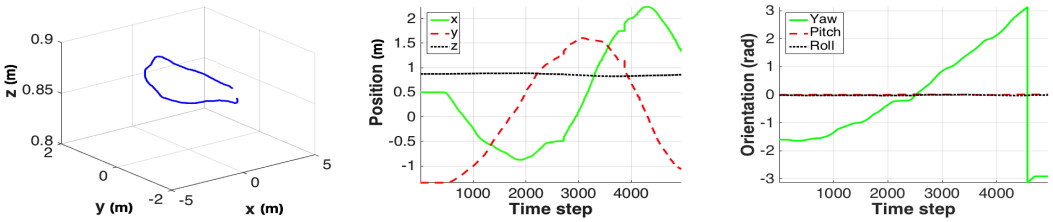
(b) Attitude error comparison (rad)

Figure 3.8: Path 2 errors using only camera (red solid plots) and using camera and LiDAR (blue dotted plots)



(a) Sensors mounted on the UAV (b) Front view (c) Top view

Figure 3.9: Sensor platform



(a) Sensor platform path (b) True position (c) True orientation

Figure 3.10: Sensor platform trajectory

Chapter 4

Fuzzy Adaptive Extended Kalman Filter for Robot 3D Pose Estimation

4.1 Introduction

Obtaining an accurate pose (position and orientation) of a mobile object is usually done by fusing the information acquired from several mounted sensors. Each of the sensors have different characteristics that make them suitable for particular applications and situations. Some sensors are noisy and inaccurate but able to work in a

variety of conditions, whereas others are more accurate but are constrained in their operating conditions. In addition to that, depending on the limitations that every specific application dictate, such as the environment of the work, size and weight of the sensor, etc., and also the available budget, it is not always possible to employ the best and most accurate sensor existing in the market. Thus, many application employ multiple sensors to obtain the best pose estimation by combining the information acquired from different sensor readings. This is called sensor fusion. It combines the strengths of multiple sensors to correct the deficiencies of single sensors [63]. Various approaches from simple averaging to more complicated algorithms such as *Extended Kalman Filters* (EKF) have been proposed to implement a reliable sensor fusion platform.

The focus of this dissertation is to obtain the pose of a robot equipped with an *inertial measurement unit* (IMU), a vision camera, and a LiDAR. IMUs are very accurate over short distances (times). However, their measurements contain bias which is integrated over time and thus causes the pose estimated from a single IMU to quickly diverge from the true value. Therefore, it is usually fused with another sensor to overcome this issue. The most commonly employed sensosr with IMU are GPS for outdoor and optical camera for indoor localization. In some applications there is also a LiDAR employed for mapping purposes, but not necessarily for localization.

Since this dissertation focuses on indoor localization, i.e., a GPS-denied environment,

the advantage of GPS in absolute localization is not given. However, due to features of indoor environments, an accurate pose estimation can be obtained from a 2D LiDAR (at least in the 2D plane). However, there are 6 *degrees of freedom* (6DOF) in a 3D environment and since the 2D LiDAR estimation only gives 3DOF, in most 3D applications, LiDAR is not used for pose estimation and its utilization remains limited to mapping. In our previous work [26], we proposed a method and verified that a 2D LiDAR indeed can be used to improve the overall 3D pose estimation of a UAV platform. However, the movement of the UAV had to be smooth and any quick movement that causes its trajectory to diverge considerably from being on a plane in the time interval between two consecutive LiDAR readings could result in large estimation error. In this chapter, we propose a method to detect the divergence of the trajectory using a *Fuzzy Inference System* (FIS) and implement a sensor fusion method, which leads to more accurate results compared to our previous work and reduces the restrictions on the robot movement. Essentially, we introduce a method that adjusts EKF parameters according to the reliability of the estimation obtained from the sensor readings.

The rest of this chapter is organized as follows. Section 4.2 is dedicated to a literature review followed by the problem statement in Sec. 4.3. Then the employed sensors and configuration setup is explained in Sec. 4.4. In the next section, the main algorithm is explained in details and the experimental results are summarized in Sec. 4.6, followed by a conclusion and proposed future work.

4.2 Previous Work

Sensor fusion was first proposed in the 1970s and was initially employed in military applications [69]. Since then, different techniques such as the *central limit theorem* (CLT), Bayesian networks, and *Extended Kalman Filters* (EKF) have been proposed to perform fusion. Fuzzy set theory, introduced by Lotfi Zadeh in 1965 [80], provided researchers with a powerful tool for a variety of applications such as multi-source information fusion in general [79], and more specifically sensor fusion.

Recently, employing fuzzy techniques for sensor fusion has become more popular. These methods can be seen from two major perspectives: (i) as the main method that can independently fuse the data from different sensors based on the their characteristics, or (ii) as a tool to be employed with another techniques, such as EKF, for their improvement and optimization.

Employing fuzzy techniques as an independent fusion method has been less common than using it associated with other algorithms, especially EKF. However, in scenarios where there is no knowledge of the mathematical system model or where the system is very complicated, such that it is hard or expensive to find a good model, fuzzy approaches can be extremely useful. Commonly used techniques in this area are various weighted averaging methods [12]. Akhoundi et al. introduced a fuzzy multi-sensor

data fusion method focusing on sensor characteristics [2], which is a good example of this approach. In the paper, the authors design a rule-based fuzzy system based on sensors' accuracy and frequency response to fuse the data acquired from various sensors with different bandwidth and accuracy properties. A similar approach was taken before this paper by Goebel et al. [31], but they did not clarify the sensor characteristics or the fuzzy rules that they employed. The methods that employ fuzzy logic as an independent fusion tool are not limited to fusion based on the sensors characteristics. Some other proposed algorithms are developed only based on the sensor measurements, without having any knowledge of their specifications. For example, fuzzy clustering was employed in [28] for sensor fusion as well as sensor fault detection. The authors suggested to use fuzzy cluster centers for signal separation followed by center-of-gravity defuzzification for sensor fusion. As a criterion for fault detection, they employed the residual between the fused signal and each sensor measurement. They ran a simulation to evaluate the proposed algorithm; however, there is no numerical comparison with any other method to verify by how much the precision was improved. A similar approach was taken by Blank et al. [12] to implement a soft voter, which was based on the original work in [36]. In this paper, the authors developed their algorithm based on the distance (dissimilarity) between each pair of sensor readings, where the sensor that does not agree with the others is more likely to be faulty and receives a lower score/weight and hence less impact on the overall output.

In a different approach, fuzzy logic is employed in concert with another fusion method to improve the functionality. A popular technique is the combination of fuzzy methods with Kalman filtering. The focus of this dissertation is also to optimize the well known EKF, which is derived from the original *Kalman Filter* (KF) proposed in 1960 [41]. The EKF is one of the most popular methods in the sensor fusion area [69] and has been employed by many researchers to fuse sensors such as GPS, IMU, cameras, etc., for both indoor and also outdoor localization [11, 15, 16, 27, 33, 35, 55, 67]. Even though the EKF allows for superior performance, its performance and optimality depends on availability of the system model and *a priori* knowledge of the process and observation/measurement noises, and in most applications their consistency. However, in many cases (such as dynamic sensor networks) the details of the system model are not available or are too complex or expensive to obtain [12]. Unfortunately, if the prior knowledge of the systems model statistics is imprecise or are dynamic during operation, there will be a significant degradation in the filter performance and the output can significantly diverge from the true value. Therefore, researchers have been proposing solutions to improve the EKF results during non-ideal conditions. Hence, the question is: is it possible to use the powerful EKF in a scenario where one or both of the sensor covariance matrices are dynamic? Fortunately, introducing fuzzy set theory provided scientists with an incredible tool to perform on-line modification of the covariance matrices.

An interesting approach for modifying the EKF can be found in [1], where authors

aimed to introduce a fuzzy-based method for GPS-IMU fusion that estimates the gyroscope and accelerometer biases of the Kalman filter error states when the GPS signal is not available. One of the most popular techniques that employ fuzzy techniques in an EKF is covariance matching [32, 78]. In this method, the theoretical and actual measurement covariances are compared and a variable called *degree of mismatch* (DoM) is defined that indicates the difference between the two. If this difference is small, it means the current observation covariance is precise and does not need a large modification, but as DoM grows, the covariance has to be modified. Therefore, the DoM is used as the input to a fuzzy system whose output is the needed covariance adjustment. The comparison process is continually performed and dynamically updates the covariance. In [29, 50], the authors employed the same method but they went one step further and developed a sensor fault diagnostic and recovery algorithm based on the ratio of the actual and theoretical values of the measurement covariance. Ideally, these two covariances are equal, and the sensor fault diagnostic and recovery algorithm checks this property and attempts to compensate when equal covariance is not the case. Even though fuzzy techniques in this application have been mostly employed for sensor fusion, their application in the sensor related area is not limited to just fusion. Some researchers have employed fuzzy systems at different levels for noise reduction and smoothing [44, 45]. The work of Kyriakoulis et al. [45] is a good example of this type of application where the authors employed a three level fuzzy system to create a hierarchical pose estimation algorithm using an IMU and a visual

camera. The first level is dedicated to the IMU noise reduction, the second level fuses the data from the first level and the data obtained from the camera, and the third level smooths the output of the prior level using the previous system state.

Using the *DoM* as the input to a fuzzifier has become a popular approach that can be used as a sensor measurement quality identifier. Nevertheless, it is not the only sensor quality identifier that can be used. Depending on the application, other quantities may be used to as input to the fuzzifier. As an example, Subramanian et al. [71] used a fuzzy logic enhanced Kalman filter to combine the information from IMU, optical camera, LiDAR, and a speed sensor to guide a vehicle in citrus groves. In their specific application, and based on the system physical characteristics and limitations, they can define if the output of each sensor is *reasonable* with respect to the distance of the sensor from a detected object, and then decide how reliable is every sensor. Then, using the sensor reliability, the fuzzy system decides on the weight of each sensor in the final fused output. This work is perhaps the most state-of-the-art in relation to our proposed method; however, it is only appropriate for 2D navigation in a ground vehicle.

In this chapter, a fuzzy modified EKF method is proposed to estimate the 3D pose of a mobile robot, i.e., a UAV. The proposed fuzzy system uses *DOM* and other application specific quality values to quantify the reliability of the sensors, which are then combined for optimal localization accuracy. Next, the problem that we are

trying to solve is explained in more detail.

4.3 Problem Statement

The problem that we are proposing a solution for in this chapter is obtaining an estimation of the 3D pose of a robot which is equipped with three sensors: IMU, vision camera, and 2D LiDAR. As mentioned in Sec. 4.2, there are already different solutions for the 3D pose estimation problem using 3D sensors. Nevertheless, employing a 2D sensor for 3D pose estimation without any prior knowledge of the environment is a new approach. There might be different reasons for using a 2D sensor instead of a 3D sensor. Typically, 2D sensors have lower cost and weigh less as compared to similar 3D models. Furthermore, there are situations in which an existing 2D sensor is already being used for a different purpose, such as when a 2D LiDAR is employed for mapping or for autonomous vehicles. In such cases, the sensor can be used to improve the 3D pose estimation at no additional cost.

In Chapter 3 [26], we proposed a solution to the aforementioned problem. However, our previous technique requires the movement of the robot to be smooth and free of quick motions so that it can be assumed that the robot almost moves on a plane during the interval between two LiDAR readings. In this chapter, we try to overcome

this limitation by proposing a fuzzy method for sensor fusion and introducing a general solution to the problem of 3D pose estimation improvement using a 2D sensor. In addition, the algorithm in the previous chapter was EKF based, which requires having *a priori* knowledge of the measurement covariance matrix and its consistency over time. The objective of this chapter is to dynamically update the measurement covariance matrix using a fuzzy inference system and achieve an optimized sensor fusion method that can work in a variety of different situations.

4.4 Robot Platform and Configuration

4.4.1 Sensors

Three sensors with different characteristics were used in this work: (i) an *Inertial Measurement Unit* (IMU), (ii) a vision camera, and (iii) a 2D *Light Detection And Ranging* (LiDAR) sensor. Pose estimation (whether 6DOF or 3DOF) can be obtained from each sensor separately. A sensor along with its pose estimation algorithm is called a sensor unit/module and is treated as a black box. In this dissertation, the raw measurement data from the IMU is used, but for the camera and LiDAR the output of the respective sensor module is a pose estimation. We now discuss these sensors and modules in more detail.

4.4.1.1 IMU

An IMU is a device that measures the velocity, orientation, and gravitational forces using a combination of data from accelerometers, gyroscopes, and magnetometers. In this work, only the raw data from the accelerometer and gyroscope are employed; it is assumed that magnetometer data is unavailable or unreliable, which is often the case in GPS-denied environements. The 3-axis accelerometer measures the acceleration along each of the x, y, and z axes in the IMU body frame, and the 3-axis gyroscope measures the angular velocities of each axis, respectively.

The IMU output rate is high compared to the other sensors in this dissertation; it has a larger update frequency and therefore it is faster. It is an accurate sensor for short periods of time but its output is subjected to bias which accumulates over time and causes quick divergence from the correct trajectory—this is often called *drift*. Mitigating this drift is accomplished by fusing this information with other sensor readings such as GPS, camera, or LiDAR. The model for the IMU is

$$\boldsymbol{\omega}_t = \boldsymbol{\omega}_m - \mathbf{b}_\omega - \mathbf{n}_\omega, \quad \mathbf{a}_t = C(\mathbf{q}_t)(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) + \mathbf{g}, \tag{4.1}$$

where $\boldsymbol{\omega}$ and $\mathbf{a}$ are angular velocity and linear acceleration, respectively, and the subscripts t and m denote the true and the measured values, respectively. The variable $\mathbf{g}$ is the gravity vector, $\mathbf{q}_t$ is the quaternion equivalent of the true attitude, and $C(\mathbf{q}_t)$

is the rotation matrix for $\mathbf{q}_t$. In (4.1), $\mathbf{n}_\omega$ and $\mathbf{n}_a$ represent the noise of the gyro and accelerometer, respectively, and $\mathbf{b}_\omega$ and $\mathbf{b}_a$ denote the relevant biases, which are random processes. More details about this IMU model can be found in [26].

4.4.1.2 Camera

For the purpose of this work, a vision camera projects a 3D point in the space onto the image plane of the camera. The most widely used model for reconstructing 3D points from their projected camera image is the pinhole model [22].

Compared to an IMU, a camera's output frequency is lower as it is a slow sensor. A complete 3D pose can be estimated using camera images; however, the estimated pose is not very accurate. It is especially difficult to estimate the distance traveled along the camera focal axis for short trajectories. This is the motivation that encouraged us to employ LiDAR—which is a highly accurate range sensor—whenever possible to improve the pose estimated by camera.

4.4.1.3 LiDAR

LiDAR measures range to nearby objects by the time interval between an emitted laser pulse and reception of the reflected pulse from an object. The measured distance

and the beam angle can be used together to calculate the coordinates of the object in the LiDAR local frame. Using a LiDAR, one can create high resolution maps and by comparing these maps from scan-to-scan, one can estimate changes in pose [22].

LiDARs are available in 2D and 3D models. A 2D LiDAR, which is the subject of the interest in this dissertation, radially scans the environment in a plane. Thus, 2D LiDAR essentially measures a slice of the 3D world. Because of this, three out of the six parameters that are necessary to fully localize an object in 3D world cannot be obtained by a 2D LiDAR.

The operation of a 3D LiDAR is similar to that of a 2D LiDAR except that instead of one measurement plane, it has two perpendicular rotational motors allowing measurement in multiple planes simultaneously. As a result, a complete 3D pose can be estimated from its data, albeit at a high computational cost. Compared to 2D LiDARs, 3D LiDARs are bulky, heavy, and expensive; these properties inhibit their application for payload-limited platforms or for applications with limited budgets. For example, small UAVs are typically limited to very small payloads, which make most 3D LiDARs inappropriate.[1]

Therefore, it can be concluded that each type of sensor has strengthes and weaknesses

[1] As technology advances and UAV payload increases and LiDARs get smaller, the limitation will be alleviated, but there will always be a platform that is too small for conventional 3D LiDARs.

and, thus, sensor fusion can be beneficial to take advantage of all the sensors' capabilities and compensate for their deficiencies to obtain more reliable, complete, and robust pose estimation.

4.4.2 Geometry definition

Before we introduce the sensor fusion algorithm, we define the geometry of the pose estimation problem. The sensors and the platform (UAV) form a rigid body which means the transformation between them remains constant and needs to be acquired only once. Hence, without loss of generalization, we assume that the sensors are co-located.

A 3D pose is fully defined by three translation coordinates: x, y and z; and three rotation angles: yaw (α), pitch (β) and roll (γ). The z axis points up, the x axis points forward, and the y axis points to the right with respect to the robot's frame. The yaw angle is defined as a rotation about the z axis, pitch as a rotation about the y axis, and roll is a rotation about the x axis.

In order to examine the robot trajectory and decide if its movement can be considered to be planar in the time interval between the two consecutive LiDAR readings, three

variables are defined as follows,

$$\Delta z = |z_i - z_{i-1}|, \tag{4.2a}$$

$$\Delta \beta = |\beta_i - \beta_{i-1}|, \tag{4.2b}$$

$$\Delta \gamma = |\gamma_i - \gamma_{i-1}|, \tag{4.2c}$$

where subscript i denotes the current time-stamp of the LiDAR measurement and $i-1$ is the previous time-stamp of LiDAR measurement. It should be noted that the changes in z, β, and γ are considered with respect to the LiDAR local frame. If the UAV moves on a plane, its pose can be fully defined by x and y coordinates and yaw angle (α); z, pitch (β), and roll (γ) will not change. Hence, if any of the z, β, or γ changes significantly in the time between the two readings, it means that the trajectory is not on a plane and the pose estimation from LiDAR has considerable error or is not valid. This will be discussed more in detail next.

4.5 Methodology

The proposed method for the sensor fusion problem described in Sec. 4.3 is an *adaptive Kalman filter* (AKF) which uses a FIS to modify the standard EKF (shown in Fig. 4.1). The EKF continuously runs at the update frequency of the IMU until a LiDAR or camera reading is obtained, then the FIS evaluates the reliability of the

LiDAR data and determine its contribution to the final estimation.

We now explain the proposed method in detail. First, the EKF, which is the foundation of our algorithm, is discussed and then the fuzzy system that modifies the EKF is introduced.

4.5.1 Extended Kalman Filter

The Kalman filter—and its variations, such as the EKF—is one of the most popular methods for sensor fusion. However, in practical applications, assumptions are frequently made that can result in filter instability or limit the application of the filter. EKFs have been widely-used for sensor fusion and pose estimation. The EKF is the non-linear version of the Kalman filter that linearizes the state about the estimation of the current mean and covariance based on the assumption that the process is fed by sensors with Gaussian noise. In order to use an EKF, the system first has to be mathematically modeled which can be summarized as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \boldsymbol{\omega}_{k-1}, \quad \mathbf{z}_k = h(\mathbf{x}_k) + \boldsymbol{\nu}_k, \tag{4.3}$$

where subscript k denotes the time step and $\boldsymbol{\omega}$ and $\boldsymbol{\nu}$ are the zero-mean Gaussian process and observation/measurement noises with covariances Q and R, accordingly. The variable $\mathbf{x}_k$ is the state at step k, $\mathbf{u}$ is the input control vector, and $\mathbf{z}_k$ is the

expected sensor measurement at time k. The function f is the model used to predict the state from the previous estimation and similarly h is computes the expected measurement from the predicted state. However, f and h cannot be applied to the covariance directly. Instead, a matrix of partial derivatives (the Jacobian) is computed at each time step which linearizes the prediction around the current estimate.

The EKF consists of two main steps, *predict* (time update) and *update* (correct/measurement update), that can be summarized as follows:

Predict:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}), \qquad \text{State estimate} \tag{4.4}$$

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}, \qquad \text{Covariance estimate} \tag{4.5}$$

Update:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}), \qquad \text{Measurement residual} \tag{4.6}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k, \qquad \text{Innovation} \tag{4.7}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1}, \qquad \text{Kalman gain} \tag{4.8}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k \tilde{\mathbf{y}}_k, \qquad \text{Update state estimate} \tag{4.9}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}, \qquad \text{Update covariance estimate} \tag{4.10}$$

where

$$F_{k-1} = \left.\frac{\partial f}{\partial x}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, u_{k-1}}, \quad H_k = \left.\frac{\partial h}{\partial x}\right|_{\hat{\mathbf{x}}_{k|k-1}}.$$

In this work, the state of the system is a 16-length vector which is represented by

$$\mathbf{x} = \begin{bmatrix} \mathbf{p}_{\{1\times3\}} & \mathbf{v}_{\{1\times3\}} & \mathbf{q}_{\{1\times4\}} & \mathbf{b}_{\omega\{1\times3\}} & \mathbf{b}_{a\{1\times3\}} \end{bmatrix}^T.$$

It contains the position of the IMU $\mathbf{p}$ with respect to the global frame, IMU linear velocity $\mathbf{v}$, and the quaternion $\mathbf{q}$, which represents the rotation of the IMU from the reference frame. The two remaining parameters in the filter state express the gyroscope and accelerometer biases, $\mathbf{b}_\omega$ and $\mathbf{b}_a$, along each axis. The details about the EKF equations and the state estimation can be found in Chapter 3 [26]. The state is *predicted* based on the IMU readings is *updated* at each the camera or LiDAR measurement. It should be noted that since the IMU has a higher frequency than the camera and LiDAR, the state vector will be updated every few (IMU) time steps rather than every single time step.

The observation noise covariance R indicates the accuracy of the measurement and in many applications is assumed to be constant. Although this assumption might be a valid approximation in some situations, it is not a good assumption in the case of using a 2D LiDAR for a 3D localization algorithm. As long as the robot is moving approximately on a plane, the pose calculated from LiDAR readings is very accurate; however, as the trajectory deviates from being on a plane, the LiDAR estimations

become more erroneous. Therefore, in order to employ the LiDAR in the fusion algorithm, we should know how close the trajectory of the robot is to being on a plane. Luckily, fuzzy set theory gives us a very powerful tool to perform on-line modification of the observation covariance matrix, resulting in an *adaptive Kalman filter* (AKF) for the stated problem. Hence, the proposed method employs fuzzy logic to update R based on the robot trajectory and covariance matching at each LiDAR measurement.

The model for the robot movement can be described by the following equations:

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{4.11a}$$

$$\dot{\mathbf{v}} = \mathbf{a}, \tag{4.11b}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \otimes \bar{\boldsymbol{\omega}}, \tag{4.11c}$$

$$\dot{\mathbf{b}}_\omega = \mathbf{n}_{b_\omega}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{b_a}. \tag{4.11d}$$

where ω and a are respectively the angular velocity and linear acceleration measured by the IMU and, by definition, $\bar{\omega} = [0 \quad \omega]$. b_ω and b_a denote the gyroscope and accelerometer biases, which are random processes, and their dynamics are modeled by n_{b_ω} and n_{b_a}. Enthusiastic readers can find more detailed explanation and derivation in [26].

The IMU and camera are fused using a standard EKF approach according to Eqs. (4.4)

to (4.10), which is fully explained in our previous work [26]. However, in order to include LiDAR in the EKF algorithm, a fuzzy system is proposed to determine its reliability and adjust its contribution accordingly.

4.5.2 Fuzzy System Design

In this section, the variables needed for the design of fuzzy system are first defined and then the fuzzy system is introduced.

4.5.2.1 Prerequisites

The purpose of the proposed *fuzzy logic system* (FLS) is to estimate LiDAR reliability and to supervise its contribution to the final output dynamically. This task is performed through modification of diagonal elements of the measurement covariance matrix R upon a LiDAR measurement. The adjustment is based on two criteria: (i) evaluation of the trajectory of the robot between two LiDAR readings to evaluate if its movement has been planar, and (ii) the difference between the actual and theoretical measurement covariances as an indication of measurement reliability, which is calculated through application of a covariance matching technique. Based on these assessments, a decision is made on how to adjust the value of R. As it can be concluded from equation. (4.7), increasing R will cause an increase in the innovation S_k, which

consequently leads to decreasing the Kalman gain K_k, according to Eq. (4.8); therefore the LiDAR has a smaller contribution to the final pose estimate. Consequently, as the measurements from the LiDAR become more unreliable, the covariance R will be increased and, hence, the final output relies less on the LiDAR readings.

A variable ΔR is defined to modify the observation noise covariance according to

$$R_i = R_{i-1} + \Delta R, \tag{4.12}$$

where i indicates the time step. The fuzzy system computes the variable ΔR.

In order to evaluate whether the trajectory of the robot is planar, three variables, Δz, $\Delta\beta$ and $\Delta\gamma$ (see Sec. 4.4.2), were introduced that accordingly indicate the change in z direction, and pitch and roll angles between two consecutive LiDAR readings. The idea is that if these values are smalle, the closer the path is to being on a plane and ultimately if all the three values are zero, it means that the robot's motion is planar between the two LiDAR readings. Large values of these variables indicate non-planar robot motion.

To implement the covariance matching technique, the actual covariance has to be calculated [56]. If Inn_k represents the innovation—the difference between the measurement and the estimated value) at time stamp k—the actual covariance is defined

as

$$C_{Inn_k} = \frac{1}{M} \sum_{j=j_0}^{k} Inn_k Inn_k^T, \tag{4.13}$$

where M is the size of the averaging window and j_0 is the first sample inside the estimation window. The window size M is usually chosen empirically; in this work $M = 10$. The theoretical value of the measurement covariance S_k is calculated from Eq. (4.7) and the difference between these values gives the *degree of mismatch* (DoM),

$$DoM_k = S_k - C_{Inn_k}. \tag{4.14}$$

The logic behind this method is that if the DoM is almost zero, this shows that both covariances are about the same and R should remain nearly unchanged. On the other hand, if the DoM is positive, it indicates that the actual covariance value is smaller than its theoretical value and R should be decreased. Finally if the DoM is negative, it means that the actual covariance is larger than the theoretical value and R has to be increased. Therefore, the fuzzy system should follow these rules:

1. $IF \quad DOM \simeq 0, \quad THEN \quad \Delta R \simeq 0;$
2. $IF \quad DOM > 0, \quad THEN \quad \Delta R < 0;$
3. $IF \quad DOM < 0, \quad THEN \quad \Delta R > 0.$

We now discuss the FLS in detail.

4.5.2.2 Fuzzy Logic System (FLS)

An FLS can be defined as a nonlinear method of mapping input data to an output. Implementing a standard FLS consists of three main steps:

1. Fuzzification: conversion of crisp input values (information sources) into fuzzy sets using fuzzy linguistic variables;
2. Inference: derivation of fuzzy output by execution of a set of rules; and lastly,
3. Defuzzification: mapping the resulting fuzzy output to a final crisp output.

In this work, the FLS inputs are Δz, $\Delta\alpha$, and $\Delta\beta$, as introduced in Eq. (4.2), and DoM, which is calculated based on (4.14). The crisp output is the variable ΔR which is used to adjust R according to Eq. (4.12). The block diagram of the FLS is shown in Fig. 4.1(b).

The *membership functions* (MFs) are all triangular and map the non-fuzzy inputs to fuzzy linguistic terms and similarly for the output. There are a total of four inputs and one output. The MFs assign a degree of membership to the universe of discourse as demonstrated in Fig. 4.2.

After fuzzification, the Mamdani *fuzzy inference system* (FIS) employs a rule base to control the output variable. The rules are simple IF-THENs that are constructed

by all possible combinations of Δz, $\Delta\beta$, $\Delta\gamma$, and DOM. To generate the rules, the deviation of the trajectory from being planar is investigated and, furthermore, the quality of the LiDAR sensor estimation is evaluated based on the value of DOM. For example, one of the rules can be stated as

$$IF\ (\Delta z = 0)\ AND\ (\Delta\beta = 0)\ AND\ (\Delta\gamma = 0)$$
$$AND\ (DOM < 0), \quad THEN\ \Delta R = 0.$$

Note that to combine the individual rules, the fuzzy set operations AND and OR are used to generate the rule base. The t-norm and t-conorm we use for these operations are min and max, respectively. After evaluating the result of each rule, these results should be combined to generate the outcome of the FIS. The employed aggregation method is max which is widely used in FISs. The complete fuzzy rules, which can be obtained from all possible combinations of the inputs, are listed in Table 4.1.

The result of the inference system is a fuzzy value. To produce a crisp final output (ΔR), the output of the FIS has to be defuzzified. The deffuzification is performed based on the output MF shown in Fig. 4.2(d) and employs the centroid method,

$$U = \frac{\int_{min}^{max} u\mu(u)du}{\int_{min}^{max} \mu(u)du}, \tag{4.15}$$

where U is the final output, u is the output variable, and μ is the output MF.

Table 4.1
Fuzzy rules

Input				**Output**
Δz	$\Delta\beta$	$\Delta\gamma$	***DOM***	ΔR
zero	zero	zero	zero	zero
			positive	small negative
			negative	small positive
		small	zero	small positive
			positive	zero
			negative	large positive
		large	_	large positive
	small	zero	zero	small positive
			positive	zero
			negative	large positive
		small	_	large positive
		large	_	large positive
	large	_	_	large positive
small	zero	zero	zero	small positive
			positive	zero
			negative	large positive
		small	_	large positive
		large	_	large positive
	small	zero	_	large positive
		small	_	large positive
		large	_	large positive
	large	_	_	large positive
large	_	_	_	large positive

In summary, the fuzzy systems is designed such that the further the deviation of the trajectory between two consecutive LiDAR readings is from being planar, the larger ΔR will be and, as a result, the value of R will be increased, leading to less contribution of the LiDAR for the pose estimation.

4.6 Experimental Results

Synthetic and real world experiments were conducted to evaluate the performance of the proposed adaptive Kalman filter algorithm. The goal was to demonstrate that employing a dynamic fuzzy tuning can improve the 3D pose estimation obtained from the fusion of 3D and 2D sensors as compared to the scenario where a standard EKF is employed.

The three sensors—IMU, camera and LiDAR—are mounted on the robot and, hence, form a rigid body. This means they are at a fixed position and orientation with respect to each other and the transformation between them is constant and predetermined (either by manual measurement or through a calibration process) only once prior to the experiment. Furthermore, the LiDAR setup on the platform is horizontal; in another words, when the platform is not rotated with respect to the reference frame, the LiDAR scans the environment in the xy plane. In this situation, the z coordinate and β and γ angles are the missing degrees of freedom from the LiDAR field-of-view.

4.6.1 Simulated Experiments

Random realistic UAV flight paths were generated in MATLAB. These paths were developed such that the trajectory is a combination of staying more or less on a plane for some times and divergence from being on a plane at others. The trajectory is shown in Fig. 4.3, as well as the true position and orientation of the UAV for the simulated path.

In Chapter 3 of this dissertation [26], we demonstrated that fusing a 2D LiDAR for planar trajectories improves pose estimation. In this chapter, two sensor fusion methods have been applied to the simulated pose data: the proposed AKF and a standard EKF. The accuracy of the proposed fuzzy AKF was compared against the EKF and the estimation errors for x, y and z and also the yaw (α), pitch (β) and roll (γ) angles were evaluated; they are reported in Table 4.2. The results show that the fuzzy AKF considerably reduces the mean error in the position and orientation estimates.

Table 4.2
Mean Errors of EKF and Fuzzy AKF for Simulated UAV Path.

Pose		**Errors**	
		EKF	**AKF**
Position	**X (m)**	0.0450	0.0026
	Y (m)	0.0186	0.0024
	Z (m)	0.0095	0.0025
Orientation	**α (rad)**	0.0029	0.0018
	β (rad)	0.0014	0.0003
	γ (rad)	0.0016	0.0005

4.6.2 Real world experiments

An experiment was conducted using a sensor platform including IMU, camera, and LiDAR to evaluate the proposed algorithm with real world data. The sensor platform developed can be mounted on any ground vehicle or UAV, or can be even used separately if desired.

The IMU used was a VectorNav VN-200s, which has an update frequency of 100 Hz. The camera was a Point Grey Blackfly BFLY-PGE-13E4C-CS, which is a single lens digital CCD. It was calibrated using the standard computer vision methods before the experiment and the whole camera module was able to output 2 frames/sec. For the LiDAR sensor unit, a Hokuyo UTM-30LX-EW was chosen. This LiDAR is a 2D scanner with frequency of 40 Hz. This LiDAR has a measurement range of 30 meters, a field of view of 270°, and an angular resolution of 0.25°. Measurements from all three sensors were time synchronized.

Table 4.3
Mean Errors of EKF and Fuzzy AKF for Real UAV Path.

Pose		**Errors**	
		EKF	**AKF**
Position	**X (m)**	0.0537	0.0024
	Y (m)	0.0353	0.0014
	Z (m)	0.0850	0.0025
Orientation	$\boldsymbol{\alpha}$ **(rad)**	0.0290	0.0030
	$\boldsymbol{\beta}$ **(rad)**	0.0040	0.0007
	$\boldsymbol{\gamma}$ **(rad)**	0.0160	0.0030

The experiment was run in a motion capture room equipped with 12 Vicon cameras, which collectively track the trajectory. The pose obtained from the motion capture system was used as ground truth data. The hexacopter was too large for flying in the motion capture room; therefore, we had to dismount the platform and proceed without using the UAV, simulating UAV movement by carrying the sensor pod by hand.

The true trajectory, as measured by the Vicon system, is shown in Fig. 4.4, as well as the true position and orientation. To evaluate the results, a similar procedure to the simulated experiment was used. The mean error results of the two algorithms, fuzzy AKF and standard EKF, are summarized in Table 4.3. From these results, it can easily be concluded that the fuzzy AKF noticeably improved the pose estimation.

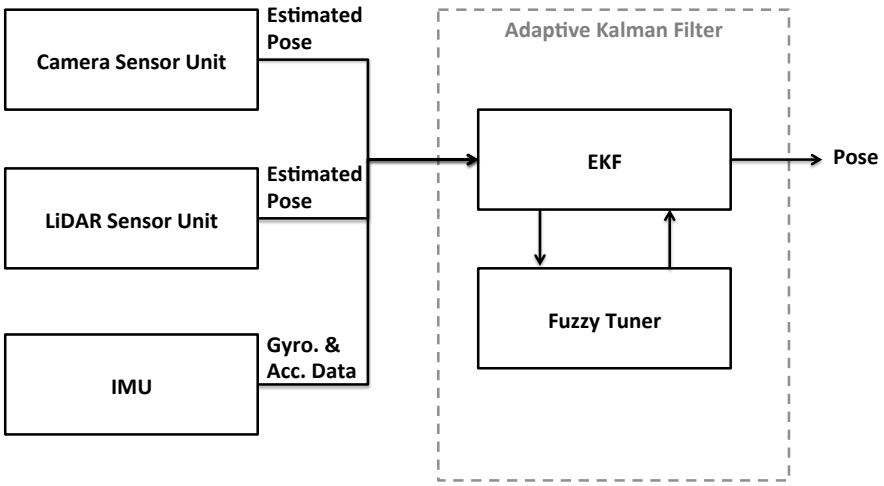
4.7 Summary and Future Work

3D pose estimation using a combination of multimodal sensors is a topic that has been investigated for several years now. However, in past work either 3D sensors were used or, if 2D sensors were used, there was an assumption on prior knowledge of the environment. In this chapter we proposed a method that uses a 2D sensor (LiDAR) along with other 3D sensors (IMU and vision camera) to improve accuracy of 3D pose estimation in an unknown, GPS-denied environment. It was shown that 2D sensors can be used for 3D pose estimation improvement as long as there is a way to monitor the quality of their estimated pose. Since the quality of estimation of a 2D sensor can degrade by changes in the trajectory or by quick movements, the algorithm needs to be able to detect this degradation and then reduce the contribution of the inaccurate sensor measurements in the final fused output. The proposed method to accomplish the task was a fuzzy logic system to supervise the sensor estimation quality. A fuzzy adaptive Kalman filter (AKF) was introduced which is a modified version of the EKF. The fuzzy AKF employs fuzzy logic to detect the reliability of the 2D LiDAR sensor module, based on the robot movement and covariance matching technique, and then to adjust the contribution of the LiDAR to the final estimated pose.

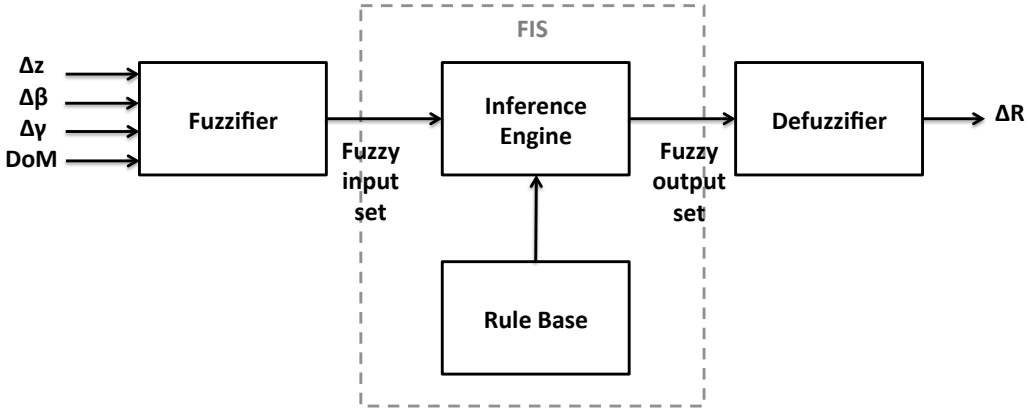
A set of simulated and real world data were used to evaluate the proposed algorithm.

The fuzzy AKF performance was compared to the standard EKF algorithm, which is widely employed for sensor fusion and pose estimation, and it was shown that the fuzzy AKF accomplishes considerably better pose estimation accuracy.

In this work, only the measurement noise covariance R was adjusted and the process noise covariance Q was assumed to be constant. In reality, this covariance is not constant in most applications. Therefore, in the future, a similar approach could be considered to to adjust the noise covariance Q.

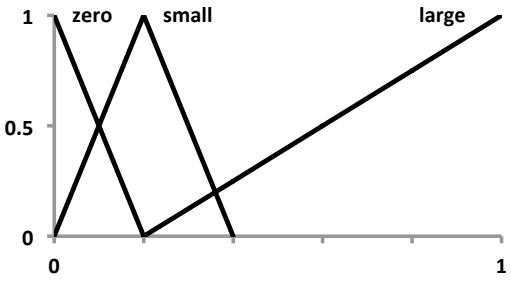


(a) AKF block diagram

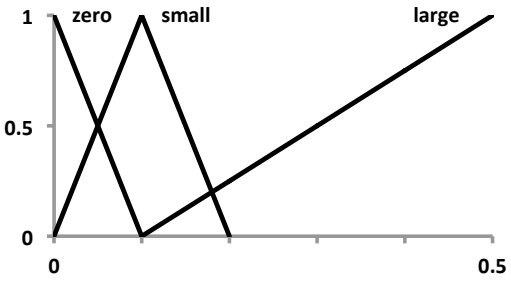


(b) FLS block diagram

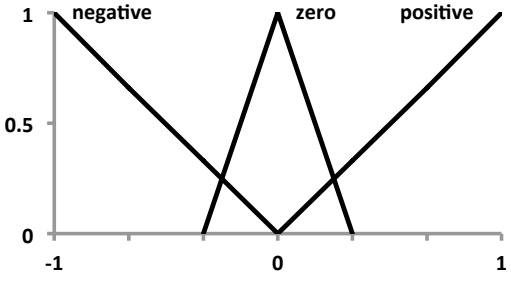
Figure 4.1: Adaptive Kalman filter for pose estimation



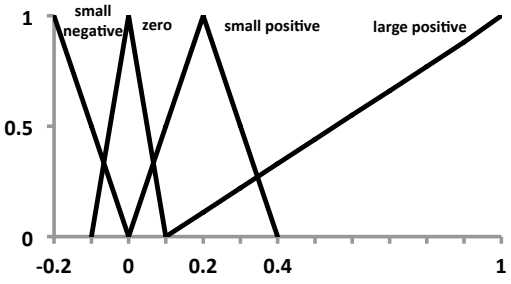
(a) Input 1: Δz (m)



(b) Input 2&3: $\Delta\beta/\Delta\gamma$ (rad)

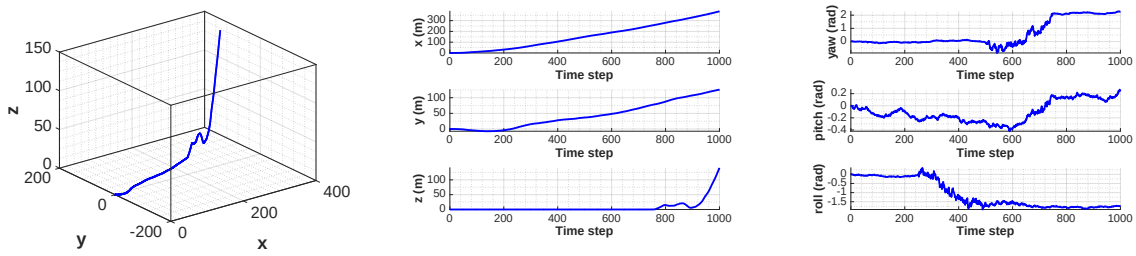


(c) Input 4: DOM



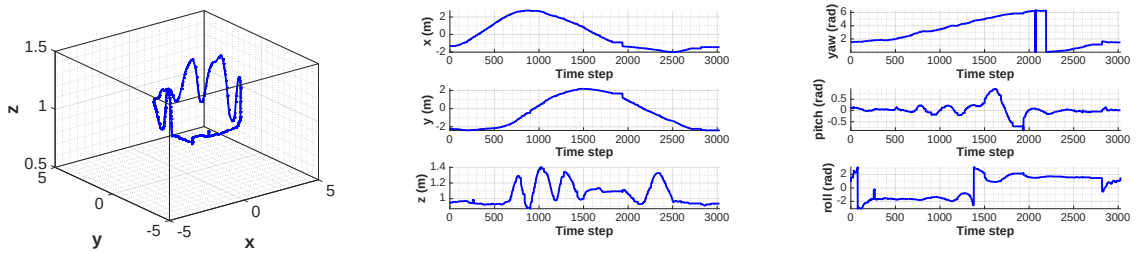
(d) Output: ΔR

Figure 4.2: Input and output membership functions



(a) UAV simulated path (b) True position (c) True orientation

Figure 4.3: Ground trajectory and state values for simulated experiment.



(a) Platform real path (b) True position (c) True orientation

Figure 4.4: Ground truth trajectory and state values for real-world experiment.

Chapter 5

Conclusions and Future Work

Sensors are widely used in all the areas of science and technology. In numerous situation and applications, one single sensor can not provide the desired output in terms of completion, precision, robustness or other criteria. This has encouraged researchers to combine the data from different sensory sources to obtain better results. Hence, sensor fusion has become the primary solution to overcome sensors deficiencies.

This dissertation provided novel solutions to two applications of the sensor fusion: an image level 2 dimensional sensor fusion for train car wheel and bearing defect detection and a three dimensional application for UAV pose estimation. The fusion techniques introduced in the first application can also potentially be employed in the UAV pose estimation by adding an extra sensor to the platform.

Chapter 2 of this dissertation was dedicated to the development of an algorithm for train car wheel and bearing defect detection. In the rail industry, two major components of rolling stock that are always of great interest when it comes to maintenance and safety related issues are car wheels and bearings. The contribution of this dissertation was proposing and implementing an automatic method to detect sliding wheel, a serious train car wheel defect, by fusing thermal and vision wayside cameras. This type of defect is not detectable at an early stage in the images taken by a vision camera, however these defects generate a distinctive heat pattern on the wheel or bearing that is clearly visible in the thermal imagery. On the other hand, thermal imagery does not contain any texture or color information which makes it difficult to detect and extract the parts of interest in the image. Therefore, the wheel and bearing segmentation from the thermal imagery is erroneous while this task can be done with a much better accuracy employing vision cameras. This makes thermal and visible spectrum imagery complementary. In this dissertation, wavelet decomposition was employed to extract the features of the thermal and vision imagery. Then the two images were merged based on their decomposition and a fused image was composed. The resulting fused image contains more information than each individual image and can be used as an input for image-based wheel and bearing defect detection algorithms. To verify the proposed method and to show an example of this application, it was demonstrated on a real data set from a Union Pacific rail line to identify sliding wheels. Currently in the railroad industry the inspection is mostly

done manually. Using the proposed method in this dissertation, the defect can be detected automatically at an early stage to prevent any further hazard.

Then we moved to a more complicated scenario which was estimating the pose of a moving platform such as a UAV in a 3D environment which is of significant importance in many areas of robotics research and in a very similar way in industries such as car industry. In order to perform this task, multi-sensor fusion has been used to improve the accuracy of the estimation and to compensate for individual sensor deficiencies. Unlike the previous works in this area that use sensors with the ability of 3D localization to estimate the full pose of a UAV, in this work we employed the data from a 2D light detection and ranging (LiDAR) sensor, which can only estimate the pose in a 2D plane. To the best of our knowledge 2D LiDARs have not been used for 3D pose estimation in an unknown environment. We fused the estimation obtained from the LiDAR with the data from visible spectrum camera and inertial sensors showing that, despite the incomplete estimation from the 2D LiDAR, the overall estimated 3D pose can be improved. This work can be used for SLAM and as the UAV moves, it can create the 3D map of environment and acquire perception of the surrounding environment. By studying the trajectory of the robot, we noticed that there are time intervals that the robots movement between the two LiDAR readings can be assumed planar. Hence, we proceeded one step further and implemented a fuzzy inference system that estimates if the UAV is almost moving on a plane or its path involves quick roll, pitch and height changes which will result in diverging

from the planar motion. Based on this idea, we adjusted sensors weights in the fusion algorithm and showed that using the introduced adaptive fuzzy sensor fusion, we were able to reduce the pose estimation error and obtain better accuracy.

In this dissertation the focus was on localization. Having a good pose estimation algorithm will help creating better and more accurate 3D map of the world. Therefore, as future research, this work can be extended to SLAM. In addition, we introduced a method for thermal and visibly spectrum fusion which in a similar way can be used for the UAV application. Visual spectrum cameras do not work very well at poor lighting conditions, at night or in some weathers such as dusty or foggy conditions. Adding a thermal or night vision camera in future can increase the capability of pose estimation in different conditions and the algorithm proposed for thermal and visual spectrum imagery can be used to create more informative images. As a result the pose estimation method potentially will become more robust and reliable.

References

[1] Walid Abdel-Hamid, Aboelmagd Noureldin, and Naser El-Sheimy. Adaptive fuzzy prediction of low-cost inertial-based positioning errors. *IEEE Transactions on Fuzzy Systems*, 15(3):519–529, 2007.

[2] Mohammad Amin Ahmad Akhoundi and Ehsan Valavi. Multi-sensor fuzzy data fusion using sensors with different characteristics. *arXiv preprint arXiv:1010.6096*, 2010.

[3] Georgios Alexandrou, Georges Kouroussis, and Olivier Verlinden. A comprehensive prediction model for vehicle/track/soil dynamic response due to wheel flats. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409715576015, 2015.

[4] V Aslantas, E Bendes, AN Toprak, and R Kurban. A comparison of image fusion methods on visible, thermal and multi-focus images for surveillance applications. 2011.

[5] Matthias Asplund, Per Gustafsson, Thomas Nordmark, Matti Rantatalo, Mikael Palo, Stephen Mayowa Famurewa, and Karina Wandt. Reliability and measurement accuracy of a condition monitoring system in an extreme climate: A case study of automatic laser scanning of wheel profiles. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409714528485, 2014.

[6] Matthias Asplund, Mikael Palo, Stephen Famurewa, and Matti Rantatalo. A study of railway wheel profile parameters used as indicators of an increased risk of wheel defects. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 230(2):323–334, 2016.

[7] Derek Barke and WK Chiu. Structural health monitoring in the railway industry: a review. *Structural Health Monitoring*, 4(1):81–93, 2005.

[8] Derek Barke and WK Chiu. Structural health monitoring in the railway industry: a review. *Structural Health Monitoring*, 4(1):81–93, 2005.

[9] CA Basca, M Talos, and R Brad. Randomized hough transform for ellipse detection with result clustering. In *EUROCON 2005-The International Conference on" Computer as a Tool"*, volume 2, pages 1397–1400. IEEE, 2005.

[10] Theodore L Bergman, Frank P Incropera, David P DeWitt, and Adrienne S Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[11] David M Bevly. Global positioning system (gps): A low-cost velocity sensor for correcting inertial sensor errors on ground vehicles. *Journal of dynamic systems, measurement, and control*, 126(2):255–264, 2004.

[12] Sebastian Blank, Tobias Föhst, and Karsten Berns. A fuzzy approach to low level sensor fusion with limited system knowledge. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–7. IEEE, 2010.

[13] A Bracciali and G Cascini. Detection of corrugation and wheelflats of railway wheels using energy and cepstrum analysis of rail acceleration. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 211(2):109–116, 1997.

[14] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[15] Francois Caron, Emmanuel Duflos, Denis Pomorski, and Philippe Vanheeghe. Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects. *Information fusion*, 7(2):221–230, 2006.

[16] Luis Rodolfo García Carrillo, Alejandro Enrique Dzul López, Rogelio Lozano, and Claude Pégard. Combining stereo vision and inertial navigation system for a quad-rotor uav. *Journal of Intelligent & Robotic Systems*, 65(1-4):373–387, 2012.

[17] Andrew Chambers, Sebastian Scherer, Luke Yoder, Sezal Jain, Stephen Nuske, and Sanjiv Singh. Robust multi-sensor fusion for micro aerial vehicle navigation in gps-degraded/denied environments. In *2014 American Control Conference*, pages 1892–1899. IEEE, 2014.

[18] Jacky CK Chow, Derek D Lichti, Jeroen D Hol, Giovanni Bellusci, and Henk Luinge. Imu and multiple rgb-d camera fusion for assisting indoor stop-and-go 3d terrestrial laser scanning. *Robotics*, 3(3):247–280, 2014.

[19] Javier Civera, Oscar G Grasa, Andrew J Davison, and JMM Montiel. 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, 27(5):609–631, 2010.

[20] James E Cline, James R Bilodeau, and Richard L Smith. Acoustic wayside identification of freight car roller bearing defects. In *Railroad Conference, 1998. Proceedings of the 1998 ASME/IEEE Joint*, pages 79–83. IEEE, 1998.

[21] Kevin D Cole, Constantine M Tarawneh, Arturo A Fuentes, BM Wilson, and L Navarro. Thermal models of railroad wheels and bearings. *International Journal of Heat and Mass Transfer*, 53(9):1636–1645, 2010.

[22] Hanieh Deilamsalehy and Timothy C Havens. Sensor fused three-dimensional localization using imu, camera and lidar. In *2016 IEEE Sensors*. IEEE, 2016.

[23] Hanieh Deilamsalehy, Timothy C Havens, and Pasi Lautala. Automatic method for detecting and categorizing train car wheel and bearing defects. In *2015*

joint rail conference, pages V001T02A007–V001T02A007. American Society of Mechanical Engineers, 2015.

[24] Hanieh Deilamsalehy, Timothy C Havens, and Pasi Lautala. Detection of sliding wheels and hot bearings using wayside thermal cameras. In *2016 Joint Rail Conference*, pages V001T02A002–V001T02A002. American Society of Mechanical Engineers, 2016.

[25] Hanieh Deilamsalehy, Timothy C Havens, Pasi Lautala, Ezequiel Medici, and James Davis. An automatic method for detecting sliding railway wheels and hot bearings using thermal imagery. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409716638703, 2016.

[26] Hanieh Deilamsalehy, Timothy C Havens, and Joshua Manela. Heterogeneous multi-sensor fusion for mobile platform 3d pose estimation. *Journal of Dynamic Systems Measurement and Control*, 2016.

[27] Long Di, Tobias Fromm, and YangQuan Chen. A data fusion system for attitude estimation of low-cost miniature uavs. *Journal of Intelligent & Robotic Systems*, 65(1-4):621–635, 2012.

[28] EE ElMadbouly, AE Abdalla, and Gh M ElBanby. Sensor fusion and sensor fault detection with fuzzy clustering. In *Computer Engineering and Systems (ICCES), 2010 International Conference on*, pages 265–268. IEEE, 2010.

[29] P Jorge Escamilla-Ambrosio and Nail Mort. Multi-sensor data fusion architecture based on adaptive kalman filters and fuzzy logic performance assessment. In *Information Fusion, 2002. Proceedings of the Fifth International Conference on*, volume 2, pages 1542–1549. IEEE, 2002.

[30] T Fiorenzani, C Manes, G Oriolo, and P Peliti. Comparative study of unscented kalman filter and extended kalman filter for position/attitude estimation in unmanned aerial vehicles. *Inst. for Systems Analysis and Computer Science (IASI-CNR), Rome, Italy, Rept*, pages 08–08, 2008.

[31] Kai Goebel and Alice M Agogino. Sensor validation and fusion for automated vehicle control using fuzzy techniques. *measurement*, 2:v1, 2001.

[32] Ramazan Havangi, Mohammad Ali Nekoui, and Mohammad Teshnehlab. Adaptive neuro-fuzzy extended kaiman filtering for robot localization. In *Power Electronics and Motion Control Conference (EPE/PEMC), 2010 14th International*, pages T5–130. IEEE, 2010.

[33] Joel A Hesch, Dimitrios G Kottas, Sean L Bowman, and Stergios I Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, page 0278364913509675, 2013.

[34] Joel A Hesch, Faraz M Mirzaei, Gian Luca Mariottini, and Stergios I Roumeliotis.

A 3d pose estimator for the visually impaired. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2716–2723. IEEE, 2009.

[35] Joel A Hesch, Faraz M Mirzaei, Gian Luca Mariottini, and Stergios I Roumeliotis. A 3d pose estimator for the visually impaired. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2716–2723. IEEE, 2009.

[36] Reza Hoseinnezhad and Alireza Bab-Hadiashar. Fusion of redundant information in brake-by-wire systems using a fuzzy voter. *J. Adv. Inf. Fusion*, 1(1):52–62, 2006.

[37] Longwen Huang, Zhengmei Li, Liuxiang Li, and Qi An. Methods to calculate accurate wheel/rail contact positions and static contact stress levels. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409714530912, 2014.

[38] Farrokh Janabi-Sharifi and Mohammed Marey. A kalman-filter-based method for pose estimation in visual servoing. *IEEE Transactions on Robotics*, 26(5):939–947, 2010.

[39] J Jergeus, C Odenmarck, R Lunden, P Sotkovszki, B Karlsson, and P Gullers. Full-scale railway wheel flat experiments. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 213(1):1–13, 1999.

[40] Pyeong-Gook Jung, Sehoon Oh, Gukchan Lim, and Kyoungchul Kong. A mobile motion capture system based on inertial sensors and smart shoes. *Journal of Dynamic Systems, Measurement, and Control*, 136(1):011002, 2014.

[41] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[42] Jonathan Kelly and Gaurav S Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1):56–79, 2011.

[43] Ferenc Kolonits. Analysis of the temperature of the rail/wheel contact surface using a half-space model and a moving heat source. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409714548101, 2014.

[44] Ujwal Koneru, Sangram Redkar, and Anshuman Razdan. Fuzzy logic based sensor fusion for accurate tracking. In *International Symposium on Visual Computing*, pages 209–218. Springer, 2011.

[45] Nikolaos Kyriakoulis and Antonios Gasteratos. On visuo-inertial fusion for robot pose estimation using hierarchical fuzzy systems. *International Journal of Optomechatronics*, 6(1):17–36, 2012.

[46] Joseph J Laviola. A comparison of unscented and extended kalman filtering for

estimating quaternion motion. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2435–2440. IEEE, 2003.

[47] Caiqin Li, Zhenzhong Wei, and Jing Xing. Online inspection system for the automatic detection of bolt defects on a freight train. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 230(4):1213–1226, 2016.

[48] Gabriele Ligorio and Angelo Maria Sabatini. Extended kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: comparative analysis and performance evaluation. *Sensors*, 13(2):1919–1941, 2013.

[49] Xiang Liu, M Saat, and Christopher Barkan. Analysis of causes of major train derailment and their effect on accident rates. *Transportation Research Record: Journal of the Transportation Research Board*, (2289):154–163, 2012.

[50] D Loebis, R Sutton, J Chudley, and W Naeem. Adaptive tuning of a kalman filter via fuzzy logic for an intelligent auv navigation system. *Control engineering practice*, 12(12):1531–1539, 2004.

[51] Todd Lupton and Salah Sukkarieh. Efficient integration of inertial observations into visual slam without initialization. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1547–1552. IEEE, 2009.

[52] Adam Mańka and Marek Sitarz. Effects of a thermal load on the wheel/brake-block subsystem: The thermal conicity of railway wheels. *Proceedings of the*

Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 230(1):193–205, 2016.

[53] Robert S McGonigal. Defect detectors: Looking for trouble in all the right places, 2006.

[54] Ezequiel Medici, Andres Cecchini, David Serrano, and Frederick Just. Damage detection in composite beam using the temperature distribution. In *ASME 2006 International Mechanical Engineering Congress and Exposition*, pages 611–615. American Society of Mechanical Engineers, 2006.

[55] Faraz M Mirzaei and Stergios I Roumeliotis. A kalman filter-based algorithm for imu-camera calibration: Observability analysis and performance evaluation. *IEEE transactions on robotics*, 24(5):1143–1156, 2008.

[56] AH Mohamed and KP Schwarz. Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4):193–203, 1999.

[57] Abdelkrim Nemra and Nabil Aouf. Robust airborne 3d visual simultaneous localization and mapping with observability and consistency analysis. *Journal of Intelligent and Robotic Systems*, 55(4-5):345–376, 2009.

[58] Anthea NGAI. *Wild Impact Load Detector*. The Institution of Railway Signal Engineers (Hong Kong Section), 2009.

[59] Gabriel Nützi, Stephan Weiss, Davide Scaramuzza, and Roland Siegwart. Fusion of imu and vision for absolute scale estimation in monocular slam. *Journal of intelligent & robotic systems*, 61(1):287–299, 2011.

[60] Edwin B Olson. Real-time correlative scan matching. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 4387–4393. IEEE, 2009.

[61] Mayorkinos Papaelias, Arash Amini, Zheng Huang, Patrick Vallely, Daniel Cardoso Dias, and Spyridon Kerkyras. Online condition monitoring of rolling stock wheels and axle bearings. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409714559758, 2014.

[62] Simon Placht, Peter Fürsattel, Etienne Assoumou Mengue, Hannes Hofmann, Christian Schaller, Michael Balda, and Elli Angelopoulou. Rochade: Robust checkerboard advanced detection for camera calibration. In *European Conference on Computer Vision*, pages 766–779. Springer, 2014.

[63] Kevin Pratt and Jeffrey Kramer. Sensor fusion for robot navigation using a fuzzy-ekf with weighted covariance. In *International Conference for Advanced Robotics*, volume 2007, 2007.

[64] Junuthula Narasimha Reddy and David K Gartling. *The finite element method in heat transfer and fluid dynamics*. CRC press, 2010.

[65] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[66] Anand Sanchez-Orta, Vicente Parra-Vega, Carlos Izaguirre-Espinosa, and Octavio Garcia. Position–yaw tracking of quadrotors. *Journal of Dynamic Systems, Measurement, and Control*, 137(6):061011, 2015.

[67] Konstantin Schauwecker and Andreas Zell. On-board dual-stereo-vision for the navigation of an autonomous mav. *Journal of Intelligent & Robotic Systems*, 74(1-2):1–16, 2014.

[68] Szabolcs Sergyán. Useful and effective feature descriptors in content-based image retrieval of thermal images. In *2012 4th IEEE International Symposium on Logistics and Industrial Informatics*, pages 55–58. IEEE, 2012.

[69] Ying Shen, Zhongxiang Zhu, and Enrong Mao. Double-fuzzy kalman filter based on gps/imu/mv sensor fusion for tractor autonomous guidance. In *2007 IEEE International Conference on Automation and Logistics*, pages 61–65. IEEE, 2007.

[70] Brant Stratman, Yongming Liu, and Sankaran Mahadevan. Structural health monitoring of railroad wheels using wheel impact load detectors. *Journal of failure analysis and prevention*, 7(3):218–225, 2007.

[71] V Subramanian, TF Burks, and WE Dixon. Sensor fusion using fuzzy logic

enhanced kalman filter for autonomous vehicle guidance in citrus groves. *Transactions of the ASABE*, 52(5):1411–1422, 2009.

[72] Christopher Torrence and Gilbert P Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.

[73] A UmaMageswari, J Joseph Ignatious, and R Vinodha. A comparitive study of kalman filter, extended kalman filter and unscented kalman filter for harmonic analysis of the non-stationary signals. *International Journal of Scientific & Engineering Research*, 3(7):1–9, 2012.

[74] Rajib Ul Alam Uzzal, Waiz Ahmed, and Rama B Bhat. A three-dimensional modeling study of wheel/rail impacts created by multiple wheel flats, and the development of a smart wheelset. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, page 0954409714545558, 2014.

[75] S Wasielewski and O Strauss. Calibration of a multi-sensor system laser rangefinder/camera. In *Intelligent Vehicles' 95 Symposium., Proceedings of the*, pages 472–477. IEEE, 1995.

[76] Stephan Weiss and Roland Siegwart. Real-time metric state estimation for modular vision-inertial systems. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4531–4537. IEEE, 2011.

[77] Yonghong Xie and Qiang Ji. A new efficient ellipse detection method. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 957–960. IEEE, 2002.

[78] T Xu, J Chudley, and R Sutton. A fuzzy logic based multi-sensor navigation system for an unmanned surface vehicle. In *Proceedings of the UKACC International Control Conference, Glasgow, UK*, 2006.

[79] Ronald R Yager. Set measure directed multi-source information fusion. *IEEE Transactions on Fuzzy Systems*, 19(6):1031–1039, 2011.

[80] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.

[81] P Zeeuw. Wavelet and image fusion. cwi, amsterdam, http, 1998.

[82] Sheng Zhao and Jay A Farrell. 2d lidar aided ins for vehicle positioning in urban environments. In *Control Applications (CCA), 2013 IEEE International Conference on*, pages 376–381. IEEE, 2013.

[83] Thi Thi Zin, Hideya Takahashi, Hiromitsu Hama, and Takashi Toriu. *Fusion of infrared and visible images for robust person detection*. INTECH Open Access Publisher, 2011.

Appendix A

Letters of Permission

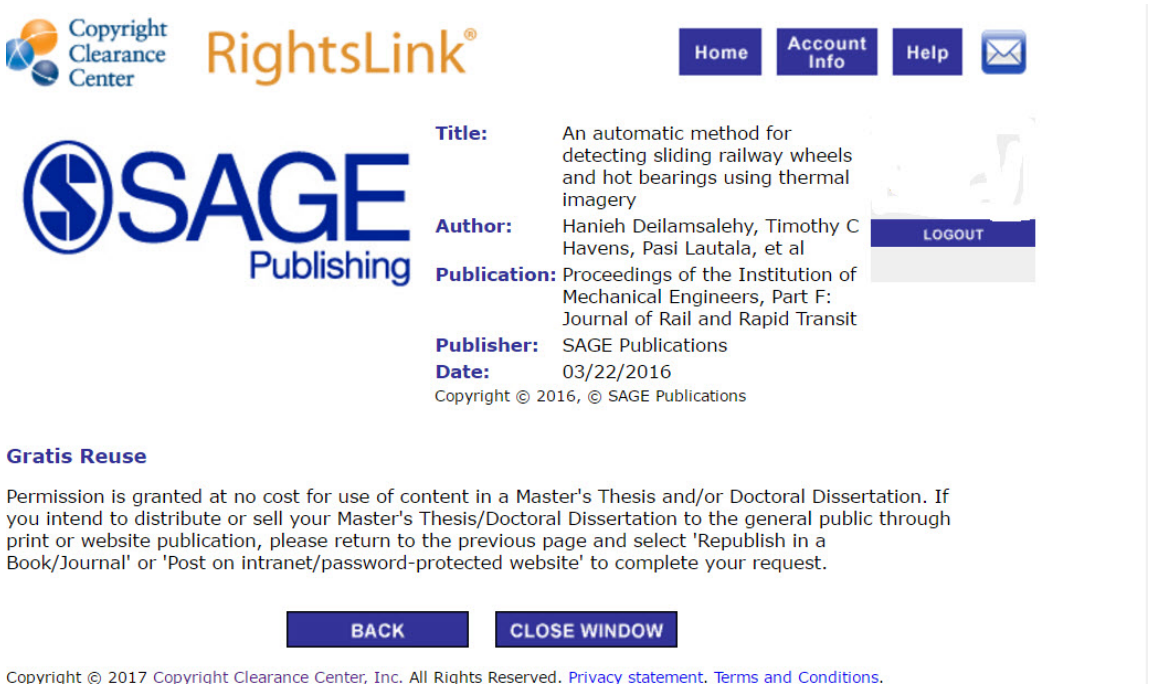


Figure A.1: SAGE permission letter for the material used in Chapter 2

Michigan Tech

Hanieh Deilamsalehy <hdeilams@mtu.edu>

Permission to use my papers in my dissertation

Beth Darchi <DarchiB@asme.org> Tue, May 23, 2017 at 3:18 PM
To: Hanieh Deilamsalehy <hdeilams@mtu.edu>

Dear Prof. Deilamsalehy,
This permission letter has been revised to reflect all requests. It is our pleasure to grant you permission to use **all or any part of** the following ASME papers:

- Automatic Method for Detecting and Categorizing Train Car Wheel and Bearing Defects, by Hanieh Deilamsalehy; Timothy C. Havens; Pasi Lautala, Paper Number JRC2015-5741
- Detection of Sliding Wheels and Hot Bearings Using Wayside Thermal Cameras, by Hanieh Deilamsalehy; Timothy C. Havens; Pasi Lautala, Paper Number JRC2016-5711
- Sensor Fusion of Wayside Visible and Thermal Imagery for Rail Car Wheel and Bearing Damage Detection, by Hanieh Deilamsalehy, Timothy C. Havens and Pasi Lautala, Paper Number JRC2017-2284

cited in your letter for inclusion in a Ph.D. dissertation entitled HETEROGENEOUS MULTI-SENSOR FUSION FOR 2D AND 3D POSE ESTIMATION to be published by Michigan Technological University.

Permission is granted for the specific use as stated herein and does not permit further use of the materials without proper authorization. Proper attribution must be made to the author(s) of the materials. **Please note:** if any or all of the figures and/or Tables are of another source, permission should be granted from that outside source or include the reference of the original source. ASME does not grant permission for outside source material that may be referenced in the ASME works.

As is customary, we request that you ensure full acknowledgment of this material, the author(s), source and ASME as original publisher. Acknowledgment must be retained on all pages where figure is printed and distributed.

Many thanks for your interest in ASME publications.

Sincerely,

Beth Darchi
Publishing Administrator
ASME
2 Park Avenue
New York, NY 10016-5990
Tel 1.212.591.7700
darchib@asme.org

From: Hanieh Deilamsalehy [mailto:hdeilams@mtu.edu]
Sent: Tuesday, May 23, 2017 3:15 PM

[Quoted text hidden]

[Quoted text hidden]

Figure A.2: ASME permission letter for the material used in Chapter 2

Title: Sensor fused three-dimensional localization using IMU, camera and LiDAR

Conference Proceedings: SENSORS, 2016 IEEE

Author: Hanieh Deilamsalehy

Publisher: IEEE

Date: Oct. 2016

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author©s approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Figure A.3: IEEE permission letter for the material used in Chapter 3