Infared Light-Based Data Association and Pose Estimation for Aircraft Landing in Urban Environments

David Akagi
Brigham Young University
Infrared Light-Based Data Association and Pose Estimation
for Aircraft Landing in Urban Environments

David Akagi

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Tim McLain, Chair
Joshua Mangelson
Andrew Ning

Department of Mechanical Engineering
Brigham Young University

Copyright © 2024 David Akagi
All Rights Reserved
Infrared Light-Based Data Association and Pose Estimation for Aircraft Landing in Urban Environments

David Akagi
Department of Mechanical Engineering
Master of Science
Abstract

In this thesis we explore an infrared light-based approach to the problem of pose estimation during aircraft landing in urban environments where GPS is unreliable or unavailable. We introduce a novel fiducial constellation composed of sparse infrared lights that incorporates projective invariant properties in its design to allow for robust recognition and association from arbitrary camera perspectives. We propose a pose estimation pipeline capable of producing high accuracy pose measurements at real-time rates from monocular infrared camera views of the fiducial constellation, and present as part of that pipeline a data association method that is able to robustly identify and associate individual constellation points in the presence of clutter and occlusions. We demonstrate the accuracy and efficiency of this pose estimation approach on real-world data obtained from multiple flight tests, and show that we can obtain decimeter level accuracy from distances of over 100 m from the constellation. To achieve greater robustness to the potentially large number of outlier infrared detections that can arise in urban environments, we also explore learning-based approaches to the outlier rejection and data association problems. By formulating the problem of camera image data association as a 2D point cloud analysis, we can apply deep learning methods designed for 3D point cloud segmentation to achieve robust, high-accuracy associations at constant real-time speeds on infrared images with high outlier-to-inlier ratios. We again demonstrate the efficiency of our learning-based approach on both synthetic and real-world data, and compare the results and limitations of this method to our first-principles-based data association approach.

Keywords: UAV landing, GPS-denied, pose estimation, infrared, data association, PointNet
Acknowledgments

This thesis would not have been possible without the unwavering support of a number of incredible individuals. I am especially grateful to my committee chair Dr. Tim McLain, who introduced me to the MAGICC Lab at BYU and guided me through three years of research with continual confidence and optimism in my efforts. I am thankful as well to my other committee members Dr. Joshua Mangelson and Dr. Andrew Ning, who contributed significantly to my interest and academic growth in the area of unmanned aircraft through numerous research and class discussions. The opportunity to pursue a graduate degree would never have been possible without the lifetime of support and guidance received from my parents, who always encouraged my academic pursuits; my brother John, who has both advised and inspired many of my academic decisions; and my sister Christine, who helped propel me through the engineering program at BYU and encouraged me to adopt healthier study and life habits. I am forever grateful to my wife Melanie, who has been immeasurably patient, understanding, and supportive through all the long nights of homework and research, and my in-laws, who have selflessly provided for our needs throughout this time. I am extremely indebted to the brilliant and hardworking students in the MAGICC Lab, particularly Aaron Brown, Ian Reid, Brandon Sutherland, and Jacob Moore, without whom quite literally none of this research would have been possible. The collaborative efforts of members of Archer Aviation including Anthony Hunter, Sid Kumar, Gary Long, Hannah Yang, and Pablo Hopman were instrumental to the research developed in this thesis, with this collaboration made possible by generous funding through the U.S. Air Force Agility Prime program.
# Table of Contents

List of Figures vii

List of Tables ix

1 Introduction 1
  1.1 Related Works 2
  1.2 Contributions 4
  1.3 Thesis Overview 5

2 Technical Approach and Background 6
  2.1 Pose Estimation Pipeline Overview 6
  2.2 IR Light Detection 7
  2.3 Pose Estimation Problem 9
  2.4 Projective Invariance 12
  2.5 Deep Learning on Point Clouds 15
  2.6 Summary 18

3 Robust IR-Based Pose Estimation for Precision VTOL Aircraft Landing in Urban Environments 19
  3.1 Abstract 19
  3.2 Introduction 19
  3.3 Related Work 21
  3.4 Proposed Landing Approach 22
  3.5 Constellation Design 23
  3.6 Data Association Algorithm 25
  3.7 Results 31
  3.8 Conclusion 36
  3.9 Acknowledgments 37

4 Deep Learning-Based Data Association in Camera Images for GPS-Denied Aircraft Landing 38
  4.1 Abstract 38
  4.2 Introduction 38
  4.3 Related Work 40
  4.4 GPS-Denied UAV Landing Overview 42
  4.5 Models and Datasets 44
  4.6 Normalization and Filtering Techniques 46
  4.7 Data Association Approaches and Results 52
List of Figures

1.1 GPS limitations in urban environments 2
2.1 Pose estimation pipeline 6
2.2 Light detection method 8
2.3 Composite IR image 9
2.4 Camera pose in inertial frame 10
2.5 Projective invariant properties 13
2.6 Ambiguous cross ratio example 14
2.7 2D point cloud formulation 15
2.8 PointNet tasks 16
2.9 PointNet++ model architecture 17
2.10 PointNet++ multiscale grouping 18
3.1 IR constellation lights 20
3.2 Pose estimation flowchart 22
3.3 Min-max difference image 23
3.4 Fiducial constellation design 24
3.5 Cross ratio diagrams 24
3.6 Polar coordinate line representation 27
3.7 Cross ratio voting scheme 30
3.8 To-scale constellation layout 32
3.9 Castle trajectory pose estimates 33
3.10 Castle flight-test errors 34
3.11 Glideslope trajectory flight test 35
3.12 Glideslope flight-test errors 36
4.1 Point IR-based pose estimation approach 43
4.2 Point IR constellation example 43
4.3 PointNet++ model architecture 45
4.4 2D point cloud normalization strategies 47
4.5 ROI normalization with outliers 48
4.6 Example PointNet++ output 49
4.7 PointNet++ output filtering 50
4.8 OutlierNet Computation Times 54
4.9 PointNet++ pose estimates for castle trajectory 56
4.10 PointNet++ pose errors for castle trajectory 57
4.11 LineNet pose estimates for castle trajectory 58
4.12 LineNet pose errors for castle trajectory  59
List of Tables

3.1 Pose estimation pipeline computation time 33
4.1 PointNet datasets 46
4.2 Data association accuracy of PointNet++ using different normalization strategies 52
4.3 OutlierNet classification accuracy 53
4.4 OutlierNet/PointNet++ accuracy on cluttered data 55
4.5 PointNet++ computation times 56
4.6 PointNet++ pose accuracy on flight data 59
1 Introduction

Uses for unmanned aerial vehicles (UAVs) have increased in recent years, with unmanned aircraft becoming a popular and safe alternative for performing jobs that are mundane or dangerous for humans to carry out. A few examples include infrastructure inspection, agricultural monitoring, and city mapping. However, while aerial automation of tasks can be an attractive approach to many problems, some important technical shortcomings exist that must be addressed before UAVs can be safely and effectively deployed in urban settings.

One such issue is a reliance on a global positioning system (GPS) that utilizes numerous satellites in orbit around the earth to send spatial information to a receiver on or near the ground. While effective and suitably accurate for outdoor UAV flight in open, rural environments, GPS information can suffer in consistency and quality in urban environments where signals can be obstructed or reflected off of surrounding buildings before arriving at a receiver as shown in Figure 1.1. Additionally, GPS signals can be subject to intentional attacks such as jamming, where the satellite signal is actively prevented from reaching the receiver, or spoofing, where an antagonist broadcasts a faked GPS signal with incorrect position information. These vulnerabilities of GPS make it an unreliable method for localizing UAVs in urban environments, especially when the UAV is operating autonomously and may be making unsupervised decisions based on its understanding of its place in the environment. An especially critical phase of autonomous flight that depends heavily on a UAV’s estimation of its position in the environment is the landing phase, when slight errors can cause catastrophic failure of the aircraft.

To address this shortcoming, we present in this thesis a vision-based method for obtaining high accuracy pose information during UAV landing based on camera views of a novel sparse infrared (IR) light fiducial. The design of this IR fiducial allows it to be robustly identified in the presence of outliers when viewed from arbitrary camera perspectives, making it suitable for vertical take-off and landing (VTOL) aircraft, which are not constrained to land from an assumed approach direction. The choice to work in the IR spectrum rather than the more commonly used visible light spectrum is motivated primarily by the reduced intensity of ambient IR light in outdoor environments compared to visible light, which reduces the number of potential outlier detections arising from IR sources or reflections in the environment. We can also easily filter the IR camera sensor to detect a
narrow band to isolate our target lights from the surrounding background.

The pose estimation approach proposed in this thesis relies on correctly associating IR light sources placed at known locations in the 3D environment with their 2D pixel locations in camera images obtained onboard the UAV. This data association problem is a critical and challenging part of the pose recovery problem that must be efficiently and robustly solved for the system to operate reliably in realistic urban environments. In this thesis we propose both geometry and learning-based approaches to the data association problem that can be applied to the class of fiducial constellation designs presented in this work.

1.1 Related Works

The research topics and approaches explored and developed in this thesis build on work done previously by other authors, and draw comparisons to other methods proposed for solving related problems. In this section we review recent literature relating to the topics of this thesis, which contain both key similarities and differences in application and approach.

1.1.1 GPS Denied Localization

The problem of unmanned navigation in environments where GPS is unreliable or unavailable has been the focus of various prior research efforts. Approaches include methods for estimating a vehicle’s global position in an environment as well as its local position relative to some prior measurement or landmark, and make use of various sensors and differing levels of knowledge or control of the environment. For instance, Bloesch et al. [1] combined monocular RGB camera information with inertia measurement unit (IMU) data in an extended Kalman filter (EKF) to obtain relative poses in an unknown environment using robust visual inertial odometry. Viswanathan et al. [2] approached GPS denied localization using image matching between panoramic photos taken onboard a vehicle and satellite imagery, updating the estimated global position of the vehicle at each time step using a particle filter. Rather than performing pose estimation onboard the vehicle, Veronese
et al. [3] placed a number of LIDAR sensors throughout the environment, using an iterative closest point method based on the Hausdorff distance between sets of data obtained by the different sensors to localize the vehicle. Yang et al. [4] outfitted a UAV with a unique camera array consisting of a wide-angle field of view camera and a stereo camera, allowing it to obtain visual and depth information to localize relative to a moving target on the ground.

An approach to the GPS denied UAV localization problem most similar to the method presented in this thesis was explored by Gui et al. [5], who placed four IR lamps on a runway for an incoming aircraft to image and extract pose information from while landing. While similar in concept to the general pose estimation approach proposed in this thesis, the focus of this thesis is primarily on an IR fiducial design and data association algorithm that can be robustly and efficiently applied to UAV flight in real-world environments. Gui et al., on the other hand, focused mostly on the light detection and pose estimation aspects of the problem. For their IR lamp layout they used a simple trapezoidal configuration, and relied on several idealized assumptions when performing data association that would limit the robustness of the approach in complicated urban environments. For example, their association method depended on all four lights being visible at all times, and could only handle a maximum of two outlier points in the frame at any time. Additionally, the aircraft was assumed to approach the lights from a known direction, thus certain geometric properties like parallelism and angular measurements were relied on for the light association problem that would not be consistent if viewed from different perspectives. In this thesis, we introduce an IR constellation design that contains significant redundancy while still remaining distinguishable from any outliers in the environment, and can be recognized and associated when viewed from any arbitrary approach direction or perspective.

1.1.2 Deep Learning Data Association Approaches

The task of data association is of vital importance to many localization and state estimation methods. For instance, in the simultaneous localization and mapping (SLAM) problem, data association is heavily relied on when tracking or reobserving landmarks in the environment, with incorrect associations resulting in high measures of uncertainty or pose estimation drift [6]. Loop closures, where a system must decide if it has reached a previously visited location on the map based on current measurements of the environment, depend strongly on proper data association as well. In our proposed sparse IR fiducial based pose estimation approach as well as other point association based approaches, a single incorrect association between the observed 2D image points and known 3D world points can severely affect the accuracy of the solution.

A wide range of existing approaches for performing data association in various environments have been proposed, including probabilistic filtering [7], multihypothesis tracking [8], Markov chain Monte Carlo data association [9], iterative closest point [10], and histogram matching [11].
Such methods are generally used for measurements taken in unfamiliar environments with unknown landmarks or targets, but may not necessarily take advantage of known information about the environment when it is available. In the case of our IR fiducial based pose estimation, we assume knowledge of the nominal constellation design beforehand and exploit that information when analyzing camera images taken onboard the UAV to perform robust data association.

With a known fiducial constellation of sparse point IR sources, it may be natural to wonder if a deep neural network (DNN) could be trained to recognize and associate the individual lights of the constellation detected in camera images. Deep learning based approaches for data association have been applied before by Yoon et al. [12] to the multiple target tracking problem, with an encoder-decoder architecture trained to positively or negatively associate each target detection with each existing track. Luna et al. [13] used graph neural networks to sort out associations between objects detected by a multi-camera array providing overlapping views of a common scene. Additionally, the classic problem of objection detection and classification for camera images using deep learning has been widely studied and explored [14], [15], [16]. However, little work has been done to apply DNNs to the problem of associating specific points observed in a 2D IR camera image with known points in the 3D environment.

If instead of analyzing an entire IR camera image we consider only the detected lights’ pixel locations, we can formulate the IR lights in the image as a 2D point cloud and apply point cloud analysis techniques to obtain desired point associations. One of the pioneering works in the area of deep learning applied to 3D point cloud analysis is the PointNet model, proposed by Qi et al. [17]. PointNet, along with its revised successor PointNet++ [18], was designed to perform object classification and part segmentation on 3D point clouds by extracting global and local features using a symmetric pooling operation. Since its inception PointNet has inspired a variety of similarly structured or derivative models, including PointNeXt [19], Point-BERT [20], and PointMLP [21]. Although designed for and demonstrated on 3D point cloud data, with some modification PointNet++ can be applied to 2D point clouds, performing classification and segmentation tasks on camera image data. In Chapter 4 we will use this modified PointNet++ model to perform both outlier detection and data association on 2D point clouds derived from IR images of our designed fiducial constellation.

1.2 Contributions

In this thesis we make the following primary contributions:

- We develop a novel sparse IR fiducial constellation design based on projective invariant properties that can be identified and associated from arbitrary camera perspectives for performing pose recovery.

- We present along with the the sparse IR constellation a geometry-based data association algorithm that efficiently associates points of
the constellation while achieving robustness to outliers and occlusions that commonly arise when operating in realistic urban environments.

- We develop a pose estimation pipeline, which requires only a monocular IR camera, that is capable of producing high accuracy pose measurements at real-time rates from images of the proposed constellation, with performance demonstrated on hardware with real-world data.

- We augment our geometry-based data association algorithm with a flexible learning-based outlier rejection and point segmentation approach based on the PointNet++ architecture. This learning-based approach offers increased efficiency when detecting and removing outlier IR detections from the camera image, and can output per-point labels predicting either each detected point’s location in the constellation or the segment of the constellation to which each point belongs.

1.3 Thesis Overview

In the following chapters we will explore the problems, approaches, and results associated with the main contributions of this thesis. In Chapter 2 we will review some of the relevant topics fundamental to the work done in this thesis as well as provide background to some of the technical approaches and methods that will be used in the later chapters. Chapter 3 is an accepted paper to the 2024 International Conference on Unmanned Aircraft Systems in which we present our IR light fiducial, pose estimation pipeline, and projective invariant-based data association approach for UAV landing in urban environments. Chapter 4 consists of a paper planned to be submitted to the 2024 American Institute of Aeronautics and Astronautics SciTech Forum which introduces our learning-based approaches to the IR camera image data association problem. Finally in Chapter 5, we conclude with a summary of the contributions made in this work as well as recommendations for further research in the areas developed in this thesis.
2 Technical Approach and Background

In this chapter we review some fundamental principles, methods, and theory that form the foundation of the work presented in this thesis. These topics, which are referenced and briefly defined in later chapters, are presented here in greater depth for the reader’s understanding.

2.1 Pose Estimation Pipeline Overview

We introduce briefly an outline of the pose estimation framework developed in this thesis for obtaining estimates of the vehicle’s position and orientation from monocular IR camera images. The specific steps of this pipeline are described in greater depth in the following sections of this chapter as well as later in Chapter 3, but we introduce them here to provide familiarity with other foundational topics that will also be discussed in this chapter. A visualization of the steps of this pose estimation pipeline is shown in Figure 2.1. As a note, the IR lights in the camera images within the figure may be difficult to see due to the restrictive IR filtering, but can be viewed more clearly in an electronic copy by zooming in.

Figure 2.1: Steps of the proposed pose estimation pipeline, beginning with acquisition of three sequential camera frames, light detection performed on the composite image, data association on the detected lights, and finally pose estimation from the calculated 2D-3D point correspondences.
The UAV is outfitted with a calibrated IR filtered camera, with IR light sources placed on the landing site in a prearranged design at precisely surveyed locations. The lights of the constellation are programmed to flash asynchronously of each other at a frequency of one third the framerate of the camera so that over a window of three camera frames, each light is on for one frame and off for the other two. The pipeline is initiated during flight as the onboard camera acquires bursts of three sequential images, which capture the lights of the constellation as they flash on. A composite image of the constellation is synthesized by subtracting the per-pixel minimum intensity values of the three frame window from the maximum intensity values of the images, effectively removing background noise and clutter. This composite image is then passed through a circle or blob detector that extracts the centroid pixel locations of each light that appears in the image.

Once the locations of the IR lights in the camera image have been obtained, the next task is to solve is the data association problem, determining which point on the landing site corresponds with a given point in the camera image. Given the solved associations between the 2D pixel locations and 3D world locations, the pose of the camera can be obtained by solving the Perspective-n-Point problem, which determines the camera rotation and translation needed to project points from the 3D world frame to their corresponding pixel locations in the image frame. The camera pose information can then be transformed to the vehicle frame to provide vehicle pose measurements for the landing aircraft.

This proposed pose estimation pipeline will be shown later in Chapter 3 to obtain high accuracy, real-time pose updates when operating on IR images obtained onboard a UAV. In a related work, Brown [22] fused pose measurements obtained using this pose estimation approach with information from an IMU using an EKF to control a UAV during landing. That work demonstrates the capacity of this proposed pose estimation pipeline to be used robustly in real world flight scenarios.

2.2 IR Light Detection

Filtered IR camera images offer advantages over RGB camera images due to the reduced amount of ambient IR light present in the environment. By selecting a camera bandpass filter specific to the wavelength emitted by the IR light sources in the fiducial constellation, we can effectively remove extraneous visual information from the background environment that distracts from the target constellation. Using this passive filtering it is still possible, however, for undesired IR light sources to appear in camera images as outlier detections. For example, active IR light sources in the environment, such as car headlights or street lights, as well as concentrated ambient IR light reflected towards the camera by reflective surfaces can appear in the filtered image as clutter, which complicates the task of identifying and associating the constellation points.

Data association methods typically rely on some sort of sampling approach as they try to find matches between sets of points. Increased clutter in the image in the form of outlier detections then increases the number of
drawn samples needed to reach a set of high confidence point associations. To mitigate this difficulty we can use an active min-max filtering technique to remove unwanted IR light sources that appear in the filtered images.

Min-max filtering can be performed when the target constellation lights are flashed at a known frequency relative to the frame rate of the camera. For example, if the camera frame rate is set to 100 frames per second, we can modulate the constellation lights to flash at a rate of 100 Hz with a 33% duty cycle such that over a sequence of three camera frames, each light is illuminated for nominally one frame and off for the next two frames (see Figure 2.2). Taking a set of three sequential camera images we can then calculate the per-pixel maximum and minimum intensity values, then compute a composite min-max image by subtracting the minimum intensities from the maximum intensities at each pixel location. Assuming that the movement of the lights in the images is small between camera frames, we then obtain a composite image that contains each of the constellation lights with any constant intensity background or outlier IR sources removed.

![Figure 2.2: Light detection method for filtering background noise and outlier light sources from IR camera images. Constellation lights are flashed at one-third the frame rate of the camera (red pixel) so that they appear in strong contrast with the static background (blue pixel) when a min-max filter is applied to three sequential images. An example of a min-max filtered image is shown in Figure 2.3.](image)

Although we assume control over the flashing frequency of each individual light we do not assume any synchronization between the other lights in the constellation or with the camera. This means that cases can arise in which a light’s illumination time is captured spread out over two frames, with a worst-case scenario of the light being on for exactly half of one camera frame and half of the following frame. However, by using a three frame window, the light is guaranteed to be off in the third camera frame, so taking the per-pixel min-max value will still result in a clear contrast between the constellation lights and the image background.
Once a min-max image has been calculated with background and constant intensity IR light sources removed (see Figure 2.3), we can extract the pixel locations of the constellation lights using a readily available circle or blob detection method. In the chapters that follow we will use the SimpleBlobDetector class provided by OpenCV to obtain centroid pixel locations for each light source that appears in the min-max image.

2.3 Pose Estimation Problem

We consider a camera coordinate frame $c$ with the position of its origin

$$
p^i_{c/o} = \begin{bmatrix} p^i_{c/o_x} \\ p^i_{c/o_y} \\ p^i_{c/o_z} \end{bmatrix}
$$

relative to an inertial point $o$ expressed in an inertial world frame $i$ and orientation

$$R_i = \begin{bmatrix} i^i_c & j^i_c & k^i_c \end{bmatrix}
$$

expressed in the inertial frame as shown in Figure 2.4. We can denote the pose of the camera by the homogeneous transformation matrix from the inertial frame $i$ to the camera frame $c$

$$T^i_c = \begin{bmatrix} R^i_c & p^i_{c/o} \\ 0 & 1 \end{bmatrix}
$$

Assuming a calibrated camera with intrinsic matrix
Figure 2.4: Example of camera frame $c$ with position $p_{c/o}$ relative to the inertial origin $o$ and coordinate vectors $i_c, j_c, k_c$ which can be expressed in either the camera frame, $i_c^c, j_c^c, k_c^c$, or the inertial frame, $i_i^i, j_i^i, k_i^i$.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where $f_x$ and $f_y$ are the camera focal lengths and $(c_x, c_y)$ is the pixel location of the optical center, we can project 3D points in the inertial world frame to a 2D camera image using the pose information. We first transform a 3D world point in homogeneous form

$$\bar{p}_i = \begin{bmatrix} p_i^i_x \\ p_i^i_y \\ p_i^i_z \\ 1 \end{bmatrix}$$

from the inertial frame to the camera frame by the homogeneous transformation

$$\bar{p}_c = T_{i}^c \bar{p}_i$$

(2.1)

where

$$T_{i}^c = T_{i}^{c\rightarrow i} = \begin{bmatrix} R_{i}^c & -R_{i}^c p_{c/o}^i \\ 0 & 1 \end{bmatrix}$$

(2.2)

noting that $R_{i}^c = R_{i}^{c\rightarrow i}$. We can return the point from its homogeneous form by the transformation
\[ \mathbf{p}^c = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \hat{\mathbf{p}}^c \] (2.3)

then convert the point expressed in the camera frame to an image-centered coordinate \( \mathbf{p}^{ic} \) using the camera calibration matrix

\[ \mathbf{p}^{ic} = K \mathbf{p}^c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbf{p}^c \] (2.4)

The final pixel coordinates \( \hat{\mathbf{p}}^c \) of the point are obtained in homogeneous form by projecting the point to the plane a unit distance from camera’s \( xy \) plane, i.e.

\[ \begin{bmatrix} \hat{\mathbf{p}}_x^c \\ \hat{\mathbf{p}}_y^c \\ 1 \end{bmatrix} = \frac{1}{p^z} \begin{bmatrix} p_x^{ic} \\ p_y^{ic} \\ p^z \end{bmatrix} = \frac{1}{\lambda} \mathbf{p}^{ic} \] (2.5)

where \( \lambda = p^z \).

If we have \( n \) homogeneous points in the 3D world frame

\[ \bar{\mathbf{P}}^i = \{ \bar{\mathbf{p}}^i_1, \bar{\mathbf{p}}^i_2, \ldots, \bar{\mathbf{p}}^i_n \} \]

with known (homogeneous) pixel locations in the 2D camera image

\[ \tilde{\mathbf{p}}^c = \{ \tilde{\mathbf{p}}^c_1, \tilde{\mathbf{p}}^c_2, \ldots, \tilde{\mathbf{p}}^c_n \} = \left\{ \begin{bmatrix} \hat{\mathbf{p}}^c_1 \\ 1 \\ \hat{\mathbf{p}}^c_2 \\ 1 \ldots, \begin{bmatrix} \hat{\mathbf{p}}^c_n \\ 1 \end{bmatrix} \right\} \]

then using Equations (2.1) through (2.5) the pose estimation problem becomes solving for the camera orientation \( R_i^c \) and position \( p^i_{c/o} \) that satisfy the relation

\[ \tilde{\mathbf{p}}^c_j = \frac{1}{\lambda_j} K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} T_i^c \bar{\mathbf{p}}^i_j \] (2.6)

for \( j = 1, \ldots, n \), or in the nonideal case with noise,

\[ \arg\min_{R_i^c, p^i_{c/o}} \sum_{j=1}^n \left\| \tilde{\mathbf{p}}^c_j - \frac{1}{\lambda_j} K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} T_i^c \bar{\mathbf{p}}^i_j \right\| \]

The above problem of solving for a camera’s pose using a set of \( n \) 3D points \( \bar{\mathbf{P}}^i \) with their associated 2D pixel locations \( \tilde{\mathbf{p}}^c \) is known as Perspective-n-Point (PnP). The PnP problem is well-known and has been widely studied,
with many existing PnP solvers offering efficient and accurate solutions to
the pose recovery problem. In most cases \( n \geq 4 \) points are needed to obtain a
unique pose solution, although fewer points can be used in certain situations.
In this work we use the iterative method of OpenCV’s \texttt{solvePnP} function to
recover camera poses. The difficulty then of performing pose estimation is
in obtaining the correctly associated sets of points \( \hat{\mathbf{P}} \) and \( \hat{\mathbf{P}}^c \) which can be
used to constrain the PnP problem. This challenge of obtaining the desired
sets of associated 3D world points and 2D pixel locations is the objective our
proposed constellation design and data association algorithm, discussed in
depth in Chapter 3.

2.4 Projective Invariance

Traditional vision-based pose recovery methods obtain 2D to 3D point
correspondences by extracting visual features (SIFT, ORB, FAST, etc.) from
the camera image, then comparing those features to known landmarks in the
environment or to previously observed features from prior frames. IR images,
however, typically have significantly less visual detail and are ill-suited for
constructing descriptors for each point of interest. Instead of relying on
visual features to identify and associate constellation points in our IR camera
images, we rely on geometric relations between the points to distinguish
them.

One prominent difficulty with determining 2D to 3D point associations
from camera images is the dependence of many geometric relations on the
viewing perspective. For example, typically useful properties like distances
between points, angles between lines, parallelism, symmetry, ratios of
distances, and areas are all variable under projective transformations such as
the camera projection model in Equation (2.6). This difficulty is made more
challenging when uncertainty is added to the image in the form of noise in
the detected light pixel locations, outlier IR light sources in the image, and
occlusion of the constellation due to obstruction or electrical/mechanical
failure of the lights.

Although geometric relations like distances and angles are ill-suited
for sorting out point associations in camera images, there do exist certain
projective invariant properties and relations between points and lines that
can be exploited when designing a fiducial constellation. First, collinearity
of points is invariant under projections. This is a fairly intuitive property, as
points that are arranged in a line in a 3D frame will always remain collinear
when projected to a 2D image frame—no translation or rotation of a camera
can cause those points to appear separate from the line. Analogously, sets
of concurrent lines, which intersect each other at a single point, remain
concurrent under projective transformations.

Figure 2.5 shows some examples of these two projective invariant proper-
ties, with the concurrent lines \( \hat{A}, \hat{B}, \hat{C}, \) and \( \hat{D} \) intersecting at point \( O \), and sets
of collinear points \( A, B, C, \) and \( D, \) and \( A', B', C', \) and \( D' \). A third property
that we will rely on is the cross ratio, or a ratio of ratios of distances, which
has analogous definitions for sets of four collinear points and four concurrent
Figure 2.5: Examples of projective invariant properties, including collinearity of points, concurrency of lines, and the cross-ratio.

given collinear points \( A, B, C, \) and \( D \) as shown in Figure 2.5, the cross ratio is defined as

\[
XR_{ABCD} = \frac{AC \cdot BD}{BC \cdot AD}
\]  

(2.7)

with \( XR_{A'B'C'D'} \) for points \( A', B', C', \) and \( D' \) defined similarly. These two example cross ratios will in fact have the same value, as will any cross ratio of points on a line that cuts the four concurrent lines \( \overline{A}, \overline{B}, \overline{C}, \) \( \) and \( \overline{D}. \) This means that if we know the cross ratio of a set of points on a line observed in one frame, we can expect to obtain the same cross ratio for the same set of points when observed from any other perspective. This is useful for the problem of data association because we can calculate and compare the cross ratio of a set of points in an image to the cross ratio of points in the constellation, and if they do not match we can assume they are not the same sets of points.

A similar definition with similar conclusions can be constructed from a set of four concurrent lines, for example lines \( \overline{A}, \overline{B}, \overline{C}, \) \( \) and \( \overline{D} \) in Figure 2.5, with the angular cross ratio defined as

\[
XR_{\angle ABCD} = \frac{\sin \angle AC \cdot \sin \angle BD}{\sin \angle BC \cdot \sin \angle AD}
\]  

(2.8)

where \( \angle AC \) is the interior angle between lines \( \overline{A} \) and \( \overline{C}, \) \( \angle BD \) is the interior angle between lines \( \overline{B} \) and \( \overline{D}, \) and so forth. It should be noted that permutation of the points or lines can result in differing cross ratios, with a total of six different cross ratios that can be observed based on the constituent ordering. Denoting the value of cross ratio \( XR_{ABCD} \) as \( \tau, \) the six possible cross ratios are obtained by the permutations.
When viewing projections of collinear points located finite distances from the camera, the only permutation that can be observed is a reverse ordering of the points, which occurs when viewing them from a mirrored perspective. Since a reversed-order cross ratio is the same as the original-order cross ratio, there is only a single value that can be obtained as the cross ratio for a given set of collinear points observed in a camera image. For sets of concurrent lines, however, there can be some ambiguity in the ordering of the lines as seen in Figure 2.6, which can result in multiple distinct cross ratios. This can cause some difficulty when attempting to assign labels to lines based on calculated angular cross ratios, however if the identity of one of the lines is known beforehand, the rest of the lines can be associated unambiguously by ordering the concurrent lines in either clockwise or counterclockwise direction starting from the known line. This line association method will be revisited later in Chapter 3 when the design of the fiducial constellation is described.

Figure 2.6: The ordering of lines used to obtain the cross ratio of concurrent lines can naturally result in two distinct cross ratios.

There exist several projective invariants besides those discussed in this chapter, most of which are based in some way on the definition of the cross
ratio, such as the cross ratio of a central projection of a twisted cubic [23], the j-invariant of the cross ratio [24], and certain area invariants [25]. Although these properties are not discussed or used in this thesis, they may be useful for expanding the utility of our projective invariant constellation design for future work.

2.5 Deep Learning on Point Clouds

In addition to a first principles projective invariance based approach for camera image data association, we will explore in Chapter 4 a learning based approach that can be used in place of or alongside the first principles approach to improve robustness and efficiency of the algorithm. To apply deep learning methods to our problem of 2D to 3D data association, we formulate our data as 2D point clouds consisting of centroid pixel locations of the detected IR lights in each composite camera image as demonstrated in Figure 2.7. Our objective is to feed the set of 2D points into the network and receive out per-point labels designating each detection as either an outlier point or one of the target constellation points. The choice to approach this problem from a point cloud perspective rather than using a convolutional neural network (CNN) based approach as in many traditional deep learning image analysis methods was motivated by several factors including computational speed, camera flexibility, accuracy limitations, and compatibility with our proposed pose estimation pipeline. These factors are discussed in greater detail in Chapter 4.

![Figure 2.7: A 2D point cloud is formulated from an IR camera image by extracting the pixel locations of the detected lights in the frame. This 2D point cloud can then be used as an input to a deep neural network in place of using the entire camera image.](image)

Point clouds present a challenge to most traditional DNN models, as the input to the network consists of numerous points that are completely invariant to set order. For example, when formulating an input vector to
feed through the DNN, the expected output from the network should be the same regardless of how the input points are ordered in the vector. Adapting traditional neural network architectures to fit this constraint is a nontrivial task that can come at the expense of compromised network performance. For example, recurrent neural networks (RNNs), which are designed and trained to operate on sequences of inputs, if assumed to be capable of disregarding sequence order would need to be trained on augmented data consisting of permutations of each set of cloud points. While perhaps feasible for datasets consisting of small or sparse point clouds, this type of data augmentation would be time consuming and computationally expensive for training on large point clouds. Alternatively, if using a nonsequential model, the number of inputs for each data instance would generally need to be fixed, meaning the model would require an unrealistic constraint that all point clouds consist of the same number of points. Additionally, achieving the input order invariance condition would likely require the input point cloud to be sorted in some sort of canonical order which itself is a challenging task, particularly with measurement noise and uncertainty.

Figure 2.8: PointNet and PointNet++ perform the tasks of part segmentation and object classification on 3D point cloud inputs.

These challenging aspects of point cloud datasets are addressed by the PointNet++ model proposed by Qi et al. [18], which was designed to perform part segmentation and object classification on 3D point clouds (see Figure 2.8), and serves as the foundation of our deep learning approach to IR camera image data association. The original PointNet model proposed by Qi et al. [17] constructed features at each input point using layers of multilayer perceptrons (MLP), then summarized all of the local information into a global feature vector using a symmetric max pooling operation. This max pooling operation is the key to PointNet achieving input sequence invariance, as the order of the inputs to this layer has no effect on its overall global features output. When performing object classification, a final MLP head is
trained to output a single class label from the global feature vector. In the case of part segmentation, a copy of the global feature vector is concatenated to each point’s feature vector, then the resulting total point-feature matrix is passed through a series of MLPs to output a per-point score labeling each point as one of the specified parts of the object. In this thesis we use the part segmentation configuration of PointNet++ to first classify each point in the cloud as either an outlier detection or constellation light, then after removing outliers and normalizing the remaining data we feed the cleaned point cloud through a second segmentation model to obtain per-point labels for each inlier detection. These labels can either be the IDs for each point in the designed constellation, or the IDs for the different lines of the constellation.

PointNet++ builds on the approach used by its predecessor PointNet by introducing a set abstraction layer that uses PointNet blocks to learn local features around sampled points in the cloud, and a feature propagation layer that interpolates learned features from a subset of sampled points to a larger subset of the point cloud as seen in Figure 2.9. Using a multiscale grouping method visualized in Figure 2.10, the set abstraction layers are able to learn local features hierarchically by iteratively grouping points at differing radii around the sampled subset of points in the cloud. By stacking multiple set abstraction layers PointNet++ is able to reduce the point cloud to a small subset of points containing high dimensional features, which are then propagated back through the feature propagation layers to each point of the original point cloud. With these adjustments PointNet++ is able to better learn geometry-based relations between points located close together in the cloud, providing better local context to accompany the global features it uses to assign point labels during the part segmentation task.
Figure 2.10: Multiscale grouping used in the set abstraction layers of PointNet++ allows local features to be captured at differing scales and concatenated into a single descriptor.

2.6 Summary

The topics presented in this chapter will be referenced and built upon in the following chapters of this thesis. In Chapter 3 we discuss in greater depth the motivation, method, and results of the pose estimation pipeline, which fundamentally relies on the IR light detection method and a projective invariant constellation design. In Chapter 4 we will use the PointNet++ architecture in various configurations to build neural networks that can perform outlier rejection and data association tasks on cluttered 2D point clouds extracted from IR camera images. The combination of these methods and principles will be shown to provide for a robust, efficient method of IR vision-based pose estimation for landing aircraft.
3 Robust IR-Based Pose Estimation for Precision VTOL Aircraft Landing in Urban Environments

This chapter is composed from a paper entitled “Robust IR-Based Pose Estimation for Precision VTOL Aircraft Landing in Urban Environments” accepted to the 2024 International Conference on Unmanned Aircraft Systems [26]. I hereby confirm that the use of this article is compliant with all publishing agreements.

3.1 Abstract

This chapter presents a novel pose estimation framework that provides high-accuracy localization for vertical take-off and landing aircraft in settings where GPS is unreliable or unavailable. The proposed framework utilizes a sparse constellation of infrared lights that can be robustly identified and associated in the presence of outliers and occlusions, making it suitable for use in realistic urban environments. This is enabled by constellation designs that exploit properties of invariance under projective transformations. Flight-test results demonstrate that the framework is capable of running in real time at speeds of over 30 Hz while providing pose information at decimeter-level accuracy at ranges of over 100 m from the landing site.

3.2 Introduction

Use of unmanned aerial vehicles (UAV) has grown significantly in recent years, with applications including agricultural monitoring [27], infrastructure inspection [28], city mapping [29], and even extraplanetary exploration [30]. With this growth comes an increased need for reliable high-accuracy localization during UAV flight and especially landing, a critical phase of UAV operation where slight errors in estimated position can result in catastrophic aircraft failure. While GPS has traditionally been used to measure vehicle position during outdoor flight, this method can be unreliable for UAV use in urban environments where GPS signals suffer from multipath interference or obstruction. Additionally, GPS can be subject to adversarial attacks such as jamming or spoofing. Robustness to these issues requires an alternative localization approach that can be used alongside or in place of GPS measurements in a state-estimation framework, which may fuse GPS measurements with other sensors or use them directly for feedback control. Vision-based approaches to UAV localization have proven effective for pose estimation
in certain scenarios [31], however, methods that rely on visible light often are dependent on consistent lighting conditions and weather that permits good visibility. These issues can be largely alleviated by using an infrared (IR) light fiducial, which contrasts strongly with ambient outdoor lighting and can be detected from significant distances.

Existing fiducial patterns such as AprilTags and ArUco markers have proven useful for obtaining estimates of a camera’s pose [32], but contain fine visual details that limit the distance from which the pattern can distinguished and interpreted. Scaling the size of these markers to improve their operable range has physical and practical limitations, restricting their use in long-range applications. Our approach is to use a sparse, simply scalable fiducial constellation composed of point IR light sources, shown in Figure 3.1, that can be unambiguously identified and provide camera pose information at far distances from any arbitrary direction or orientation. Robustness to arbitrary camera perspectives allows our method to be used for aircraft of any type approaching the landing site from any given trajectory, making it suitable for fixed-wing aircraft that typically approach a runway from a designated direction, as well as for vertical take-off and landing (VTOL) vehicles that are able to approach a landing site from any direction.

![Figure 3.1: IR-filtered image of lights laid out on a landing site in the proposed constellation design.](image)

In addition to handling arbitrary views of the constellation, a pose estimation approach intended to function in real outdoor environments must be robust to non-ideal viewing conditions, including occlusions of constellation lights due to environmental obstructions or partial failure of the lights. Using a sparse light constellation in an urban setting also demands robustness to outlier light sources that arise as a result of devices in the environment actively emitting IR light, as well as reflective surfaces passively transmitting concentrated ambient IR sunlight at the UAV. Thus, the fiducial
used for this approach must have sufficient redundancy to handle occlusions while remaining distinguishable from any other IR light sources existing in the environment that may clutter the camera images.

In this work we propose a pose estimation framework, including a novel IR fiducial constellation and accompanying data association algorithm, for UAV landing that produces high-accuracy camera poses at real-time rates while being robust to partial occlusions and outlier light sources that occur in realistic urban environments. We additionally report results from hardware tests using a VTOL UAV that demonstrate the accuracy and efficiency of our approach, supporting its use as an alternative to GPS for urban flight.

### 3.3 Related Work

Given the significance of the problem of UAV landing in GPS-denied environments, several existing approaches have been proposed and explored. Zhang et al. [33] used a combined infrared-inertial system which fused poses calculated from detected runway lines in infrared images with inertial navigation system (INS) measurements in a square-root unscented Kalman filter. Yang et al. [34] used a ground-based IR camera array to detect an IR laser source on the nose of a fixed-wing UAV as it approached a runway. In addition to requiring multiple cameras to produce a pose estimate, their approach relies on a UAV approaching the landing site from an assumed direction to be observed by the camera array, which may be appropriate for fixed-wing aircraft but is a limiting assumption for VTOL aircraft.

A novel visual fiducial consisting of concentric circles with unique thickness ratios was proposed by Lange et al. [35]. Their vision-based localization approach was robust to partial occlusion and could be scaled with the UAV distance from the fiducial. While providing highly accurate height estimates, this fiducial alone was not suitable for producing horizontal position estimates, which they achieved by relying on additional angular measurements from the drone sensors. Martínez et al. [36] used both a ground-based camera array tracking colored landmarks on a UAV and an aircraft-mounted camera viewing visual tags on the ground to obtain pose estimates of the UAV.

An approach most similar to ours was explored by Gui et al. [5] who used airborne images of IR lamps placed on a runway to obtain pose estimates for a UAV when approaching for landing. Their work, while demonstrating the feasibility of the method, left the problems of IR light layout and data association largely unaddressed, relying on a four-point square light configuration that could be correctly associated only with all the lights functional and two or fewer outliers in the environment. These assumptions may be suitable in ideal rural environments with few extraneous light sources and perfectly functioning light sources, however we extend this approach to uncertain urban environments, where robustness to light occlusions and ambient IR light reflections are imperative for a vision-based approach.
3.4 Proposed Landing Approach

Our proposed pose estimation framework for landing in GPS-denied environments is shown in Figure 3.2. Point IR light sources are laid out in a specific constellation design on the landing site (see Section 3.5), while the UAV is outfitted with a forward-facing camera equipped with an IR bandpass filter set to capture bursts of three sequential images. The light sources are programmed to flash at the same frequency as the camera frame rate with a one-third duty cycle so that the lights are nominally illuminated in one of every three camera images, with a worst-case scenario of a light being captured at half intensity over two sequential frames. As the aircraft comes into viewing range of the IR light sources, the camera begins acquiring bursts of three consecutive images which are used to calculate pixel-by-pixel minimum and maximum intensity images over the three frames. Subtracting the minimum from the maximum intensity image produces a difference image which has ambient IR lighting removed while displaying the flashing light sources in high contrast as demonstrated in Figure 3.3.

![Figure 3.2: Overview of the proposed pose estimation framework which takes in IR camera images of the fiducial constellation and produces camera pose estimates.](image)

The high-contrast difference image is processed by a circle or blob detection algorithm that calculates the centroid location of each light source in the image, providing pixel coordinates in the camera image frame. In our implementation of the proposed pose estimation framework we used OpenCV’s computationally efficient SimpleBlobDetector class to carry out the task of light detection on the difference image. With the detected lights’ pixel locations, the next step requires assigning associations between detected lights in the camera image and the constellation lights laid out on the landing site, while appropriately identifying and ignoring any outlier light detections which may have appeared in the difference image. We present in Sections 3.5 and 3.6 respectively a novel constellation design and data association approach that solves this association problem at real-time speeds while offering robustness to both outlier light detections and obstructed or absent light sources.

Once the associations between the detected lights’ 2D pixel locations and 3D world locations have been established, the recovery of camera pose information relative to the constellation becomes the well-known problem of perspective-n-point (PnP), which can be solved using a number of existing algorithms and approaches. We again relied on OpenCV’s solvePnP function in our framework, using the iterative Levenberg-Marquardt optimization-based method to obtain the 3D rotation and translation that transforms points from the 3D landing site frame to the 2D camera frame. Taking the inverse of this transformation gives the full pose of the camera relative to the
landing site, which after applying any necessary camera-to-vehicle frame transformations produces the pose of the UAV relative to the landing site.

3.5 Constellation Design

The design of the IR light constellation is crucial when attempting to identify associations between detected lights in camera images and the points of the constellation laid out on the landing site. Projective transformations of points from a 3D world frame to a 2D camera image distort many important geometric relationships such as distances, angles, and even parallelism, making association based on such relationships unreliable. Additionally, when operating in real-world settings the IR light constellation must be designed in such a way that points can be associated in instances of obstructed or missing lights, such as in cases of mechanical or electrical failure, as well as in the presence of outlier IR light sources in the environment.

To design a constellation with distinguishable associations that is suitably robust to occlusions and cluttered conditions under projective transformations, we rely on a handful of useful projective invariant properties. In particular we focus on properties that are invariant for coplanar points and lines, since the landing site where the constellation lights will be laid out is nominally flat, although other more general invariants exist that could be applied to the design of nonplanar constellations. Our designed constellation, as shown in Figure 3.4, relies on the invariance of planar collinearity of points, concurrency of lines, ordering of points along a line or lines about an intersection, as well as the cross ratio, a ratio of ratios of distances or angles which serves as one of the primary invariants in projective geometry [23].

The cross ratio is defined analogously for sets of points along a line and for sets of concurrent lines as shown in Figure 3.5. For a set of four collinear
Figure 3.4: Fiducial constellation design, with 25 IR point light sources arranged along five extending legs.

Figure 3.5: Cross ratios can be calculated for sets of concurrent lines (angular) or collinear points (linear).

points, the linear cross ratio is defined as

$$ XR_{ABCD} = XR_{DCBA} = \frac{AC \cdot BD}{BC \cdot AD} $$  \hspace{1cm} (3.1)$$

while the angular cross ratio for a set of four concurrent lines is defined as

$$ XR_{\angle ABCD} = XR_{\angle DCBA} = \frac{\sin \angle AC \cdot \sin \angle BD}{\sin \angle BC \cdot \sin \angle AD} $$  \hspace{1cm} (3.2)$$

These values are constant for ordered sets of points and lines under any arbitrary perspective transformation, and can thus be used as identification tags of sorts to positively associate points and lines found in 2D camera
images regardless of the camera’s position and orientation relative to the constellation.

The constellation design, shown in Figure 3.4, is arranged with five concurrent lines extending radially from the intersection point, fanned out asymmetrically in a 180° semicircle. Each leg of the constellation contains five point light sources asymmetrically spaced apart, for a total of 25 points in the constellation. With four interior angles between the lines of the constellation, there are two distinct angular cross ratios embedded in the constellation, $XR_{\angle ABCD}$ obtained by taking the angles starting from the left side of line $A$ and moving clockwise, and $XR_{\angle ADCB}$ obtained by starting from the right side of line $A$ and moving counterclockwise. The specific angular spacing of the legs of the constellation used in our tests was chosen using a Monte Carlo simulation that selected interior angles that would maximize the difference between the two angular cross ratios of the constellation to reduce any ambiguity in correctly associating lines during the data association step. The linear spacing between the points on each line was similarly determined using a Monte Carlo approach that maximized the difference between linear cross ratios when calculated for each subset of four points on the line. A fifth point is included on each line for redundancy so that linear cross ratios can still be calculated to help identify points in a line in the event that a point on the line is occluded.

3.6 Data Association Algorithm

With the IR lights arranged in the constellation design detailed above, 2D point clouds of lights detected in the camera images can be uniquely associated with points in the constellation by our proposed data association approach outlined by Algorithm 3.1.

**Algorithm 3.1 Data Association Algorithm**

1: Use RANSAC to find a best-fit set of four concurrent lines that matches the angular cross ratios of the constellation.
2: Using the known angular cross ratios of the constellation, determine the identity of each detected line in the constellation.
3: For each detected line, use the known linear cross ratios of the constellation to cast votes for each subset of four detected points.
4: Associate detected points which meet a voting threshold with their corresponding constellation points.

3.6.1 RANSAC Model Fitting

The algorithm begins by using a Random Sample Consensus (RANSAC) approach to find the lines of the constellation from among the potential clutter by randomly sampling four pairs of points at a time from the detected points in the image. When searching for lines in the camera image, we avoid using the standard slope-intercept form of lines $y = mx + b$, as computer precision deteriorates and eventually is lost as lines approach the vertical
with an infinite slope. We instead use the polar coordinate form of a line, used also for the Hough line transform,

\[ p = x \cos \theta + y \sin \theta \quad (3.3) \]

where \( p \) is the perpendicular distance from the line to the coordinate system origin and \( \theta \) is the angle from the horizontal to the normal of the line (see Figure 3.6). This polar parameterization, allows us to fit and evaluate lines without any loss of precision based on the slope of the line. Denoting the pixel locations of a pair of detected lights as \((x_1, y_1)\) and \((x_2, y_2)\), we solve for \( \theta \) in Equation 3.3 as

\[ \theta = -\arctan2(x_1 - x_2, y_1 - y_2) \quad (3.4) \]

which we can plug back into Equation 3.3 to solve for \( p \). After solving for \( p_i \) and \( \theta_i \) for \( i = \{1, \ldots, 4\} \), we end up with a system of equations

\[
\begin{align*}
p_1 &= x \cos \theta_1 + y \sin \theta_1 \\
p_2 &= x \cos \theta_2 + y \sin \theta_2 \\
p_3 &= x \cos \theta_3 + y \sin \theta_3 \\
p_4 &= x \cos \theta_4 + y \sin \theta_4
\end{align*}
\]

(3.5)

We can check if these four lines formed by the randomly sampled pairs of points intersect at a single point by calculating the least-squares solution to Equation 3.5 and evaluating the magnitude of the residual. Letting

\[
b = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \quad A = \begin{bmatrix} \cos \theta_1 & \sin \theta_1 \\ \cos \theta_2 & \sin \theta_2 \\ \cos \theta_3 & \sin \theta_3 \\ \cos \theta_4 & \sin \theta_4 \end{bmatrix}, \quad \text{and} \quad z = \begin{bmatrix} x \\ y \end{bmatrix}
\]

we have the system of lines in the form \( b = Az \), which can be solved using the pseudoinverse

\[ z = (A^T A)^{-1} A^T b \quad (3.6) \]

which is guaranteed to exist as long as \( \text{rank} A = 2 \), meaning that at least two of \( \theta_i \) for \( i = \{1, \ldots, 4\} \) must be unique up to a 180° rotation. Although this condition technically produces a solution, we are interested only in cases where all \( \theta_i \) are unique up to a 180° rotation, which can be checked by manual inspection of the angles. \( z \) represents the point which minimizes the distance to each line, which is the intersection point if the four lines are concurrent. In the case of noiseless images the least squares solution \( z \) perfectly satisfies Equation 3.5 and the magnitude of the residual \( r = Az - b \) is zero. In realistic scenarios with noise in detected pixel locations and IR light placement, \( \|r\| \) is nonzero but small, thus a threshold value \( \alpha \) can be set,
Figure 3.6: Polar coordinate representation of a line used for line detection and fitting in the data association algorithm. \( p \) is the distance from the line to the origin, and \( \theta \) is the angle from the horizontal to the normal to the line.

with sampled lines having a residual magnitude \( \| r \| > \alpha \) being considered nonconcurrent and hence not the correct lines of the constellation.

If the sampled points do not form concurrent lines, the RANSAC iteration terminates early and a new set of four pairs of points is sampled and checked for concurrency. When a set of concurrent lines with residual magnitude \( \| r \| \leq \alpha \) is found, we calculate an angular cross ratio from the set of sampled lines which is then compared to the two known angular cross ratios of the constellation. Denoting the set \( \Theta = \{ \theta_i \} \) for \( i = \{1, \ldots, 4\} \), let the set \( \Theta^* = \{ \theta_i^* \in \Theta \mid \theta_i^* < \theta_{i+1}^* \} \) for \( i = \{1, \ldots, 4\} \) be the sorted set of monotonically increasing angles in \( \Theta \). Defining \( \theta_{i/1}^* = \theta_i^* - \theta_1^* \), the angular cross ratio of the sampled lines is calculated with Equation 3.2 using the set \( \Theta^* \),

\[
XR_{\angle 1234} = \frac{\sin \theta_{1/3}^* \cdot \sin \theta_{2/4}^*}{\sin \theta_{1/3}^* \cdot \sin \theta_{1/4}^*}
\]  

(3.7)

If the calculated angular cross ratio is within some threshold \( \gamma \) of one of the known angular cross ratios of the constellation, the sampled set of lines is checked for number of inlier points among the detected lights in the camera image. Otherwise, the RANSAC iteration again terminates early and a new set of four pairs of points is sampled.

To determine the number of inlier points captured by the set of sampled lines, we calculate the distance from each point to each sampled line. Defining \( \bar{\theta}_i = \frac{\pi}{2} - \theta_i \) as the angle between line \( i \) and the horizontal, we apply to the homogeneous coordinates of the points \([x \ y \ 1]^T\) a set of modified SE(2) transformations \( T_i \) defined by
\[ T_i = \begin{bmatrix} 0 & 1 & 0 \\ \cos \tilde{\theta}_i & -\sin \tilde{\theta}_i & 0 \\ \sin \tilde{\theta}_i & \cos \tilde{\theta}_i & -p_i \\ 0 & 0 & 1 \end{bmatrix} \] (3.8)

which selects the transformed \( y \) components, \( y' \), of the points after rotating by the angle \( \theta_i \) and subtracting the vertical offset \( p_i \). The absolute value of these \( y' \) values is the distance from the points to line \( i \), so we look for the distance from each point to its closest line. Letting

\[ A = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & \ldots & x_n \\ y_1 & y_2 & \ldots & y_n \\ 1 & 1 & \ldots & 1 \end{bmatrix} \]

be a \( 4 \times n \) matrix, we search for the column-wise minimum absolute valued elements of \( A \), which we denote \( U \) defined by

\[ u_j = \min_{i \in \{1, \ldots, 4\}} |a_{ij}| \quad \forall j \in \{1, \ldots, n\} \] (3.9)

where \( n \) is the number of detected points. The number of inliers to the sampled lines then is the number of elements of \( U \) that are smaller than some outlier distance threshold \( \delta \). The specific line to which each detected point is closest is the row of \( A \) which produces the minimum absolute value for each column:

\[ v_j = \text{argmin}_{i \in \{1, \ldots, 4\}} |a_{ij}| \quad \forall j \in \{1, \ldots, n\} \] (3.10)

where \( V \) is the vector of closest line indices. Using the elements of \( V \) that correspond to elements \( \{u \in U \mid u < \delta\} \), we can partition the total set of points \( P = \{(x_j, y_j) \mid \forall j \in \{1, \ldots, n\}\} \) into sets \( P_{\text{outliers}} \), which contains points which are not within distance \( \delta \) from any line, and \( P_i \), which contain points that are closest to the lines \( i = 1, \ldots, 4 \) respectively.

Once the model of four concurrent lines with a known angular cross ratio and the largest number of inliers is determined, we fit lines to each set of inlier points \( P_i \) for \( i = 1, \ldots, 4 \). Using Equation 3.3 to fit \( p_i \) and \( \theta_i \) to set \( P_i \) of size \( m \) with pixel values \((x_j, y_j)\) for \( j = 1, \ldots, m \) gives

\[ p_i = x_j \cos \theta_i + y_j \sin \theta_i \] (3.11)

which can be rearranged to the form \( A z = 0 \) where

\[ A = \begin{bmatrix} x_1 & y_1 & -1 \\ x_2 & y_2 & -1 \\ \vdots & \vdots & \vdots \\ x_m & y_m & -1 \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} \cos \theta_i \\ \sin \theta_i \\ p_i \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \] (3.12)
In both noiseless and noisy images we can solve for $z$ by finding the null space of $A$, or its closest approximation, using a singular value decomposition. Let $z^* = \begin{bmatrix} z_1^* & z_2^* & z_3^* \end{bmatrix}^T$ be the right singular vector corresponding with the smallest singular value of $A$, which is equal to $z$ scaled by some factor $\beta$, $z^* = \beta z$. To recover the scale $\beta$ note that $z_2^* / z_1^* = \tan \theta_i$, so we can solve for $\theta_i$ using

$$\theta_i = \text{atan2}(z_2^*, z_1^*)$$

Then $\beta = z_1^* / \cos \theta_i$ (or $z_2^* / \sin \theta_i$ in the case that $\cos \theta_i = 0$) and $p_i = \frac{z_3^*}{\beta}$. The use of $\text{atan2}$ is necessary to ensure the sign of $p_i$ is correct, as wrapping $\theta_i$ to $[-\frac{\pi}{2}, \frac{\pi}{2}]$ can cause $p_i$ to take on the wrong sign.

### 3.6.2 Line Association

Once lines have been fit to each set of points $P_i$ for $i = 1, \ldots, 4$ we must associate each detected line with a line of the designed constellation. Following the notation of Figure 3.4, we assume line $A$ can be distinguished as the set containing the most points, as it consists of ten points while each other line has only five. In the case of occluded lights in line $A$ combined with outlier lights detected on another line, it is possible for this assumption to be violated, so we can also check that the supposed line $A$ coincides with the convex hull of the detected points, or in other words, the inlier points in the image are bounded on one side by line $A$. This can be done by applying the transformation in Equation 3.8 using $\theta_i$ and $p_i$ for the supposed line $A$ and checking the sign of the $y'$ values. If the line is correct, the sign of $y'$ values with magnitude $|y'| > \epsilon$ should be either all positive or all negative.

Once the set of points for line $A$, $P_A$, is known along with its fitted line parameters $p_A$ and $\theta_A$, we must identify the remaining lines $B$, $C$, and $D$. Let $\theta_1^*, \theta_2^*, \theta_3^*$ be the angles and $P_1^*, P_2^*, P_3^*$ be the point sets for the remaining unidentified lines. We wrap and sort the angles so that they are monotonically increasing in the range $[\theta_A, \theta_A + \pi]$. Denoting these sorted angles as $\theta_1^*, \theta_2^*, \theta_3^*$, we calculate the angular cross ratio as in Equation 3.2

$$XR_{A123} = \frac{\sin \theta_2^* \cdot \sin \theta_1^*}{\sin \theta_1^* \cdot \sin \theta_3^*}$$

Since we have already checked that the angular cross ratio of the fitted lines matches a cross ratio of the constellation, the value calculated in Equation 3.14 will be close to constellation cross ratio $XR_{ABCD}$ or $XR_{ACDB}$. We can then associate $\{\theta_1^*, \theta_2^*, \theta_3^*\} \rightarrow \{\theta_B, \theta_C, \theta_D\}$ or $\{\theta_1^*, \theta_2^*, \theta_3^*\} \rightarrow \{\theta_D, \theta_C, \theta_B\}$ depending on the matched cross ratio, with point sets $P_i^*$ accordingly associated for $i = 1, 2, 3$.

### 3.6.3 Point Association

With the four constellation lines and their corresponding point sets identified, the remaining task is to identify each detected point within the fitted lines
as a specific point in the constellation or an outlier. We first recalculate the intersection point of the concurrent lines, which may have changed since performing RANSAC due to the addition of inlier points on each line, using Equation 3.6. In the steps that follow we treat line $A$ as two separate lines of five points each.

![Figure 3.7: Visual representation of the voting process used to determine the associations for points on a detected line. Cross ratios are calculated using combinations of the unknown points (1 through 6 above) along the detected line, and are compared to the known cross ratios of combinations of points in the constellation. A voting matrix is used to keep track of votes for each possible association between points on the line and points in the constellation.](image)

To associate individual points within each detected line, we use a cross ratio-based voting scheme, a visual representation of which is shown in Figure 3.7. Since calculation of the linear cross ratio defined in Equation 3.1 requires four collinear points, no attempt is made to associate points belonging to detected lines with fewer than four points. For each line with four or more detected points, we sort the points in order of increasing distance $d$ from the intersection point, $D = \{d_1, \ldots, d_n\}$. We then take every combination $C_j$ of four points and calculate their linear cross ratio using Equation 3.1. If a combination’s cross ratio is within some threshold $\lambda$ of one of the constellation’s $\binom{5}{4} = 5$ linear cross ratios, each of the points in the combination receives a vote for the associated point in the constellation line. Once the cross ratios have been calculated and voted on for each combination, those points which received a number of votes greater than some threshold...
\( \mu \) become associated with the points for which they received the most votes. Added robustness to the voting scheme against potential outliers can be achieved by assigning associations based on a confidence metric calculated as the ratio of top received votes to second-most received votes for each points, positively associating only points which are identified unambiguously.

3.7 Results

We tested the efficacy and computational efficiency of our proposed pose estimation framework in several outdoor landing scenarios with the framework running real-time onboard the UAV. Our testing consisted of several different types of landing trajectories approaching the constellation from different directions to validate the robustness of our data association approach and pose estimation framework to arbitrary projective transformations. Flight tests took place in areas with reflective surfaces like parked cars and metal signs nearby, which introduced outlier light detections similar to those that might be expected in an average urban setting.

3.7.1 Test Setup

Flights were conducted using an Aurelia X6 Standard hexacopter drone, which carried the camera and computer used for operating the pose estimation pipeline. The computer used was an Intel NUC with a 12th Gen Intel Core i1-1270p processor, consisting of 12 cores/16 threads with a max frequency of 4.80 GHz. The drone was also equipped with a u-blox ZED-F9P real-time kinematic (RTK) GPS receiver that measures position with centimeter-level accuracy and was used to obtain ground-truth position measurements to which we compared our estimated poses. RTK GPS is able to achieve high accuracy measurements using a reference base station that detects and corrects errors in position measured by the GPS receiver [37]. While useful for our flight tests, which were performed in open areas away from buildings that could obstruct satellite signals, in dense urban settings RTK GPS would suffer from the same shortcomings as traditional GPS.

Imagery was acquired using an Alvium 1800 U-240m monochrome camera equipped with an 850-860 nm bandpass filter that passively removed ambient IR light outside of the target spectrum of the light sources. The camera framerate was set to 100 frames per second, with the IR light sources programmed to flash asynchronously at the same 100 Hz frequency with a duty cycle of 33% as described in section 3.4. The light sources were each composed of 32 high power 850 nm surface mount LEDs laid out in an 8×4 array on a metal core printed circuit board, mounted externally on the lid of a Pelican Vault case which housed the battery and driver board for the light. Prior to each set of flight tests we laid the lights out on the landing site in the designed shape of the constellation, then precisely surveyed the location of each light using the drone’s RTK GPS receiver. These surveyed light locations, after being transformed to a local coordinate system relative to the RTK GPS base station, were used as the known 3D world frame coordinates for the PnP solver.
The fiducial constellation used for these flight tests was designed using the approach outlined in Section 3.5 with the concurrent lines arranged at angles of 0°, 31.59°, 89.78°, 121.35°, and 180° as shown in Figure 3.8. During the castle trajectory flight test described below in Section 3.7.2, the distances from the intersection point to the five points on each line were 4.71 m, 10.70 m, 14.27 m, 18.25 m, and 23.12 m respectively, while these distances were scaled by 1/2 for the glideslope trajectory test. These specific distances and angles were selected using the Monte Carlo approach described in Section 3.5.

### 3.7.2 Estimated Pose Accuracy and Computational Time

The results from two of our test flight trajectories are summarized in Figures 3.9 through 3.12. The castle trajectory consists of a circular path of 100 m radius around the constellation, modulated with periodic altitude changes of 10 m. This trajectory was designed to verify the capability of our framework to produce pose estimates from views of the constellation produced by landing approaches from any arbitrary direction. The second tested trajectory is a traditional landing glide slope from approximately 200 m away approaching the landing site at a 7° slope, similar to the approach trajectory of a fixed-wing aircraft. This flight test was intended to demonstrate how error in the estimated position increases with distance from the constellation.

During the castle trajectory flight test the position errors in the north and east directions averaged 0.232 and 0.248 m, with standard deviations of 0.212 and 0.242 m respectively as shown in Figure 3.10. The combined horizontal error of 0.340 m represents a 0.34% error compared to the UAV’s horizontal distance of 100 m from the center of the constellation. The vertical error during the castle flight test averaged 0.146 m, with a standard deviation of
0.146 and a maximum error 1.96 m.

![Figure 3.9: Castle trajectory test flight with RTK GPS ground truth plotted with our position estimates. The IR light constellation is plotted in red.](image)

The castle trajectory consisted of 6,000 sets of three images, of which 4,566 were able to have the lights properly associated and pose estimated. Data association using the algorithm described in Section 3.6 proved robust to both occlusions and outliers, as pose estimates were produced when detecting as few as 14 points and as many as 31 points in the camera image. Average computation times over the flight test for each component of the pose estimation framework are listed in Table 3.1, with the average total time of the framework taking a little more than 30 ms, which is equivalent to a 33 Hz update rate, much faster than standard GPS update rates.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Computation Time (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Subtraction</td>
<td>4961.3</td>
</tr>
<tr>
<td>Light Detection</td>
<td>10064.9</td>
</tr>
<tr>
<td>Data Association</td>
<td>15104.5</td>
</tr>
<tr>
<td>Pose Estimation (PnP)</td>
<td>213.2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>30343.9</strong></td>
</tr>
</tbody>
</table>

Failures to produce pose estimates during the flight occurred primarily when noise in the pixel locations of the lights, sometimes from jerky motions of the UAV accentuated by the asynchronous light flashing, caused points to
be outside the thresholds set in the data association algorithm. This issue precluded points from being associated for the following PnP step, but can be partially resolved by adjusting the threshold values in Section 3.6 to more suitable values.

The glide slope trajectory test shown in Figure 3.11 was performed using the same constellation as the castle trajectory test, but with the distances between points scaled by 1/2. This limited the range from which lights could be detected and distinguished, but we were still able to perform data association and obtain pose estimates on images of the constellation taken from over 175 m away. Pose estimate errors in the north and east directions averaged 0.314 and 0.665 m with standard deviations of 0.302 and 0.319 m respectively. As seen in Figure 3.12 the error in each axis decreases with increased proximity to the constellation. Maximum errors

Figure 3.10: Error magnitudes of our position estimates compared to RTK GPS ground truth for the castle trajectory.
in the respective north and east directions were 3.21 and 2.52 m. Vertical error during the glideslope test averaged 0.116 m with a standard deviation of 0.329 and maximum error value of 3.81 m. A nearly constant error offset in the north and east components of the pose estimates can be seen in Figure 3.12, which may suggest a suboptimal camera calibration or imprecise calculation of the transformation between the camera and RTK GPS receiver on the aircraft. A video showing the onboard camera imagery of the light constellation during the glideslope descent can be found at https://scholarsarchive.byu.edu/streaming/7/.

Figure 3.11: Glide slope trajectory test flight with RTK GPS ground truth plotted with our position estimates.

The average errors for both of our reported tests were decimeter level in all directions, which outperforms the expected 1 to 5 m accuracy of traditional GPS sensors, and our pose estimation method provided updates an order of magnitude faster than typical GPS update rates [38]. In some situations where IR visibility in the environment is poor due to weather conditions like dense fog or rain, enhancements to the light detection method may be necessary to reliably use our proposed approach, whereas GPS can be expected to operate normally in a wide range of weather conditions. However, the dependence of GPS on open-sky visibility remains a limitation that can in general be solved with additional improvements in speed and accuracy using our pose estimation method.
3.8 Conclusion

We have presented a pose estimation approach based on a novel IR light constellation design and corresponding data association algorithm that allows for high accuracy, real-time pose estimates in GPS denied environments. Our sparse IR light constellation design is easily scalable for long or short distance viewing, and is designed such that it can be identified and associated robustly in the presence of outlier IR light sources and occluded constellation lights, making it suitable for use in cluttered urban environments. We demonstrated through our hardware test results that the pose estimation approach is able to run real-time on board a UAV in the presence of outliers and occlusions, achieving sub-meter accuracy from ranges of over 100 m from the landing site.

Although convenient for placement on a landing site, planar fiducials have some weaknesses that make them unreliable in certain scenarios. For example, approaching the landing site from a shallow angle makes planar objects appear nearly linear, making distinguishing points or designs on the fiducial very challenging. Additionally, a pose ambiguity problem exists that arises occasionally when solving the PnP problem using coplanar points...
viewed from certain perspectives [39]. In these cases, even if the points of the constellation are associated correctly, the pose produced by the PnP solver may be incorrect. Both of these weaknesses of planar fiducials can be resolved by using cross ratios and other projective invariants to extend a planar constellation to a sparse 3D fiducial constellation, which would allow for more reliable PnP solutions under extreme viewing conditions.

3.9 Acknowledgments

The research described in this chapter was supported by the AFWERX Agility Prime program through an STTR Phase II Award to Archer Aviation and Brigham Young University.
This chapter is composed from a paper entitled “Deep Learning-Based Data Association in Camera Images for GPS-Denied Aircraft Landing” planned to be submitted to the 2025 American Institute of Aeronautics and Astronautics SciTech Forum. I hereby confirm that the use of this article is compliant with all publishing agreements.

4.1 Abstract

This chapter explores a learning-based approach to the problem of associating infrared detections in a 2D camera image to known 3D point sources present in the environment. We pose this data association problem in the context of pose estimation for aircraft landing in GPS-denied urban environments, where infrared light sources are placed on the landing site to provide pose information for approaching aircraft. Although sampling-based methods are able to solve this problem with some robustness to outliers, their computation time often scales poorly with increased image clutter and can become a bottleneck when used in real-time frameworks. We present an approach based on the PointNet++ architecture that solves the data association problem using point cloud analysis and can be trained using simply generated synthetic data. We show that both fully learned and hybrid PointNet++ data association approaches are able to achieve the desired outlier robustness, obtaining high accuracy data associations at real time speeds in the presence of high levels of image clutter. We demonstrate the effectiveness of these learning-based data association methods on real world flight-test data, and show that vehicle pose estimates obtained using the output associations from our learned approaches outperform traditional GPS in both accuracy and update rate.

4.2 Introduction

Cameras are inexpensive and effective sensors for many autonomous tasks including observing the surrounding environment, detecting and classifying objects, and estimating relative positions. In a previous work, we demonstrated the use of a monocular infrared (IR) filtered camera to obtain high accuracy pose estimates of a vertical take-off and landing (VTOL) aircraft...
during its landing approach [26]. Our approach relied on a novel sparse IR-point constellation designed with projective invariant properties that could be used to robustly associate detected IR points in camera images. The data association approach in our prior work depended on random sampling consensus (RANSAC) to identify lines of the constellation among potential clutter resulting from outlier IR sources or reflections in the environment. While the randomized nature of this approach makes it robust to outliers in the camera image, the computation time needed to acquire a high confidence solution scales poorly with the number of outliers. As a result, in cases of high outlier-to-inlier ratios the RANSAC step of the data association algorithm can slow the update rate of the pose estimation pipeline, leading to inconsistent and possibly slower than real time pose updates.

To address this issue, we explore learning-based solutions by reformulating the problem of IR camera image data association as the task of 2D point cloud analysis. Recent advances in deep learning approaches to point cloud classification and segmentation have enabled unique methods for analyzing groups of points based on their geometric relationships. These methods are particularly suited for working with IR-filtered camera images that lack the feature-rich details of RGB camera images, and can operate at near constant speeds independent of clutter levels in the camera images. In particular, the PointNet++ model [18] has become state-of-the-art for performing classification and segmentation tasks on 3D point clouds, using a novel multilayer set abstraction approach to capture and learn both local and global properties of the data.

One potential problem that arises when applying 3D point cloud methods to 2D camera point clouds is that many geometric relationships like distances between points and angles between lines are not consistent when subject to projective transformations. This means that geometric features observed by a deep neural network (DNN) from one perspective of a 2D projected point cloud may not appear the same when viewed from a different perspective. In our previous work, we solved this problem by relying on a set of projective invariant properties including collinearity, concurrency, and cross ratios to obtain useful information about the arrangement of detected IR points in a camera image. It may be natural to wonder then if a DNN can learn to recognize similarly useful projective invariant features of a 2D point cloud to robustly distinguish constellation points from outlier points and associate the supposed inlier points with known 3D world locations. We show experimentally using our previously introduced projective invariant constellation design that a deep learning network is capable of learning to associate 3D point clouds viewed under projective transformations.

In this chapter we explore various data association approaches based on a slightly modified 2D PointNet++ architecture that demonstrate the model’s ability to both detect outliers and perform data association on 2D point clouds derived from IR camera images. First, we apply PointNet++ directly to the aforementioned task of point-wise data association, training the network to assign labels to constellation points while identifying outlier points in the camera image. Based on improvements to the network performance observed when normalizing the image points without outliers present, we
next design and report on a nested series of PointNet++ networks, with a smaller, lightweight OutlierNet tasked with identifying the subset of constellation points from among outliers, and a full PointNet++ network performing per-point data association on the cleaned and normalized point cloud. Results from these approaches demonstrate a high level of accuracy, comparable to the method in our previous work, while maintaining consistent speeds over a range of outlier levels.

The third approach we investigate in this work is motivated by a demand for robustness to variations to the IR constellation layout. A limitation of using neural networks to perform point-wise data association is that once trained to recognize points in a specific arrangement, the network can reliably perform inference only on images of that single constellation. Rearrangement, addition, or removal of points in the constellation results in test data that is not representative of the training data. To address this issue, we introduce a hybrid learning approach to the data association problem that is applicable to the entire class of constellation designs introduced in our previous work. Similar to the previously described nested network approach, our hybrid method involves using OutlierNet in series with a PointNet++ model trained to recognize broad geometric features of the constellation. In our approach we train this model, which we denote as LineNet, to identify the individual lines within the point cloud that compose the target constellation. We then follow the steps of the approach from our previous work to label constellation lines using angular cross ratio values, then associate the individual points on each line using linear cross ratio values. This partially learned data association method is able to be broadly applied to arbitrary configurations of our projective invariant constellation and is shown to also operate at constant speeds on significantly cluttered point clouds.

In this chapter we present and discuss the configurations, training methods, and results for the three described PointNet++ based data association approaches. We report on both the accuracy and computational efficiency of the approaches tested on both synthetic and real-world flight-test data. Since the ultimate objective of our data association methods is to recover camera pose information for landing aircraft, we additionally investigate the quality of the estimated camera poses obtained using the output associations from each network. We show that both the data association and resulting pose estimation accuracy improve on the RANSAC based approach of our previous work, and the computational speed allows for real-time updates when used in an end-to-end pose estimation pipeline.

4.3 Related Work

A large majority of deep learning approaches for point cloud analysis are designed for working with 3D point clouds, taking in $xyz$ coordinates of the points in the cloud and outputting either an object class for the cloud or per-point labels. Few approaches are designed for working with 2D point clouds, which can arise from planar sensors such as cameras, radars, and 2D lidar. Among these sensors, cameras are unique in that information from
the surrounding environment is subjected to a projective transformation, altering many useful geometric properties like angles and distances between points that are invariant in Euclidean spaces. In this section we review recent work in the area of 3D point cloud analysis, existing approaches for analyzing 2D point clouds, and established deep learning approaches to analyzing camera images.

As a pioneer in deep learning methods for point cloud processing, PointNet [17] and its successor PointNet++ [18] have been applied to many types of 3D classification and segmentation problems. The original PointNet model operated on point cloud inputs by regressing the points to a single vector of global features using a series of multilayer perceptrons (MLP). When performing part segmentation, the global features are concatenated to each individual point, and the resulting point-feature matrix is passed through another MLP layer to obtain per-point labels. A key concept in the PointNet model is the use of a symmetric maxpool function when constructing the global feature vector, as this solves the problem of dealing with unordered inputs that arises when analyzing sets of points, which are typically permutation invariant. The more recent PointNet++ model uses sampling with multi-scale grouping to hierarchically learn both local and global geometric features of the point cloud. As a note, PointNet++ was technically applied to a 2D classification problem, performing digit identification on planar point clouds contrived from the standard MNIST dataset of handwritten numbers. However, these images and corresponding 2D point clouds are acquired from a top-down view and carefully scaled to a uniform size, essentially eliminating the challenge of dealing with geometric variations that arise from perspective changes [40].

Since their inception, PointNet and PointNet++ have inspired numerous other similarly structured point-based architectures aimed at increasing computational efficiency or improving specific aspects of the network, such as sampling methods and feature construction [41]. Specific to the problem of learning-based data association, Chu et al. [42] used PointNet for multiple target data association, associating noisy multitarget radar detections observed from multiple agents to track targets' movement over time. Chen et al. [43] predicted 6D poses of objects in RGB-D images using PointNet as the backbone of their PointPoseNet, which used a 2D object detection convolutional neural network (CNN) to predict pixel depth information around regions of interest to synthesize a 3D point cloud to which PointNet could be applied. A recent point cloud analysis model, PointMLP [21], was designed to hierarchically group 3D points similar to PointNet++, but employed a residual feed-forward connection to the MLP blocks to allow for greater network depth.

Deep learning analysis of 2D camera images has traditionally been performed using CNNs, with networks like the popular YOLO model performing object detection, classification, and segmentation [14]. End-to-end pose estimation networks, such as PoseNet [44] proposed by Kendall et al., train CNNs to predict 6 degree of freedom camera poses from RGB or RGB-D camera images. Valada et al. [45] approached the image-based pose-regression problem by simultaneously solving the auxiliary problem
Deep Learning-Based Data Association in Camera Images

of estimating the vehicle’s visual odometry. These end-to-end approaches however typically achieve worse accuracy than structure from motion (SFM) approaches that perform Perspective-n-Point (PnP) on 2D-to-3D point correspondences to find the camera pose that minimizes the reprojection error of the points [46], [47]. One notable recent end-to-end model that has achieved improved performance is Differential RANSAC (DSAC) [48], a hybrid CNN approach that regresses 2D points from camera images to a 3D coordinate scene representation. The regressed 3D coordinates are then used to calculate a pose using a modified RANSAC scheme that replaces non-differentiable functions in the traditional algorithm with probabilistic model parameters that can be optimized during training.

Besides a general decrease in accuracy seen with end-to-end CNNs compared to SFM approaches, CNNs have some limitations that make them less ideal for the task of pose estimation using IR constellations. For example, CNN training data and inputs are generally required to be the same fixed resolution, which potentially limits the model’s application to different vehicle sensor suites. Additionally, like other neural networks, CNNs do not generalize well to images taken in environments outside their training dataset, so the IR constellation imaged in a novel setting would likely result in a less accurate solution. Finally, while PnP solutions are inherently equipped with a reprojection error metric to quantify the accuracy of an estimated pose, CNNs in general do not provide such a metric, which complicates the task of evaluating the quality of the output pose. For these reasons we formulate the problem of pose estimation from a 2D point cloud perspective rather than an image-to-pose approach.

4.4 GPS-Denied UAV Landing Overview

The overarching purpose of our learning-based data association approach is to enable real time IR light-based pose estimation as a replacement for GPS measurements in GPS-denied landing scenarios for UAVs. These situations can arise when operating a UAV in urban environments where buildings interfere with GPS satellite signals, or in hostile environments where GPS signals can be jammed or spoofed by antagonists. Our pose estimation approach, which is described in detail in [26], involves placing IR lights on the landing site in a specific constellation design at precisely measured locations. The aircraft is fitted with an IR-filtered camera, which is used to acquire images of the constellation lights as the aircraft approaches the landing site. If the IR light sources that appear in the camera image can be associated with their 3D real-world locations, we can use these 2D-to-3D point associations to solve the Perspective-n-Point (PnP) problem to obtain a precise measurement of the pose of the camera relative to the landing site. A visualization of this point IR-based pose estimation approach is shown in Figure 4.1.

The problem of associating the IR lights observed in a 2D camera image points with their 3D world locations is made challenging by the inconsistency of fundamental geometric properties like distances, angles, and areas when subject to a projective transform. In our previous work, we addressed
this issue by designing a class of IR constellation layouts, pictured in Figure 4.2, that contains specific projective invariant properties including collinear points, concurrent lines, and specific cross-ratios, that allow it to be identified and associated when viewed from arbitrary camera perspectives. The constellation class can be configured with custom lengths $L_1$ through $L_5$ from the constellation vertex to the points along each line, and custom angles $\angle AB$ through $\angle DE$ that make up a 180° semicircle.

Motivated by the need for robustness to outlier IR sources that may arise in camera images taken in crowded urban environments, we previously relied on a RANSAC approach to identify the set of five concurrent lines that make up the target constellation from among potential camera clutter. Once these lines were identified, we could use a linear cross ratio-based voting scheme to associate the individual points on each line. These point associations were then used to solve the PnP problem, providing pose
While RANSAC is shown to be effective at associating points in IR images with a low to moderate number of outlier detections, the number of iterations required to find a high confidence consensus set of points is a combinatorial problem and can grow factorially with the number of outlier points. In images with high levels of outliers then, the RANSAC line detection step would either slow the processing speed of the entire pose estimation pipeline, or else produce a low-confidence consensus set of points that would not likely be useful for obtaining the correct point associations. Due to the importance of receiving consistent, high confidence pose measurements during the UAV landing phase, we explore alternative learning-based methods to the IR image data association problem that can operate at constant computational speeds across varying outlier levels. As described earlier, rather than operating on the entire IR image, we first extract the pixel locations of the detected IR lights in the frame, then take these points as a 2D point cloud that can be analyzed using an appropriate deep learning model.

4.5 Models and Datasets

The theoretical backbone of our 2D point cloud data association approach is the PointNet++ model, proposed by Qi et al. for analyzing 3D point clouds [18]. The model architecture consists of blocks of set abstraction layers that iteratively cluster neighbors around a number of sampled points to obtain high-level descriptors for the point cloud, as shown in Figure 4.3. After the point cloud has been reduced by a given number of set abstraction layers, the regressed features for the reduced point set are propagated back through the network using feature propagation layers that interpolate features to nearby points. By the end of the feature propagation layers, each point of the original point cloud has a descriptor made of features interpolated from different set abstraction levels, providing both local and global information that is used by an MLP head to assign labels to each point.

In this work we make use of both full-sized PointNet++ models consisting of four set abstraction and feature propagation layers, as well as reduced-size models consisting of only three set abstraction and feature propagation layers. The reduced-size PointNet++ models are given two different roles to perform simplified tasks on the input point cloud. OutlierNet is tasked with assigning each point in the cloud a binary classification as either a constellation inlier point or an outlier, while LineNet labels points according to the line of the constellation to which they belong. These lightweight models are used in our approaches to either preprocess the point cloud data for PointNet++ to then operate on or to perform a reduced part of the data association problem in coordination with some other companion method. Because of the reduced size of these networks they are able to operate on inputs at higher computational speeds than the full PointNet++ model, helping the data association approaches to achieve high accuracy performance while maintaining real time speeds.
4.5.1 Training and Test Data

Training data used for our data association approaches consists of synthetic data produced by projecting constellation points, visualized in Figure 4.2, from 3D world positions to 2D pixel locations using a calibrated pinhole camera model. Poses for these projections were generated by first randomly sampling a horizontal and vertical position within 300 m and 150 m respectively of the origin of the constellation in the inertial world frame. These ranges represent typical bounds for aircraft landing trajectories as well as nominal sensing distances for the IR light sources. Since the constellation points must be at least partially visible in the camera frame for the PointNet++ model to perform data association, we set the orientation of the simulated camera to the unit vector pointing from the camera’s position to the constellation center, then add normally distributed noise to both the camera elevation and azimuth. We then add normally distributed noise to the projected pixel locations to simulate measurement noise in the detected light locations.

To simulate outlier detections in the IR camera image, we add a random number of outlier points to the synthetic point cloud, with an upper bound set as some ratio of the number of points in the constellation. These outlier detections are distributed throughout the point cloud by randomly sampling x and y pixel coordinates from a uniform distribution. We also simulate occlusion of the constellation lights by removing projected constellation points from the generated point cloud at a 15% drop rate. Given the computational simplicity of generating these 2D point clouds we are able to create relatively large datasets, with one million point cloud instances for our training dataset and ten thousand instances for our test dataset.

Using the steps listed above we generate four synthetic datasets for
training and testing the accuracy and robustness of our data association approaches. The details of the construction parameters of the datasets are listed in Table 4.1. Our Clean dataset contains no outliers and is used for training models that will operate under the assumption that a preprocessing step has been used to remove any outliers from the point cloud. The Standard Cluttered, or Cluttered, dataset is generated with a maximum outlier-to-inlier ratio of 0.2, which is similar to ratios observed in real-world images obtained in previous flight tests. The Outlier50 and Outlier150 datasets have maximum inlier-to-outlier ratios of 0.5 and 1.5 respectively, and are meant to test the robustness of our data association approaches in terms of accuracy and computational speed to what we consider extreme cases of image clutter.

Table 4.1: Composition of datasets used to train PointNet++ models

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Max Outlier Ratio</th>
<th>Occlusion Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>0</td>
<td>.15</td>
</tr>
<tr>
<td>Cluttered</td>
<td>0.2</td>
<td>.15</td>
</tr>
<tr>
<td>Outlier50</td>
<td>0.5</td>
<td>.15</td>
</tr>
<tr>
<td>Outlier150</td>
<td>1.5</td>
<td>.15</td>
</tr>
</tbody>
</table>

In addition to evaluating our trained models on synthetic data, we also test their performance on real-world 2D point clouds derived from flight-test images obtained onboard a UAV. These images were captured by a VTOL aircraft equipped with an IR bandpass filtered camera during a flight trajectory around the constellation of points laid out on the ground. The specific trajectory, referred to afterwards as the castle trajectory, is an approximately 100 m radius circle around the constellation at a height of between 20 and 30 m, with periodic 10 m changes in altitude. The objective of this data set is to test the data association methods with camera views of the constellation points from a variety of viewing perspectives, as well as to verify the performance on real-world camera images of models trained on simulated data.

4.6 Normalization and Filtering Techniques

The normalization of inputs to a deep learning model is a non-trivial task that can have significant implications on the performance of the network. In this section we discuss some alternative methods for normalizing inputs to the PointNet++ model and propose situations for their potential use. We also discuss in this section methods for filtering the outputs from the PointNet++ model to avoid infeasible solutions and prevent overconfidence in incorrect outputs.

4.6.1 Normalization Schemes

One of the important preprocessing steps for training deep neural networks is normalizing the inputs to the network, which scales the different input features to a common range and has been shown to have a positive effect on both training speed and performance [49]. When training PointNet++ models
we apply two straightforward normalization strategies to scale coordinates of the points in the cloud to values between 0 and 1, with both approaches visualized in Figure 4.4.

![Diagram](image)

Figure 4.4: Normalization strategies used for scaling the 2D point cloud inputs to the PointNet++ model, with the normalization achieved by Equation 4.1 shown in red and Equation 4.2 shown in green.

The first approach is to simply divide all $x$ coordinates by the width of the image and all $y$ coordinates by the height of the image. For example, given a 2D point cloud $B = \{b^1, b^2, \ldots, b^n\}$ composed of $n$ points defined as

$$b^i = \begin{bmatrix} b^i_x \\ b^i_y \end{bmatrix}$$

derived from a camera image of size $(width, height)$, we normalize the input points using the equation

$$\bar{b}^i = \frac{b^i_x}{width} \begin{bmatrix} 1 \\ \frac{b^i_y}{height} \end{bmatrix}^T \quad (4.1)$$

This normalization essentially reduces the camera image boundary to a unit length square, but does not necessarily incorporate any relevant information about the arrangement of points in the image. A generally more useful normalization is obtained by scaling the point coordinates by the minimum and maximum values of the cloud. Letting $b^\text{max}_x$, $b^\text{min}_x$, $b^\text{max}_y$, and $b^\text{min}_y$ be the respective maximum and minimum $x$ and $y$ coordinate values of the point cloud, we normalize according to

$$\bar{b}^i = \begin{bmatrix} \frac{b^i_x - b^\text{min}_x}{b^\text{max}_x - b^\text{min}_x} \\ \frac{b^i_y - b^\text{min}_y}{b^\text{max}_y - b^\text{min}_y} \end{bmatrix}^T \quad (4.2)$$
This normalization method narrows the values of the point cloud to a unit sized region of interest around the points, essentially scaling the points to a bounding box around just the constellation. This is intuitively a useful preprocessing step, as the layout of the constellation points are normalized to a standard size regardless of how large the constellation appears in the camera image. A problem arises, however, when outliers are present in the camera image point cloud, as shown in Figure 4.5. When the outliers affect the maximum and minimum pixel coordinates of the point cloud, the normalization scheme in Equation 4.2 bounds the points to a region of interest that is larger than the actual constellation. This can have a detrimental effect on the model during training, as it may learn to rely on certain spatial relations between normalized points that are subject to change based on the presence of outliers.

Figure 4.5: The normalization strategy described in Equation 4.2 is useful for narrowing the scale of the constellation points when the point cloud is free of outliers (a)(b) but is less effective when outlier points are present that affect the minimum and maximum pixel coordinates of the point cloud (c).

We will make use of both the full-frame normalization method from Equation 4.1 and the region of interest (ROI) normalization method from Equation 4.2 in the data association approaches that follow. It will be shown that preprocessing the point cloud with a reduced-size PointNet++ model using full-frame normalization followed by PointNet++ evaluation of the resulting point cloud using ROI normalization produces a robust, high accuracy data association solution in the presence of significant image clutter.

4.6.2 PointNet++ Output Filtering

Our ultimate objective for exploring deep learning approaches to data association for IR camera images is to obtain the 2D to 3D point associations needed to recover camera pose information by solving the PnP problem. We
assume that the IR light sources of the fiducial constellation are precisely
surveyed and the camera is calibrated beforehand, so given the output point
labels from the PointNet++ model we can immediately use the known 3D
coordinate locations of the lights to compute the position and orientation of
the camera.

The PnP problem is sensitive to incorrect 2D to 3D associations, as a
single incorrect association can severely degrade the accuracy of the solution.
We addressed this sensitivity in our previous work using the projective
invariant cross ratio property as a metric for a voting scheme to assign point
associations. This allowed us to apply a confidence level threshold to the
labeling process, only assigning an association to a point if it received a given
percentage of votes. When using a deep learning model to assign these
associations, however, the confidence that can be placed in the labels output
by the model is less straightforward. Deep learning models are typically
unaware of when they are performing inference on novel input data that is
different from the data used to train model, so they often continue to output
incorrect labels with high confidence scores [50]. To verify the quality of
the data association solution output from PointNet++, we first check the
confidence and feasibility of the output labels, then filter a best solution
subset of labels using the reprojection error obtained from solving PnP with
reduced subsets of the associated points.

![Figure 4.6](image)

**Figure 4.6:** Given an input point cloud, PointNet++ outputs a vector (columns) of
scores for each input point, with the highest score for each column (yellow) being
the label assigned to the point.

The segmentation configuration of PointNet++ outputs a vector of scores
for each input point designating the likelihood that the specific point fits
into each of the possible classes, visualized in Figure 4.6. Our specific
segmentation problem is somewhat unique from a typical point cloud
segmentation task in that except for the outlier class, each label can appear at
most once in the cloud. The PointNet++ model does not inherently enforce
Deep Learning-Based Data Association in Camera Images

this constraint upon its outputs, so we manually check that each input point’s assigned label indeed has the best score for that class label. If multiple input points are assigned the same label, the point with the highest score receives the label and the other points receive their next highest scoring labels. Additionally, to ensure that labels are only assigned to points when the model is reasonably confident, we set a minimum score threshold above which a point’s label score must reach for it to receive that label. These manually imposed confidence and feasibility constraints on the PointNet++ outputs are demonstrated in Figure 4.7.

<table>
<thead>
<tr>
<th>Class Scores</th>
<th>Input Points</th>
</tr>
</thead>
</table>
|              | 1     | 2     | 3     | 4     | ...
| Outlier      | -9.4  | -10.2 | -9.7  | -10.6 | ...
| Point 1      | -0.3  | -0.2  | -2.3  | -4.5  | ...
| Point 2      | -0.4  | -0.7  | -2.1  | -4.4  | ...
| Point 3      | -3.5  | -1.5  | -1.7  | -3.1  | ...
| Point 4      | -3.7  | -4.1  | -1.8  | -2.8  | ...
| Point 5      | -3.4  | -4.1  | -2.2  | -0.7  | ...
| ...          | ...   | ...   | ...   | ...   | ...

Figure 4.7: Example of cases when an input point’s maximum score does not match the maximum score of a class (orange), and when a point’s maximum score matches the class maximum score but is below some acceptable threshold, in this case -1.5 (red).

To address the problem of potentially overconfident incorrect associations, we filter the PointNet++ associations based on the reprojection error obtained when using the output associations to solve the PnP problem. An outline of this filtering approach is listed in Algorithm 4.1, with the set of points \( \hat{B} = \{ \hat{b}^1, \hat{b}^2, \ldots, \hat{b}^m \} \) denoting the \( m \leq n \) labeled inlier points from PointNet++ that remain after resolving any infeasible or unconfident point associations. The process begins by calculating a baseline reprojection error by solving PnP using the points in \( \hat{B} \). We iteratively remove one point at a time from \( \hat{B} \), calculate the resulting PnP reprojection error, then determine the percent increase or decrease in error compared to the baseline value. If the percent error decreases with a given point removed by more than some error improvement threshold \( \gamma \), then we denote that point a bad association. After iterating through all of the points, we remove all bad associations and calculate a new baseline reprojection error. If that error is less than some acceptable reprojection error threshold \( \epsilon \), we take the remaining point associations as our final set. Otherwise we iterate again through the points until either there are no more bad associations, the PnP reprojection error is
less than $\epsilon$, or the remaining number of points is less than six, which is the minimum number of points needed to solve PnP using the iterative method of OpenCV’s solvePnP function.

Algorithm 4.1 Point Association Filtering

- reprojection error threshold $\epsilon$
- error improvement threshold $\gamma$
- $B^* \leftarrow \hat{B}$
- $k \leftarrow |B^*|
- e \leftarrow$ PnP reprojection error using $B^*$

while $e > \epsilon$
do
- $bad points \leftarrow \{\emptyset\}$
- for $i = 1$ to $k$ do
  - $\hat{B} \leftarrow B^* \setminus b^*$
  - $e_i \leftarrow$ PnP reprojection error using $\hat{B}$
  - if $\frac{e_i}{e} < \gamma$ then
    - append $i$ to $bad points$
  - end if
- end for
- if $|bad points| = 0$ then
  - return $B^*$
- end if
- $B^* \leftarrow B^* \setminus b^*$ for $j \in bad points$
- $k \leftarrow |B^*|
- if k < 6 then
  - return $\{\emptyset\}$
- end if
- $e \leftarrow$ PnP reprojection error using $B^*$
end while
- return $B^*$

Using this filtering method we are able to identify a subset of the labeled outputs from PointNet++ that are consistent with a single view of the constellation. It should be noted that using this approach we can end up with an empty solution set if the algorithm keeps removing points without ever reaching the desired PnP reprojection error threshold $\epsilon$. However, we prefer this case of no acceptable solution to the alternative of an overconfident but incorrect solution. Using the IR fiducial design constellation shown in Figure 4.2 also provides a significant level of redundancy to loss or misassociation of points, with over four times the minimum number of points needed to solve PnP included in its design. Another potential issue with this method is the computation time, as in the worst case scenario the filtering can require $O(m^2)$ calculations of PnP. However, the design of the algorithm lends itself naturally and simply to multithreading, which allows for reduction of the computation time by several factors. Also, in practice we find that it rarely takes more than a few iterations through the point cloud to either find an acceptable solution or remove all of the associations from the solution set. Computation time for this filtering step are discussed along
with the other steps of the PointNet++ approach in the following section.

4.7 Data Association Approaches and Results

In this section we discuss the various model configurations and approaches used to solve the problem of data association in 2D point clouds derived from IR camera images. Motivation and explanation of these different design choices are presented along with results demonstrated on both synthetic and real-world data.

4.7.1 PointNet++ Data Association

Our baseline data association approach is to directly apply a 2D PointNet++ model in the part segmentation configuration to 2D IR point clouds. Given the class of IR constellation designs shown in Figure 4.2, we assign one segmentation class to each specific point and reserve one additional class for labeling outliers, resulting in 26 part labels. In Section 4.6 we discussed two possible methods for normalizing the input point cloud data, and so we begin our analysis by training the PointNet++ model on the Cluttered and Clean datasets using both the full-frame and ROI normalization schemes.

Results from training PointNet++ using the two normalization methods are summarized in Table 4.2. The accuracy reported is defined as the number of correct associations divided by the total number of attempted associations, which may be less than the total number of points in the cloud due to the filtering out of unconfident associations. In these preliminary tests we do not filter poses using the iterative PnP approach. PointNet++ trained using the full-frame normalization scheme is shown to work somewhat well, achieving a 85.44% accuracy on the test datasets described in Section 4.5. A similar performance was observed when training the model with the ROI normalization scheme. However, significant improvement was seen when training and testing the PointNet++ model on a clean dataset with no outliers, with the ROI normalization scheme resulting in an impressive 96.6% accuracy.

Table 4.2: Data association accuracy of PointNet++ using different normalization strategies.

<table>
<thead>
<tr>
<th></th>
<th>Cluttered</th>
<th>Clean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Frame</td>
<td>85.44</td>
<td>87.25</td>
</tr>
<tr>
<td>ROI</td>
<td>84.67</td>
<td>96.56</td>
</tr>
</tbody>
</table>

This difference in performances suggests that if the input point cloud can be cleaned of outliers prior to being passed through the PointNet++ network, the model can achieve high data association accuracy using the ROI normalization scheme. To this end, we next modeled and trained OutlierNet, a lightweight, reduced-scale version of PointNet++ that can quickly perform part segmentation, labeling each point in the cloud as either a constellation inlier or an outlier. The points that are identified as inliers are then normalized using Equation 4.2 and passed to a PointNet++ model that has been trained on the Clean dataset using the same normalization strategy.
4.7.2 OutlierNet/PointNet++ Data Association

Given our three datasets with differing levels of outlier clutter, we train an OutlierNet model on each dataset and compare performance of the resulting models on each dataset. Results for these models are shown in Table 4.3, where the standard OutlierNet model was trained on the Cluttered dataset, OutlierNet50 was trained on the Outlier50 dataset, and OutlierNet150 was trained on the Outlier150 dataset. All models are trained using the full-frame normalization approach, which was shown to have slightly better performance in cluttered environments than ROI normalization. As could be expected, each model achieved the best test performance on the clutter level it was trained on, although the performance of the OutlierNet and OutlierNet50 models on the Cluttered and Outlier50 datasets are seen to be nearly identical. It is interesting to note that although the OutlierNet model was trained on data containing a maximum inlier-to-outlier ratio of 0.2, it was still able to perform exceptionally well on the Outlier50 dataset that had more than double the amount of outliers. However, there appears to be a limit to how much the OutlierNet and OutlierNet50 models can generalize beyond the outlier ratios they were trained on, as OutlierNet150 proved to be the only model capable of retaining a high classification accuracy on the severely cluttered OutlierNet150 dataset.

<table>
<thead>
<tr>
<th></th>
<th>Cluttered</th>
<th>Outlier50</th>
<th>Outlier150</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutlierNet</td>
<td>99.63</td>
<td>99.06</td>
<td>80.75</td>
</tr>
<tr>
<td>OutlierNet50</td>
<td>99.61</td>
<td>99.44</td>
<td>89.39</td>
</tr>
<tr>
<td>OutlierNet150</td>
<td>98.20</td>
<td>98.34</td>
<td><strong>96.63</strong></td>
</tr>
</tbody>
</table>

Although OutlierNet150 was outperformed at lower outlier ratios by the models trained specifically on those datasets, the model is still able to achieve high classification accuracy over the full range of clutter levels. For its high accuracy and robustness to varying outlier levels, we choose to use OutlierNet150 as our default OutlierNet model for experiments moving forward. It should be noted that just as the OutlierNet and OutlierNet50 models performed well until reaching an outlier level that they could not generalize well on, if we were to keep increasing the outlier ratio past 1.5 there would likely be a drop off in performance by the OutlierNet150 model as well. However, we view this as more a general limitation of deep learning methods rather than a specific weakness of this approach, and so we move forward under the assumption that the data association models we propose will be operating on suitably similar distributions of 2D point clouds.

One of the motivating reasons for exploring a deep learning approach to IR camera image data association was to achieve consistent computational speed over a range of outlier-to-inlier ratios. This is a necessary requirement for a pose estimation pipeline to be able to run at consistent real-time speeds when operating in non-ideal environments on cluttered IR images. We therefore test the computation time of the three OutlierNet models on data with an increasing number of outliers to observe the effect of increasing the
size of the input point cloud. The results, shown in Figure 4.8, demonstrate that the OutlierNet models are indeed able to maintain nearly constant computation time despite significant increases in outlier levels. As a note, computation for this test was performed using Python implementations of OutlierNet, which resulted in overall slower computation times than would be observed using a C++ implementation of the same models, as is shown later in Table 4.5. However, the importance of this specific result is the consistent time of computation over varying outlier levels rather than the magnitude of the OutlierNet computation time.

Figure 4.8: Computation times for OutlierNet models on data with varying outlier-to-inlier ratios.

With a suitable OutlierNet model selected and trained, we test its performance as a preprocessing step for the PointNet++ model. We continue to use the full-frame normalization method with the OutlierNet model on the cluttered point cloud inputs, and as suggested by the PointNet++ normalization results from Section 4.6 we use the ROI normalization strategy for the PointNet++ model on the cleaned point cloud. The results of the combined OutlierNet/PointNet++ network compared to a lone PointNet++ model on the different cluttered datasets are summarized in Table 4.4. In these results we report classification accuracy of the models as before, but also include a percent labeled metric equal to the number of associations assigned by the network divided by the total number of points in the cloud. This metric could be considered an efficiency or confidence measure, with higher percentages indicating that the model was confident enough to assign labels for more of the points in the cloud. As would be expected, the combination of OutlierNet with PointNet++ far outperforms the solitary PointNet++ in both accuracy and labeling efficiency at every outlier level.

Although demonstrating promising performance on simulated data, the objective of our OutlierNet/PointNet++ network is to obtain high accuracy
associations from real-world IR camera image data. We thus turn our attention to the castle trajectory flight-test data described in Section 4.5. As a baseline comparison for our learning-based data association method, we turn to the cross ratio-based data association approach (XRDA) developed and demonstrated in our prior work [26]. Since we are now concerned not only with accuracy of associations but also the quality of poses obtained using those associations, we use the PnP filtering method outlined in Section 4.6 with the solution set of point associations output by PointNet++.

The estimated poses along the castle trajectory obtained by both the OutlierNet/PointNet++ and XRDA methods are shown in Figure 4.9. The pose error levels shown in Figure 4.10 are nearly identical for the two approaches, however over the course of the flight trajectory, which consists of 6,000 images of the constellation, the XRDA approach associated enough points to obtain a PnP pose estimate in 4,566 frames, or 76.1% of the flight, while the OutlierNet/PointNet++ network was able to associate sufficient points for a PnP pose estimate in 5,438 frames, or 90.6% of the flight. This significant increase in pose estimation output can be attributed to parts of the flight where enough of the lights were missed by the light detection algorithm that the cross ratio-based voting scheme of XRDA was unable to obtain enough information to confidently associate the points. Since OutlierNet/PointNet++ does not rely on any hand-tuned parameters or thresholds it was able to continue associating points with limited information from the lights, labeling enough points to be able to perform PnP despite severe occlusions. In Figure 4.9 these areas of improved robustness can be seen as the yellow OutlierNet/PoseNet++ segments of the trajectory that continue to output poses while the green XRDA approach is unable to estimate pose information.

In addition to obtaining high accuracy point associations, the OutlierNet/PointNet++ network implemented in C++ is able to operate at real-time speeds. Computation times for each part of the data association process are listed in Table 4.5, with times included for the PnP filtering step with and without multithreading. When using multithreading, the data association process takes on average only 12.7 ms, which is faster than the 15.1 ms needed to compute point associations using our prior cross ratio-based approach. Without using multithreading for the PnP filtering step the total computation time increases to 17.8 ms, which is slower than our XRDA method but still fast enough to produce real-time or near-real-time pose updates. An end-to-end pose estimation pipeline designed similar to [26] that uses the OutlierNet/PointNet++ data association approach can produce pose updates at a rate on the order of tens of hertz, which is an order of magnitude faster than traditional GPS.
4.7.3 OutlierNet/LineNet Data Association

One drawback of using PointNet++ to learn and assign pointwise associations is that like other deep learning approaches, it does not generalize well to constellations that are different from the one used to train it. For example, if we trained a PointNet++ model to associate points of a specific constellation layout designed following the example in Figure 4.2 but later wanted to adjust the spacing between points or angles between lines of the constellation, we would need to retrain the PointNet++ model on a new dataset synthesized for the specific desired constellation layout. A more desirable feature of a deep learned association approach would be flexibility to function on the entire class of fiducials rather than a single specific design.

Flexibility of the deep learning approach could be improved if instead of outputting the specific point associations for each point in the input cloud, we trained the network to simply identify the sets of collinear points within the cloud. Once the points of the constellation are separated into lines, we

---

**Figure 4.9:** Pose estimates produced using OutlierNet/PointNet++ data associations vs. the XRDA method on flight-test data.

**Table 4.5:** Average computation times for PointNet++ data association in C++

<table>
<thead>
<tr>
<th>Operation</th>
<th>Computation Time (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutlierNet</td>
<td>3990.3</td>
</tr>
<tr>
<td>PointNet++</td>
<td>5262.9</td>
</tr>
<tr>
<td>PnP Filtering - 8 threads</td>
<td>3470.3</td>
</tr>
<tr>
<td>(PnP Filtering - 1 thread)</td>
<td>(8561.9)</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>12723.5</strong></td>
</tr>
</tbody>
</table>
can associate the lines using knowledge of the angular cross ratios embedded in the constellation, then associate individual points on each line using knowledge of the linear cross ratios within the constellation. In our previous work the line detection step was performed using a RANSAC approach, which while theoretically capable of detecting target features among clutter, can be computationally expensive when operating on images with a high number of outliers.

To achieve consistent real-time computational speed in the presence of high outlier-to-inlier ratios while maintaining flexibility to handle arbitrary configurations of the constellation, we propose replacing the RANSAC line-fitting step of our previous cross ratio-based method with an OutlierNet/LineNet network trained to clean and segment a 2D point cloud into the five lines of the constellation. We train the LineNet model, which like OutlierNet is a reduced-size PointNet++ model, on the Clean dataset, with the points labeled by the line on which they lie rather than their individual associations. To simplify the line association step of the cross ratio method we employ a loss function in the training of LineNet that allows reversed ordering of the line labels as the only acceptable permutation of the truth labels.

As the outputs of the LineNet model are per-point labels designating the
specific line of the constellation to which the points belong, to determine the individual point associations we must take the extra step of performing the cross ratio-based voting scheme of the XRDA approach. This extra step however is computationally inexpensive, and in practice contributes on average less than a millisecond to the total computation time of our hybrid XRDA/learning-based data association.

Figure 4.11: Pose estimates produced using LineNet data associations vs. the XRDA method on flight-test data.

With a trained OutlierNet/LineNet network, we again compare pose estimates obtained on real-world images using this hybrid data association approach to our prior RANSAC cross ratio-based approach. The estimated trajectory of the two approaches are shown in Figure 4.11, with the error levels for each approach plotted in Figure 4.12. Similar to the OutlierNet/PointNet++ approach, the error magnitudes of the estimated poses obtained using OutlierNet/LineNet are slightly higher than those of the XRDA approach, however the OutlierNet/LineNet network produced more pose estimates over the course of the trajectory than the XRDA approach. Results for both the OutlierNet/LineNet and OutlierNet/PointNet++ approaches compared to the XRDA method are summarized in Table 4.6. We can see that in all three cases, the poses obtained using the different data association methods are accurate to a decimeter level, which is a significant improvement over the meter-level accuracy obtained using traditional GPS [38].

The volume of pose outputs produced by the OutlierNet/LineNet approach was less than the OutlierNet/PointNet++ approach, however as discussed previously, by using the LineNet approach we gain the flexibility
To perform data association on a variety of constellation designs instead of a single specific layout. The slightly higher pose error levels observed in both deep learning approaches compared to the XRDA method can be attributed to the limited accuracy of the models, which results in points being removed from the solution set by the filtering approach described in Section 4.6. With fewer point associations available to solve the PnP problem, the precision of the estimated poses can deteriorate, leading to an overall reduction in pose accuracy. However, it is worth noting that although the deep learning approaches do achieve slightly reduced pose accuracy, the difference compared to our cross ratio-based approach is on the order of a few centimeters, which is a small fraction of the overall distance of the vehicle from the constellation.

### 4.8 Conclusion

In this work, we have presented a deep learning approach to data association for 2D point clouds based on the PointNet++ architecture that is able to obtain a consensus set of high confidence 2D to 3D point associations in...
the presence of outliers. We show that for the task of associating points of a sparse fiducial constellation, we can obtain good results on real-world data from training the model on synthetic data that is straightforward to generate in large volumes and trivially simple to label. We propose the use of a lightweight preprocessing OutlierNet model to remove outlier points from the input cloud to improve data association accuracy, and explore options for normalizing 2D camera image point clouds to achieve robust performance. We also show that PointNet++ trained as a line segmentation model can be integrated neatly into a first-principles-based data association approach, achieving robustness to outliers at real-time speeds while retaining important guarantees on the data association outputs.

Although demonstrated to function well in somewhat predictable environments, this data association method is limited by some shortcomings common to most deep learning approaches. One limitation is a weakness to generalizing to novel inputs, as a model trained on a dataset with a given outlier density or distribution will exhibit suboptimal performance on inputs that do not resemble the training data. While we were able to obtain good results on real-world test-flight data by training the networks on data containing uniformly distributed outliers at levels of up to 150% the constellation points, if this model were to be tested in environments with higher outlier-to-inlier ratios or significantly different outlier distributions, it would likely not perform as well as desired. Robustness to this issue, either by expanding the capabilities of the network or detecting when unfamiliar data is being fed into the model, would be a useful area of further work for this application as well as for deep learning in general.

PointNet++ was originally designed to analyze 3D point clouds using assumptions of local and global feature relevance based on geometric relationships in a Euclidean space. Monocular camera images, however, represent a projective space where Euclidean based assumptions may approximately hold true in restrictive cases but are not generally valid. We showed that despite these differences in domain, PointNet++ was able to achieve good performance on 2D point clouds derived from IR camera images. However, application of this approach to scenarios with extreme changes in perspective where Euclidean assumptions are no longer good approximations would likely require adjustments to the architecture to obtain consistent and reliable results. Some possible changes could include an alternative sampling and grouping approach to the radial grouping method, which assumes points close together in the cloud are in fact related, as well as a revised feature propagation method that does not rely on an inverse distance weighted interpolation. These adjustments to the PointNet++ approach could further improve the results obtained in this work as well as open the door to wider applications of point cloud analysis models.

Acknowledgments

The research described in this chapter was supported by the AFWERX Agility Prime program through an STTR Phase II Award to Archer Aviation and Brigham Young University.
5 Conclusion

In this thesis we have presented a monocular IR camera-based pose estimation framework to address the technical shortcoming of relying on GPS for UAV flight in urban environments. As a part of that framework, we introduced a novel fiducial pattern composed of sparse IR point light sources that is easily scalable and contains projective invariant properties in its design that allows it to be identified and associated from arbitrary viewing perspectives. The design of the constellation contains significant redundancy to ensure that pose information can be obtained by an approaching UAV in spite of possible occlusions or failure of the constellation lights. Additionally, use of projective invariant properties in the fiducial design allows the points of the constellation to be robustly distinguished from outlier IR light sources that may be present in the environment.

We demonstrated the ability to distinguish the constellation lights from possible image clutter and identify individual lights with their corresponding 3D world locations using a novel data association method. Association occurs in two steps, with the lines of the constellation first being distinguished from outlier light detections using a RANSAC approach and identified using knowledge of the angular cross ratios embedded in the constellation. Next, the individual points of each line are associated with their corresponding 3D world locations using a voting scheme based on calculations of linear cross ratios for combinations of points along each line. This data association approach is tested on real-world IR camera images obtained onboard a UAV, and is shown to be able to operate at real-time speeds in the presence of a moderate number of outliers.

A difficulty of operating in urban environments is the possibility of large numbers of outlier IR detections cluttering the camera image. While sampling-based approaches like RANSAC are theoretically still able to find the target points from among the outlier points, the number of sampling iterations required to achieve a high confidence solution grows significantly with the outlier-to-inlier ratio. This can become a bottleneck in the pose estimation pipeline which slows the update rate of the estimated pose outputs and prevents the pipeline from running at real time speeds. To manage this possibly expensive computational cost, we proposed the training and insertion into the pose estimation framework of OutlierNet, a lightweight PointNet++ deep learning model that was shown to be capable of detecting and isolating target constellation points from noisy point clouds with high levels of outliers in constant time. We additionally demonstrated the ability
of PointNet++ in its part segmentation configuration to perform line and point associations on the cleaned and normalized IR detections, allowing it to assist or replace bottleneck sections of the data association algorithm.

The entire pose estimation pipeline using our cross ratio voting-based data association algorithm, our learning-based association approach using PointNet++, and a hybrid combination of the two methods were demonstrated to provide highly accurate poses at real time speeds on multiple flight test datasets gathered over differing trajectories from up to 200 m away from the landing site. Our results demonstrated decimeter level accuracy at around a 30 Hz update rate, which is an order of magnitude faster and more accurate than traditional GPS sensors [38].

5.1 Future Work

Given the methods and approaches used in this thesis, there exist several areas in which overall performance and utility could be enhanced with further work and exploration. For example, our proposed fiducial constellation design suffers from certain shortcomings common to all planar visual fiducials, including difficulty distinguishing the pattern when viewed from specific perspectives. When the viewing angle approaches parallel to the plane of the constellation and the vertical distance from the camera to the constellation is small relative to the horizontal distance, distinguishing even projective invariant properties like concurrent lines or collinear points can become challenging. In the case of our sparse point constellation, as the camera reaches ground level where the lights are placed, the viewing perspective of the lights causes them to appear compressed into a single line, making the data association problem substantially more difficult. This weakness could be fixed by extending the planar constellation from a 2D fiducial to a 3D arrangement of lights, which would ideally allow points in the constellation to be distinguished and associated when viewed at the same elevation as the constellation. This could likely be accomplished by either building on the set of projective invariant properties used in our planar constellation design, or using a different set of projective invariants that are more naturally suited to 3D spaces.

While shown to be useful and generally effective for detecting constellation points among outliers in IR camera images, deep learning approaches to outlier rejection and data association do not always have the same performance guarantees as a sampling-based approach like RANSAC. For instance, when faced with a camera image containing an unusually large number of outliers, given a sufficient number of sampling iterations RANSAC is still capable of obtaining a set of high confidence inlier points. However if the number of outliers or their distribution in the image does not resemble the data used to train the deep learning network, the performance of the network could unexpectedly deteriorate or the network could exhibit overconfidence in an incorrect solution, causing potentially disastrous errors in the later stages of the pose estimation process. A desirable solution to this problem would then be to have a non-sampling-based method for finding the target constellation points that scales well with increased outlier levels and can
offer more reliable performance guarantees than learning-based approaches. This solution could conceivably build on existing computer vision methods for pattern recognition like Hough transforms or Canny edge detection, or could require a more novel approach that may generalize to other similarly formulated problems.

In this thesis we somewhat naively applied PointNet++, which was designed and shown to work well for analyzing 3D point clouds, to the related but fundamentally different domain of 2D point clouds derived from IR camera images. The approach used to design PointNet++ assumed that the distances between points induced by the metric space of the cloud would define useful local features that could be used to learn details about the point cloud. While this is true for 3D Euclidean spaces in which rotation or translation of the cloud does not produce any change in local structures, this is not the case for 2D points observed in camera images, which may appear to be locally related from one perspective but distant from each other when viewed from a different perspective. This does not necessarily invalidate the PointNet++ approach to camera image data association, as it is experimentally shown to still work quite well in Chapter 4, but should be considered when applying PointNet++ to scenarios that are more removed from the assumptions of 3D Euclidean space. Some adjustments to the model that could be made to generalize it to non-Euclidean spaces might include an adaptive grouping method in the set abstraction layers instead of simply clustering points within a set radius, or an added branch of the network dedicated to estimating pointwise depth information. Additionally, although we were able to achieve good performance of PointNet++ on our data by cleaning the data with a lightweight OutlierNet before normalizing and segmenting the point cloud with the full PointNet++ model, a more concise and efficient method would be to combine these two models into a single architecture that can simultaneously recognize and reject outliers while assigning labels accurately to the inlier points.

Some general adjustments to the test setup of the drone and IR light constellation could also help improve the performance of the overall pose estimation pipeline presented in this thesis. Flashing the lights synchronously such that they all appear on and off in the same camera frames rather than each point randomly lighting up during the three frame window would help reduce noise in the pixel locations of the detected lights due to vehicle motion between camera frames. Additionally, use of a camera gimbal and/or vibration isolation mechanism onboard the drone, while adding some complexity to the extrinsic transformation from the camera frame to vehicle frame, could help remove camera jitter and free the UAV from the constraint of always flying with the camera faced towards the constellation. These suggestions, as well as the previously addressed areas of improvement and future research, would help contribute to the work done in this thesis to advance the safe use of UAVs in urban environments.
References


