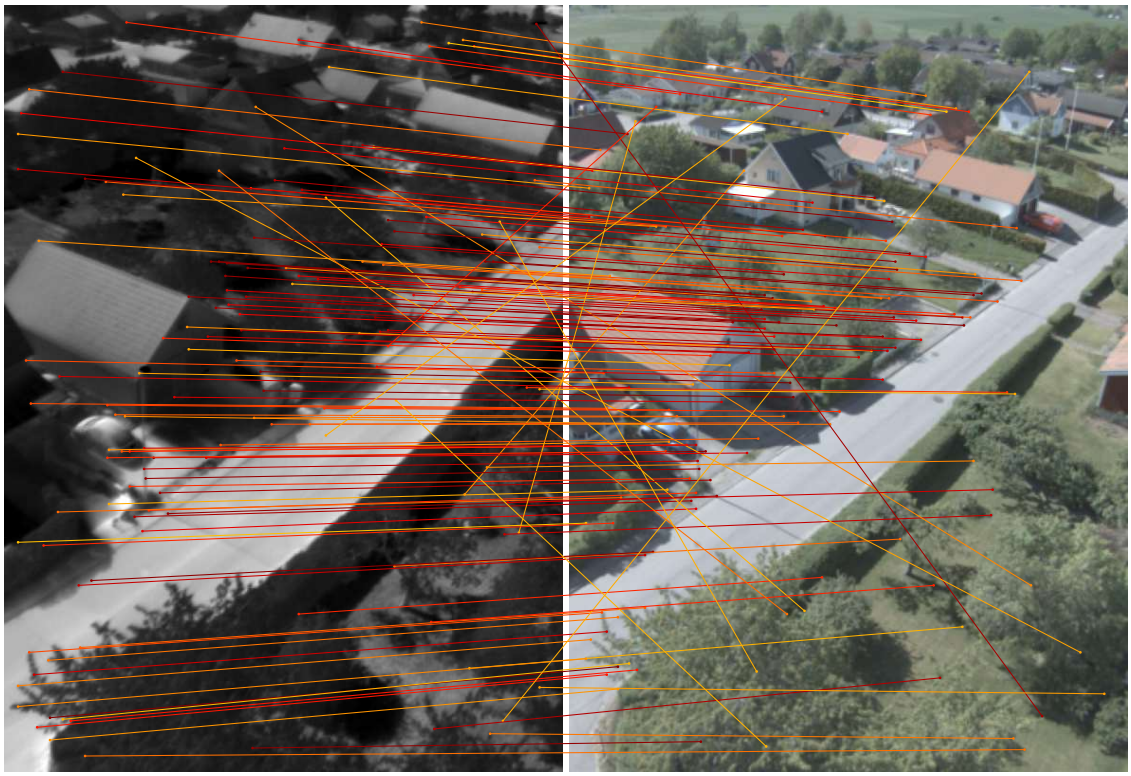
CHALMERS
UNIVERSITY OF TECHNOLOGY



Cross-modal feature matching between infrared and visual images

Adapting intra-modal feature matching models for cross-modal matching

Master's thesis in Data Science and AI

TOMMY ALEXANDER RÄJERT

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

Master's thesis 2023

Cross-modal feature matching between infrared and visual images

Adapting intra-modal feature matching models for cross-modal matching

TOMMY ALEXANDER RÄJERT

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Cross-modal feature matching between infrared and visual images
Adapting intra-modal feature matching models for cross-modal matching
TOMMY ALEXANDER RÄJERT

Supervisor: Yaroslava Lochman, Department of Electrical Engineering
Supervisor: Viktor Ringdahl, Saab Dynamics AB
Examiner: Christopher Zach, Department of Electrical Engineering

Master's Thesis 2023
Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: An example of automatic feature point extraction and matching between an infrared and visual image pair depicting a suburb, with line colors representing match confidence.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2023

Cross-modal feature matching between infrared and visual images
Adapting intra-modal feature matching models for cross-modal matching
TOMMY ALEXANDER RÄJERT
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Image feature matching is an essential part to various computer vision applications. Many modern solutions apply machine learning techniques to achieve state-of-the-art results. A lesser studied problem is matching image features between images of different modalities. This thesis investigates this problem for the visual–LWIR (long-wave infrared) case by utilizing the matching capabilities of the pre-trained intra-modal models SuperPoint and SuperGlue. This is done by adding interfacing models and additional layers to mitigate problems such as catastrophic forgetting and data biasing in the pre-trained models. These techniques prove only marginally successful compared to the pre-trained models themselves. For training these models, a method for sparse pseudo ground truth point correspondence is proposed, and evaluation is done via pose estimation. This thesis provides insight into some specific methods of transfer learning for the SuperPoint and SuperGlue models, methods for ground truth estimation, and discusses the difficulties faced in this problem. Further studying of this problem may be able to construct improved models for LWIR–visual matching, which would enable more reliable methods for cross-modal camera calibration & registration, localization, and image retrieval, with numerous applications in the automotive, defense, and healthcare industries.

Keywords: feature matching, deep learning, computer vision, pose estimation, multi-modal, infrared imaging, graph neural networks.

Acknowledgements

I would like to extend my gratitude to my two supervisors Yaroslava Lochman and Viktor Ringdahl for their invaluable support and advice throughout the project. I would also like to thank Saab Dynamics for the opportunity to work on this project, as well as Fredrik Lundell at Saab for his much appreciated help and for enabling the project by collecting much of the data used. Finally, I wish to thank my very patient examiner Christopher Zach for his valuable feedback and suggestions.

Tommy Räjert, Linköping, August 2023

Glossary

Below are the lists of acronyms and terminology that have been used throughout this thesis, listed in alphabetical order:

Acronyms

ANN	*Artificial neural network.* xv, xvii, 1, 2, 7, 8, 16, 37
AUC	*Area under curve.* xxiii, 36, 41, 44
CNN	*Convolutional neural network.* xii, xiii, xv, xvii, xxi, 5, 8, 9, 23, 37, 45
ECEF	*Earth-centered, earth-fixed coordinate system.* 3, 4, 22, 23, 34
GAN	*Generative adversarial network.* 6, 45
GCN	*Graph convolutional (neural) network.* 9
GNN	*Graph neural network.* xv–xvii, xxi, 5, 9, 17, 23, 33, 40, 42–45
IMU	*Inertial measurement unit.* 2, 4, 22, 48
LWIR	*Long-wave infrared.* xxi, 2, 29
ML	*Machine learning.* 4, 7, 16, 30
MLP	*Multilayer perceptron.* 7, 8, 10, 20
NUC	*Non-uniformity correction.* 4, 25
PnP	*Perspective-n-Point.* xxi, V, 14, 15, 35, 36, 41, 43, 44
RANSAC	*Random sample consensus.* 6, 15, 26, 35, 39–43, 48

Terminology

FPN noise	Short for fixed pattern noise—a type of noise present in certain cameras such as uncooled microbolometers. In this case it is caused by internal heating in the camera and manifests itself as a low-frequency noisy pattern in the image, which is constant over small time intervals. xvi
Kalman filter	A mathematical model used to estimate the state of a system based on noisy measurements over time. 4, 22
Shannon entropy	A measure of the randomness in a set of data. 23
Base-line	The distance between the centers of two pinhole cameras in epipolar geometry. (See Figure 2.2). 12, 15, 20, 22, 30, 33–36, 39, 41, 44, 50
Baseline	Used to refer to the model used as baseline for comparison of experimental results.. 26, 38–42, 44
Cheirality check	A test to determine which of a set of solutions for a camera pose is permissible, based on a set of 3D points in a epipolar geometry. Conversely, this may also refer to determining the sign of 3D points in the same setting. 14
Disparity	The distance a feature in an image moves in the image plane between two cameras in stereo vision. I.e. the pixel distance between the projections of a 3D point onto different image planes. 28, 39
Feature map	The output of a layer in a CNN. 8, 17, 20, 22, 23
Hyperparameter	Parameters (such as layer depth, learning rate, etc.) which cannot be optimized during model training. 8, 17
Image plane	The plane onto which points are projected by the pinhole camera model. (See Figure 2.2). 10–13, 15, 26, 27
Intrinsic	Intrinsic rotations are rotations in 3D described as a sequence of rotations along orthogonal axes relative to the object/camera rotated. This is in contrast to extrinsic rotations, where the rotational axes stay fixed according to an outside reference frame. 22

Kernel	A real matrix used when convolving a feature map in CNNs. 8
Occlusion	The problem of image features not being persistently visible over time in a scene. 23
Attention	A mechanism in some neural networks that allows the model to selectively focus on specific parts of the input. 5, 9
Overfitting	Fitting a model such that it learns the specific distribution of the sampled training data and fails to generalize. 8
Receptive field	The range of inputs a neuron is sensitive to in a neural network. 8, 9

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables which are used throughout this thesis.

Vectors, Matrices, and Tensors

$\boldsymbol{R}$	3×3 rotation matrix	$[\mathbb{R}^{3\times3}]$
$\boldsymbol{K}$	Intrinsic matrix (calibration matrix)	$[\mathbb{R}^{3\times3}]$
$\boldsymbol{C}$	Camera matrix	$[\mathbb{R}^{3\times4}]$
$\boldsymbol{t}$	Translation vector	$[\mathbb{R}^{3}]$
$\boldsymbol{z}, \boldsymbol{Z}$	Point(s) in three dimensions	$[\mathbb{R}^{3}, \mathbb{R}^{3\times N}]$
$\hat{\boldsymbol{z}}, \hat{\boldsymbol{Z}}$	Point(s) in three dimensions projected onto the image plane	$[\mathbb{R}^{2}, \mathbb{R}^{2\times N}]$
$\boldsymbol{p}, \boldsymbol{P}$	Point(s) in image plane	$[\mathbb{R}^{2}, \mathbb{R}^{2\times N}]$
$\boldsymbol{W}^{(i)}$	Weight parameter for the ith layer in an ANN	$[\mathbb{R}^{M\times N}]$
$\boldsymbol{b}^{(i)}$	Bias parameter for the ith layer in an ANN	$[\mathbb{R}^{M}]$
$\boldsymbol{X}$	Input data tensor	$[\mathbb{R}^{N\times K}]$
$\boldsymbol{Y}$	Target data tensor	$[\mathbb{R}^{N_{\text{out}}\times K}]$
$\boldsymbol{F}$	Fundamental matrix	$[\mathbb{R}^{3\times3}]$
$\boldsymbol{E}$	Essential matrix	$[\mathbb{R}^{3\times3}]$
$\boldsymbol{\mathcal{K}}$	CNN kernel matrix	$[\mathbb{R}^{K\times K}]$
$\boldsymbol{l}$	GNN node embedding	$[\mathbb{R}^{V\times N}]$
FoV	Field of view in the x and y axes, as a vector	$[\mathbb{R}^{2}]$
$\boldsymbol{T}$	Affine transformation matrix	$[\mathbb{R}^{4\times4}]$
$\boldsymbol{e}$	Epipole: position of a camera projected onto the image plane of another camera, in epipolar geometry	$[\mathbb{R}^{2}]$

$\boldsymbol{J}$	Jacobian matrix	$[\mathbb{R}^{N\times M}]$
$\boldsymbol{r}$	Vector of reprojection errors	$[\mathbb{R}^{N}]$
$\mathbf{1}, \mathbf{0}$	Constant vector of ones or zeros, respectively	$[\mathbb{R}^{N}]$
$\boldsymbol{\mathcal{H}}$	Homography matrix	$[\mathbb{R}^{3\times 3}]$
$\boldsymbol{d}$	Feature descriptor vector	$[\mathbb{R}^{256}]$
$\mathbf{P}, P$	Collection of points in a 2D graph as a vector and set, respectively	$[\mathbb{R}^{N\times 2}], \mathbb{R}^2$
$\mathbf{E}, E$	Collection of 2D graph edges as a vector and set, respectively	$[P^{N\times 2}], P^2$
$\boldsymbol{A}$	Adjacency matrix for a graph	$[\mathbb{R}^{N_1\times N_2}_{\geq 0,\leq 1}]$
$\boldsymbol{\mathcal{I}}$	Image	$[\mathbb{N}^{N\times M}]$
$\boldsymbol{\mathcal{E}}$	FPN noise component of an image	$[\mathbb{N}^{N\times M}]$

Variables and functions

d	Network depth/number of layers	$\mathbb{R}$
a, b	Input and output vector sizes	$\mathbb{N}$
m, n, i	Indices	$\mathbb{N}$
M, N	Tensor dimension sizes	$\mathbb{N}$
k	Batch index	$\mathbb{N}$
v, u	GNN nodes	–
f_x	Focal length parameter in horizontal (x) image axis	$\mathbb{R}$
f_y	Focal length parameter in vertical (y) image axis	$\mathbb{R}$
c_x	Principal point in the image plane, along the horizontal (x) axis, in pixels	$\mathbb{R}$
c_y	Principal point in the image plane, along the vertical (y) axis, in pixels	$\mathbb{R}$
s_x	Image width in pixels	$\mathbb{N}$
s_y	Image height in pixels	$\mathbb{N}$
κ_i	Radial distortion parameters	$\mathbb{R}$
$\Delta\theta$	Angle (error) between two vectors	$[0, 2\pi)$
$\mathcal{L}$	Loss function	$\mathbb{R}^{N_{\text{out}}\times K} \to \mathbb{R}$
g	Activation function	$\mathbb{R} \to \mathbb{R}$

F	Artificial neural network as a function	$\mathbb{R}^{N \times K} \to \mathbb{R}^{N_{\text{out}} \times K}$
K	CNN kernel size	$\mathbb{N}$
r	Radius from the principal point in the image	$\mathbb{R}$
τ	Function returning a triangulated point $\boldsymbol{z}$, given two corresponding points $\boldsymbol{p}, \boldsymbol{p}'$, and their respective camera poses $\mathcal{C}, \mathcal{C}'$	$\mathbb{R}^2 \times \mathbb{R}^2 \times [-\pi, \pi)^6 \times [-\pi, \pi)^6 \to \mathbb{R}^3$
m_p, m_n	Cosine loss threshold for positive and negative samples, respectively	$\mathbb{R}$
λ_d	Hyperparameter	$\mathbb{R}$
$\mathcal{P}$	SuperPoint neural network as a function	$\mathbb{R}_{[0,1]}^{s_x \times s_y} \to \mathbb{N}^{2 \times k} \times \mathbb{R}^{k \times 256}$
$\mathcal{G}$	SuperGlue neural network as a function	$\left(\mathbb{N}^{2 \times k} \times \mathbb{R}^{k \times 256}\right)^2 \to P^k \times P^k \times E^k$
$\mathcal{M}$	Neural network model interfacing SuperPoint and SuperGlue, as a function	$\mathbb{R}^{k \times 256} \to \mathbb{R}^{k \times 256}$, $\left(\mathbb{R}^{k \times 256}\right)^2 \to \left(\mathbb{R}^{k \times 256}\right)^2$
$\mathcal{T}$	Image translation model as a function	$\mathbb{R}^{N \times M} \to \mathbb{R}^{N \times M}$
$\mathbb{i}$	Image pixel intensity	$\mathbb{N}$
t	Time	$\mathbb{N}, \mathbb{R}$
$\mathcal{R}$	Function mapping a rotation matrix to the angle which it rotates a point around the origin	$\mathbb{R}^{3 \times 3} \to [0, 2\pi)$
d	Distance in meters	$\mathbb{R}$
C	Number of channels in the feature map and kernel	$\mathbb{N}$
ϵ	Small value	$\mathbb{R}_+$

Notation

$\underline{\boldsymbol{z}}$	Homogeneous form of $\boldsymbol{z}$
$\bar{\boldsymbol{t}}$	Normalized vector of $\boldsymbol{t}$
$\pi(\boldsymbol{\mathcal{H}}\underline{\boldsymbol{p}})$	$\boldsymbol{p}$ transformed by the homography $\boldsymbol{\mathcal{H}}$
$\hat{\boldsymbol{t}}$	Estimated $\boldsymbol{t}$
$\hat{\mathcal{P}}$	Modified network $\mathcal{P}$
$g : \mathbb{R} \to \mathbb{R}$	Type membership
$\boldsymbol{X} \in \mathbb{A}$	Set membership
$v \sim u$	Adjacency of GNN nodes
IR $\sim$ Vis	Specifies the modalities compared in feature matches
$\mathcal{C}$	Indicates the pose of a camera

$\mathcal{C}^{ab}$	Indicates the pose of a camera a relative to the reference frame of camera b
Vis, IR, C, Vis+IR, A, B	Indicates visual, LWIR, calibration board, fused (visual+LWIR), and two generic modalities (A and B), respectively

Contents

List of Figures

List of Tables

1

Introduction

Detecting, describing, and matching image features are essential tasks for many important computer vision applications such as structure from motion (SfM) [1], [2], simultaneous localization and mapping (SLAM) [3], retrieval, image registration, object recognition and tracking, etc. [4]–[8]. Sets of image features are used to represent the image and condense the information contained therein in order to enable efficient processing and model training [7].

Modern techniques replace analytical approaches (such as SIFT[†] [9] and ORB[‡] features [10]) with neural networks to attain state-of-the-art results [5], [6], [11]–[13], but these often suffer from being sensitive to changes to the input distribution [14]. In some cases one might want to reliably compare images from very different data distributions, such as for cross-domain or cross-modal image retrieval (including applications like retrieving faces with a phantom sketch, finding photos of buildings from paintings, or matching shoeprints imprinted in different media), multimodal camera calibration, or remote sensing [5], [11], [15], [16]. In these scenarios both traditional and modern generalized (intra-modal)[§] techniques will most likely perform poorly [15], [17], since they were designed (and trained) to work with samples from a single data distribution. As such, specialized models have been proposed for some of these specific applications.

For industries such as the automotive, defense, and healthcare industries that utilize computer vision assisted systems there is an interest in utilizing the information from several modalities to gain more accurate information about the environment [18], or to use data from one modality to make inferences inside another. In the latter case, one modality could be used for model training, while another is used in the end system for inference. This can be advantageous when high-quality reference data is best collected in one modality, while another is more practicable for the end solution for reasons related to cost, speed, reliability, etc., or simply because a pre-trained system trained on another modality is already available.

Different environments present different visual challenges for computer vision sys-

[†]**S**cale-**i**nvariant **f**eature **t**ransform

[‡]**O**riented FAST and **R**otated **B**RIEF

[§]In Section 4.3 a state-of-the-art trained image feature matching model is demonstrated to perform below expectations using similar modalities.

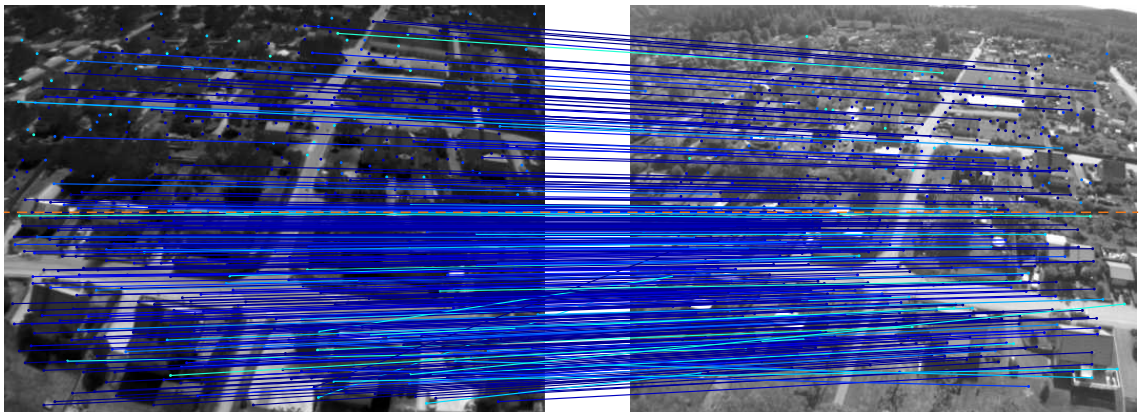


Figure 1.1: An example of a good prediction using the final trained neural network model. The matches are colored according to the network confidence with `jet` colormap where dark blue corresponds to the highest confidence. For better visibility, the upper half of the figure shows a random 20% sample of the matches; the two halves are separated by a dashed line.

tems and calls for different sensing equipment (such as cameras capturing different wavelengths). Reliably operating, discovering relations, and making inferences between these modalities can be difficult [15]. In cases where useful reference data is primarily available from one modality, while a system operates on a different modality—for example, a system for visual localization may use an infrared camera for sensing, but only have a reference map for the visual spectrum—being able to establish useful correspondences between these data sources is therefore valuable.

This work addresses the general problem of matching image features from different modalities (See Figure 1.1), and investigates how this can be done to match features between visual and thermographic cameras in particular, in order to facilitate accurately extrinsically calibrating these cameras, as well as retrieve images from one modality corresponding to an image from the other modality.

1.1 Problem Formulation

This work aims to adapt pre-trained neural network models to the cross-modal task of finding point correspondences between images taken with a pair of cameras—one in the visual domain, and the other in the LWIR domain. Given a pair of such images (one visual and one infrared) depicting the same scene at the same point in time, the algorithm should yield a set of point correspondences that are sufficiently accurate, numerous, and spread across the scene to be useful for calibrating the two cameras.

The problem of relating the visual and thermographic image modalities can be analyzed in many different domains and for an array of different purposes. In this work, drone-based aerial photography will be considered.

The drone rig is fitted with a visual and a thermographic camera, an *inertial mea-*

Properties	Visual	Thermal
height (px)	1024	480
width (px)	1280	640
bit depth	8	16
color channels	3*	1
FPS† (hz)	10	10
FoV_x (deg)	68.3437	56.8880
FoV_y (deg)	57.0105	44.2094
Spectral range (μm)	-	8–14

Table 1.1: Parameters of the two cameras used to collect the image data.

surement unit (IMU), and a global positioning system (GPS) unit for the purpose of creating 3D maps of its environment using SLAM. In order to leverage all the data coming from both visual and thermographic cameras, a proper correspondence between the images of different modalities (i.e. cross-modal matching) needs to be established. Solving this fundamental problem will unlock the possibilities to develop reliable cross-modal vision algorithms, such as (1) general offline and, perhaps, more challenging online calibration algorithm of the cross-modal camera system and (2) cross-modal SLAM, where it is possible to do both intra- and cross-modal matching in order to determine location relative to prior images of the environment, regardless of which modality of the data and of the camera is available. For both of these tasks (cross-modal matching and extrinsic calibration), accurately describing and matching image features across the two modalities is a vital requirement, and since features can appear very different in visible light compared to infrared light, matching across these modalities is not a trivial problem [17].

A solution to the offline calibration problem is to calibrate the cameras with a thermal checkerboard pattern, and perform a calibration routine before flight. However, a more seamless and flexible approach would be to do the calibration automatically by image feature matching, using the information available in the scene.

To facilitate these needs, a reliable and efficient way to compute and match these cross-modal features is required. Intra-modal feature matching has been extensively studied but is still an active field [13], [19], [20]. Cross-modal feature matching, on the other hand, has been studied to a lesser degree, likely in part because it is difficult to generalize and requires a high level of information abstraction.

The data collected consists of a set of series of visual and infrared images, as well as ECEF coordinates and Euler rotations for each image pair. For each series of images there is also a difference in pose between the cameras, as well intrinsic parameters for each camera, which were calibrated using a thermal checkerboard. See Section 3.2 for details. The properties of the cameras are shown in Table 1.1.

The thermal camera is an uncooled microbolometer capturing light in the long wave

*The visual images are in most cases converted to gray scale in this report.
†The frequency at which images are recorded to the dataset.

(a) LWIR Image (b) Visual Image

Figure 1.2: A pair of images from the dataset, showing a suburb.

infrared (LWIR) range. Because it is uncooled, a noticable artifact in the image data are the (vertical) bands of noise that gradually increase in intensity over time until the camera performs a *non-uniformity correction* (NUC). The camera performs this NUC in intervals of 250 seconds, and briefly halts the capture of images during the correction.

The visual camera is synchronized with the thermal camera to capture images at the same time the thermal camera does, and it is stored in a raw Bayer-filter format. An example of an image pair is shown in Figure 1.2.

The pose data was collected using an *inertial measurement unit* (IMU) and a pair of GPS receivers—one stationary, and the other mounted on the drone—and subsequently processed by a Kalman filter to get poses for the IMU in terms of ECEF coordinates and Euler angles. The pose data was then sampled (by interpolation) at times corresponding to when images were captured.

The pose calculated for the IMU is used also for one of the cameras, resulting in an error in pose for the two cameras that becomes apparent whenever the drone rotates around an axis not parallel to the axis between the IMU center and the optical center of a camera. See Section 5.1 for discussion on the potential impact of this.

1.2 Related Work

The problem of relating images between different modalities or domains has been studied in a number of different contexts and situations, including infrared and visual wavelength images; although to a limited extent. This section will detail some notable examples of this related work. In the infrared–visual context, there are notably a few well-known resources for image data, such as the FLIR [21] and KAIST [22] datasets for autonomous driving ML tasks. These are however rather biased in terms of image content, since they are intended for autonomous driving

tasks.

Kong et al. [11] investigated cross-domain image feature matching for matching pictures of shoeprint indentations left in various materials at crime scenes, to a database of known shoeprints. As the database of images were systematically collected (with high contrast, clear and complete prints, and white background) and thus constitutes a very narrow distribution, the distribution of pictures taken at crime scenes is far from representative of this known distribution. They attempt to solve this problem by training a Siamese network to project corresponding image features onto the same feature space, using CNNs to reduce the feature space of the input, then applying learned importance weighting and a three-channel variant of normalized cross-correlation to pairwise image patches, and scoring performance by l2-regularized hinge loss. By this, they obtain state-of-the-art results, even though there is a lot of room for improvement with 79.0% top-1% accuracy. An issue they discuss concerning their cross-correlation is that there is no natural three-channel analogue to the regular grayscale implementation; their implementation is one of several proposed multi-channel generalizations. Another limitation is the CNN architecture they use, which has been shown to not relate spatial relations and features far apart as well as architectures utilizing attention [23].

Sarlin et al. [19] and Detone et al. [5] detail a pair of models called SuperGlue and SuperPoint, respectively, that together learn an intra-domain feature matching, using a combination of CNNs, GNNs, self-attention, and cross-attention. In order to train their SuperPoint model to find good interest points, they train on a labeled set of image pairs—each pair related by a homography—using a Siamese architecture (similarly to Kong et al. [11] or Liu et al. [6]) to simultaneously generate both interest points and point descriptors. This labeled data is generated from a separate model which has been trained on simple randomly synthesized images. The process of generating the labels for the training data consists of letting this separate model predict points on several random cropped homographies of an image from the training data, and using a subset of the union of these points as the training labels (named "homographic adaptation" by the authors). This self-supervised Siamese model is then trained with cross-entropy loss for the set of points and hinge loss for their descriptors. The SuperGlue model is subsequently trained to solve the matching problem with the feature points obtained by SuperPoint, and the two models are then jointly trained end-to-end [19].

The combination of the two models, SuperPoint and SuperGlue, show clear advancements of the state-of-the-art in image matching [19]. The work is also a stepping stone towards a fully end-to-end deep visual SLAM pipeline [19].

The SuperPoint–SuperGlue model works well for intra-modal matching, for which it was designed and trained, but has not been adapted to cross-modal matching. As shown in [5], the model is able to generalize from being trained on synthetic data, and should therefore be adaptable and perform well with intra-modal matching for thermographic images. While this work focuses on "front-end" feature point extraction and their "middle-end" matcher, Kong et al. [11] relates the cross-modal

aspect to the "middle-end" and "back-end." The former work would be important in extracting feature points without supervision, learned point description, deep and attention-based feature embedding, and interpreting the scene, and the latter for ideas on how to relate cross-modal features using a common embedding space, relating importance of features in this aspect, and gains of whitening in this process.

There are a few other papers tackling similar cross-domain problems, such as the one by Shrivastava et al. [15], which presents alternate approaches to that of Kong et al., that may also be considered for our purposes. In this paper, SVM is used in order to discriminate feature importance and promote feature uniqueness.

The topic of cross-modal matching has been studied in the context of closely-related field of multi-modal image registration [17], [18], [24]–[30], which has numerous applications in the medical, remote sensing, and 3D vision research areas [27].

Wachinger and Navab [25] analyze the advantages of using the so-called structural representations for multi-modal registration. A possibility to achieve modality-independent representations allows to use similarity metrics related to euclidean space, such as L1 or L2 distances, rather than more computationally expensive metrics like mutual information when comparing the features. The authors propose two versions of structural representations—entropy images and Laplacian images—that fulfill certain requirements of the theoretical properties for efficient optimization. However, each type of representation comes with a drawback, either inferior registration performance or higher computational cost.

Pielawski et al. [24] proposes another way to achieve efficient modality-independent representations. Referred to as COMIRs, the representations are obtained through training a dense U-Net-like [31] model with contrastive loss and additional regularization to enforce rotation equivariance. The proposed approach outperforms registration methods based on mutual information maximization or *generative adversarial networks* (GANs) [32], however, for training, it requires aligned pairs of multi-modal images that are rarely available in the real SLAM cases.

Zhang et al. [17] cover domains closer to the ones of interest for this work, where they consider the problem of image registration by homography estimation between visual and infrared modalities. Their approach consists of a novel image-to-image translation approach, which they use to translate the infrared image into the visual modality. After translating the images, they extract features by traditional techniques such as SIFT [9] and ORB [10], perform feature matching, and estimates a homography with RANSAC. Their image-to-image translation is based on the GAN architecture and incorporates a wavelet transform into a variational autoencoder, and only considers the low-frequency components of the image for inferring the variational variables.

2

Background

This chapter serves to provide some background to topics in machine learning relevant to this work, as well as theory on stereo vision, and previous work on image feature detection, description and matching. Variables and functions will be introduced with their corresponding type, or set membership for clarity. These can also be referenced in the Nomenclature.

2.1 Artificial Neural Networks

Artificial neural networks (henceforth "neural networks" or "ANNs") are a class of computational models designed to be very general and expressive while being able to be optimized to fit to some set of data according to some *loss function* describing the heuristic quality of the fit. The model takes inspiration from the structure of the brain by modeling individual neurons and their connections [14]. It does this by assigning a simple parameterized function to each of a set of nodes, taking a weighted sum of the values of a subset of the set of nodes as input [14], [33].

The relatively simple *multilayer perceptron* (MLP) is often used to introduce the general structure of ANNs [33]. An MLP consists of d layers of nodes (neurons): an input layer, an output layer, and $(d-2)$ *hidden layers*. The middle layers are called "hidden" because the network is treated as a black-box function. The value of the mth node of layer i in the MLP, $L_{m\cdot}^{(i)}$, is given by Equation 2.1a, and the MLP network, as a function $F : \mathbb{R}^{A\times K} \to \mathbb{R}^{B\times K}$ of the input data $\boldsymbol{X}$, is given by Equation 2.1c. Equation 2.1b uses vector notation instead of the indexed sum notation and will be used in this thesis for simplicity. Notice that the sum of $\boldsymbol{W}^{(i)}\boldsymbol{L}^{(i-1)} : \mathbb{R}^{M\times K}$ and $(-\boldsymbol{b}^{(i)}) : \mathbb{R}^{m}$ do not have matching dimensions; the vector $\boldsymbol{b}$ is implicitly replicated over dimension k. This dimension is the *batch* dimension, which is sometimes left out, but included here for completeness, and represents an improper subset of the data $\boldsymbol{X}$. The data is split in batches in order to enable parallel computations. The data itself is a vector of vectorized *features*, (i.e. a matrix, or more generally, a tensor).

In Equation 2.1, $W_{mn}^{(i)} : \mathbb{R}^{M\times N}$ is a matrix of all weights applied to each connection between every node in the preceding layer with every node in the current layer. Each

node has its own *bias* $b_m^{(i)} : \mathbb{R}^M$ that is used to threshold* or offset the value of the node, which makes this equivalent to an affine transformation between layers. This alone would only constitute an affine transformation between the input and output of the network, making the hidden layers and all their parameters redundant. In order to introduce non-linearity into the network—and thereby increase the generality of the network—an *activation function* $g : \mathbb{R} \to \mathbb{R}$ is defined as some non-linear function which is quick to compute when applied element-wise to a tensor of values.

In this MLP network, the variables $\boldsymbol{W}$ and $\boldsymbol{b}$ are optimized to minimize some objective (loss function) $\mathcal{L}(F(\boldsymbol{X}), \boldsymbol{Y})$.

$$L_{mk}^{(i)} = g\left(\sum_{m=1}^{M} W_{mn}^{(i)} L_{nk}^{(i-1)} - b_m^{(i)}\right) \tag{2.1a}$$

$$\boldsymbol{L}^{(i)} = g\left(\boldsymbol{W}^{(i)}\boldsymbol{L}^{(i-1)} - \boldsymbol{b}^{(i)}\right) \tag{2.1b}$$

$$F(\boldsymbol{X}) = g_{\text{out}}\left(\boldsymbol{W}^{(d)} g\left(...g\left(\boldsymbol{W}^{(1)}\boldsymbol{X} - \boldsymbol{b}^{(1)}\right)...\right) - \boldsymbol{b}^{(d)}\right) \tag{2.1c}$$

Neural networks are by design non-linear, and a globally optimal solution can generally not be found analytically. Therefore numerical optimization is used, which is usually done with some variation on *stochastic gradient descent* (SGD) [14], [33], or *Bayesian optimization* in the case of hyperparameter optimization [33].

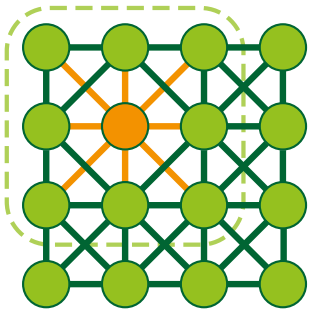
2.1.1 Convolutional Neural Networks

A fundamental problem in deep learning is the rapidly increasing number of trainable parameters [34]: as the number of parameters increases, so does the training time and risk of overfitting. *Convolutional neural networks* (CNNs) limit the receptive field of each neuron to an $K \times K$ region in the previous layer. They also assume translational invariance accross the data, meaning that the network does not have to learn to recognize the same feature in each position in the input tensor. In doing these two things, the number of parameters is significantly decreased, and convergence is sped up. [14], [16], [33] The CNN architecture is described layer-wise as a discrete convolution operation between a feature map and a *kernel* $\boldsymbol{\mathcal{K}} : \mathbb{R}^{K\times K}$, according to Equation 2.2 [14], [33]:

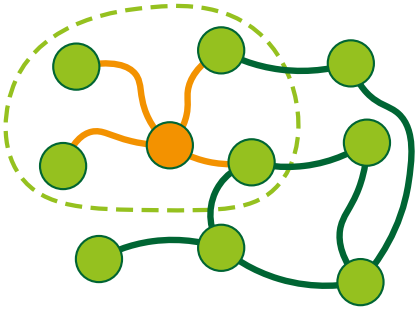
$$L_{ijk} = g\left(\sum_{m=1}^{M}\sum_{n=1}^{N} \mathcal{K}_{mn} L_{m+i-1,n+j-1,k} - b\right) \tag{2.2}$$

Although CNNs are applicable to data of any number of dimensions [35], we will only consider the two-dimensional case in this report, for simplicity.

*If this behaves like a threshold or not depends on the activation function. For an activation function like ReLU it acts like a threshold, while in general it acts like an offset.



(a) 2D convolution as a graph, with a 3×3 kernel.



(b) A general graph.

Figure 2.1: A comparison between the convolutional operation of a conventional 2D CNN (a) and the convolutional operation of a GNN (b). The orange node depicts how information is aggregated from the receptive field from one layer to the next. The dashed line indicates the receptive field of the orange node.

2.1.2 Graph Neural Networks

There are a number of different formulations for *graph neural networks* (GNNs) [19], [35], [36], but the general principles are the same. While convolutional neural networks—as well as many other architectures—treat data in an ordered manner, some kinds of data are better viewed as (unordered) sets. Placing relations between set elements then naturally leads to graphs. The GNN architecture considered in this work is an instance of graph convolutional networks (GCNs or ConvGNNs) [19], [36]. This variant of GNNs can intuitively be viewed as a generalization of convolutional networks [36].

While conventional CNNs by definition have a limited receptive field [14], in GCNs the receptive field is broadened to an arbitrary size, since it is not necessarily determined by proximity. This is illustrated in Figure 2.1, in which the receptive field for a node in a lattice and a general graph is shown, representing parts of CNN and GCN, respectively.

The convolution in a GCN is often referred to as *message passing*, since encoded information travels back and forth along the edges of the graph. As noted by Sarlin et al. [19], message passing in a graph can also be viewed as an attention mechanism.

Micheli [37] describes the message passing in a GCN as the following [36]:

$$\boldsymbol{l}_v^{(i)} = g\left(\boldsymbol{W}_1^{(i)}\boldsymbol{x}_v + \sum_{u \sim v} \boldsymbol{W}_2^{(i)}\boldsymbol{l}_u^{(i-1)}\right) \tag{2.3}$$

This amounts to each node in a layer being a linear combination of its neighbors and the input vector, (ignoring g). In Equation 2.3, node v in layer (i) is defined as the sum of the input features $\boldsymbol{x}$ weighted by $\boldsymbol{W}_1^{(i)}$, and the sum of adjacent node embeddings $\boldsymbol{l}_u$ weighted by $\boldsymbol{W}_2^{(i)}$. The sum $\sum_{u \sim v}$ denotes the sum for all nodes u

adjacent to node v. As in previous architectures, the linear combination of feature vectors is passed through an element-wise non-linear activation function $g : \mathbb{R} \to \mathbb{R}$.

Many variations of this definition exist that incorporate explicit attention mechanisms, add MLPs to each message pass, etc. [36]. Section 2.3 will introduce one of these in more detail.

2.2 Computer Vision

Computer vision is a field that studies problems involving the extraction of high-level information from visual input, such as object detection and structure from motion (SfM) [7], [38], [39]. This section aims to provide theoretical background relevant to the problem at hand. Stereo vision is an area of computer vision concerned with problems involving two cameras. The relation between the two views are described in epipolar geometry; see Section 2.2.2.

2.2.1 Camera Model

Cameras, points in the world, and the picture are described in terms of linear algebra with homogeneous coordinates. A *camera matrix* is a projection matrix $\boldsymbol{C} \in \mathbb{R}^{3\times4}$ that relates a point $\boldsymbol{z} \in \mathbb{R}^3$ in the world to a point $\hat{\boldsymbol{z}} \in \mathbb{R}^2$ in the image plane of the camera, by the relation in Equation 2.4 [7], [8]:

$$\underline{\hat{\boldsymbol{z}}} = (\hat{\boldsymbol{z}} \;\; 1)^\intercal \sim \boldsymbol{K}\,(\boldsymbol{R} \mid \boldsymbol{t})\,(\boldsymbol{z} \;\; 1)^\intercal = \boldsymbol{C}\,(\boldsymbol{z} \;\; 1)^\intercal = \boldsymbol{C}\underline{\boldsymbol{z}} \tag{2.4a}$$

$$\boldsymbol{p} \approx \hat{\boldsymbol{z}} = \pi\,(\boldsymbol{C}\underline{\boldsymbol{z}}) \tag{2.4b}$$

where $\underline{\boldsymbol{z}} \overset{\text{def}}{=} (\boldsymbol{z} \;\; 1)^\intercal$ denotes $\boldsymbol{z}$ in *homogeneous coordinates*, and $\pi : \mathbb{R}^{n+1} \to \mathbb{R}^n$ normalizes the homogeneous coordinates.

The projection matrix is usually decomposed into an *intrinsic* matrix (also known as *calibration matrix*) $\boldsymbol{K} \in \mathbb{R}^3$ and *extrinsic* matrix $\left(\boldsymbol{R} \;\middle|\; \boldsymbol{t}\right) \in \mathbb{R}^{3\times4}$ [7]. The extrinsic matrix constitutes the camera pose, where $\boldsymbol{R}$ is a rotation matrix and $\boldsymbol{t}$ is a translation matrix, whereas the intrinsic matrix is built from parameters determined by the specific camera. The intrinsic matrix is an upper-triangular matrix and in this work is modeled assuming zero skew as in

$$\boldsymbol{K} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

where f_x and f_y are the focal length of the camera in pixels in the x and y direction respectively, and c_x and c_y are the x and y coordinates of the principal point in pixels.

The focal length components are computed from the field of view components (listed in Table 1.1) according to $(f_x \ f_y)^\intercal = \boldsymbol{C} / \left(2 \tan\left(\mathbf{FoV}/2\right)\right)$.

The camera model explained so far is referred to as the *pinhole camera model* [7], [8]. It is an idealized model of the camera, and it is often not enough to model the real cameras due additional distortions introduced by the lens systems and sensor plane misalignment [8]. A deviation from an ideal pinhole point occurs during the projection of the world point onto the sensor plane. Hence, the image measurements need to be corrected or *undistorted*. In this work, we use the commonly used radial distortion model [7], as it is good enough for various applications.

The distortion model [40] is defined by

$$r' = r\left(1 + \sum_{i=1}^{i_{\max}} \kappa_i r^i\right) \tag{2.6}$$

where r' is the radius of the projected point before multiplying by $\boldsymbol{K}$ and r is the radius of an ideal pinhole point $\boldsymbol{p}$ [7], [41]. Normally, around two or three parameters κ_i are used, and only even exponents of r are used since terms with odd exponents can lead to less stable estimates [41]. For the data used in this work, distortion parameters are however expressed in terms of a polynomial with also odd exponents of r (except the linear term) and the order of a polynomial is $i_{\max} = 4$.

When describing the camera matrix, the transformations in three-dimensional space expressed by the extrinsic matrix are described in terms of transforming points in some reference frame into the reference frame of the camera. In other words, applying $(\boldsymbol{R} \mid \boldsymbol{t})$ to the point where the camera is located in the world's reference frame, will transform it to the origin. The camera matrix is given by

$$\boldsymbol{C} = \boldsymbol{K}\boldsymbol{T}^{ec} = \boldsymbol{K}\left(\boldsymbol{R} \mid \boldsymbol{t}\right) \tag{2.7}$$

We denote the camera pose (position and orientation), in relation to the world coordinate system, by $\mathcal{C}^{ce}$, and a transformation matrix $\boldsymbol{T}$ (such as the extrinsic matrix or a rotation matrix) by $\boldsymbol{T}^{ce}$ if it is of the camera c in relation to the world e, or by $\boldsymbol{T}^{c_2c_1}$ if it is of the camera c_2 in relation to the camera c_1.

2.2.2 Epipolar Geometry

Stereo vision is an area of computer vision concerned with problems involving two cameras. The two-view geometry describing the relation between the two cameras and the world—often called *epipolar geometry* [7], [8]—is described in this section.

Figure 2.2 depicts the components of epipolar geometry. The relationship of individual points between the two image planes is dependent on their corresponding 3D point in the world. Given a projected point $\hat{\boldsymbol{z}}$ in the image plane of the camera at

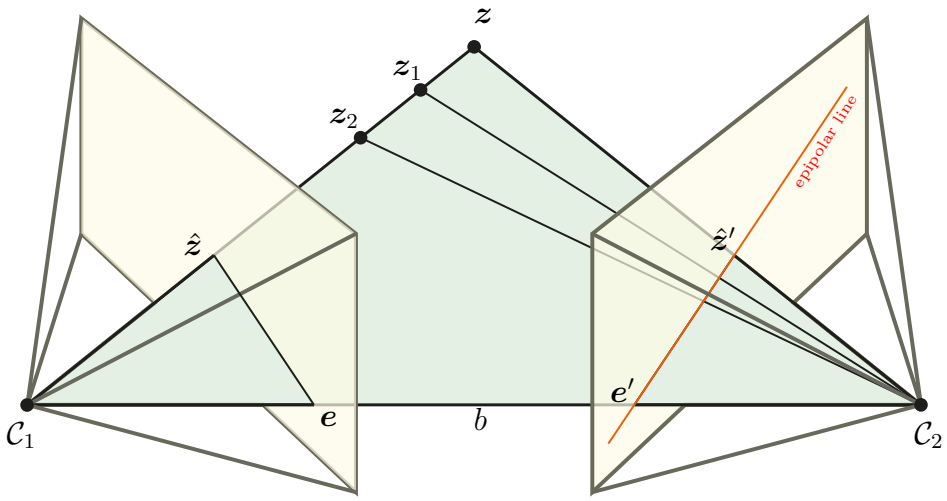


Figure 2.2: Two pinhole cameras are angled towards each other, separated by their base-line b. $\mathcal{C}_1$ and $\mathcal{C}_2$ denote the pose of the two respective cameras and are depicted as points at the optical center of the cameras. $\boldsymbol{z}$ is a point in the world viewed by the two cameras, and is projected onto the image planes (colored yellow) as $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{z}}'$. $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are other possible positions for $\boldsymbol{z}$, given $\hat{\boldsymbol{z}}$, and the projection of these all fall onto the epipolar line of the camera to the right. The camera center of one of the cameras projected onto the image plane of the other is known as the *epipolar point*, and is labeled $\boldsymbol{e}$ and $\boldsymbol{e}'$ for the two respective cameras. Every possible epipolar line will cross the epipolar point.

$\mathcal{C}_1$, the location of $\boldsymbol{z}$ is constrained to a line $\overline{\mathcal{C}_1\boldsymbol{z}}$. The projection of this line onto the image plane of the camera at $\mathcal{C}_2$ is called an *epipolar line* $\overline{\boldsymbol{e}'\hat{\boldsymbol{z}}'}$, and the requirement that $\hat{\boldsymbol{z}}'$ lies on this line is called the *epipolar constraint* [7].

The *fundamental matrix* $\boldsymbol{F} \in \mathbb{R}^{3\times3}$ encapsulates the epipolar constraint between points in the two images. It is defined by Equation 2.8 [7], [8]:

$$\underline{\hat{\boldsymbol{z}}}'^{\intercal}\boldsymbol{F}\underline{\hat{\boldsymbol{z}}} = 0 \tag{2.8}$$

and holds for any 3D point $\boldsymbol{z}$, given $\mathcal{C}_1$ and $\mathcal{C}_2$. In the special case where the intrinsic matrices are identity $\boldsymbol{I}$, $\boldsymbol{F}$ is called the *essential matrix*, denoted $\boldsymbol{E} \in \mathbb{R}^{3\times3}$, and they are related by Equation 2.9 [7], [8]:

$$\boldsymbol{F} = \boldsymbol{K}_2^{-\intercal}\boldsymbol{E}\boldsymbol{K}_1^{-1} \tag{2.9}$$

Because of Equation 2.8, both the fundamental and essential matrix is only defined up to a scale.

The epipolar constraint is often used for outlier rejection or as a form of weak supervision [8], [42], but the epipolar error $\boldsymbol{p}^{\intercal}\boldsymbol{F}\boldsymbol{p}$ is a biased [43]. Instead, the *Sampson*

distance is used [44], which is a first-order approximation of the reprojection error* [43]. The Sampson distance [8], [43] SD is defined according to

$$\mathrm{SD}(\boldsymbol{p}, \boldsymbol{p}') = \frac{(\boldsymbol{p}'^{\intercal}\boldsymbol{F}\boldsymbol{p})^2}{\boldsymbol{J}\boldsymbol{J}^{\intercal}} = \frac{(\boldsymbol{p}'^{\intercal}\boldsymbol{F}\boldsymbol{p})^2}{(\boldsymbol{F}\boldsymbol{p})_1^2 + (\boldsymbol{F}\boldsymbol{p})_2^2 + (\boldsymbol{F}^{\intercal}\boldsymbol{p}')_1^2 + (\boldsymbol{F}^{\intercal}\boldsymbol{p}')_2^2} \tag{2.10}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the error in the image points $\boldsymbol{p}$ and $\boldsymbol{p}'$.

2.2.2.1 Triangulation

Triangulation is the problem of determining the 3D point $\boldsymbol{z}$ of which the respective points $\boldsymbol{p}^{(n)}$ in N different image planes approximate the projections $\hat{\boldsymbol{z}}^{(n)}$ [8], [45]. This requires prior knowledge about the pose of the different cameras, as well as the location of the points in the different image planes [45]. In other words, in the stereo case with several points to estimate, it assumes the two cameras have calibrated relative extrinsics and the (feature) points have good matches between the images. In this work we only consider the case where $N = 2$.

Triangulation can be formalized as the minimization of the sum of squared reprojection errors—as in Equation 2.11—while keeping the epipolar constraint (Equation 2.8) precisely fulfilled [45], [46]. In this report, triangulation is denoted as a function τ, as in Equation 2.12, similarly to Hartley [45].

$$\boldsymbol{z} \approx \arg\min_{\boldsymbol{z}} \sum_n \left\| \boldsymbol{p}^{(n)} - \pi\left(\boldsymbol{C}_n \underline{\boldsymbol{z}}\right) \right\|_2^2 \tag{2.11}$$

$$\boldsymbol{z} = \tau(\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{p}^{(1)}, \boldsymbol{p}^{(2)}) \tag{2.12}$$

2.2.2.2 Relative camera pose estimation

There are two general settings we will primarily consider in this report. First, when $\boldsymbol{P}$, $\boldsymbol{P}'$, $\mathcal{C}_1$, and $\mathcal{C}_2$ are known, and $\boldsymbol{Z}$ is unknown: this is described in Section 2.2.2.1. Second, when $\boldsymbol{P}$ and $\boldsymbol{P}'$ are known, but $\mathcal{C}_1$ and $\mathcal{C}_2$ are unknown. In this second setting, we know there exists a fundamental matrix such that $\hat{\underline{\boldsymbol{Z}}}'^{\intercal}\boldsymbol{F}\hat{\underline{\boldsymbol{Z}}} = \boldsymbol{0}$, and the same can be said for $\boldsymbol{E}$ using normalized image coordinates [8], given the intrinsic matrices.

The essential matrix $\boldsymbol{E}$ can also be expressed in terms of the relative pose between the cameras, as in Equation 2.13 [8]:

$$\boldsymbol{E} \sim [\boldsymbol{t}^{c_2 c_1}]_\times \boldsymbol{R}^{c_2 c_1} \tag{2.13}$$

*The reprojection error here is that of the reprojection of the optimal triangulation of the two points, i.e. the reprojection error with the minimum sum of squared errors.

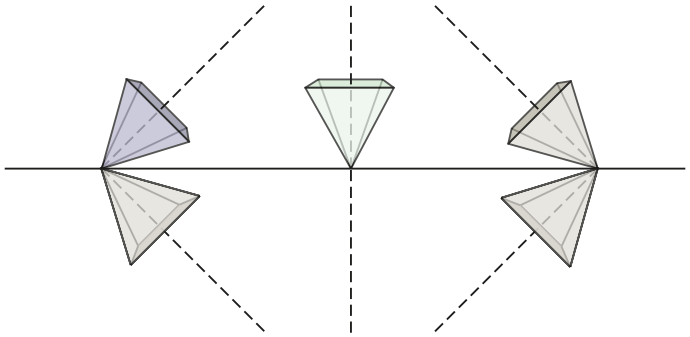


Figure 2.3: A camera (blue) is placed one unit away from its stereo partner (green). The cheiral ambiguity when recovering its pose from their essential matrix $\boldsymbol{E}$ means there are three additional false solutions (gray) for the pose of the camera.

where $[\boldsymbol{t}]_\times$ denotes the cross-product matrix of $\boldsymbol{t}$. Fortunately, $\boldsymbol{t}$ and $\boldsymbol{R}$ can be determined from $\boldsymbol{E}$ alone, but only up to a scale. This means that $\boldsymbol{R}$ can be completely determined, since $|\boldsymbol{R}| = 1$, and that the direction of translation $\bar{\boldsymbol{t}} \stackrel{\text{def}}{=} \boldsymbol{t}/|\boldsymbol{t}|$ can be determined [7]. The relative pose is determined by

$$\begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix}^{c_2 c_1} = \begin{pmatrix} \boldsymbol{R}_2 & \boldsymbol{t}_2 \\ \boldsymbol{0} & 1 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{R}_1 & \boldsymbol{t}_1 \\ \boldsymbol{0} & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_2^\intercal \boldsymbol{R}_1 & \boldsymbol{R}_2^\intercal (\boldsymbol{t}_1 - \boldsymbol{t}_2) \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{2.14}$$

Specifically, the relative $\boldsymbol{R}$ and $\bar{\boldsymbol{t}}$ are determined by the singular vector decomposition of $\boldsymbol{E}$, as in Equation 2.15 [7]:

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal && \text{(2.15a)} \\ (\boldsymbol{u}_0 \ \boldsymbol{u}_1 \ \pm\bar{\boldsymbol{t}}) &= \boldsymbol{U} && \text{(2.15b)} \\ \pm\boldsymbol{R} &= \boldsymbol{U}\boldsymbol{R}_{\pm 90^\circ}\boldsymbol{V}^\intercal && \text{(2.15c)} \end{aligned}$$

In Equation 2.15a, the diagonal matrix $\boldsymbol{\Sigma} = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{smallmatrix}\right)$ in the ideal case where there is no noise, and when noise is present, $\bar{\boldsymbol{t}}$ is selected as the vector with the smallest corresponding singular value [7]. Because the scale is lost in $\boldsymbol{E}$, so is also the sign, and as such there are two possible values for $\bar{\boldsymbol{t}}$ and four possible values for $\boldsymbol{R}$. These four solutions are illustrated in Figure 2.3. A *cheirality check* can then be performed to determine these signs. This involves looking at whether the points $\boldsymbol{Z}$ inferred from $\hat{\boldsymbol{Z}}$ and $\hat{\boldsymbol{Z}}'$ are both in front of the camera, and within its view [7], [8].

2.2.3 Perspective-n-Point

Essential matrix estimation is not sufficient for many applications of determining relative camera pose. In these cases *Perspective-n-Point* (PnP) is used instead. Here, in addition to $\boldsymbol{P}$, the world points $\boldsymbol{Z}$ are also assumed to be known, which introduces a known scale into the problem.

PnP is not formulated using epipolar geometry, since only one camera is concerned, and can simply be viewed as an optimization problem of minimizing the reprojection error $\boldsymbol{r}$, as defined by Equation 2.16:

$$\boldsymbol{r}_i = \|\pi(\widehat{\boldsymbol{C}\boldsymbol{z}_i}) - \boldsymbol{p}_i\|_2 \tag{2.16}$$

This minimization is done in different ways by different algorithms [47]. In this project, we consider an implementation using RANSAC for initialization (i.e. outlier rejection) and EPnP to solve the final n-point optimization.

The minimum number of points to determine a pose is 3 [7], [8], [47]. The problem of determining camera pose from a set of three points is referred to as P3P, and usually involves solving an 8th-degree polynomial with even exponents [48]. The RANSAC-based method samples random sets of three point pairs and counts the number of inliers—points whose reprojection error is below some threshold [49]. The maximal inlier sets are then optimized with EPnP, which is an efficient n-point solver that finds a near-minimum for the mean squared error $\texttt{mean}(\boldsymbol{r}^2)$ [48].

2.2.4 Homography

The extrinsic matrix of a camera is often written in homogeneous form, which is a 3D affine transformation matrix $\boldsymbol{T} \in \mathbb{R}^{4\times4}$. Similarly, a 2D affine transformation matrix $\boldsymbol{T} \in \mathbb{R}^{3\times3}$ translates, rotates, shears, and scales points in the plane, and has a total of 6 degrees of freedom. The generalization of this transformation is the *homography*, which utilizes all 8 possible degrees of freedom available for homogeneous coordinates, which breaks the invariance of parallel lines under affine transformations [7]. This transformation is also referred to as *projective*, since they are equivalent to projecting the points in a plane onto another plane in 3D [7].

In stereo vision, these are useful for (among other things) describing the projection of planar surfaces in the world onto the image plane, as well as describing the relation between two different image planes, for a region of 3D points that lie on a plane. Objects in the world are often not planar, but for some applications this can be a good enough estimation [17], [18], [26]–[29]. A homography is also a good approximation when the base-line is much smaller than the distance to the 3D points, where the homography can then be determined by the intrinsic and extrinsic camera parameters, using Equation 2.17, assuming the translative part of the relative camera pose $\approx \boldsymbol{0}$ [50].

$$\boldsymbol{\mathcal{H}} = \boldsymbol{K}_2\boldsymbol{R}_2\boldsymbol{R}_1^\intercal\boldsymbol{K}_1^{-1} \tag{2.17}$$

2.2.5 Image feature extraction

Algorithms such as PnP, triangulation algorithms, or the eight-point algorithm for essential matrix (and fundamental matrix) estimation assume that there is an al-

ready established correspondence of points $\boldsymbol{P}, \boldsymbol{P'}$ [8]. These points are in many applications automatically selected and described—either by a hand-crafted feature extractor such as SIFT [9], ORB [10], etc., or by a machine learning model such as SuperPoint [5]—and then matched by some matching algorithm [4], [5], [7], [8].

SIFT solves the detection problem by computing a Gaussian pyramid of the image—a series of iteratively sub-sampled and Gaussian blurred variants to the image—on which local optima (in three degrees of freedom) are considered detected interest points [9]. These points are then described in a way which is (roughly) invariant to scale*, rotation, and brightness, derived from the local surroundings of the interest point [9]. How the machine learning model SuperPoint approaches this is covered in Section 2.3.

2.3 SuperPoint and SuperGlue

SuperGlue and SuperPoint are two interfacing neural network models designed for automated extraction and matching image features between pairs of images.

2.3.1 Feature Extraction with SuperPoint

SuperPoint, by DeTone et al. [5], is a fully convolutional model, meaning that it scales well for different-sized images. It is designed to extract useful image features that are consistent between multiple camera views. We denote the SuperPoint model by a function $\mathcal{P} : \mathbb{R}_{[0,1]}^{s_x \times s_y} \to \mathbb{N}^{2\times k} \times \mathbb{R}^{k\times 256}$ in Equation 2.19a†.

Because of the difficulty of achieving true ground-truth for both interest points detection and matching points between images, the authors employ different methods for self-supervision. First, they trained a separate model for feature detection on exclusively synthetic data [51]. This data was constructed of various geometric shapes augmented with noise and could as such be automatically and efficiently labeled with interest points at intersections and corners. This model was then applied to random homographies of non-synthetic images, in a process they call *homographic adaptation*, in order to enforce invariance to (not consistency between) changes in perspective [5].

SuperPoint is (also trained to consistently describe the same feature points across homography transformations – already mentioned) uses cosine similarity to determine similarity between points, as described by Equation 2.18 [5]:

*To some degree also affine transformations in general

†We use set notation and vector notation interchangeably for simplicity. The order of points along the vector is arbitrary, but matches the order of its corresponding vector of descriptors.

$$\mathcal{L}_d(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{H}}) = \left(\frac{1}{(s_x/8)(s_y/8)}\right)^2 \sum_{i,j,i',j'} l(\boldsymbol{d}_{ij}, \boldsymbol{d}'_{i'j'}, \pi(\boldsymbol{\mathcal{H}}\underline{\boldsymbol{p}_{ij}}), \boldsymbol{p}_{i'j'}) \quad (2.18a)$$

$$l(\boldsymbol{d}, \boldsymbol{d}', \boldsymbol{p}, \boldsymbol{p}') = \begin{cases} \lambda_d \max\{0, m_p - \boldsymbol{d}_i \cdot \boldsymbol{d}_j\}, & \text{if } \boldsymbol{p} \approx \boldsymbol{p}' \\ \max\{0, \boldsymbol{d}_i \cdot \boldsymbol{d}_j - m_n\}, & \text{otherwise,} \end{cases} \quad (2.18b)$$

where $\pi(\boldsymbol{\mathcal{H}}\underline{\boldsymbol{p}})$ denotes a point $\boldsymbol{p}$ transformed by homography $\boldsymbol{\mathcal{H}}$, $\boldsymbol{d} : \mathbb{R}^{\frac{s_x}{8} \times \frac{s_y}{8} \times 256}$ are the descriptors of the original image $\boldsymbol{\mathcal{I}}$, and $\boldsymbol{d}'$ are the descriptors of the image transformed by $\boldsymbol{\mathcal{H}}$. The weighting term λ_d, as well as the (positive and negative) margins of the hinge loss, m_p and m_n, are hyperparameters, s_x and s_y denote the width and height of $\boldsymbol{\mathcal{I}}$, and indices i and j index the 8×8 px subdivisions of $\boldsymbol{\mathcal{I}}$. These subdivisions of the image are used to construct the feature descriptors, and allow at most one interest point.

The subdividing of the image is a result of the pooling layers in the VGG-like [52] encoder (see Figure 3.1 or [5] for an overview of the structure). To counteract the inaccuracy that this introduces, the network does two things. First, the *interest point decoder* is applied to the encoded feature map to recover the pixel-level positions of the interest points instead of performing upsampling. This achieved by applying a convolutional layer with $8 \times 8 + 1 = 65$ channels (one extra channel for the "no interest point" dustbin) followed by an explicit decoder also known as pixel-shuffle. Second, the *descriptor decoder* performs a convolves the encoded feature map to obtain feature descriptors followed by a bicubic interpolation, according to where the feature point is deemed to be located by the point decoder [5].

2.3.2 Feature Matching with SuperGlue

SuperGlue [19] subsequently builds on top of the feature extraction method (in this work, we are solely concerned with the SuperPoint feature extractor) to compute matches between the sets of points in the two images. In the matching context (see Section 2.2.5), there is generally no inherent order between image features. As such, Sarlin et al. [19] argue that graph neural networks (see Section 2.1.2) are well suited for the matching of image features.

The SuperGlue model alternates between *self-attention* and *cross-attention* between layers, meaning that in odd-numbered layers, descriptors are embedded using the attention graph of their respective images only, and in even-numbered layers, a bipartite graph between the two images is used instead. This is motivated by the need to embed information into each descriptor about the rest of its image, as well as the need to compare descriptors between the images [19].

We denote the SuperGlue model by a function $\mathcal{G} : (\mathbb{N}^{2\times k_1} \times \mathbb{R}^{k_1 \times 256}, \mathbb{N}^{2\times k_2} \times \mathbb{R}^{k_2 \times 256}) \to (\mathbb{N}^{2\times k_1}, \mathbb{N}^{2\times k_2}, \mathbb{N}^{2\times n} \times \mathbb{N}^{2\times n})$, where $n \leq k_1, k_2$, such that composing $\mathcal{P}$ and $\mathcal{G}$ maps a pair of images to a bipartite graph of interest points, as shown in Equations 2.19b–c:

$$\mathcal{P}(\boldsymbol{\mathcal{I}}) = \left\{ (\boldsymbol{p}, \boldsymbol{d}) \in \mathbb{N}^2 \times \mathbb{R}^{256} \,\middle|\, \begin{array}{l} \boldsymbol{p} \text{ is the location of a feature in } \boldsymbol{\mathcal{I}}, \\ \boldsymbol{d} \text{ describes the feature at } \boldsymbol{p} \end{array} \right\} \tag{2.19a}$$

$$\mathcal{G}(\mathcal{P}(\boldsymbol{\mathcal{I}}_1), \mathcal{P}(\boldsymbol{\mathcal{I}}_2)) = (P_1, P_2, E_{1\sim 2}) \tag{2.19b}$$

$$E_{1\sim 2} = \{(\boldsymbol{p}_1, \boldsymbol{p}_2) | \boldsymbol{p}_1 \in P_1 \wedge \boldsymbol{p}_2 \in P_2\} \tag{2.19c}$$

SuperGlue is trained in a supervised manner using pseudo-ground truth labeling of the correspondences obtained from the estimated poses and depth maps It is optimized by maximizing the log-likelihood of the estimated adjacency matrix $\boldsymbol{A}$ for points expected to be matched and the log-likelihood of $(1 - \boldsymbol{A}_i \cdot \mathbf{1})$ for those not expected to be assigned a match [19].

3

Methods

This section will detail the approaches taken in this project, what tools were used, and how tasks were carried out. This includes processing of pose and image data, construction of pseudo-ground truth, model design, and training.

3.1 Model design

A combination of the SuperPoint feature extractor and SuperGlue matcher have been demonstrated [19], [53] to work well on real outdoor scenes and have a useful geometric model of the world. As such, we would like to preserve this in a derivative cross-modal network. The most straightforward approach would likely be to fine-tune the SuperPoint and SuperGlue networks on the target dataset; however, because we are trying to make the model perform the additional task of converting image features to a common feature space, we run the risk of disturbing their internal models and biasing them to the new data (i.e. *catastrophic forgetting* [54]). With this in mind, a natural way to approach the problem would be to instead train an interfacing model or add additional trainable layers to the two models.

There are several points in the SuperPoint-SuperGlue pipeline at which trainable layers could be inserted. Since SuperGlue and SuperPoint are trained on visual images exclusively, it would be natural to train only SuperGlue and the instance of SuperPoint that processes the thermal image. Equation 3.1 gives the general form of such a model, where $\hat{\mathcal{G}}$ and $\hat{\mathcal{P}}$ denote modified and/or fine-tuned instances of SuperGlue and SuperPoint, respectively, and $\mathcal{M}$ denotes a separate model that interfaces between SuperPoint and SuperGlue.

$$(P_{\text{Vis}}, P_{\text{IR}}, E_{\text{Vis}\sim\text{IR}}) = \hat{\mathcal{G}}(\mathcal{M}(\mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{Vis}})), \hat{\mathcal{P}}(\boldsymbol{\mathcal{I}}_{\text{IR}}))) \tag{3.1}$$

Another possible general architecture is based on training image translation models $\mathcal{T}_{\text{Vis}}, \mathcal{T}_{\text{IR}}$, shown below

$$(P_{\text{Vis}}, P_{\text{IR}}, E_{\text{Vis}\sim\text{IR}}) = \hat{\mathcal{G}}(\mathcal{M}(\hat{\mathcal{P}}_{\text{Vis}}(\mathcal{T}_{\text{Vis}}(\boldsymbol{\mathcal{I}}_{\text{Vis}})), \hat{\mathcal{P}}_{\text{IR}}(\mathcal{T}_{\text{IR}}(\boldsymbol{\mathcal{I}}_{\text{IR}})))) \tag{3.2}$$

In that case, the models $\mathcal{T}_{\text{Vis}}$ and $\mathcal{T}_{\text{IR}}$ would try to translate directly to the visual domain (in which case $\mathcal{T}_{\text{Vis}} = \boldsymbol{\mathcal{I}} \mapsto \boldsymbol{\mathcal{I}}$), or instead try to map both image modalities to a new common modality, in which case $\mathcal{T}_{\text{Vis}} = \boldsymbol{\mathcal{I}}_{\text{Vis}} \mapsto \boldsymbol{\mathcal{I}}_{\text{Vis+IR}}$ and $\mathcal{T}_{\text{IR}} = \boldsymbol{\mathcal{I}}_{\text{IR}} \mapsto \boldsymbol{\mathcal{I}}'_{\text{Vis+IR}}$ and Vis+IR refers to a common modality or latent representation, where $\boldsymbol{\mathcal{I}}_{\text{Vis+IR}}$ and $\boldsymbol{\mathcal{I}}'_{\text{Vis+IR}}$ should ideally have indistinguishable distributions.

The former image translation approach was recently tried for image registration and shows some promise [17]. However, because Ran Zhang et al. [17] use paired data, dense correspondences are essentially assumed. This can be a problem since there is parallax between the pair of images, and may result in less accurate placement of features or biasing the model to a certain base-line. The exact placement of features is important in camera calibration and pose estimation [47], [55], [56], so this is likely not an optimal approach. An alternative to this is to use an unpaired method such as CycleGAN [57].

The choice of architecture for $\mathcal{M}$ can be divided into two groups: (1) a model applied to each separate descriptor—either restricted to one of the modalities, or by applying one version to each modality—which does not take the relation of different features in the image into account. A suitable architecture for this kind of model would be a simple MLP, or some variation on this. The relation between features in the same image is likely important also for translating them between modalities, which motivates using (2) a model applied to the set of descriptors for an isolated modality; i.e. using a GNN model. Unlike the GNN of SuperGlue, this model would not consider cross-attention. Such a network could also be conceivable, but would limit any options for pre-training of $\mathcal{M}$ and would no longer separate feature translation and feature matching. We will consider both the per-feature model and the feature graph model.

3.1.1 SuperPoint modification

The general structure of SuperPoint is shown in Figure 3.1 along with added trainable layers. The illustrative layers shown represent convolutional layers, and reduction in size represents the downsampling layers in the network. It is quite easy to imagine training additional layers either preceding, succeeding, or interleaving the original layers in any of the three sub-modules of SuperPoint, as suggested in Figure 3.1.

The effect of these different layers would however be different. Like mentioned earlier in Section 3.1, the layers preceding the encoder of SuperPoint would be essentially equivalent to an image translation module $\mathcal{T}$ (and can as such be disregarded in $\hat{\mathcal{P}}$). Layers succeeding the encoder would serve to transform feature maps sampled from the LWIR modality. Similarly, interleaving layers could serve to do a similar transformation, but at different levels of information abstraction and feature scale.

Modifications to the interest point decoder would probably be unnecessary, assuming that the encoder can encode features sampled in LWIR in the same way it encodes features from the visual modality, and that interest point locations do not change

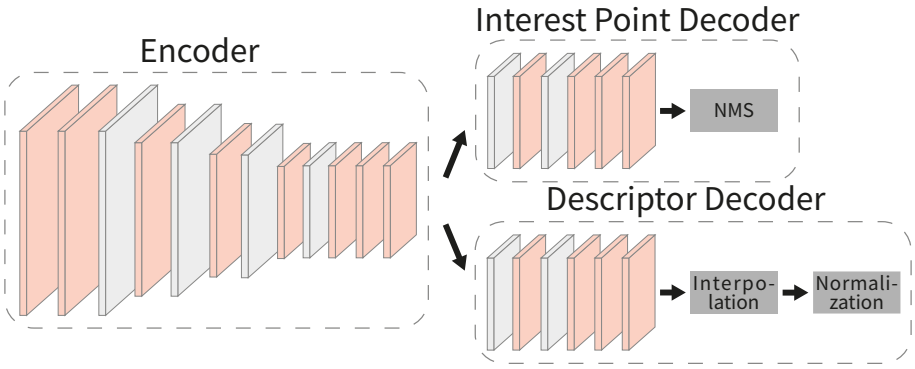


Figure 3.1: Illustration of structural modification of SuperPoint. Layers in gray represent those in the original network, while those in red represent (possible) added layers. The exact number of layers is not accurate.

considering the basic geometric shapes the model is pre-trained on. Even though the exact location of interest points is very important for applications such as camera calibration, it only has a small effect on how points are matched in SuperGlue. As such, it is likely better not to modify the interest point decoder. Furthermore, adding trainable layers to this module increases the risk that SuperPoint falls into one of the failure modes of the training (see Section 4.6.1).

Modifying the descriptor decoder is probably much more productive, since these are the tensors that carry on through SuperGlue, and are subsequently used to calculate the loss. Much like the layers preceding the encoder, the layers succeeding the descriptor decoder can be thought of as being part of the interfacing model $\mathcal{M}$. If $\mathcal{M}$ is applied element-wise on the set of descriptors, these two formulations are equivalent.

In summary, this leaves $\hat{\mathcal{P}}$ with added layers interleaved in the encoder and description decoder, in addition to the models $\mathcal{T}$ and $\mathcal{M}$.

3.1.1.1 SuperPoint loss

An issue with modifiying SuperPoint is that it is not trivial how to train it without completely reimplementing the training pipeline used by DeTone et al. [5], which would be impractical. Instead, both end-to-end training and a simplified training pipeline was tried.

In the simplified pipeline the pseudo-ground truth derived with the methods in Section 3.3.1 and Section 3.3.2 was used. Because the recall is limited in the pseudo-ground truth, a triplet loss is used instead of the loss in Equation 2.18, used by DeTone et al. [5]. This places less negative weight on what the ground truth estimation considers mismatches. This loss is given by

$$\mathcal{L} = \sum_{(\boldsymbol{p},\boldsymbol{d})} l(\boldsymbol{d}, \boldsymbol{d}', \boldsymbol{p}, \boldsymbol{p}') + l(\boldsymbol{d}, \boldsymbol{d}'', \boldsymbol{p}, \boldsymbol{p}''), \text{where } \boldsymbol{p} \approx \boldsymbol{p}' \text{ and } \boldsymbol{p} \neq \boldsymbol{p}'' \tag{3.3}$$

where l refers to the function defined in Equation 2.18, and the sum iterates over each point $\boldsymbol{p}$ with matching point $\boldsymbol{p}'$, and samples an unsuitable match $\boldsymbol{p}''$.

3.1.2 Weight initialization

When training layers in SuperPoint with end-to-end training it is particularly important to initialize new layers with weights that correspond to an identity transform of the feature map. Small perturbations to the weights can have a large impact on how points are described, and consequently quickly lead to interest points no longer being extracted, or non-interest points appearing. Since there is no gradient between the binary selection of points and the output tensors of SuperPoint, the training is unlikely recover from this.

Because of how the attention mechanism is formulated in GNNs, there is no way initialize the weights of $\mathcal{M}$ such that $\mathcal{M} = \boldsymbol{d} \mapsto \boldsymbol{d}$, without introducing some kind of skip connector. This can make the training a bit more challenging (using regular random initialization), but it does not suffer the same problems as the trainable layers in $\mathcal{P}$ do, since no gradient information is lost between $\mathcal{M}$ and the score matrix of $\mathcal{G}$. This makes a random weight initialization scheme a feasible approach.

3.2 Data

The data used in this project was collected by Saab prior to the start of this project. The positional data (from IMU and GPS) was processed with a Kalman filter and expressed in the form of drone poses. The relative pose (including base-line scale) between the two cameras was calibrated before each flight; see Section 1.1. Both positional data and image data was saved with a frequency of 10Hz, and were synchronized in time.

The pose data consists of per-frame *absolute poses* and *relative poses* of the calibrated system. Absolute poses are the camera poses at which each of the pairs of images were taken. Camera positions were expressed in the *Earth-centered, earth-fixed coordinate system* (ECEF) coordinates in meters and camera orientations were expressed with Euler angles relative to the ECEF reference frame (i.e., as $\mathcal{C}^{ce}$). Furthermore, Euler angles used yaw-pitch-roll order, which is equivalent to intrinsic rotations around axes Y-X-Z in the OpenCV formulation [49]. Conversions between representations of rotations were made with the spatial transform package from SciPy.

Relative poses between the cameras were expressed in the visual camera's reference frame. It was considered fixed throughout each series of images and manually calibrated before each flight using a thermal checkerboard (see Section 4.1.1.1). Due to the difficulty of determining the camera poses relative to the IMU, this offset was assumed to be zero, which could have an impact when working with drone poses with a significant relative difference in rotation.

3.2.1 Data filtering

In order to sift out bad data that could have a negative impact on the results, two separate measures were employed.

First, for any images close to the ground it would be more difficult to ensure good ground-truth correspondences because the parallax would be much larger, occlusion would become a bigger problem, cameras would sometimes be out of focus, and the information useful for feature matching present would be of much more varying quality. Because ECEF coordinates were used, it is easy to simply exclude images below a certain threshold, relative to the lowest position recorded throughout the flight.

The second measure was to filter out images containing a low amount of useful information in the infrared modality. This was mainly a problem where images contained a lot of water, and when the temperature was too low for thermal noise not to drown out any useful information. In order to target specific images within a series, some kind of heuristic is needed. The heuristic chosen is based on the Shannon entropy of the image, and is described by Larkin [58]. Larkin extends the notion of image entropy to two dimensions, however, because of the fixed-pattern noise (FPN) in the infrared images, the one-dimensional formulation was found to work better. Additionally, this step had to be performed on distorted images in order to preserve the FPN invariance of the entropy estimator. The equation used for image entropy is given by

$$H(\boldsymbol{\mathcal{I}}) = -\sum_{\mathbb{i}=0}^{2^{16}-1} \Pr\left(\frac{\partial \boldsymbol{\mathcal{I}}(i,j)}{\partial j} = \mathbb{i}\right) \log_2\left(\Pr\left(\frac{\partial \boldsymbol{\mathcal{I}}(i,j)}{\partial j} = \mathbb{i}\right)\right) \tag{3.4}$$

where the upper bound of summation is the maximum discrete gradient, which equals the maximum pixel value.

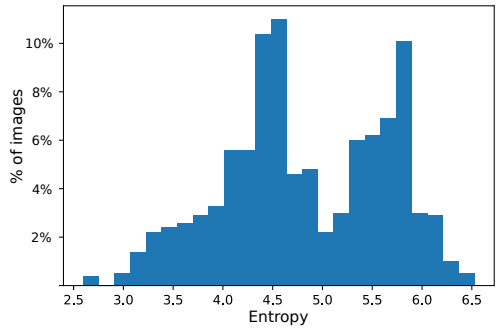
When displaying the distribution of image entropy over the dataset (see Figure 3.2), we see that it is distinctly bimodal, where the mode with high entropy predominantly consists of images of mostly water or other* low-detail images, and the mode with lower entropy consists of images with little to no water.

3.2.2 Image processing

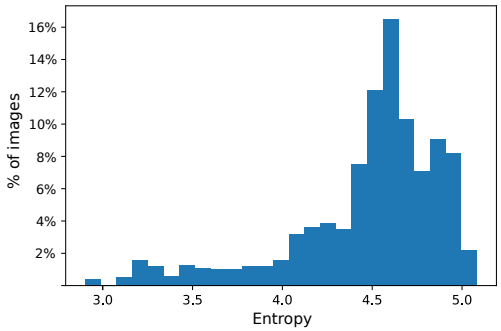
Since CNNs can generally be applied to any image size (that is sufficiently large), we are free to choose the resolution of our images. However, both inference and training has a quadratic† computational complexity [59], so small images are preferred. Additionally, GNNs are, as mentioned in Section 2.1.2, quadratic in terms of the number of nodes—i.e. feature points in this case. On the other hand, image features clearly decline in visibility with lowered resolution. In order to keep maximal information retention while reducing the image size, scaling factors $2^{-n}k$ were used. It

*These include images of empty fields, images with high low-frequency FPN noise, etc.

†The computational complexity of a 2D convolution operation is on the order of $\mathcal{O}(MNK^2C^2)$ for a feature map of dimension $M \times N$, kernel size $K \times K$, and channel depths C [59].



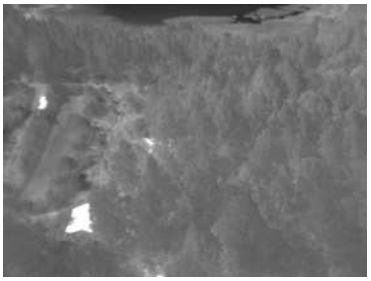
(a) Entropy density among a random set of 1000 thermal images, sampled from the full dataset.



(b) Entropy density among a random set of 1000 thermal images, sampled from only image series not showing any bodies of water.

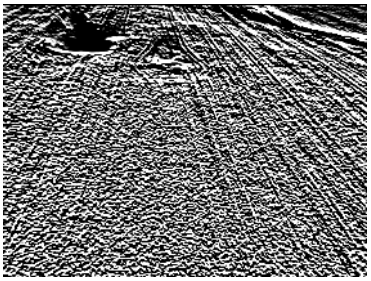
Figure 3.2: Two histograms showing the entropy density of thermal images in the dataset, according to the entropy definition given in Equation 3.4. In (a) the distribution has two distinguishable modes that appear to correspond with images containing a lot of water, or by other means lacking in useful detail. This reinforced by (b) appearing to have only the lower entropy mode, and that sampling images with entropy $H(\boldsymbol{\mathcal{I}})$ either below or above 5.1 appears to separate the two classes well.

(a) Image with $H(\boldsymbol{\mathcal{I}}) < 3$.



(b) Image with $H(\boldsymbol{\mathcal{I}}) \approx 4.5$.

(c) Image with $H(\boldsymbol{\mathcal{I}}) > 6.5$.

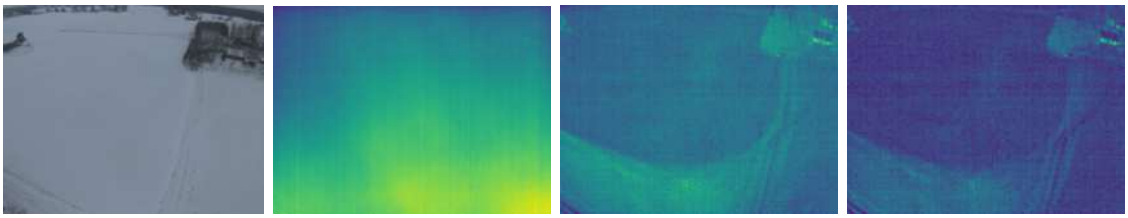


(d) Gradient image of (a).

(e) Gradient image of (b).

(f) Gradient image of (c).

Figure 3.3: Three randomly selected images from different intervals of the entropy (as defined by Equation 3.4) distribution and their respective vertical gradients. Image (a) is sampled among images with low entropy; its corresponding gradient image contains a lot of high frequency detail, whereas image (c) on the other end of the distribution has less such information. Image (b) is sampled from around the first peak seen in the entropy distribution (Figure 3.2a). Due to the low amount of information available from the water, these images generally yield higher entropy.



(a) Visual image. (b) LWIR image. (c) Denoising with constant model. (d) Denoising with linear model.

Figure 3.4: An image pair from a cold winter day shown in (a–b). All useful information is seemingly drowned out in the LWIR image by low- and high-requency FPN noise, but a significant amount of information is shown to be recoverable with the two different noise models in (c–d). All three thermal images are normalized to the same interval, and we can see that higher contrast is achieved for the linear model. The lesser pronounced high-frequency FPN noise is also noticeable as they are only as pronounced as any perceived horizontal lines in the noise.

was found that reducing the size of the thermal image quickly diminished the quality and numerosity of intra-domain matches, and was as such kept at the original image size. The visual image was reduced to 50 % of its original size, placing it at around the same pixel density as the thermal image. Keeping the angular size of pixels roughly the same also probably helps in matching features (this is supported by the findings in Section 4.3). Reducing the size of the visual image further had the same effect as for the thermal image, although less severe.

A subset of the thermal images was heavily affected by thermal noise, which appeared to void these images of any useful information. However, these images did still retain a lot of information about the scene due to the high bit depth, as can be seen in Figure 3.4. It was found that by modeling the noise as a constant term or linear term (as described in Equation 3.5, where t_0 is the time of the most recent NUC) in time t for some interval of t that does not contain NUC, the noise could be isolated and removed. The noise was isolated by assuming that the pixel-wise minimum over time of the true images would be roughly homogeneous and have small values: $\min \hat{\boldsymbol{\mathcal{I}}} \approx \epsilon \mathbf{1}$. Using Equation 3.5 as a model, $\boldsymbol{\mathcal{E}}$ can be estimated as $\min \boldsymbol{\mathcal{I}}/(t - t_0)$.

$$\boldsymbol{\mathcal{I}} = \hat{\boldsymbol{\mathcal{I}}} + (t - t_0)\boldsymbol{\mathcal{E}} \tag{3.5}$$

An example of this denoising is shown in Figure 3.4. Additional examples of successful and less successful noise removal is shown in Appendix B, with the original image ($\boldsymbol{\mathcal{I}}$) to the left and processed image ($\hat{\boldsymbol{\mathcal{I}}}$) to the right. In the second example, the method fails because some regions of pixels in $\boldsymbol{\mathcal{E}}$ do not reach ambient levels, and consequently leaves an afterimage from bright features.

3.3 Pseudo-ground truth construction

Obtaining true ground truth for point correspondences in images is generally impracticable. As such, many networks—such as SuperPoint [5]—employ some kind of pseudo-ground truth as a form of supervision (see Section 2.3). Others use data estimated by SLAM or stereo depth estimation techniques [19]. As there are no freely available multi-modal datasets that we can use for this, we must instead resort to self-supervision (similar to that in [5]) or weak supervision (like done in [44]).

Two forms of pseudo-ground truth were tried in this project. The first method (Section 3.3.1) uses weak supervision and assumes the 3D points associated with interest points in the image lie on a plane, and associates a corresponding homography between the image planes. The other method uses a triangulation scheme to find corresponding points. Although the triangulation scheme is seemingly more accurate, the homography-based method was found to yield significantly more point matches, while keeping a reasonably high accuracy, and was therefore more suited for training. Additionally, the triangulation-based method was a lot slower to run, and could not be integrated into the training pipeline. Instead, it was run on the dataset as a pre-processing step.

3.3.1 Homography-based ground truth estimation

This method does not depend on absolute nor relative poses from the dataset, and so does not utilize the data fully. It is however both quicker to run, easier to implement, and yields many matches.

The SuperPoint and SuperGlue pipeline was run on bimodal pairs of images with network weights pre-trained on the MegaDepth [60] "outdoor" dataset, SuperGlue matching threshold set to 0.7, SuperPoint keypoint threshold set to 0.0003, and the maximum number of matched point pairs set to 1024, yielding putative matches $E_{\text{Vis}\sim\text{IR}}$, as given by Equation 3.6. This assumes that the baseline matcher already works somewhat well also cross-modally.

$$(\mathbf{P}_{\text{Vis}}, \mathbf{P}_{\text{IR}}, \mathbf{E}_{\text{Vis}\sim\text{IR}}) = \mathcal{G}(\mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{Vis}}), \mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{IR}})) \tag{3.6}$$

Weak supervision is applied to the putative matches, similar to Yi et al. [44] and Pang et al. [42], by first estimating a homography between the matched points of the two images using RANSAC and keeping only the inlier matches. This assumes that the 3D points in the scene all lie in a plane, which can be a reasonable approximation in our case, when far enough off the ground.

3.3.2 Triangulation-based ground truth estimation

Given that we have good matches within one modality, we can potentially achieve pseudo-ground truth correspondences with much higher accuracy by utilizing the known camera poses and use triangulation to determine expected locations of points

in other image planes. Reprojection errors can then be utilized to exclude outliers to further improve precision.

The SuperPoint and SuperGlue pipeline was run on pairs of images within the same modality in order to find good matches. Network weights pre-trained on the were used. The SuperPoint interest point threshold was set to zero, while the SuperGlue matching threshold was set to 0.8 and the maximum number of matched point pairs was set to 1024. This gives more confident matches, but requires more processing time and memory. The pre-trained SuperPoint and SuperGlue models work quite well in the visual modality out of the box, but worse on thermal images, as expected. This makes it more important to adjust the networks to find as many points as possible in order to find enough confident inter-modal matches.

The matching in the two modalities is described by

$$\begin{aligned}(\mathbf{P}_{\text{Vis}}^{(1)}, \mathbf{P}_{\text{Vis}}^{(2)}, \mathbf{E}_{\text{Vis}\sim\text{Vis}}^{(1\sim2)}) &= \mathcal{G}(\mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{Vis}}^{(1)}), \mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{Vis}}^{(2)})) \\ (\mathbf{P}_{\text{IR}}^{(1)}, \mathbf{P}_{\text{IR}}^{(2)}, \mathbf{E}_{\text{IR}\sim\text{IR}}^{(1\sim2)}) &= \mathcal{G}(\mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{IR}}^{(1)}), \mathcal{P}(\boldsymbol{\mathcal{I}}_{\text{IR}}^{(2)})),\end{aligned} \tag{3.7}$$

where Vis and IR denote the visual and LWIR modalities, respectively, and $^{(t)}$ denotes a point in time, for some $t \in \mathbb{N}$.

After finding putative intra-modal matches, these are triangulated using OpenCV's `triangulatePoints`. The resulting 3D points are then projected onto their corresponding image from the opposite modality. If after being projected the point has no intra-modal neighbors, they are rejected. Finally, if nearest-neighbor matches do not correspond to the same points between modalities, these points are also rejected. The resulting four sets of points are all associated by bijections.

The triangulation is described by

$$\boldsymbol{Z}_{\text{Vis}} = \tau(\boldsymbol{C}_{\text{Vis}}^{(1)}, \boldsymbol{C}_{\text{Vis}}^{(2)}, \boldsymbol{p}_{\text{Vis}}^{(1)}, \boldsymbol{p}_{\text{Vis}}^{(2)}) \tag{3.8a}$$

$$\boldsymbol{Z}_{\text{IR}} = \tau(\boldsymbol{C}_{\text{IR}}^{(1)}, \boldsymbol{C}_{\text{IR}}^{(2)}, \boldsymbol{p}_{\text{Vis}}^{(1)}, \boldsymbol{p}_{\text{Vis}}^{(2)})$$

$$\hat{\boldsymbol{Z}}_{\text{Vis}\to\text{IR}} = \pi\left(\boldsymbol{C}_{\text{IR}}\underline{\boldsymbol{Z}_{\text{Vis}}}\right) \tag{3.8b}$$

$$\hat{\boldsymbol{Z}}_{\text{IR}\to\text{Vis}} = \pi\left(\boldsymbol{C}_{\text{Vis}}\underline{\boldsymbol{Z}_{\text{IR}}}\right),$$

where $\hat{\boldsymbol{z}}_{\text{A}\to\text{B}}$ denotes a point triangulated in modality A, and reprojected in modality B, at the same point in time.

If a pair of intra-modal matches, corresponding to the same 3D point $\boldsymbol{z}$, and if the triangulation is accurate enough, the reprojected points should then approximate their corresponding points $\boldsymbol{P}$, as given by:

$$\boldsymbol{p}_{\text{Vis}} \approx \pi\left(\boldsymbol{C}_{\text{Vis}}\underline{\boldsymbol{z}}\right) \wedge \boldsymbol{p}_{\text{IR}} \approx \pi\left(\boldsymbol{C}_{\text{IR}}\underline{\boldsymbol{z}}\right) \quad \Longrightarrow \quad \hat{\boldsymbol{z}}_{\text{Vis}\to\text{IR}} \approx \boldsymbol{p}_{\text{IR}} \wedge \hat{\boldsymbol{z}}_{\text{IR}\to\text{Vis}} \approx \boldsymbol{p}_{\text{Vis}} \tag{3.9a}$$

$$\boldsymbol{p}_{\text{Vis}} \approx \pi\left(\boldsymbol{C}_{\text{Vis}}\underline{\boldsymbol{z}}\right) \wedge \boldsymbol{p}_{\text{IR}} \approx \pi\left(\boldsymbol{C}_{\text{IR}}\underline{\boldsymbol{z}'}\right) \wedge \boldsymbol{z} \not\approx \boldsymbol{z}' \Longrightarrow \hat{\boldsymbol{z}}_{\text{Vis}\to\text{IR}} \not\approx \boldsymbol{p}_{\text{IR}} \vee \hat{\boldsymbol{z}}'_{\text{IR}\to\text{Vis}} \not\approx \boldsymbol{p}_{\text{Vis}} \tag{3.9b}$$

This follows from the epipolar constraint, as long as the intra-modal epipolar lines are not parallel to the inter-modal epipolar lines. To see why, consider the case shown in Figure 3.5. Given an interest point $\boldsymbol{p}$ with putative matching points $\boldsymbol{p}_1, \boldsymbol{p}_2$ in two other images, and assuming the epipolar lines $\overline{\boldsymbol{e}_1\boldsymbol{p}}$ and $\overline{\boldsymbol{e}_2\boldsymbol{p}}$ are not approximately parallel, there are two cases: either the two points $\boldsymbol{p}_1, \boldsymbol{p}_2$ correctly point to the same point in the scene—in which case $\boldsymbol{p}$ is correctly matched with $\boldsymbol{p}_1, \boldsymbol{p}_2$ if and only if it conforms with the two epipolar constraints*—or they do not match, in which case the probability that $\boldsymbol{p}$ conforms with the epipolar constraints is $4\omega^2 \sin\theta/(s_x s_y)$ where $4\omega^2 \sin\theta$ is the area of the region in which $\boldsymbol{p}$ would be accepted, $s_x s_y$ is the image area, ω is the accepted error from the epipolar line, and θ is the angle at which the two epipolar lines cross†.

Another potential way to check this cross-modal point correspondence is to directly compare the triangulated points in 3D space. While this would achieve the same effect without the extra computation of reprojection, one would instead have to account for different threshold values along different axes and at different distances from the camera, since the reprojection error is inversely proportional to the square of the distance between the camera and 3D point $\boldsymbol{z}$ [61].

In order for triangulation to yield accurate 3D point estimates, intra-modal pairs must be sufficiently far apart to create enough disparity (parallax) to be able to gauge the distance to the point. On the other hand, the camera scenes at the two different points in time must have sufficient overlap.

For each point along a series in the dataset, another point was selected in the same series that was estimated to be at a distance d apart, and have a relative change in rotation less than $\Delta\theta_{\max}$. Optimally, a method to determine the percentage overlap between scenes would be used, but for simplicity this method was opted for.

Assuming the drone is well above the ground, the distance d was chosen to be the temporally closest pose in the range $5 < d < 10$, and the maximum rotation difference was set to $\Delta\theta_{\max} = 10°$. Given two camera poses $\mathcal{C}^{ce}_{(1)}$, $\mathcal{C}^{ce}_{(2)}$, with corresponding affine transformation matrices $\boldsymbol{T}^{ec}$ as in Equation 2.7, the relative transformation $\boldsymbol{T}^{c_2c_1}$ is given by Equation 3.10a, and the distance d and angle $\Delta\theta$ is calculated from this with Equations 3.10b, 3.10c, where $\mathcal{R} : \mathbb{R}^{3\times3} \to \mathbb{R}^3$ represents the axis-angle (Rodrigues) representation of rotations.

*up to some error threshold

†for sufficiently large angles, since for very small angles the region of acceptance will extend outside the confines of the image

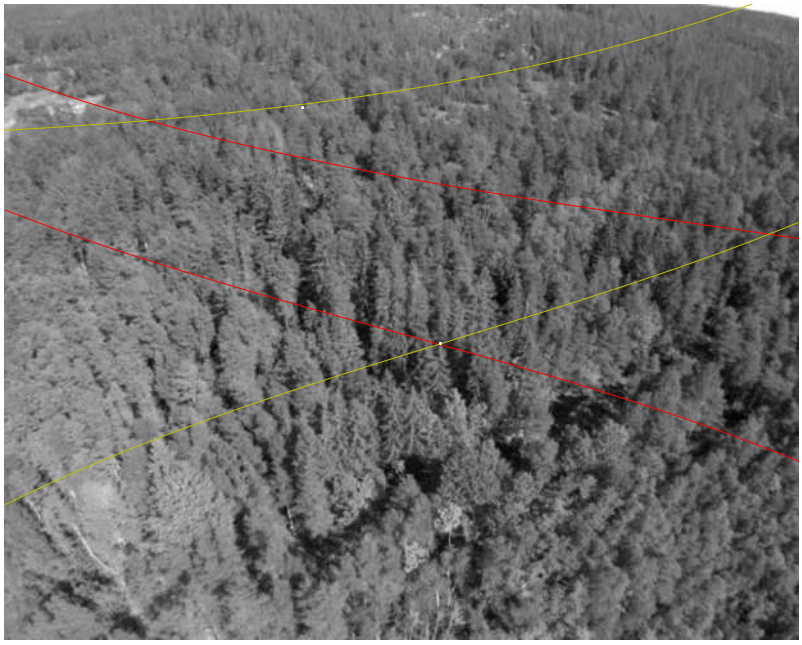


Figure 3.5: An image taken with the drone over a forested area (in the visual range), overlayed with a few selected epipolar lines of matched interest points from the corresponding LWIR image (yellow), and epipolar lines of matched interest points from the visual image in the subsequent time frame (red). Interest points are shown as white dots. Lines are warped according to the camera parameters. Near the center of the image there is a well matched interest point, as it lies within a very small margin of error from the corresponding intersection of epipolar lines. In contrast, the interest point shown towards the top of the image is not well matched, since it is far from its corresponding intersection, and can therefore be filtered out.

$$\boldsymbol{T}^{c_2c_1} = \begin{pmatrix} \boldsymbol{R}^{c_2c_1} & \boldsymbol{t}^{c_2c_1} \\ 0 & 1 \end{pmatrix} = \boldsymbol{T}^{ec}_{(2)}\boldsymbol{T}^{ce}_{(1)} = \boldsymbol{T}_{(2)}\boldsymbol{T}^{-1}_{(1)} \tag{3.10a}$$

$$d = |\boldsymbol{t}^{c_2c_1}| \tag{3.10b}$$

$$\Delta\theta = |\mathcal{R}(\boldsymbol{R}^{c_2c_1})| \tag{3.10c}$$

3.3.2.1 Alternative triangulation scheme

There are a couple of potential problems with the ground-truth estimation method described above. First, it relies on the assumption that the feature detectors, descriptors, and matchers used will work well in the infrared modality. This can especially be a problem for machine learning models, since they will have a bias towards the data it was trained on. Second, it relies on feature points being detected in the same locations as in the visual modality.

While the latter is difficult to mitigate without changing the feature detector/descriptor method, the former can be helped by relying more on the visual images.

In the method described above (Section 3.3.2), matching and triangulation is carried out in both modalities in parallel, yielding $\boldsymbol{Z}_{\text{Vis}}$ and $\boldsymbol{Z}_{\text{IR}}$. An alternative approach would be to do both triangulation and the reprojection consistency check in the visual modality, at different points in time, yielding instead $\boldsymbol{Z}^{1\sim2}_{\text{Vis}}$, which is reprojected onto $\boldsymbol{\mathcal{I}}^{(3)}_{\text{Vis}}$ and checked against $E^{(2\sim3)}_{\text{Vis}\sim\text{Vis}}$. This is a bit less restrictive, as we only have to perform triangulation for a single point set pair. On the other hand, this does not generally impose the same constraint of crossing epipolar lines, since consecutive camera poses usually lie on a line and do not change its rotation. As a result, epipolar lines from the two adjacent camera poses (with sufficiently large base-line), are close to being parallel. In Figure 3.5, we can see an example of this.

An example of the pseudo-ground truth matches generated by this alternative approach is shown in Figure 3.6.

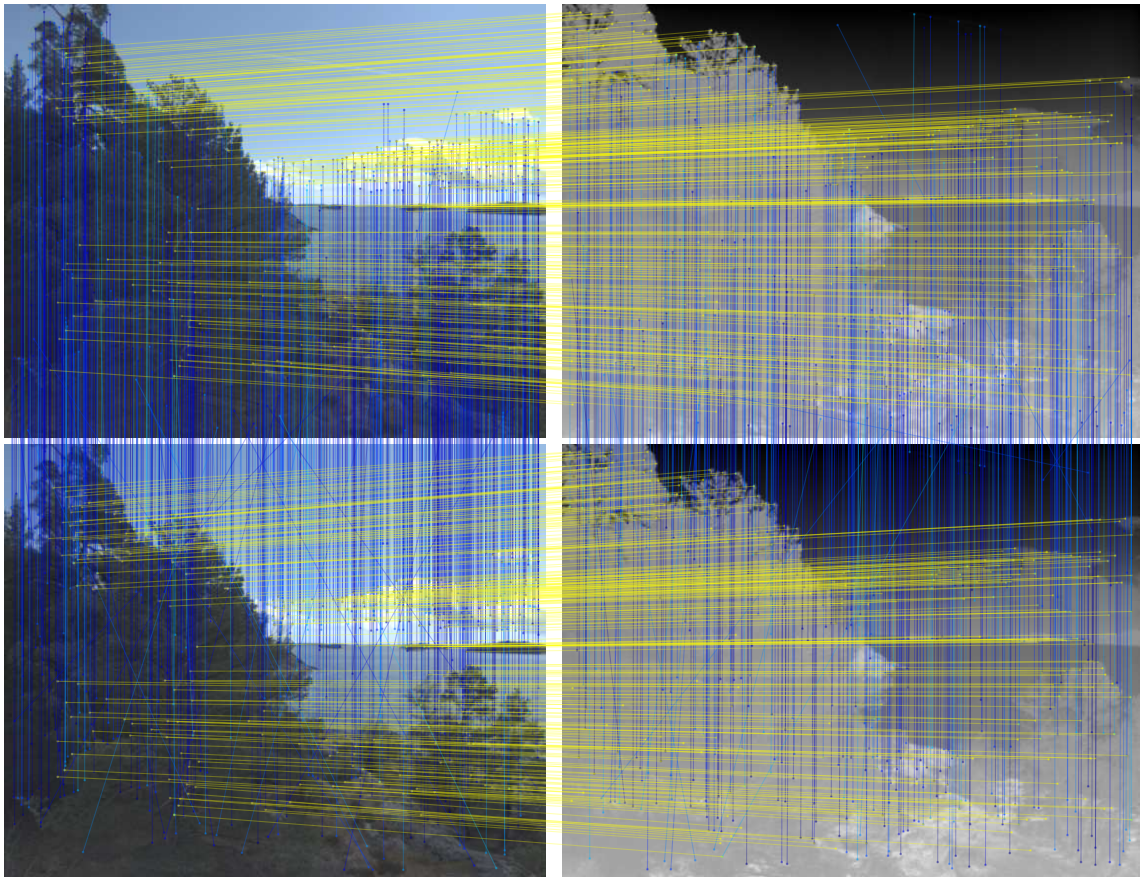


Figure 3.6: Two pairs of images over a forest. In blue hue (by match confidence), intra-modal SuperGlue matches are shown, which are then used to obtain the inter-modal matches (yellow), which are considered pseudo-ground truth.

4

Experiments

This section will present the results of experiments, evaluation metrics, and model predictions produced in this project, as well as discuss the effectiveness of the different methods tried. The first section of this chapter will present details of data preparation, selected models, and training. The second section of this chapter will detail how evaluation was set up. Section 4.3 will then cover some tangential results to the primary goal of this project. The subsequent sections will cover ground truth estimation and model evaluation, respectively.

4.1 Implementation details

The selected model design evaluated below is a feature translation model described by Equation 3.1 from Section 3.1, using a GNN architecture as described in Section 2.1.2. Each feature is represented by 256 32-bit floating-point numbers, and the model $\mathcal{M}$ uses four cross-attention layers in four attention heads. Optimization was done with the Adam optimizer with a learning rate of $\eta = 10^{-5}$ and weight decay of $\lambda_{wd} = 10^{-5}$.

The pseudo-ground truth used for training was that of the homography-based method. This is because it was found to perform better than the first method described in Section 3.3.2. The method later described in Section 3.3.2.1 is shown in Section 4.4 to likely yield better performance, and is suggested as the method of choice in further work, but was unfortunately not implemented into the training pipeline in this project due to time constraints.

4.1.1 Data preparation

Two datasets are used in evaluation. The dataset referred to as the "outdoor dataset" (or simply "dataset") is the primary data the model is exposed to (and constitutes the combined training and validation set). The other dataset, referred to as the "calibration dataset," is a small dataset collected indoors, for calibrating the relative ground truth camera pose (described in Section 4.1.1.1). This dataset is useful because it comprises a different data distribution—not only in terms of visual domain, but also in base-line—because we can utilize the calibration board pictured to achieve scale, and because we can assume the points on said board all lie in the

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

Figure 4.1: Selected sample from calibration dataset. Images (a–b) show an image pair with a well-defined board pattern. Images (c–d) show an image pair where the calibration board is overexposed in the visual domain.

same plane. A sample of this set is shown in Figure 4.1.

4.1.1.1 Camera Calibration

Camera calibration was performed before the start of the project, and will as such only be very briefly covered here for the purpose of providing a more complete context.

A thermal checkerboard was used both to establish correspondences between the two modalities and within each modality. A contrast filter was applied to the visual images in order to make the checkerboards appear more similar. First, the intrinsic camera parameters were calculated for each modality over a sample of images. Then, using these parameters for each calibration instance, bundle adjustment was used to determine the relative pose and base-line scale between the cameras. Unfortunately, the original unprocessed images were not saved, which means model inference on these images (described in Section 4.2) will not entirely reflect the performance of the intended distributions.

The reflection in the metal surface of the checkerboard resulted in some failure cases where the checkerboard pattern was no longer completely distinguishable. This is one of the downsides of using this kind of calibration method, and is yet another reason why a cross-modal matcher would be preferable. A well distinguished pair, as well as a less distinguished pair of images from the calibration dataset is shown in Figure 4.1.

4.1.1.2 Positional data pre-processing

The positional data for the captured images was used in the training pipeline in order to map points between the images, in order to generate the training data used. Because of the large difference in scale between relative and absolute poses, and because 32-bit floating-point numbers are favored in PyTorch, the origin of the ECEF reference frame was translated to the drone's starting position for each series of poses. It was found that if this was not part of the preprocessing (or alternatively, back-and-forth conversion between 32-bit and 64-bit tensors), a simple 3D affine transformation of a pose or a point would have an error of around 5cm. This error

would then have further accumulated in triangulation.

4.1.1.3 Image pre-processing

A couple of pre-processing steps were used for the image data. First, visual images were converted to a color channel format from the native Bayer layout, after which it was converted to grayscale using the luminance channel of its YUV encoding [62]. This would optimally be done in a single step, but was easier to implement as separate steps. This conversion is made because SuperPoint was designed for and trained on grayscale images.

Second, as introduced in Section 2.2.1, a non-standard model of radial lens distortion was used to describe the cameras. Because of this incompatibility with the OpenCV camera model, the code needed for distorting images had to be reimplemented based on the camera model used here.

After this initial processing, undistorted grayscale images that SuperPoint will accept are obtained.

4.2 Evaluation strategy

The model proposed in Section 3.1 is evaluated against a baseline model in Section 4.5. The baseline model considered in this report is the pre-trained SuperGlue and SuperPoint networks [63].

Due to the potential biases, misclassifications, and failure modes in the kind of self-supervised learning detailed in Section 3.3, evaluation using these pseudo-ground truth point correspondences is dubious at best, and should therefore only be used as a heuristic loss. Instead, as in previous work [12], [19], [20], the matches are used to estimate camera poses.

This is done in two ways: first by essential matrix estimation, and then by *Perspective-n-Point* (PnP)*. An essential matrix can be minimally estimated using the calibrated intrinsic parameters of both cameras and a set of at least five point pairs [7], [8]. The minimal five-point algorithm is used within *random sample consensus* (RANSAC) robust estimation algorithm [64] to find an essential matrix with a maximal number of inliers conforming with the matched points.

As mentioned in Section 2.2.2, the essential matrix can be computed from the relative pose between the cameras, and can similarly be decomposed to recover the relative pose, but only up to a scale. This means we can recover the relative rotation $\boldsymbol{R}_{\text{rel}}$ and the direction $\bar{\boldsymbol{t}} \stackrel{\text{def}}{=} \boldsymbol{t}/\left|\boldsymbol{t}\right|$ of the relative translation. In [12], [19], [20] the error $\Delta\theta$ between ground truth pose and estimated pose is defined as the maximum of the respective angular errors of rotation $\Delta\theta_{\boldsymbol{R}}$ and translation $\Delta\theta_{\boldsymbol{t}}$, as in Equation 4.1.

*It should be noted that a fiducial matcher is in general a better solution for this kind of problem. For the sake of having an evaluation method that estimates relative pose along with base-line, we use this somewhat unorthodox problem solution.

Because the accuracy of the estimated rotation is likely to be much better than the accuracy of the estimated translation when using a small base-line, both of these are presented in Section 4.5.1.

$$\Delta\theta_{\boldsymbol{R}} = |\mathcal{R}(\boldsymbol{R}_{\text{rel}}^{-1}\hat{\boldsymbol{R}}_{\text{rel}})| \tag{4.1a}$$

$$\Delta\theta_{\boldsymbol{t}} = \cos^{-1}(\bar{\boldsymbol{t}} \cdot \hat{\bar{\boldsymbol{t}}}) \tag{4.1b}$$

$$\Delta\theta = \max\{\Delta\theta_{\boldsymbol{R}}, \Delta\theta_{\boldsymbol{t}}\} \tag{4.1c}$$

Because we cannot determine the base-line scale along with the relative pose from an essential matrix, PnP is also used as an evaluation method. As introduced in Section 2.2.3, PnP assumes 3D points are available. These can either be estimated by triangulation—as is often done along with bundle adjustment in SLAM applications [3], [7]—or be provided a priori from a known object, such as a calibration board.

The goal in this evaluation is to estimate the scaled relative pose between the cameras from the cross-modal matches provided by the models. In order to use PnP to estimate ${}^{c_{\text{Vis}}}\mathcal{C}^{c_{\text{IR}}}$, the 3D points on the calibration board P_{C} in $\boldsymbol{\mathcal{I}}_{\text{IR}}$ must be known. As such, $(\mathbf{P}_{\text{IR}}, \mathbf{P}_{\text{C}}, \mathbf{E}_{\text{IR}\sim\text{C}})$ is predicted using the original SuperPoint/SuperGlue model, and $(\mathbf{P}_{\text{Vis}}, \mathbf{P}'_{\text{IR}}, \mathbf{E}_{\text{Vis}\sim\text{IR}})$ is predicted using the trained model, which is subsequently used to establish matches $E_{\text{Vis}\sim\text{IR}\sim\text{C}}$ that assign matches in the calibration board to the visual image, using intermediate matches $P_{\text{IR}} \cap P'_{\text{IR}}$.

Applying PnP to P_{IR}, P_{C}, and $E_{\text{Vis}\sim\text{IR}\sim\text{C}}$ yields an estimated camera pose ${}^{c_{\text{Vis}}}\mathcal{C}^{e}$. In order to find the (scaled) relative pose, we need to then apply PnP a second time to $(P_{\text{Vis}}, P_{\text{C}}, E_{\text{Vis}\sim\text{IR}\sim\text{C}})$ to find ${}^{c_{\text{IR}}}\mathcal{C}^{e}$, with which we find ${}^{c_{\text{Vis}}}\mathcal{C}^{c_{\text{IR}}}$.

As in [19], [20], the AUC (area under curve) of pose accuracy for thresholds @5°, @10°, and @20° is also calculated. This is calculated according to

$$AUC@\theta^\circ = \int_0^{\theta} \texttt{mean}\left(\begin{cases} 1, & \text{if } \Delta\theta_{\boldsymbol{R}} < \Delta\theta_{\boldsymbol{R}}^{\texttt{max}} \\ 0 & \text{otherwise} \end{cases}\right) \mathrm{d}\Delta\theta_{\boldsymbol{R}}^{\texttt{max}} \tag{4.2}$$

for the error in $\boldsymbol{R}$, and similarly for $\boldsymbol{t}$.

4.3 Note on fiducial matching with SuperGlue

In Section 4.5.1, pose estimation is performed by using PnP with a calibration board, which would normally be accomplished with a fiducial matching algorithm [65]. SuperGlue and SuperPoint can be used to find visual matches to arbitrary features in the calibration board, and indirectly accomplish a similar goal. To this end, matching was performed between images from one of the two cameras and a digital image of the calibration board.

Sarlin et al. [19] show that SuperGlue is able to find matches between large perspective transforms with high confidence. They also show that their method performs well on certain domain changes, such as matching between nighttime and daytime images. Additionally, they demonstrate that the models work well even when the image contains a lot of similarly looking features, such as for feature points in the pattern of a checkerboard. When performing matching between images and the calibration board in this project, it was, on the contrary, found that matches were very sensitive to both changes in rotation and scale. This is illustrated in Figure 4.2, where good matches could be obtained only when the images were (very roughly) pre-registered. Oftentimes, when only the rotation was different (as in Figure 4.2b), the network would find homographies for the board that were off by a multiple of 90°. By comparison, Sarlin et al. [19] present results that show confident matches on images of square tiles, with similar 90° changes in perspective. They also show similarly well-performing matches of images with a checkerboard, although without the large perspective change.

This indicates that the model can perform relatively well between new modalities or domains if other conditions are forgiving. But when for example perspective or lighting changes too much, it may fail completely. These kinds of problems are a common phenomenon with CNN-based feature matchers since CNNs are intrinsically translation invariant, but not invariant to other transformations such as rotation and scaling [66].*

Even though SuperGlue can find decent matches when given some help, these results further support the need for dedicated cross-modal or cross-domain matching models.

4.4 Ground truth estimation

The ground truth estimation methods tried were successful to varying degrees. Empirical accuracy measures of ground-truth estimates are difficult to construct. Statistics about other qualities that have effects on the reliability of the method can be used to get a picture of the quality of the pseudo-ground truth, while theoretical reasoning can hint at the accuracy of individual matches. The latter is discussed in Sections 3.3.1 and 3.3.2.

The density of matches is generally significantly greater for the homography-based method, since it does not rely on accurate triangulation and feature matching between additional images. Instead, it relies on a random sample consensus being a good estimate for a local homography, as well as the 3D feature point being recognized in both modalities, between only two images. On the other hand, the homography-based method will in many cases only be able to approximate a specific region in the image, where a plane can approximate the 3D points, which is not the case for the triangulation-based methods. It is also much more prone to errors but,

*This has inspired work in group equivariant neural networks for feature matchers among other areas [66].

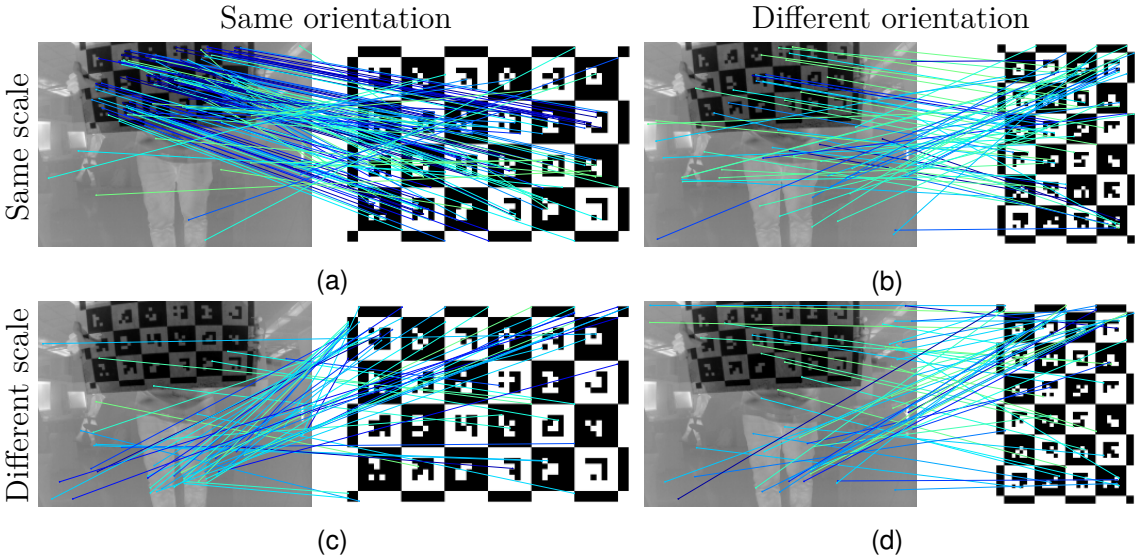


Figure 4.2: Infrared image matched with digital calibration board image. In (d) the board is not changed; in (b–c) the scale and rotation, respectively, are modified to approximate that of the board in the LWIR image. In (a) both scale and rotation are corrected for. The difference in scale is a factor of 2. Only (a) successfully matches the features in the board.

as mentioned, this is difficult to show empirically.

4.5 Benchmarks

The training graph of the trained model is shown in Figure 4.3. Validation loss is expectedly higher, but closely follows the downward trend of the corresponding training loss. Curiously, it was found that although the validation loss does not quickly diverge from the training loss, performance often deteriorated significantly if training was not stopped early in the convergence. This misalignment is likely an effect of bad matches in the pseudo-ground truth used for training, and the loss function regarding unknown true positive matches as incorrect. This is likely a major bottleneck in achieving better results, which would probably be partly remedied by a better ground-truth estimation like the method suggested in Section 3.3.2.1. However, despite this misalignment, transfer learning still has the potential to improve performance over the baseline by using an early stopping policy.

4.5.1 Pose estimation

In a similar fashion to how previous work on the topic has evaluated their methods [12], [19], [20], pose estimation was conducted using the point matches from the models. This was done both with the outdoor dataset used for training, and the set of images used for calibrating the cameras before each flight. The outdoor dataset was used since this is the kind of data the model is trained to work best with. The other is used because the distance between the cameras and the calibration board

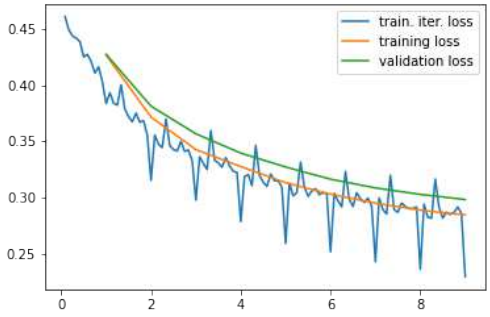


Figure 4.3: Training and validation loss against time (labeled by epoch).

is much smaller, and therefore generates greater disparity, which means noise will have much less of an impact on the estimates.

The baseline model achieved pose errors indicated by Table 4.1 using the essential matrix estimation method on the calibration image set. Due to varying quality of images in addition to noise and the errors and biases in the model, some image pairs yielded very low support for the essential matrix found by RANSAC. This high variance impacts the mean a lot, which can be seen by the relatively low median error for $\Delta\theta_{\boldsymbol{R}}$. In Figure 4.4, the error is plotted against support. We see that many image pairs ended up with low support, and consequently have a large variance (as also seen in Table 4.1), while the variance in error approach zero for image pairs with large support. Because of the small base-line, there is a large difference between the errors $\Delta\theta_{\boldsymbol{R}}$ and $\Delta\theta_{\boldsymbol{t}}$. For this reason both errors are presented, as opposed to how in previous work [12], [19], [20] the errors were assumed to be of similar magnitude.

The same procedure was performed with the trained model, which achieved errors in pose indicated by Table 4.1, for the calibration dataset, and Table 4.2 for the outdoor dataset. Similarly, the conditional distribution is compared to the baseline model in Figure 4.5.

Indoor	mean		median		std. dev.	
	baseline	trained	baseline	trained	baseline	trained
$\Delta\theta_{\boldsymbol{R}}$	4.28	**4.21**	**2.10**	2.61	18.4	**16.18**
$\Delta\theta_{\boldsymbol{t}}$	**40.65**	49.96	**16.11**	42.95	40.91	**39.29**
support	81.18	**100.08**	80	**89**	**53.89**	66.57

Table 4.1: Errors in pose—split into the errors in $\boldsymbol{R}$ and $\boldsymbol{t}$—listed in terms of their mean, median, and standard deviation, obtained using the calibration (indoor) dataset. The distribution of the support found by RANSAC also presented in the same way. The values are compared between the trained and baseline models, where the better value is highlighted by bold font.

The error is very low for $\Delta\theta_{\boldsymbol{R}}$ compared to $\Delta\theta_{\boldsymbol{t}}$, which is expected. However, there is no visible improvement of the trained model.

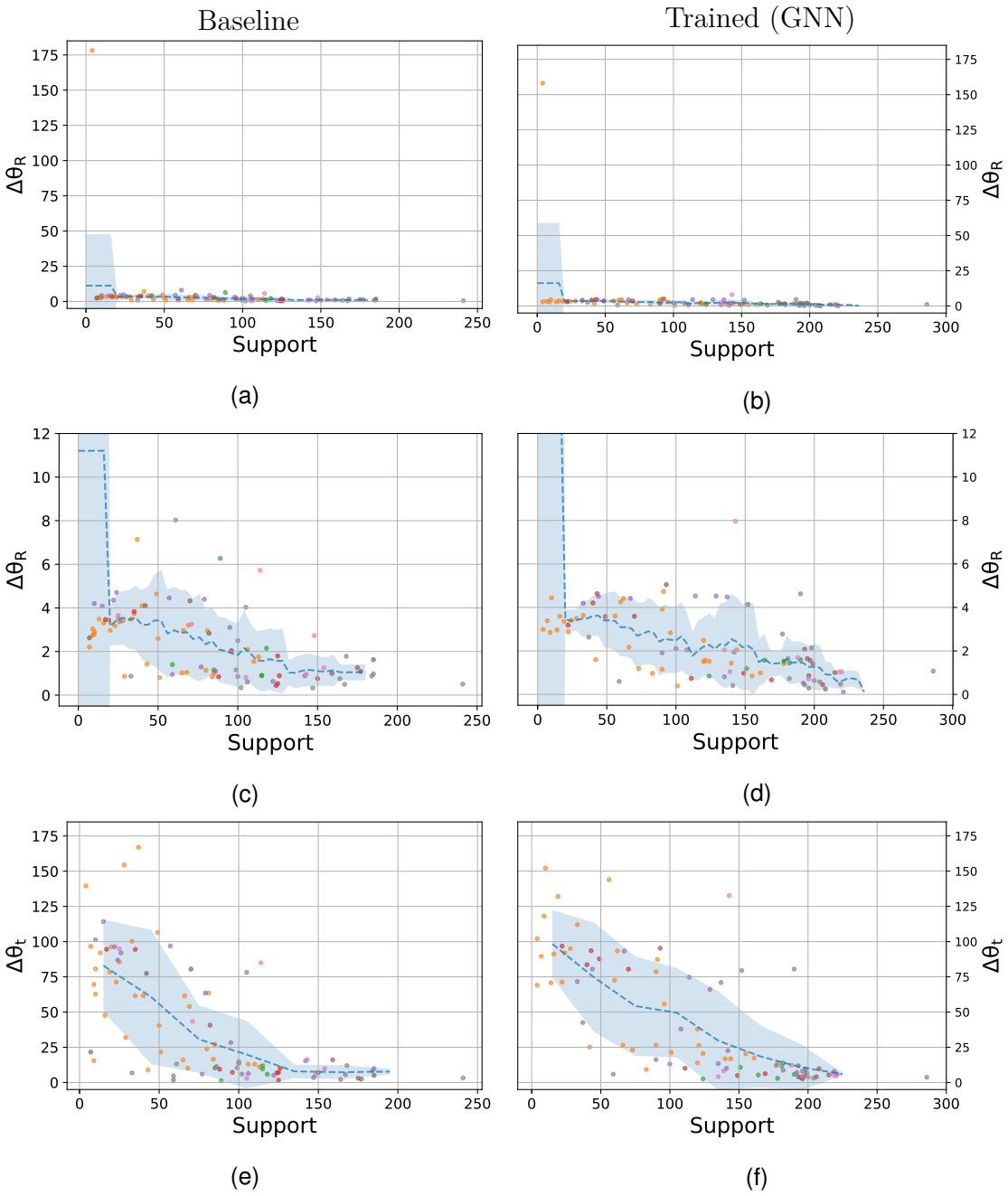


Figure 4.4: Pose errors using essential matrix estimation on the calibration dataset plotted against RANSAC support for the baseline model and trained GNN model. Plots (a-d) show the error in the rotation component of the relative camera pose, where (c-d) display the range of errors in more detail. Plots (e-f) show the angular error in translation. Colors of points represent the different calibration sets, and the blue region shows the running average error and standard deviation.

Outdoor	mean		median		std. dev.	
	baseline	trained	baseline	trained	baseline	trained
$\Delta\theta_{\boldsymbol{R}}$	**18.54**	32.86	1.57	**0.42**	**39.18**	56.83
$\Delta\theta_{\boldsymbol{t}}$	**78.26**	91.66	**73.68**	90.14	37.26	**29.83**
support	**20.62**	9.38	**14**	6	16.64	**9.92**

Table 4.2: Errors in pose—split into the errors in $\boldsymbol{R}$ and $\boldsymbol{t}$—listed in terms of their mean, median, and standard deviation, obtained using the outdoor dataset. The distribution of the support found by RANSAC also presented in the same way. The values are compared between the trained and baseline models, where the better value is highlighted by bold font.

Similarly, pose estimation was performed on the outdoor dataset. The corresponding error plots are shown in Figure 4.5. In this data—in comparison to the calibration dataset—the camera base-line is even smaller in relation to the distance to the 3D points. As expected, $\Delta\theta_{\boldsymbol{t}}$ is much worse. The rotational error $\Delta\theta_{\boldsymbol{t}}$ is not as good as in Figure 4.4, as it has a larger variance where there is low support, but the trend quickly approaches zero as support grows.

For both datasets the support is slightly smaller for the trained model, which is likely attributable to the homography-based method for pseudo-ground truth biasing the model to a smaller number of matches.

It should be noted that since pose estimation from essential matrix must consider cheirality (see Section 2.2.2.2), in cases with low support the cheirality check may fail and yield an additional error in $\boldsymbol{R}$ or $\boldsymbol{t}$ of 180°. So errors around 90° likely indicate more error in matches than errors around 180°, given the support is relatively low.

As is done by Sarlin et al. [19], *pose accuracy* is plotted against a threshold $\Delta\theta_{\boldsymbol{R}}^{\texttt{max}}$ in Figure 4.6. We see that the models perform very similarly, though the trained model appears to have a slight edge. However, it is disadvantaged by slightly worse performance for lower thresholds of $\Delta\theta_{\boldsymbol{t}}^{\texttt{max}}$, and any improvements are only very small. AUC values for these graphs are tabulated in Table 4.3. Once again, the trained model appears to have a very small edge over the baseline model, for all but one of the threshold values. The corresponding monomodal values presented by Sarlin et al. [19] are added for comparison. As mentioned, their numbers do not distinguish between $\Delta\theta_{\boldsymbol{R}}$ and $\Delta\theta_{\boldsymbol{t}}$, and they presumably have a larger base-line. As such, these values are expected to fall somewhere in between the pose accuracy AUC for $\Delta\theta_{\boldsymbol{R}}$ and $\Delta\theta_{\boldsymbol{t}}$.

The pose estimation using PnP, as described in Section 4.2, was performed on the calibration dataset. Similar graphs to the previous evaluations in this section is shown in Figure 4.7. Due to how this method was constructed, the number of matched points—and consequently the RANSAC support—was significantly reduced. Specifically, this was as a result of only the intersection of the matched points for both pairs of images could be used, and the added match inaccuracy this transitive matching introduces. As a result of this, the pose estimation became less

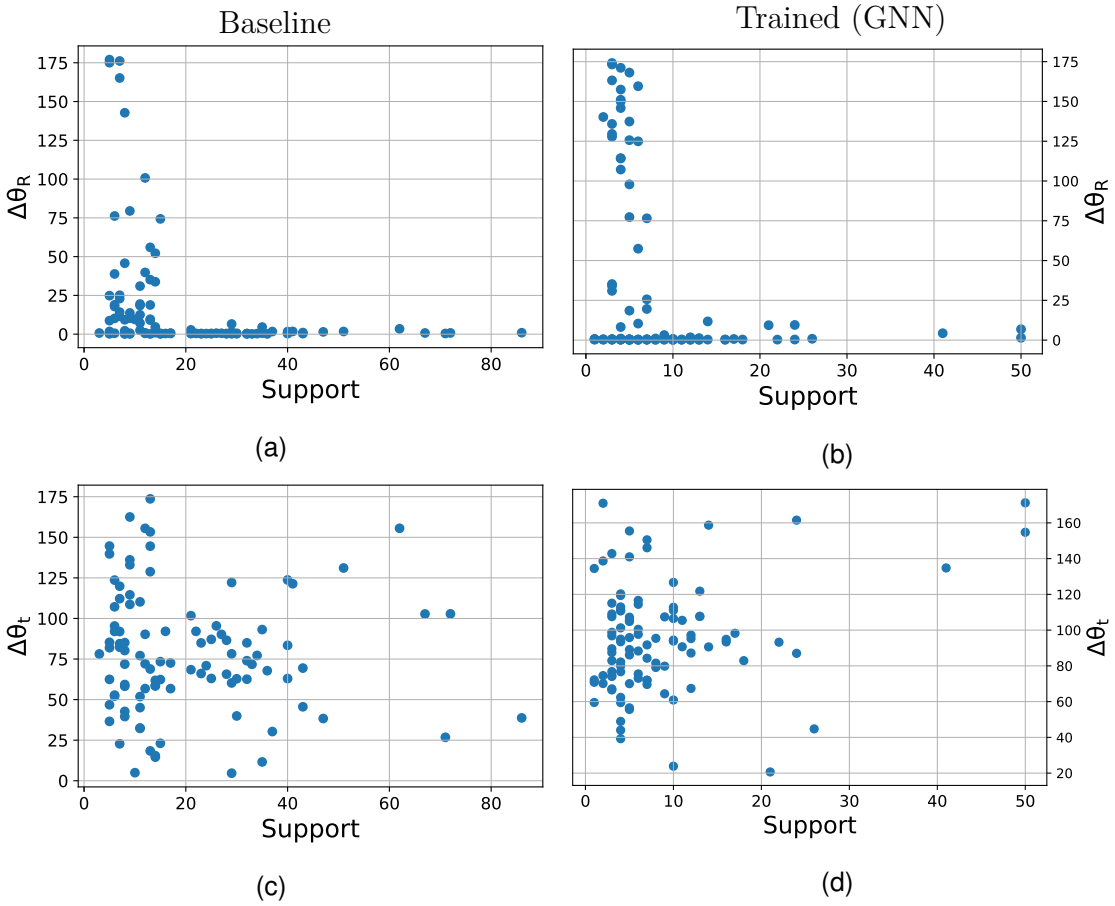


Figure 4.5: Pose errors using essential matrix estimation on the outdoor dataset plotted against RANSAC support for the baseline model and trained GNN model. Plots (a-d) show the error in the rotational component. Plots (c-d) show the angular error in translation.

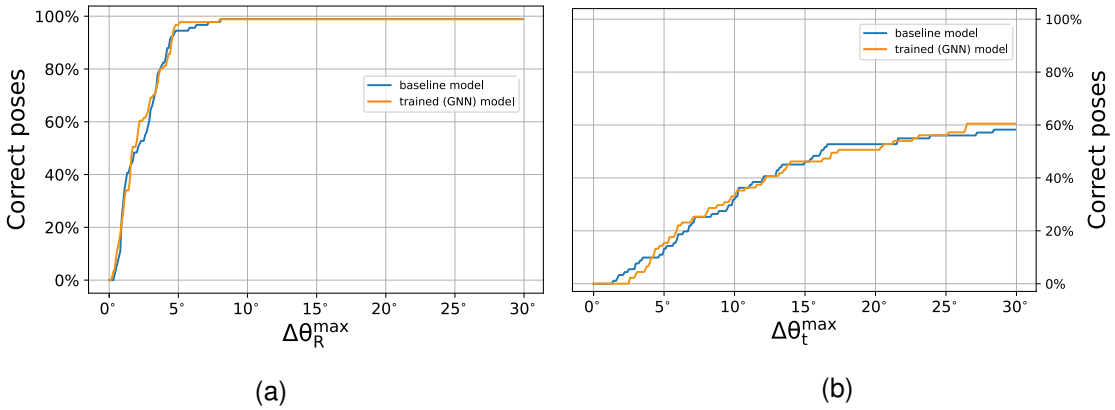


Figure 4.6: The percentage of correctly estimated poses using essential matrix estimation, plotted against the error threshold $\Delta\theta_{\boldsymbol{R}}^{\mathtt{max}}$ below which a pose is deemed correct. The baseline model is plotted with a blue line, while the trained GNN model is colored orange.

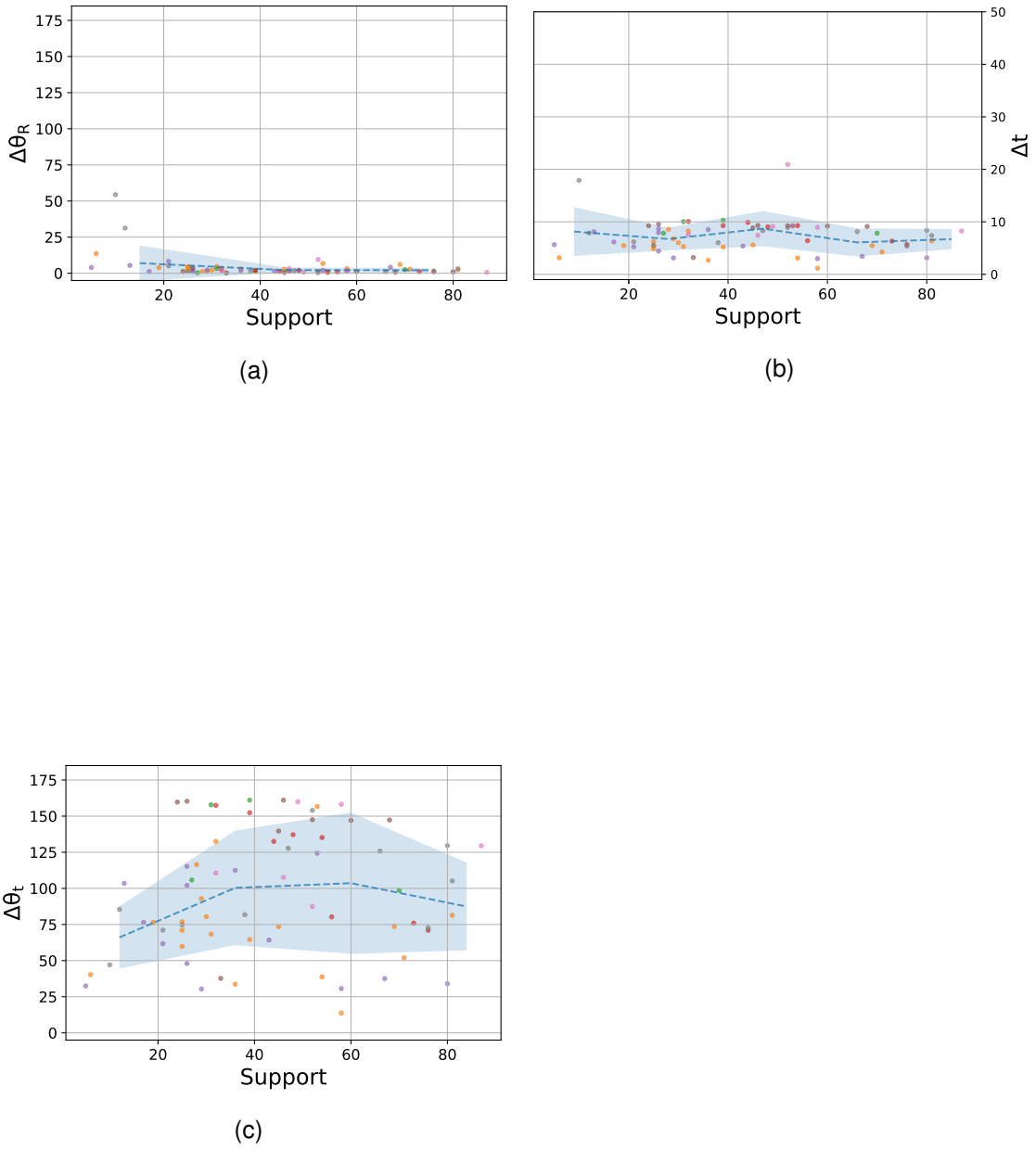


Figure 4.7: Pose errors using PnP on the calibration dataset plotted against RANSAC support for the trained GNN model. Plots (a–b) show errors in $\boldsymbol{R}$ and $\boldsymbol{t}$, respectively. For comparison to results in Figures 4.4 and 4.5, the angular error in $\boldsymbol{t}$ is shown in (c).

Indoor	@5°		@10°		@20°	
	$\Delta\theta_{\boldsymbol{R}}$	$\Delta\theta_{\boldsymbol{t}}$	$\Delta\theta_{\boldsymbol{R}}$	$\Delta\theta_{\boldsymbol{t}}$	$\Delta\theta_{\boldsymbol{R}}$	$\Delta\theta_{\boldsymbol{t}}$
Trained	**54.43**	3.56	**76.32**	**14.21**	**87.61**	**29.36**
Baseline	52.97	**5.03**	75.12	13.90	87.01	22.81
SuperGlue*	16.16		33.81		51.84	

Table 4.3: AUC scores of Figure 4.6 at three different thresholds of $\Delta\theta_{\boldsymbol{R}}^{\texttt{max}}$. The cross-modal models (*Trained* and *Baseline*) are split into errors for $\boldsymbol{R}$ and $\boldsymbol{t}$ respectively. Values are compared to the values presented in [19] for indoor wide base-line pose estimation. The largest AUC scores for each threshold are highlighted in bold font. Values are presented as percentages.

accurate, even though PnP is generally more accurate than essential matrix pose recovery, given the same conditions.

A sample of matches constructed using the PnP based method on the calibration datasetis included in Appendix A.

4.6 Unsuccessful designs

A few different architectures were tried in the project, among which the GNN-based model worked the best. This section will briefly present the unsuccessful designs and discuss their performance.

4.6.1 Training SuperPoint

The attempts at training SuperPoint were all unsuccessful for various reasons. As mentioned in Section 3.1.2, training was found to be unstable when training end-to-end—particularly when using trainable layers in the encoder section of the network. This is likely because this affected the score matrix, making points no longer be detected while perturbing their description.

Training instead only a modified descriptor decoder remained stable, but did not decrease loss either. The same behavior was observed in both end-to-end training and training only SuperPoint according to Equation 2.18.

Furthermore, training while freezing the pre-trained layers could only perturb the last layers in the model to any significant degree. This, in combination with the above conclusions, seems to indicate that the added interleaving layers are not conductive to the transfer learning objective.

4.6.2 Interfacing model

The interfacing model $\mathcal{M}$, as described in Section 3.1, was implemented both as a per-descriptor translator (i.e. the added layers preceeding the interpolation step

*monomodal and wide base-line

in Figure 3.1), and as a graph neural network. The former of the two yielded no improvement in loss, likely due to each descriptor not containing enough information in isolation—in part because SuperPoint was trained to describe points to a GNN that considers the entire (multi-)graph of descriptors and pixel coordinates. Another reason could be that the model was simply not complex enough to parse and translate the necessary information available.

4.6.3 Image translation

A simplistic model for image translation was tried in the form of modifying the SuperPoint encoder, which is equivalent to adding a convolutional neural network as a step before SuperPoint, as discussed in Section 3.1. This is therefore covered by Section 4.6.1.

An approach with a simple UNet [67] GAN architecture was also briefly attempted, but not thoroughly developed due to time constraints in the project.

The setup used to train a model $\mathcal{T}_{\text{IR}} = \boldsymbol{\mathcal{I}}_{\text{IR}} \mapsto \boldsymbol{\mathcal{I}}'_{\text{Vis+IR}}$ was to define four UNet models $\mathcal{T}_{\text{IR}}$, $\mathcal{T}_{\text{Vis}}$, $\mathcal{T}'_{\text{IR}}$, and $\mathcal{T}'_{\text{Vis}}$ (where $\mathcal{T}'_{\text{A}}$ is meant to approximate the inverse of $\mathcal{T}_{\text{A}}$) to fulfill the requirements:

$$\mathcal{T}'_{\text{IR}} \circ \mathcal{T}_{\text{IR}} \approx \text{id} \tag{4.3a}$$
$$\mathcal{T}'_{\text{Vis}} \circ \mathcal{T}_{\text{Vis}} \approx \text{id} \tag{4.3b}$$

under the assumption that all (or most) information in the image is retained when going from the source modality to the latent modality, or vice versa.

The models $\mathcal{T}_{\text{IR}}$ and $\mathcal{T}_{\text{Vis}}$ make up the generating part of the GAN, and a discriminator $\mathcal{D}$ is defined to distinguish which modality the source image originates from. In other words, $\mathcal{D}$ would optimally have the following behavior:

$$\mathcal{D}(\mathcal{T}_{\text{IR}}(\boldsymbol{\mathcal{I}}_{\text{IR}})) = 1 \tag{4.4a}$$
$$\mathcal{D}(\mathcal{T}_{\text{Vis}}(\boldsymbol{\mathcal{I}}_{\text{Vis}})) = 0 \tag{4.4b}$$

while $\mathcal{T}_{\text{IR}}$ and $\mathcal{T}_{\text{Vis}}$ are instead optimized to maximize the cross-entropy.

It was discovered that this kind of training was difficult to stabilize since the function image of both $\mathcal{T}_{\text{Vis}}$ and $\mathcal{T}_{\text{IR}}$ are constantly changing. The usual way of defining a GAN leaves the distribution of one of the discriminated classes fixed [32] so that the discriminator is able to continually learn. One way to get around this problem is to let the target domain of the image translation functions $\mathcal{T}_{\text{IR}}, \mathcal{T}_{\text{Vis}}$ be one of the source domains Vis, IR, as mentioned in Section 3.1, and like Zhang et al. [17] demonstrated. This way, the target stays fixed and known, but comes with its own

problems. Apart from the problems mentioned earlier in Section 3.1, there is also the problem of the different modalities containing mutually exclusive information, which might encourage such a model to make up information where there is none.

5

Discussion

In this section the results and methods used are discussed.

5.1 Data

The thermographic camera used to collect image data is an uncooled microbolometer. For this reason, there is a larger influence of noise—and in particular FPN noise. This may have had a biasing factor on training, and it made some domains difficult to observe. On the other hand, it is likely has a positive influence on training in that it builds robustness to high-frequency noise, as well as the kind of high- and low-frequency FPN noise present in images from this kind of camera unit. The actual effect of this is difficult to measure however, since we cannot easily isolate the FPN noise in order to compare the effect it has on the model.

In practice, both cooled and uncooled cameras could potentially be used. If an uncooled camera is preferred, and cold environments can be expected, some kind of image processing like what is suggested in Section 3.2.1 could potentially be used to increase robustness. A possible issue with this is that these processed images would not be representative of the rest of the data, and could therefore make the model perform worse. This, in addition to not being able to consistently get good denoising, led to this not being integrated as a pre-processing step.

At no point in the project was the dataset used not large enough. However, the number of individual flights is not very large, and so is quite biased. This is one good reason for relying on transfer learning, but would also be a reason to want to augment the data somehow. Diversifying the data without introducing other data sources or simply collecting more data is not easy, but can be done to some extent in order to reduce certain biases and improve the robustness and generality of the model. One such method is homography augmentation, which was successfully employed in training both SuperPoint [5] and SuperGlue [19] among others [28]. In addition to extending the effective size of the training dataset, this method encourages the model to choose matches that are (locally) consistent with a common homography, which can be very useful since a homography describes a perspective transform of a plane [8].

One significant bias in the data used in this project is that the base-line is roughly the same for all images. This may instill the expectation that feature point matches should not deviate too much from the approximate homography in Equation 2.17 in the trained model. To get an effectively much larger base-line, image pairs could instead be taken at slightly different points in time. In doing this, however, the homography-based method for pseudo-ground truth (Section 3.3.1) becomes much less reliable, and one must ensure that images contain the same scene, similar to how the triangulation-based method (Section 3.3.2) for pseudo-ground truth was constructed. This increases the need for accurately determining whether camera views intersect sufficiently. A good estimate for this could be to estimate the intersection of the two camera view frustums, like done by Zamani et al. [68]. A problem in this formulation is that we cannot know the range of distances from the camera to consider, and according to Zamani et al., calculating the intersection alone is non-trivial.

Another common way to augment image data is to add noise to the images in order to increase robustness to this noise [69]. While adding Gaussian noise can be beneficial [69], the kind of FPN noise often prevalent in these kinds of images [70] is not Gaussian. Further work could try to model this noise, or instead use methods to reduce it, as done by Guan et al. [70].

The pose data used is generated according to the reference frame of the IMU unit,* however, this is used directly to represent camera pose, without accounting for the offset between IMU and cameras. This can be an issue in relative pose accuracy between poses at different times. In most cases it should not be a significant problem since the drone travels roughly straight most of the time.

5.1.1 Ground truth

While the homography-based estimation method is more dense, it has several limitations such as often failing to cover the entire image, being less accurate, and possibly biasing the model to homographies. This last issue seems to already be an issue with the SuperPoint and SuperGlue models—as can be seen in many of the mismatches in Appendix A.1, in Figure 4.2, or in the examples provided in [19]—where complete mismatches often find small sections of the two images related by a homography, which can be a problem for RANSAC-based algorithms.

Since the triangulation-based methods do not yield as dense correspondences as the homography-based method does, a trade-off could be made in further work where after finding the triangulation-based pseudo-ground truth correspondences, RANSAC or similar is used to find local homographies with a strict error threshold, for which extra points can be augmented.

Another potential way to improve the triangulation-based methods would be to use bundle adjustment methods like in SLAM [3], in order to find correspondences using fewer constraints.

*Also the GPS data neglects any offset between IMU and GPS.

A third potential improvement that could be tried in the pseudo-ground truth construction would be to do a more sofisticated matching than just selecting the closest neighbor projected from the other modality, within a certain radius. Because this often leads to multiple points receiving the same match, it could be possible to do optimal matching on such subsets of the image.

5.2 Choice of method

The particular choice of models is not clear whether they were the best suited, even though there likely is good promise in the models elected.

As pointed to in Section 3.1, an approach based on a generative model such as the GAN used by Ran Zhang et al. [17] for image registration could be a good alternative approach. However, due to the greater sensitivity to errors in $\boldsymbol{P}$ for camera pose estimation and SLAM, compared to homography estimation and image registration, it is likely that these methods may not be accurate enough in placement of image features.

Further, a derivative model of the original SuperGlue model could have been used. For example, the model introduced by Shi et al. [20] in late April of this year could have helped in bringing down the hardware requirements for training and inference, and made for a more scalable solution.

There was not a lot of effort put towards developing modality-specific feature point extractors, which could potentially be a direction for further work, as a lot of feature points do not in fact overlap. Increasing this overlap could help in creating more dense and accurate matches, assuming that there is significant common information between these feature points.

5.3 Evaluation

One of the more noteworthy observations from the pose estimation evaluation of the different matchers was the lack of increased performance despite better matches. The SuperPoint model only estimates feature locations to, at best, pixel-level. In intrinsic and extrinsic camera calibration with calibration boards sub-pixel precision is achieved by modeling saddle points to achieve more accurate pose or parameter estimates [7], [71]. This is likely at least a contributing factor to the poor improvement in pose accuracy. Further work could potentially integrate sub-pixel estimates into SuperPoint by, for instance, modeling the distribution of scores per 8×8 cell.

5.4 Conclusion

The objective of this work has been to investigate approaches to the cross-modal image feature matching problem for the visual and LWIR modalities, utilizing the pre-trained SuperPoint and SuperGlue models. A number of approaches for transfer

learning have been investigated, and while some have shown marginal improvements in some aspects, none have been able to clearly outperform the baseline model. This is likely due to the complex relationship between the weights in the early layers—which are arguably the ones most closely tied to changes in modality in the input—and the final loss. This highlights some of the limits of transfer learning and complex end-to-end training, and gives some insight where to focus related further work.

As mentioned earlier in this chapter, another limiting factor to the success of any proposed models in this report is the indirect way of evaluation, and the errors introduced by interest points' discrete coordinates in trying to estimate pose with high accuracy (and small base-line).

Furthermore, the report concludes some limitations to the pre-trained models themselves; such as in Section 4.3 where SuperGlue predicts completely wrong results with high confidence under certain circumstances, and is highly inconsistent under presumed invariant transformations of the input data.

Bibliography

[1] P. Lindenberger, P.-E. Sarlin, V. Larsson, and M. Pollefeys, "Pixel-Perfect Structure-from-Motion with Featuremetric Refinement," pp. 5967–5977, Feb. 2022, ISSN: 15505499. DOI: 10.1109/ICCV48922.2021.00593.

[2] S. Agarwal, Y. Furukawa, N. Snavely, *et al.*, "Building Rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, Oct. 2011, ISSN: 0001-0782. DOI: 10.1145/2001269.2001293. [Online]. Available: https://dl.acm.org/doi/10.1145/2001269.2001293.

[3] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015, ISSN: 15523098. DOI: 10.1109/TRO.2015.2463671.

[4] M. Hassaballah, A. A. Abdelmgeid, and H. A. Alshazly, "Image Features Detection, Description and Matching," *Studies in Computational Intelligence*, vol. 630, pp. 11–45, 2016, ISSN: 1860949X. DOI: 10.1007/978-3-319-28854-3_2. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-28854-3_2.

[5] D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 337–349, Dec. 2017, ISSN: 21607516. DOI: 10.1109/CVPRW.2018.00060. [Online]. Available: https://arxiv.org/abs/1712.07629v4.

[6] W. Liu, X. Shen, C. Wang, Z. Zhang, C. Wen, and J. Li, "H-Net: Neural Network for Cross-domain Image Patch Matching," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018, pp. 856–863. [Online]. Available: https://www.ijcai.org/proceedings/2018/0119.pdf.

[7] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., ser. Texts in Computer Science. Springer International Publishing, 2022, ISBN: 978-3-030-34371-2. DOI: 10.1007/978-3-030-34372-9. [Online]. Available: https://link.springer.com/10.1007/978-3-030-34372-9.

[8] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge university press, 2003, ISBN: 9780521540513.

[9] D. G. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999. DOI: 10.1109/ICCV.1999.790410.

[10] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011. DOI: 10.1109/ICCV.2011.6126544.

[11] B. Kong, J. Supancic, D. Ramanan, and C. C. Fowlkes, "Cross-Domain Image Matching with Deep Feature Maps," *International Journal of Computer Vision*, vol. 127, no. 11-12, pp. 1738–1750, Apr. 2018. DOI: 10.1007/s11263-018-01143-3. [Online]. Available: http://arxiv.org/abs/1804.02367.

[12] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-Free Local Feature Matching with Transformers," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 8918–8927, Apr. 2021, ISSN: 10636919. DOI: 10.48550/arxiv.2104.00680. [Online]. Available: https://arxiv.org/abs/2104.00680v1.

[13] J. Edstedt, M. Wadenbäck, and M. Felsberg, "Deep Kernelized Dense Geometric Matching," Feb. 2022. DOI: 10.48550/arxiv.2202.00667. [Online]. Available: https://arxiv.org/abs/2202.00667v1.

[14] B. Mehlig, *Machine learning with neural networks : an introduction for scientists and engineers*. Cambridge University Press, Oct. 2021, ISBN: 9781108494939.

[15] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros, "Data-driven Visual Similarity for Cross-domain Image Matching," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, pp. 1–10, Dec. 2011, ISSN: 15577368. DOI: 10.1145/2024156.2024188. [Online]. Available: http://graphics.cs.cmu.edu/projects/crossDomainMatching/abhinav-sa11.pdf.

[16] H. Li, X.-J. Wu, and J. Kittler, "Infrared and Visible Image Fusion using a Deep Learning Framework," *Proceedings - International Conference on Pattern Recognition*, vol. 2018-August, pp. 2705–2710, Apr. 2018. DOI: 10.1109/ICPR.2018.8546006. [Online]. Available: http://arxiv.org/abs/1804.06992.

[17] R. Zhang, J. Bin, Z. Liu, and E. Blasch, "Image translation to enhance IR2VIS image registration," *https://doi.org/10.1117/12.2588034*, vol. 11733, pp. 30–40, Apr. 2021, ISSN: 1996756X. DOI: 10.1117/12.2588034. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11733/117330A/Image-translation-to-enhance-IR2VIS-image-registration/10.1117/12.2588034.full.

[18] D. Zhao, "Rapid Multimodal Image Registration Based on the Local Edge Histogram," *Mathematical Problems in Engineering*, vol. 2021, 2021, ISSN: 15635147. DOI: 10.1155/2021/5598177.

[19] P. E. Sarlin, D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching with Graph Neural Networks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4937–4946, Nov. 2019, ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.00499. [Online]. Available: https://arxiv.org/abs/1911.11763v2.

[20] Y. Shi, J.-X. Cai, Y. Shavit, T.-J. Mu, W. Feng, and K. Zhang, "ClusterGNN: Cluster-based Coarse-to-Fine Graph Neural Network for Efficient Feature Matching," Apr. 2022. DOI: 10.48550/arxiv.2204.11700. [Online]. Available: https://arxiv.org/abs/2204.11700v1.

[21] *FREE - FLIR Thermal Dataset for Algorithm Training | Teledyne FLIR*. [Online]. Available: https://www.flir.com/oem/adas/adas-dataset-form/.

[22] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral Pedestrian Detection: Benchmark Dataset and Baselines," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[23] H. Zhao, J. Jia, and V. Koltun, "Exploring Self-attention for Image Recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 10 073–10 082, Apr. 2020, ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.01009. [Online]. Available: https://arxiv.org/abs/2004.13621v1.

[24] N. Pielawski, E. Wetzer, J. Öfverstedt, *et al.*, "CoMIR: Contrastive Multimodal Image Representation for Registration," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 18 433–18 444. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/d6428eecbe0f7dff83fc607c5044b2b9-Paper.pdf.

[25] C. Wachinger and N. Navab, "Entropy and Laplacian images: Structural representations for multi-modal registration," *Medical Image Analysis*, vol. 16, no. 1, pp. 1–17, Jan. 2012, ISSN: 1361-8415. DOI: 10.1016/J.MEDIA.2011.03.001.

[26] Y. Ye, L. Bruzzone, J. Shan, F. Bovolo, and Q. Zhu, "Fast and Robust Matching for Multimodal Remote Sensing Image Registration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9059–9070, Nov. 2019, ISSN: 15580644. DOI: 10.1109/TGRS.2019.2924684.

[27] X. Jiang, J. Ma, G. Xiao, Z. Shao, and X. Guo, "A review of multimodal image matching: Methods and applications," *Information Fusion*, vol. 73, pp. 22–71, Sep. 2021, ISSN: 1566-2535. DOI: 10.1016/J.INFFUS.2021.02.012.

[28] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep Image Homography Estimation," Jun. 2016. DOI: 10.48550/arxiv.1606.03798. [Online]. Available: https://arxiv.org/abs/1606.03798v1.

[29] M. Arar, Y. Ginger, D. Danon, A. H. Bermano, and D. Cohen-Or, "Unsupervised multi-modal image registration via geometry preserving image-to-image translation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 13 407–13 416, Mar. 2020, ISSN: 10636919. DOI: 10.1109/CVPR42600.2020.01342. [Online]. Available: https://arxiv.org/abs/2003.08073v1.

[30] R. Windsor, A. Jamaludin, T. Kadir, and A. Zisserman, "Self-supervised Multi-modal Alignment for Whole Body Medical Imaging," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12902 LNCS, pp. 90–101, Jul. 2021, ISSN: 16113349. DOI: 10.1007/978-3-030-87196-3_9. [Online]. Available: https://arxiv.org/abs/2107.06652v2.

[31] S. Jegou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2017-July, pp. 1175–1183, Nov. 2016, ISSN:

21607516. DOI: 10.48550/arxiv.1611.09326. [Online]. Available: https://arxiv.org/abs/1611.09326v3.

[32] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative Adversarial Nets," *Advances in neural information processing systems*, vol. 27, Jun. 2014. [Online]. Available: http://arxiv.org/abs/1406.2661.

[33] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press, 2016, http://www.deeplearningbook.org.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, ISSN: 00189219. DOI: 10.1109/5.726791.

[35] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, Jun. 2021, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2021.3084827.

[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, Jan. 2021, ISSN: 21622388. DOI: 10.1109/TNNLS.2020.2978386.

[37] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009, ISSN: 10459227. DOI: 10.1109/TNN.2008.2010350.

[38] T. S. Huang, "Computer Vision: Evolution And Promise," *1996 CERN School of Computing*, vol. 19, pp. 21–25, 1996. DOI: 10.5170/CERN-1996-008.21. [Online]. Available: http://cds.cern.ch/record/400313.

[39] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, 2018, ISSN: 16875273. DOI: 10.1155/2018/7068349.

[40] D. C. Brown, "Close-Range Camera Calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855–866, 1971.

[41] F. Bukhari and M. N. Dailey, "Automatic radial distortion estimation from a single image," *Journal of Mathematical Imaging and Vision*, vol. 45, no. 1, pp. 31–45, Jan. 2013, ISSN: 09249907. DOI: 10.1007/S10851-012-0342-2.

[42] S. Pang, A. Du, M. A. Orgun, and H. Chen, "Weakly supervised learning for image keypoint matching using graph convolutional networks," *Knowledge-Based Systems*, vol. 197, p. 105 871, Jun. 2020, ISSN: 0950-7051. DOI: 10.1016/J.KNOSYS.2020.105871.

[43] M. E. Fathy, A. S. Hussein, and M. F. Tolba, "Fundamental Matrix Estimation: A Study of Error Criteria," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 383–391, Jun. 2017. DOI: 10.1016/j.patrec.2010.09.019. [Online]. Available: http://arxiv.org/abs/1706.07886.

[44] K. M. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, and P. Fua, "Learning to Find Good Correspondences," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2666–2674, Nov. 2017, ISSN: 10636919. DOI: 10.48550/arxiv.1711.05971. [Online]. Available: https://arxiv.org/abs/1711.05971v2.

[45] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, Nov. 1997, ISSN: 1077-3142. DOI: 10.1006/CVIU.1997.0547.

[46] P. Lindstrom, "Triangulation made easy," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1554–1561, 2010, ISSN: 10636919. DOI: 10.1109/CVPR.2010.5539785.

[47] S. Pan and X. Wang, "A Survey on Perspective-n-Point Problem," *Chinese Control Conference, CCC*, vol. 40, pp. 2396–2401, Jul. 2021, ISSN: 21612927. DOI: 10.23919/CCC52363.2021.9549863.

[48] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision 2008 81:2*, vol. 81, no. 2, pp. 155–166, Jul. 2008, ISSN: 1573-1405. DOI: 10.1007/S11263-008-0152-6. [Online]. Available: https://link.springer.com/article/10.1007/s11263-008-0152-6.

[49] *OpenCV: Camera Calibration and 3D Reconstruction*. [Online]. Available: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html.

[50] T. Pajdla, *Elements of Geometry for Computer Vision*. Czech Technical University in Prague, 2017.

[51] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Toward Geometric Deep SLAM," Jul. 2017. [Online]. Available: https://arxiv.org/abs/1707.07410v1.

[52] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Sep. 2014. DOI: 10.48550/arxiv.1409.1556. [Online]. Available: https://arxiv.org/abs/1409.1556v6.

[53] U. Efe, K. G. Ince, and A. Alatan, "DFM: A Performance Baseline for Deep Feature Matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2021, pp. 4284–4293.

[54] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio, "An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Dec. 2013. DOI: 10.48550/arxiv.1312.6211. [Online]. Available: https://arxiv.org/abs/1312.6211v3.

[55] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997, ISSN: 01628828. DOI: 10.1109/34.601246.

[56] W. Chojnacki, M. J. Brooks, A. Van Den Hengel, and D. Gawley, "Revisiting Hartley's normalized eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1172–1177, Sep. 2003, ISSN: 01628828. DOI: 10.1109/TPAMI.2003.1227992.

[57] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-October, pp. 2242–

2251, Mar. 2017, ISSN: 15505499. DOI: 10.48550/arxiv.1703.10593. [Online]. Available: https://arxiv.org/abs/1703.10593v6.

[58] K. G. Larkin, "Reflections on Shannon Information: In search of a natural information-entropy for images," Sep. 2016. DOI: 10.48550/arxiv.1609.01117. [Online]. Available: https://arxiv.org/abs/1609.01117v1.

[59] C. Cheng and K. K. Parhi, "Fast 2D Convolution Algorithms for Convolutional Neural Networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1678–1691, May 2020, ISSN: 15580806. DOI: 10.1109/TCSI.2020.2964748.

[60] Z. Li and N. Snavely, "MegaDepth: Learning Single-View Depth Prediction from Internet Photos," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, Apr. 2018, ISSN: 10636919. DOI: 10.48550/arxiv.1804.00607. [Online]. Available: https://arxiv.org/abs/1804.00607v4.

[61] M. Hess-Flores, S. Recker, and K. I. Joy, "Uncertainty, baseline, and noise analysis for l1 error-based multi-view triangulation," *Proceedings - International Conference on Pattern Recognition*, pp. 4074–4079, Dec. 2014, ISSN: 10514651. DOI: 10.1109/ICPR.2014.698.

[62] M. Dwairi, Z. Alqadi, A. Abujazar, and R. Abu Zneit, "Optimized True-Color Image Processing," *World Applied Sciences Journal*, vol. 8, no. 10, pp. 1175–1182, May 2010, ISSN: 1818-4952.

[63] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, and W. Li, *GitHub - Magic Leap / SuperGlue Pretrained Network*, 2020. [Online]. Available: https://github.com/magicleap/SuperGluePretrainedNetwork.

[64] M. A. Fischler and R. C. Bolles, "Random sample consensus," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, ISSN: 15577317. DOI: 10.1145/358669.358692. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/358669.358692.

[65] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial Markers for Pose Estimation," *Journal of Intelligent & Robotic Systems 2021 101:4*, vol. 101, no. 4, pp. 1–26, Mar. 2021, ISSN: 1573-0409. DOI: 10.1007/S10846-020-01307-9. [Online]. Available: https://link.springer.com/article/10.1007/s10846-020-01307-9.

[66] G. Bökman and F. Kahl, "A Case for Using Rotation Invariant Features in State of the Art Feature Matchers," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 5110–5119, 2022. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022W/IMW/html/Bokman_A_Case_for_Using_Rotation_Invariant_Features_in_State_of_CVPRW_2022_paper.html.

[67] W. Weng and X. Zhu, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *IEEE Access*, vol. 9, pp. 16 591–16 603, May 2015, ISSN: 21693536. DOI: 10.48550/arxiv.1505.04597. [Online]. Available: https://arxiv.org/abs/1505.04597v1.

[68] Y. Zamani, H. Shirzad, and S. Kasaei, "Similarity measures for intersection of camera view frustums," *Iranian Conference on Machine Vision and Im-*

age Processing, MVIP, vol. 2017-November, pp. 171–175, Apr. 2018, ISSN: 21666784. DOI: 10.1109/IRANIANMVIP.2017.8342343.

[69] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019, ISSN: 21961115. DOI: 10.1186/S40537-019-0197-0. [Online]. Available: https://link.springer.com/article/10.1186/s40537-019-0197-0.

[70] J. Guan, R. Lai, A. Xiong, Z. Liu, and L. Gu, "Fixed Pattern Noise Reduction for Infrared Images Based on Cascade Residual Attention CNN," *Neurocomputing*, vol. 377, pp. 301–313, Oct. 2019. DOI: 10.1016/j.neucom.2019.10.054. [Online]. Available: http://arxiv.org/abs/1910.09858.

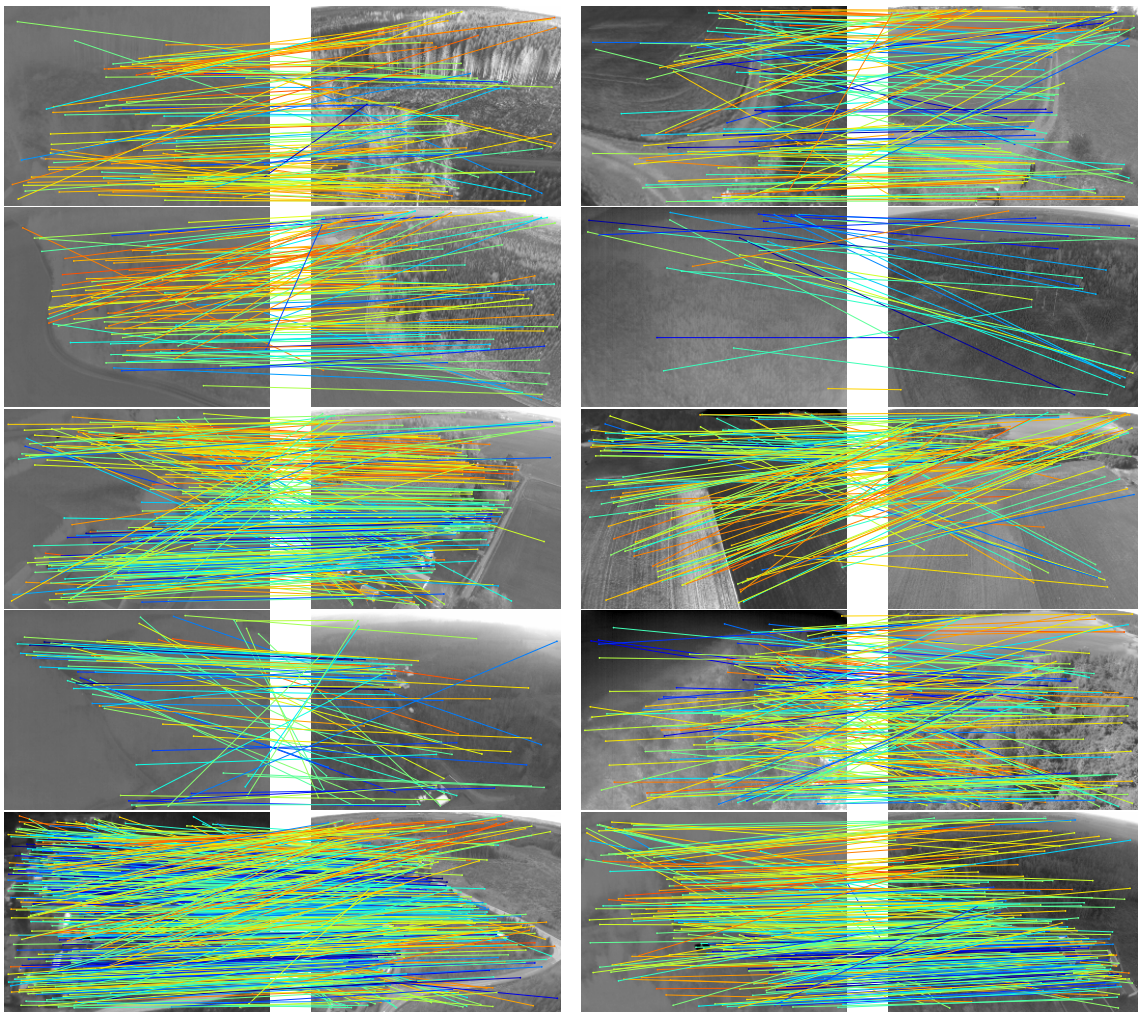
[71] L. Lucchese and S. K. Mitra, "Using saddle points for subpixel feature detection in camera calibration targets," *IEEE Asia-Pacific Conference on Circuits and Systems, Proceedings, APCCAS*, vol. 2, pp. 191–195, 2002. DOI: 10.1109/APCCAS.2002.1115151.

A

Appendix of Matching Plots

This appendix list examples of matched images from the test set.

Figure A.1: Visual–thermal image pairs which were run through the baseline model to create feature point matches. The thermal (LWIR) image is placed to the left, and the visual image to the right. Pairs are tabulated in two columns. Line colors indicate the confidence of each match, with blue being most confident, and red being the least confident.



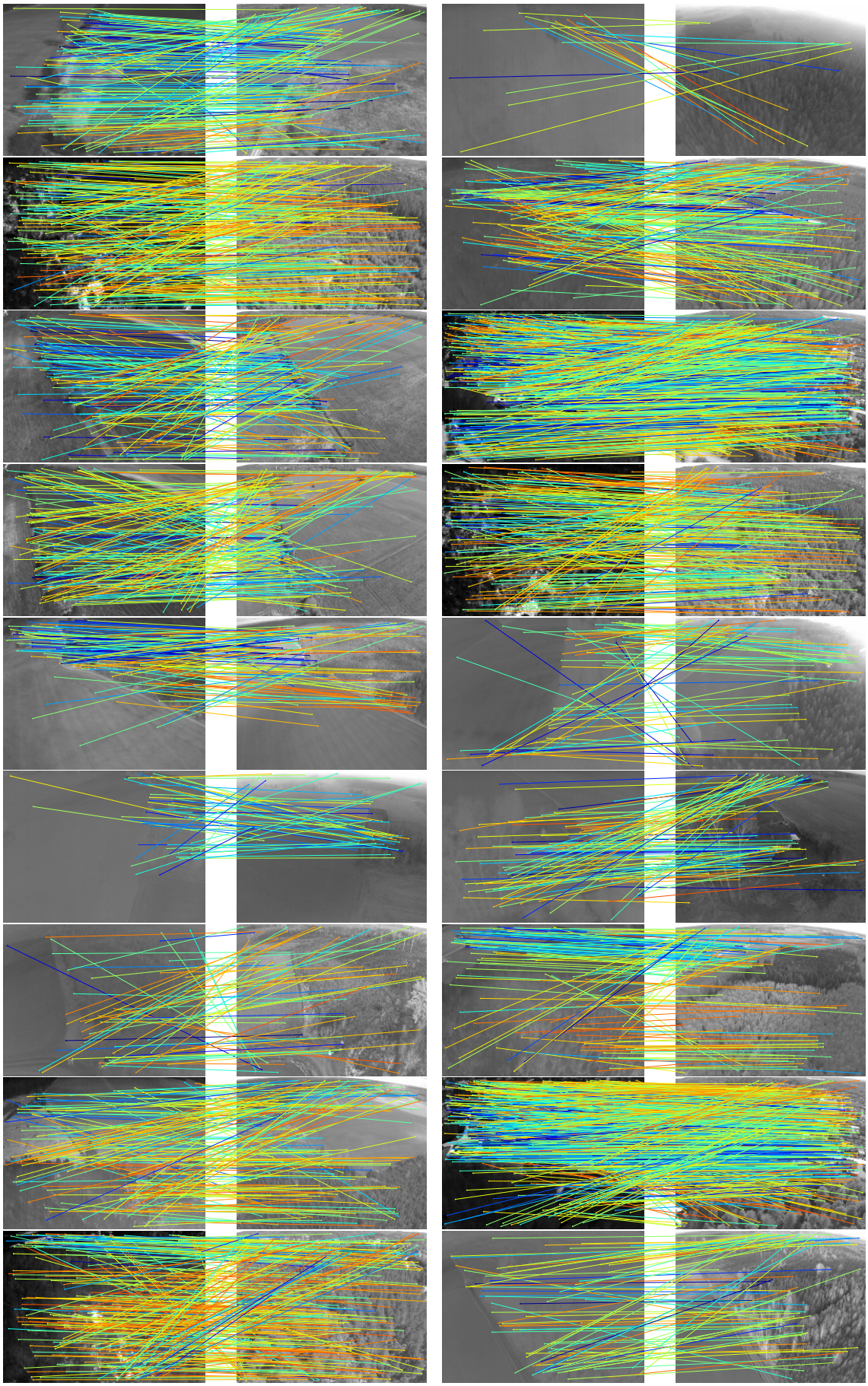
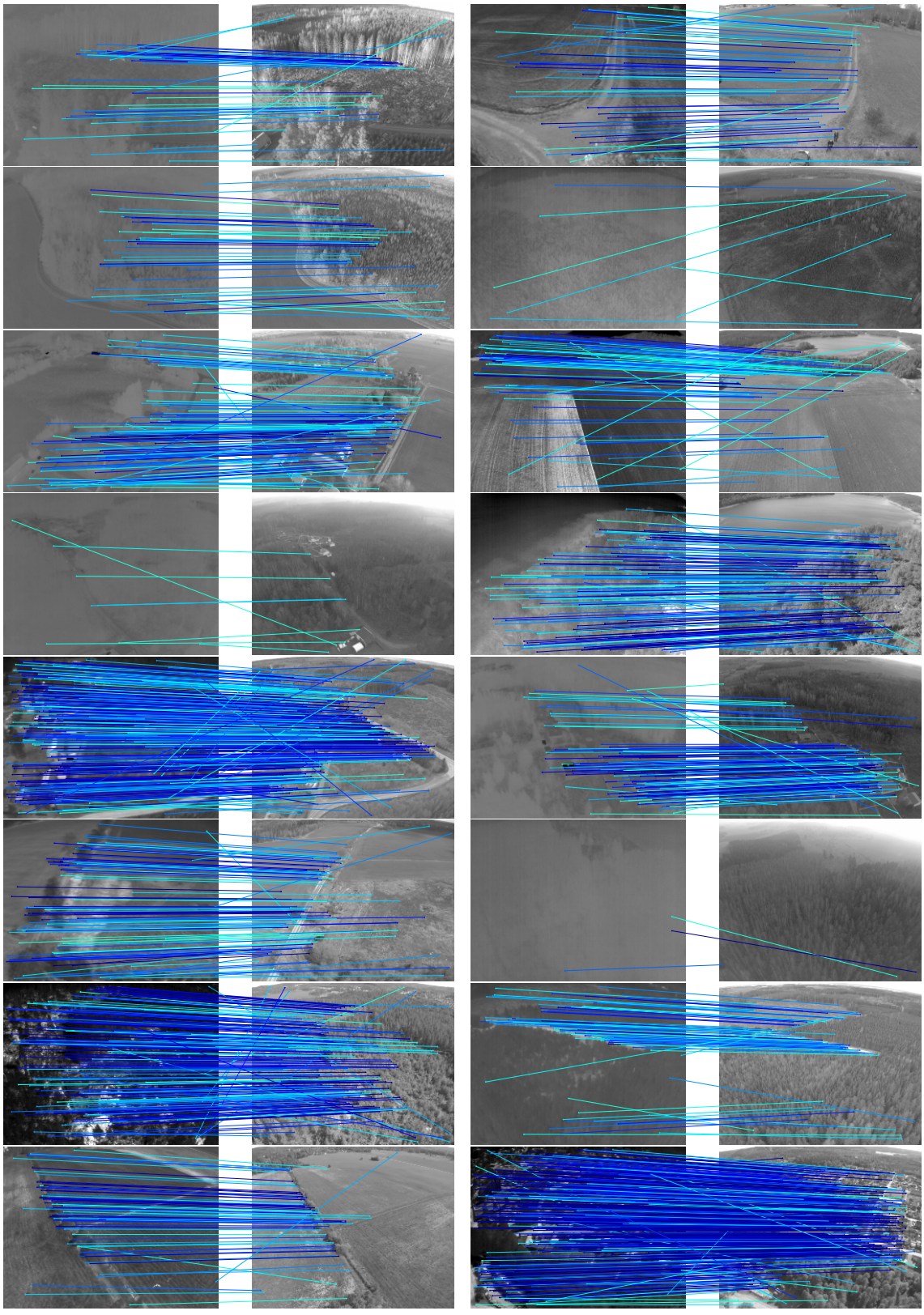
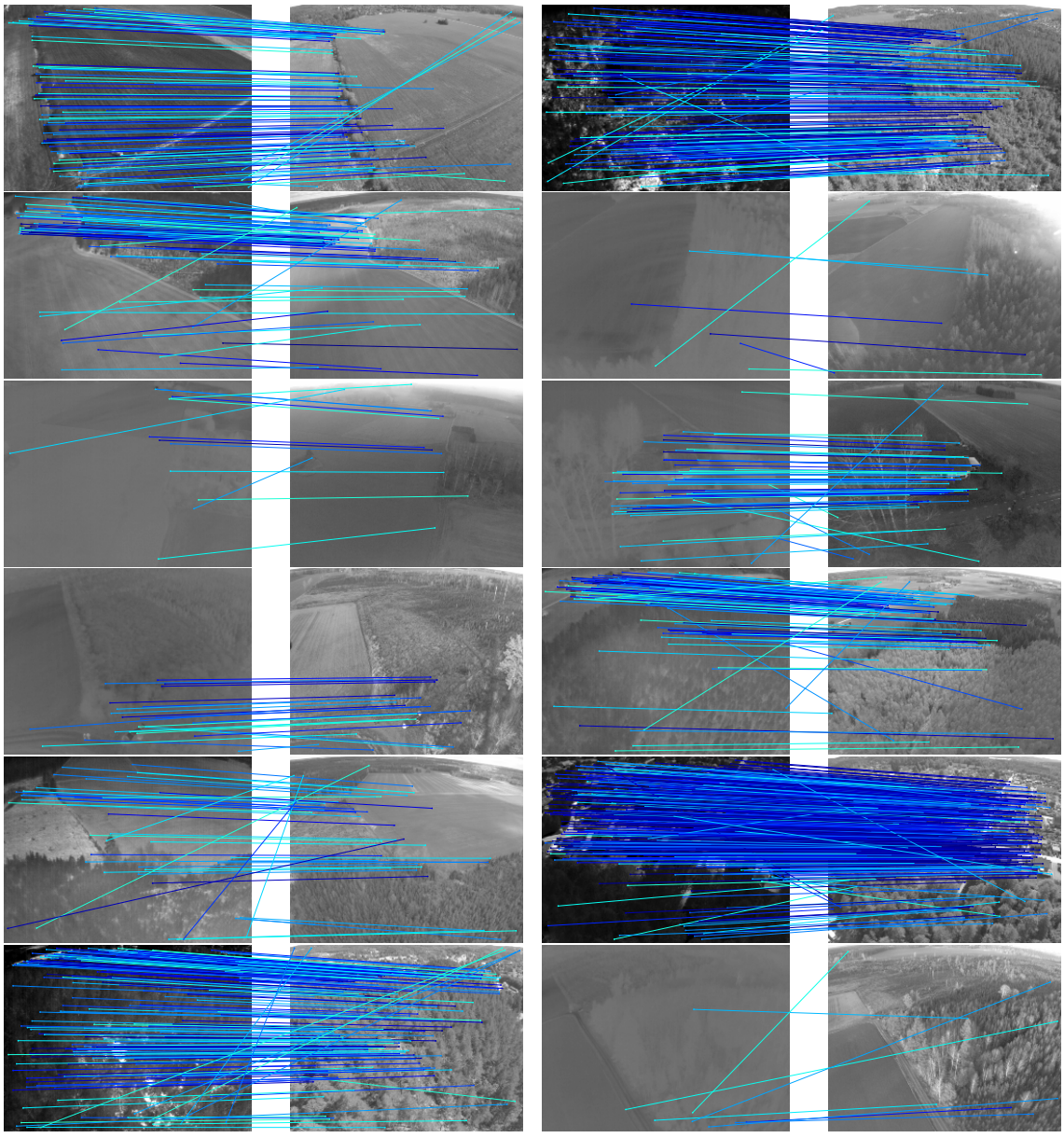


Figure A.2: Visual–thermal image pairs which were run through the trained GNN model to create feature point matches. The thermal (LWIR) image is placed to the left, and the visual image to the right. Pairs are tabulated in two columns. Line colors indicate the confidence of each match, with blue being most confident, and red being the least confident.





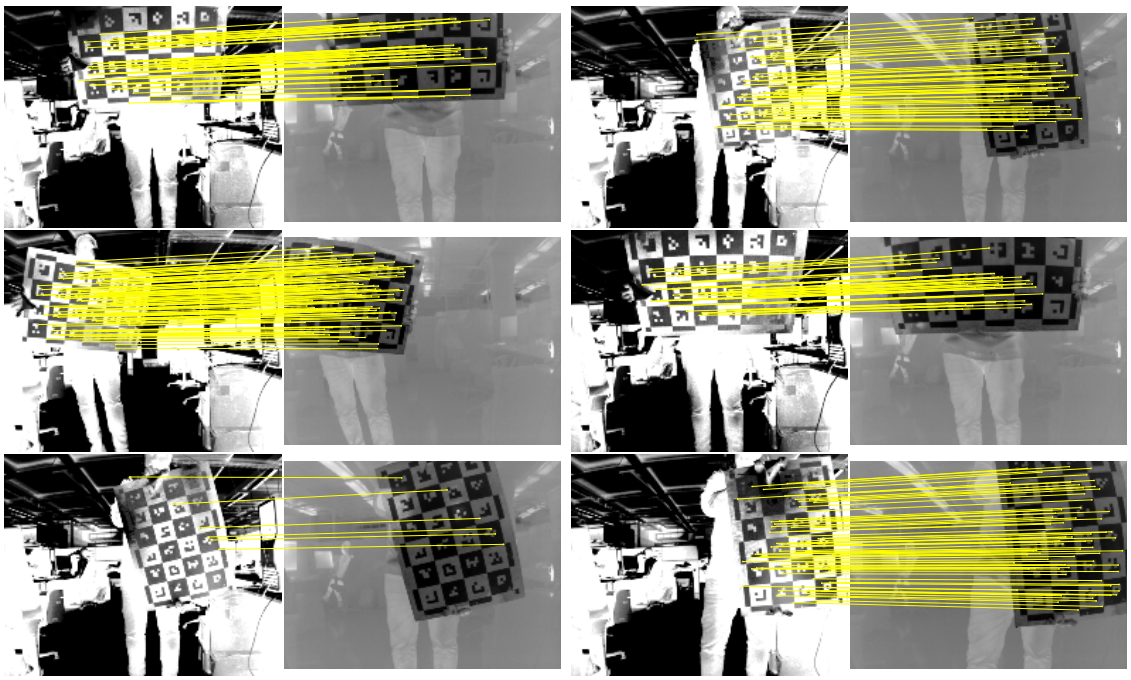


Figure A.3: A sample of image matchings over the calibration datasetusing the PnP method.

B

Appendix of Image Processing Examples

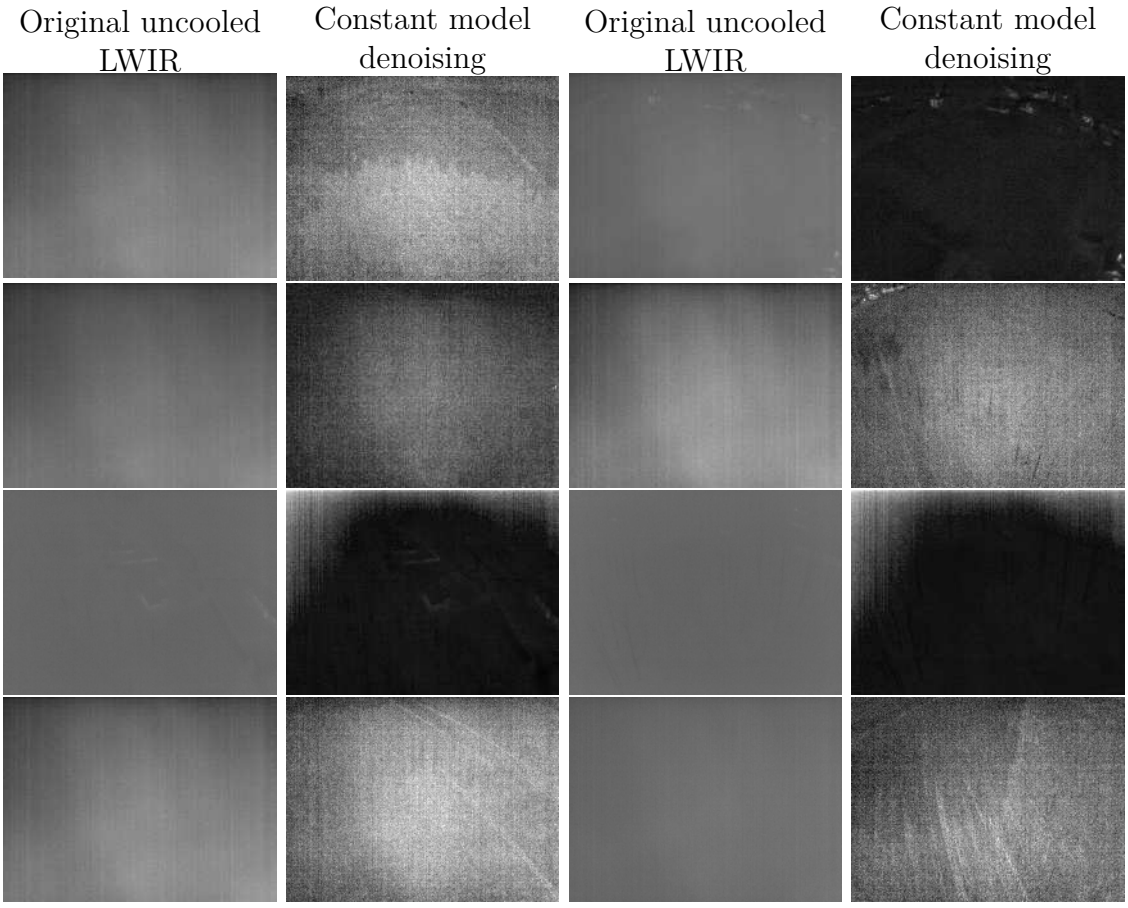


Figure B.1: Additional samples from denoising method. The images are ordered from best to worst (from top to bottom). The top row of images are well denoised without obvious artifacts. The image(s) to the right on the second row as well as the images on the last rows exhibit some "smearing" artifacts due to objects overlapping in consecutive frames. The images in the second to last row show regions where the denoising has worked well, but due to the FPN pattern shifting too rapidly, other areas are instead obscured by noise.

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY