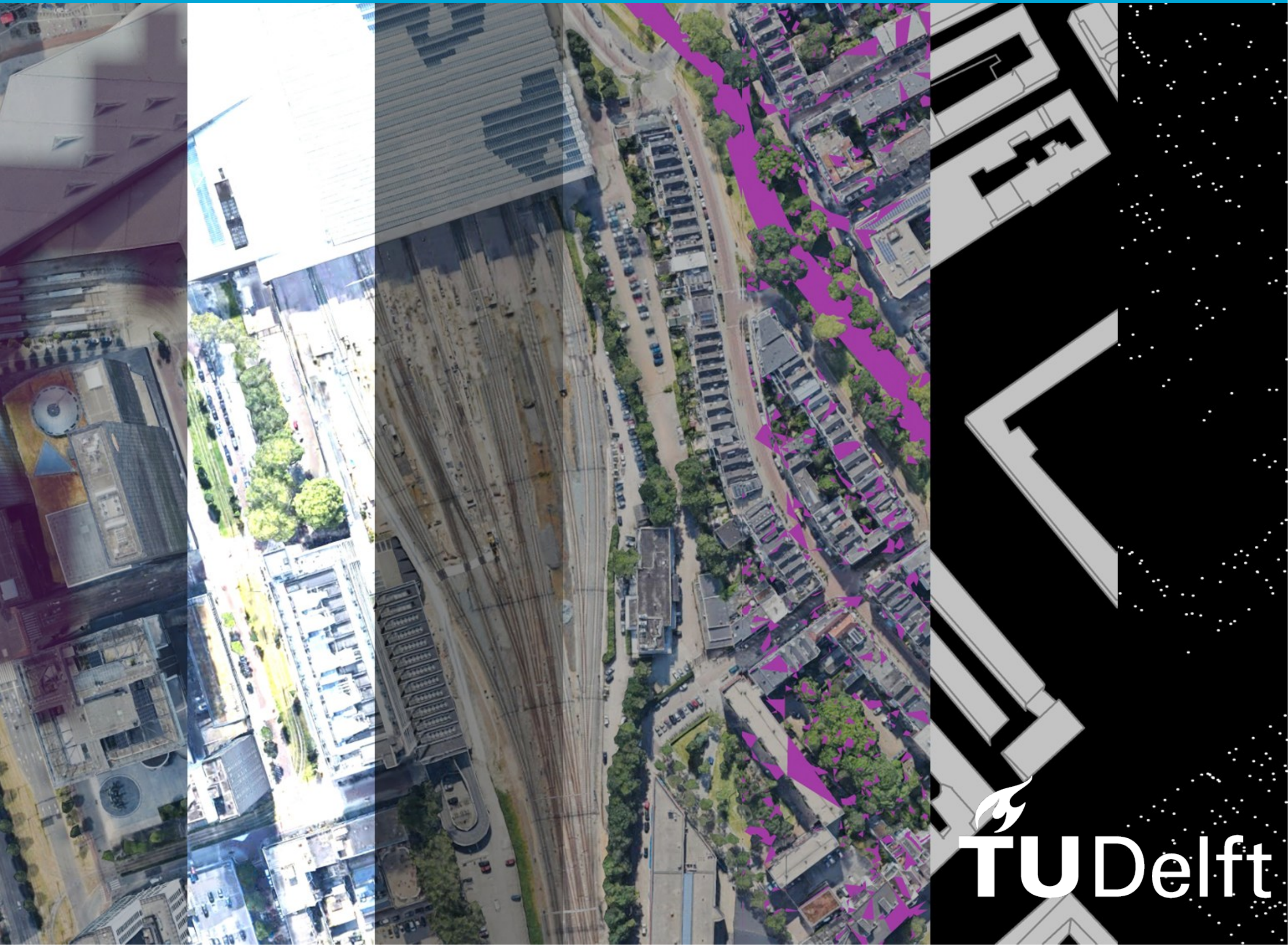
MSc thesis in Geomatics

Working Towards Oblique Aerial Adjustment through the Creation of Synthetic Testcases

Walter Hugo Johannes Kahn

2025



MSc thesis in Geomatics

Working Towards Oblique Aerial Adjustment through the Creation of Synthetic Testcases

Walter Hugo Johannes Kahn

June 2025

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

Walter Hugo Johannes Kahn: *Working Towards Oblique Aerial Adjustment through the Creation of Synthetic Testcases* (2025)

The work in this thesis was carried out in the:

Delft University of Technology
Geo-Database Management Centre
Julianalaan 134, 2628 BZ Delft, Zuid-Holland, The Netherlands

Ingenieursbureau Geodelta B.V.
Kanaalweg 4, 2628 EB Delft, Zuid-Holland, The Netherlands

Supervisors:	ir. E. Verbree
	Dr. ir. B.M. Meijers
Co-reader:	Dr. ir. A.A.Verhagen
Company supervisor:	ir. A. Verbraeck

Disclaimers

All aerial data used within this report (further referred to as the Miramap 2024 flight campaign) was collected by Miramap Aerial Surveys website Miramap during the 2024 flight season. This data was provided by the Municipality of Rotterdam municipal website, specifically the Stadsbeheer department and its Basisinformatie subdepartment, for the exclusive purpose of conducting the analyses required for this research. Consequently, the data cannot be redistributed by the author. Interested parties may request access to this dataset via the Municipality of Rotterdam's Open Data Portal website Open data Portal Rotterdam.

In addition, Google Maps 3D Tiles data was used in this research. This dataset is semi-open and can be accessed through the Google Maps API. To use this service, a valid and secure API key is required, which can be obtained via the Google Developer Portal Google API. Note that creating an API key requires a registered Google account with an associated credit card.

The coded implementation of the analytical framework developed by the author was created specifically for this thesis. This implementation is not intended for distribution and will not be made publicly available.

The thesis used ChatGPT and Grammarly as a form of spell checking.

The thesis contains a number of slide shows, which are best viewed using Adobe Acrobat. The thesis can be viewed using any

In this thesis, datasets have been used that were made available for research purposes by the Municipality of Rotterdam (Gemeente Rotterdam). The acquisition of these datasets was carried out by MiraMap on behalf of the Municipality of Rotterdam. The Municipality of Rotterdam is the data owner.

This thesis does not imply that the datasets are owned by Geodelta, nor that the datasets can be requested from Geodelta. Any requests regarding access to the original datasets should be directed to the Municipality of Rotterdam and/or MiraMap, in accordance with the applicable agreements and conditions.

This disclaimer is added after the thesis was assessed and uploaded, to clarify the provenance and ownership of the data used.

Abstract

This thesis explores the development and validation of a synthetic framework for oblique aerial image adjustment and object point detection, with the goal of improving photogrammetric workflows in complex urban environments. The research is motivated by the inherent challenges of oblique imagery, such as occlusion, perspective distortion, and variable visibility, which complicate traditional adjustment procedures. To address these issues, the study employs a novel approach by generating synthetic test cases that emulate real-world oblique aerial data, enabling controlled experiments and sensitivity analyses.

Utilizing data from recent aerial campaigns over Rotterdam, including both nadir and oblique images, the research implements and evaluates various adjustment and feature detection algorithms, including Bundle, DISK, SIFT, and LightGlue. The synthetic framework allows systematic testing of key parameters and environmental conditions, such as occlusion and lighting variations, providing insights into the robustness and limitations of different methods. Although the results demonstrate promising potential for synthetic data to replicate key geometric and photogrammetric behaviors, challenges remain in achieving full photorealism and seamless transferability to real-world applications.

The findings underscore the importance of synthetic data in advancing urban geospatial systems and support the early-stage design of aerial collection systems, with particular relevance for municipal maintenance, planning, and infrastructure management in the Netherlands. The study concludes with recommendations for future research directions, emphasizing the integration of more photorealistic synthetic imagery and improved synthetic-to-real transfer methods to enhance the accuracy and reliability of oblique aerial mapping workflows. Overall, this work contributes to the growing body of knowledge on synthetic data use in photogrammetry and opens pathways for more resilient and efficient urban mapping solutions.

Key words: HTW, Adjustment, Aerial Imagery, Synthetic Data

Acknowledgements

This thesis report partially reflects the results of my graduation project, completed at Ingenieursbureau Geodelta between November 2024 and June 2025, as part of the curriculum of the Master's in Geomatics program at Delft University of Technology. It also marks the end of nearly ten years of academic pursuit, including my Bachelor's in Geodesy at Hogeschool Utrecht and the Master's in Geomatics at TU Delft.

I could not have completed this thesis without the invaluable supervision of Edward Verbree and Martijn Meijers, who guided me conceptually throughout the process. I would also like to thank my co-reader, Sandra Verhagen, for her technical insights and valuable feedback during the P4 phase, which were instrumental in the final development of this thesis.

Furthermore, I would like to thank all my colleagues at Geodelta for their support and collaboration during this period. In particular, I am grateful to Annemieke Verbraeck for her general supervision, Jeroen Kappé for his technical assistance, and Sjoerd Staats for his guidance on the overall direction of the thesis and his help in acquiring the necessary data. A special thanks goes to Niels Treffers, who introduced the concept of synthetic adjustment, which served as the foundation for the framework developed in this research.

Finally, I would like to express my heartfelt gratitude to my family and friends for their continuous support and encouragement over the past decade.

Walter Hugo Johannes Kahn

Delft, June 2025

Contents

List of Figures

List of Tables

Acronyms

2D	Two Dimensional	18
3D	Three Dimensional	9
401	representing the nadir direction (corresponding to a 0 0 0/180 oriented camera)	18
402	representing the forward direction (corresponding to a 0 45 0/180 oriented camera)	19
403	representing the right direction (corresponding to a 90 45 0/180 oriented camera)	19
404	representing the backward direction (corresponding to a 180 45 0/180 oriented camera)	19
405	representing the left direction (corresponding to a -90 45 0/180 oriented camera)	19
AHN	Actueel Hoogtebestand Nederland	14
ALS	Aerial Laser Scanning	14
API	Application Programming Interface	35
AR	Augmented Reality	29
ASCII	American Standard Code for Information Interchange	29
BAG	Basisregristratie Adressen en Gebouwen	13
BSDF	Bidirectional Scattering Distribution Function	35
BGT	Basisregristratie Grootschalige Topografie	13
BLOSM	Blender Open Street Map	55
BIM	Building Information Model	29
BM5	Beeld Materiaal 5	14
BOR	Beheer Openbare Ruimte	13
CAD	Computer Aided Design	29
CM	Centimeters	20
CNN	Convolutional Neural Network	14
COLLADA	Collaborative Design Activity	29
COLMAP	The pipeline of Structure-From-Motion and Multi-View Stereo	34
CPU	Computer Processing Unit	35
DISK	DIScrete Keypoints	11
DOF	Degrees of Freedom	38
FBX	Filmbox	29
GCP	Ground Control Points	14
GPU	Graphics Processing Unit	35
GSD	Ground Sampling Distance	17
GNSS	Global Navigation Satellite System	14
GNSS-IMU	combination of a Global Navigation Satellite System and Inertial Motion Unit	13
HTW	Handleiding Technische Werkzaamheden	13
IMU	Inertial Motion Unit	22
KM	Kilometer	47
LightGlue	LightGlue Keypoint Matching	10
LOD	Level of Detail	29
MM	Millimeter	10
M	Meter	50
NAP	Normaal Amsterdams Peil	22
NDVI	Normalized Difference Vegetation Index	18
NIR	Near Infra Red	18
OBJ	Object File	29
OpenGL	Open Graphics Library	35
OSM	Open Street Map	10
PPA	Principle Point Average	23
RD	Rijksdriehoeksstelsel	22
RTK	Real Time Kineticism	48
SfM	Structure from Motion	9
SIFT	Scale-Invariant Feature Transformation	10
STL	Standard Tessellation Language	29
UI	User Interface	35
TPI	Terrestrial Panoramic Imagery	14
USD	Universal Scene Description	29
VR	Virtual Reality	29
WOZ	Waardering Ontroerende Zaken	13
XML	Extensible Markup Language	29

1 Introduction

Adjustment theory, from a geodetic perspective, is a deterministic statistical process used to fit sets of measurements by reducing stochastic noise and identifying and removing human errors. Various methods exist for collecting geodetic measurements; one widely used by Dutch municipalities is the collection of aerial imagery. These images support the maintenance of multiple geo-related base registration for which various Dutch governmental organizations are responsible, such as the Basisregristratie Grootschalige Topografie (BGT), Basisregristratie Adressen en Gebouwen (BAG), Beheer Openbare Ruimte (BOR), and Waardering Ontroerende Zaken (WOZ).

The aerial imagery is captured from airplanes at various angles, producing either nadir images (taken at a near-vertical angle, roughly 0°) or oblique images (taken at approximately 45°). Image adjustment is necessary because the position and orientation of the aircraft at the moment of image capture are subject to uncertainty.

Aircraft positioning and orientation data are typically gathered using an onboard combination of a Global Navigation Satellite System and Inertial Motion Unit (GNSS-IMU) system. Despite this, the observations are still affected by stochastic noise caused by positioning errors and sensor drift. Adjustment serves to mitigate this noise, resulting in a more stable measurement network and more accurate image placement. To date, this process has been successfully applied only to nadir imagery. Adjusting oblique images remains problematic due to connection difficulties, primarily caused by point occlusion, where objects in the built environment obscure key features.

The existing process for collecting and adjusting nadir aerial images is documented in Chapter 7 of the Handleiding Technische Werkzaamheden (HTW) [Polman and ir. M.A. Salzmann, 1996, p. 417], and this standard pipeline will be referred to throughout the thesis as the aerial imagery collection pipeline.

Due to their angled perspective, oblique images provide access to additional information, such as building façades, which can be valuable for mapping. However, since the HTW currently applies only to nadir imagery, the feasibility of establishing a corresponding pipeline for oblique imagery presents an important research question. This thesis aims to develop an oblique image processing framework modeled on the nadir aerial image collection pipeline as defined in the HTW. A semantic distinction is made here, differentiating pipeline and framework. The pipeline refers to a complete process that describes the choices that can be made, and a framework is the experimental process of working towards a pipeline.

To validate such a framework, it is necessary to design a set of mathematically defined test cases. These tests will assess the impact of key parameters involved in both the object point detection and adjustment stages. Lessons drawn from synthetic tests can then be applied to real-world data. The quality of the adjustment will be evaluated using the criteria described in Chapters 3 [Polman and ir. M.A. Salzmann, 1996, p. 61] and 4 [Polman and ir. M.A. Salzmann, 1996, p. 151] of the HTW.

1.1 Problem Statement

Currently, the collection and processing of oblique imagery is not described in the HTW. As a result, pipelines for oblique images often mimic those of nadir imagery, even though the parameters involved affect oblique imagery differently. This leads to unclear influences on image adjustment outcomes and complicates error analysis. Therefore, the primary goal of this thesis is to develop a framework for oblique image adjustment.

For such a framework to be considered successful, the images must be geometrically aligned—meaning the relative scale, rotation, and X, Y, Z positions of the camera sensors must be consistently related across the image network, without introducing warping, which is defined as the geometric distortion of an image to achieve alignment, but instead adjustment focuses on maintaining the integrity of the image structure. Thus, transformations are limited to uniform scaling, rotation, and translation.

In a theoretical ideal, the aircraft's GNSS-IMU would provide perfect position and orientation data, eliminating the need for adjustment. However, in reality, these measurements are subject to noise and drift, making adjustments necessary. While this adjustment process is well-established for nadir images, its extension to oblique images remains unexplored in practice.

Furthermore, image parameters depend not only on external sensors like the GNSS-IMU, but also on the camera's intrinsic parameters (e.g., focal length, principal point, lens distortion). These are themselves subject to noise. The HTW mandates annual camera calibration for nadir systems [Polman and ir. M.A. Salzmann, 1996, p. 435], but even calibrated systems introduce uncertainties that complicate adjustment.

Consequently, camera parameters must be included in the bundle adjustment, which is a form of non-linear least squares optimization used in aerial image processing. This increases the number of variables and the complexity of solving the system. Occlusion, especially prevalent in oblique imagery, presents another challenge: when object points are obstructed, they cannot be matched across images, weakening the overall network geometry. Addressing these interconnected challenges is essential before a standardized workflow can be proposed.

Once the images are collected, a key step is the detection and matching of object points, which are used in the system construction, traditionally performed using feature-based algorithms such as SIFT. These methods offer a degree of interpretability and allow users to trace why particular matches were made. However, recent developments in deep learning have led to the adoption of feature detectors like DISK, which rely on Convolutional Neural Network (CNN)-based architectures. While these models often outperform traditional methods, they function as "black boxes," making their internal decision-making hard to interpret. This limits transparency and complicates debugging, which is particularly problematic in precision-critical tasks such as aerial image adjustment.

A final challenge in developing an oblique image framework lies in data availability. Aerial data collection is expensive and complex, and uncontrolled environments introduce variables that can obscure cause-and-effect relationships. A viable solution is to begin with synthetic data. This enables the creation of controlled, noise-free scenarios where different types of noise and errors can be introduced systematically, allowing for targeted analysis of their effects.

1.2 Project Relevance

The development of an oblique image collection framework has broader implications beyond governmental base registration maintenance. The insights gained from this research can contribute to various geo-spatial applications, not all of which are airborne.

One such application is the design of signal plans [Polman and ir. M.A. Salzmann, 1996, p. 420], which includes Beeld Materiaal 5 (BM5) points maintained by the Dutch Kadaster as well as local control points. These serve as Ground Control Points (GCP)s (see Section 2.10) used during the second phase of adjustment, where the relative image network is transformed into real-world coordinates, thus allowing for real-world mapping. Currently, most GCPs are optimized for nadir imagery and may be inadequate for oblique adjustment.

The methodology proposed in this thesis allows for the testing and validation of oblique-compatible GCPs, enhancing the feasibility and reliability of an oblique adjustment workflow. This, in turn, supports the broader applicability of oblique imagery in both public and private geospatial initiatives.

Another application for a successful oblique framework is related to the Aerial Laser Scanning (ALS) that is collected for the Dutch Government, called the Actueel Hoogtebestand Nederland (AHN) [Rijkswaterstaat, 2021]. Currently, the AHN is available with colors. The coloring takes place using the Beeld Materiaal nadir adjusted images collected by the Dutch Kadaster. The usage of nadir leads to the sides of buildings being incorrectly colored due to the building side occlusion present in nadir. The use of oblique images would solve this problem, but would require successful adjustment of oblique images to be done to reflect the actual position of colors.

Another application that relates to its use in ALS scanning is more abstract. Certain adversarial countries cause Global Navigation Satellite System (GNSS) interference along their shared border. This interference causes GNSS observations along a border to become unusable when trying to reconstruct the point cloud. This problem could be solved by letting the positioning take place through the reconstruction of a photogrammetric system using SfM. This SfM could then be used to adjust oblique images which are taken at the same time as the ALS, thus allowing the continuation of laser scanning in such vital areas.

Another form of adjustment in photogrammetry, which has so far been less explored, is in Terrestrial Panoramic Imagery (TPI). The successful creation of such a pipeline would lead to similar advantages as the adjustment of oblique imagery. Meaning that base registration could be maintained using even more methods, including the case that foliage might provide full cover regardless of aerial angle. Having such a case could be solved by street-based imagery. This research could be used to help towards the creation of such a framework.

The final advantage brought forward by the research is that the creation of the synthetic test data would allow for the testing of object point detection algorithms and the testing of Bundle algorithm extensions.

1.3 Research Scope and Objectives

The full nadir aerial imagery collection pipeline is explained in Section 2 and delves into the different aspects of its components, which are necessary for the completion of the research. The most important part is which parameters influence the pipeline. This, in combination with the various problems identified before, led to the creation of the following research question:

What parameters are key in the reconstruction accuracy of adjustment theory applied to RGB aerial images using synthetic test cases?

This research question covers two different aspects of the pipeline that are identified as being key initial steps in the creation of an oblique framework. To accomplish the steps for both the detection and adjustment steps of the research, the main question has been split into different phases (see Section 3). These phases each deal with their stage of the framework and can be used to answer various sub-questions stemming from the main question. Each sub-question might be related to multiple phases. The RGB in the main question refers to the three-channel colors Red Green and Blue.

The sub-questions are as follows:

- What is the mathematical pipeline for aerial adjustment theory?
- What hyperparameters are present in aerial adjustment, and what are their effects when changed in synthetic tests?
- What are the main differences between nadir and oblique image adjustment?
- How do the different parameters of oblique imagery influence the errors present in the final adjusted images?
- How does the choice between object point extraction algorithms such as SIFT/LightGlue and DISK/LightGlue affect the reconstruction accuracy?
- How do simulated external factors such as sun position, wind, clouds, and foliage affect the adjustment?

The objective of the research is to create a starting workflow for adjustment and object point detection aimed at oblique imagery using various test cases at the hand of self-created synthetic data, which is done to allow for the testing through statistical independence of the various parameters that are associated with each stage. The research will not delve into the creation of an adjustment algorithm or the further development of a detection and matching algorithm, as those already exist. Instead, the choice has been made for more commonly used algorithms such as Bundle, DISK, SIFT, and LightGlue, each of which is explained in Section 2. Furthermore, as mentioned previously, adjustment entails merely the scaling, rotating, and translation of images, not warping (changing the geometric structure of an image) or morphing (combining two images). This distinction is made as it would remove the measurability of the images.

1.4 Advantages of Synthetic Data

Most of the sub-questions focus on the use of synthetic data as a means to solve the problems tackled through the research. The choice for synthetic data mainly stems from the ability to generate test cases on a variable independence level. Furthermore, the use of synthetic data also allows for the mimicking of edge cases.

The conceptual idea can be metaphorically described using two trees as shown in Figure 1. On the left is an image of a real tree that represents a complex system of leaves, a trunk, and roots, all interconnected to represent what we call a tree. On the left is an artistic rendition of a tree, a so-called synthetic tree that also has leaves, a trunk, and roots, but is a made-up representation of a tree. The real tree on the left is more difficult to alter as its interconnected system is more constrained. Elements can be removed but not necessarily added, same for moving parts of the tree. However, the synthetic tree does allow for changes to be made, meaning that even though it is a modeled tree, more kinds of trees/ systems can be modeled. This same logic applies to the methodological creation of a synthetic aerial workflow as proposed in Chapter 3.

Figure 1: Analogous example of the advantages of synthetic data

1.5 Focus Area

While synthetic data offers a promising foundation, meaningful validation requires applying the developed workflow to real-world data. Only through such comparison can it be determined whether the synthetic approach merits further development. For this purpose, data from the Miramap 2024 aerial campaign over Rotterdam (provided by Geodelta, Miramap, and the municipality of Rotterdam) has been made available. The campaign took place in late March and early April 2024 and captured approximately 75,000 images, including both nadir and oblique views. As this research is a proof of concept, a smaller subset of around 800 images was selected from a 4 km^2 area in central Rotterdam. This location was chosen due to its complex urban features, including tall buildings and the River Maas, which present realistic challenges such as occlusion and weak image connections. Agricultural and mountainous areas were excluded, as they lack the dense structures necessary to replicate these urban-specific issues.

1.6 Thesis Outline

The thesis report is divided into multiple chapters, the first of which is the introduction, which is used to explain the basic concepts of the research and the problems that need to be solved. Chapter 2 lays down a foundation of related work and commonly used terms for the understanding of research. Chapter 3 explains the proposed novel methods to solve the sub-questions. Chapter 4 shows the result of the different methods, and Chapter 5 gives phase-separated case-by-case analysis of the produced data, which results from the adjustment. Chapter 6 and Chapter 8 discuss the research and provide conclusions to it.

2 Related Work and Theoretical Background

Chapter 2 describes the related work and theoretical background in which the experimental research is placed. This is used to lay the groundwork that is needed in Chapters 3, 4, and 5 to answer and relate the main and sub-research questions. Each section defines a separate element of the required background. In addition to the related work, a glossary is located at the end of the thesis for quick hand review of discussed topics.

2.1 The Aerial Imagery Collection Pipeline

This first section provides an overview of the photogrammetric production pipeline used in large-scale topographic data acquisition of the Dutch Kadaster through aerial imagery encoded in the HTW. The pipeline consists of three primary steps:

1. Aerial imagery planning and collection
2. Photogrammetric object point detection, matching and adjustment
3. Photogrammetric detail measurement

Each step plays a crucial role in ensuring the accuracy and consistency of the resulting digital maps.

The collection process begins with the selection of a target area to be photographed at predefined scales Ground Sampling Distance (GSD) (see Section 2.9). Before the aerial survey, GCP's are chosen (see Section 2.10), or must be identified. These points serve as spatial references for geo-referencing the imagery and creating an adjustment system (see Section 2.21).

The points are established either terrestrially or via detection and must be visibly identifiable on the aerial images. These are either:

- *Control points*: Marked in the field before the survey, typically using ground disks (see Section 2.10).
- *Object points*: Identifiable features chosen post-survey through image detection from the images themselves (see Section 2.11).

To ensure all control points are visible, a signal plan (see Section 2.4) is created, detailing the locations and types of control points. Based on this a flight plan is created which entails the planned flight lines that define how the area will be captured in overlapping image strips. The corresponding flight plan (see Section 2.4) is plotted over a topographic base map and used during the aerial survey using calibrated photogrammetric cameras.

Following image acquisition, the imagery is processed to relate overlapping photos through aerial triangulation and block adjustment geometrically. This phase reconstructs the spatial geometry of the image block as it existed during the flight.

Key components include:

- Aerial Triangulation: Measurement of control points and connecting features across overlapping photos (see Section 2.12 and Section 2.18).
- Block Adjustment: Least-squares adjustment of all image coordinates, incorporating control point constraints to align the block with a national coordinate system (see Section 2.20 and Section 2.21).

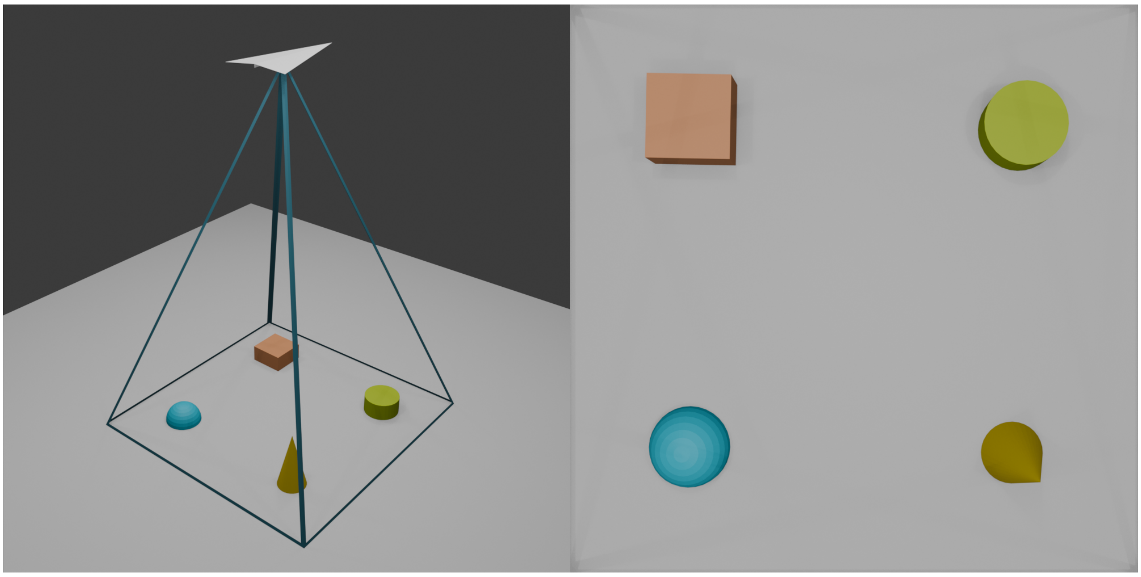
The result is a densified control network, which serves as the geometric basis for further detailed measurements.

Measurements are performed using overlapping photo pairs, also known as stereo pairs. Through a process of relative and absolute orientation, these models are aligned internally in what is called the first phase of adjustment (see Section 2.20) and then georeferenced to the terrestrial coordinate system in what is called the second phase of adjustment (see Section 2.21).

Photogrammetry is also used for maintaining existing topographic datasets (see Section 1). Stereoscopic inspection tools allow for the projection of the existing map into the stereo model, making any changes in the topography immediately visible.

2.2 Nadir Imagery

As mentioned in Chapter 1, this research focuses on the two most common types of aerial imagery collected for Dutch municipalities. The first of which are nadir images, so named for being taken at zero-degree angle facing away from the airplane along the nadir axis, pointed towards the center of the photographed mass (Earth). A simple example scene of what nadir imagery capture looks like is given in Figure 2a and Figure 2b. Most plane-based aerial images are taken as three-channel 24-bit colored images, the three 8-bit channels corresponding to R G B (Red, Green and Blue) colour scales.



(a) Overview of a Singular Nadir Image Frustum Projected over a Surface with Sightlines

(b) Singular Nadir Image

Possible use cases for nadir imagery are described in [Pang et al., 2023] and [Wang et al., 2022], which often involve mapping the footprints of buildings within the built environment from above. The HTW describes this process as being applied to the maintenance of Dutch base registrations such as the GBKN (Grootschalige Basis Kaart Nederland). Nowadays, nadir imagery is being applied by Dutch municipalities for the maintenance of the current Dutch base registration (BGT, BAG, BOR, and WOZ) [Dufour et al., 2022].

Other, more modern nadir applications are described in [Shorter and Kasparis, 2009], which outline the use of nadir imagery to detect vegetation and its change over time. Another vegetation-based application for nadir imagery is to use the R channel in combination with Near Infra Red (NIR) imagery to create Normalized Difference Vegetation Index (NDVI) imagery, which can be used to map the healthiness of vegetation in a photographed region [Tucker, 1979].

One of the advantages that aerial imagery offers, when compared to more traditional terrestrial methods, is its lower cost of collection versus a full year of terrestrial measurement. One of the drawbacks of nadir imagery is the occlusion of building facades (see Section 2.19). The mapping of facades is currently not part of the base registrations, as they currently only represent Two Dimensional (2D) data [Dukai et al., 2024]. This, might however, change as more mapping applications require the use of 3D data to model the real world correctly.

The nadir images are marked in the research using the following convention:

- 401 representing the nadir direction (corresponding to a 0 0 0/180 oriented camera) (401)

This naming convention is based on the Miramap 2024 Rotterdam flight campaign.

2.3 Oblique Imagery

The facade-based drawback described in Section 2.2 can be alleviated by expanding the collection of imagery through oblique images. Oblique images are taken at a 45° angle relative to the photographed mass (Earth) [Höhle, 2008]. The data collected for Dutch municipalities is often captured in four directions at the same time. This results in each oblique image set consisting of at least four subsets. Similar to nadir, this situation can be portrayed using a simplified overview given in Figure 3.

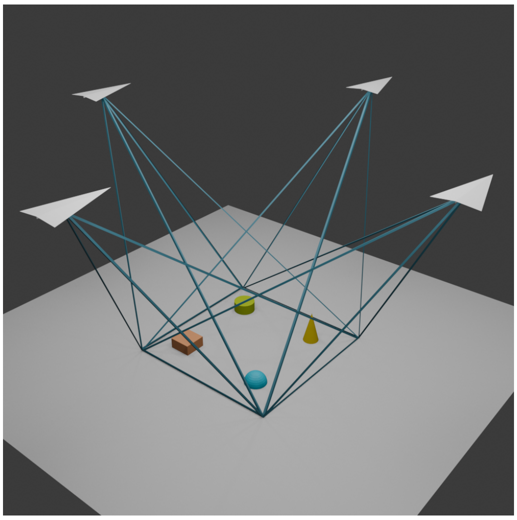
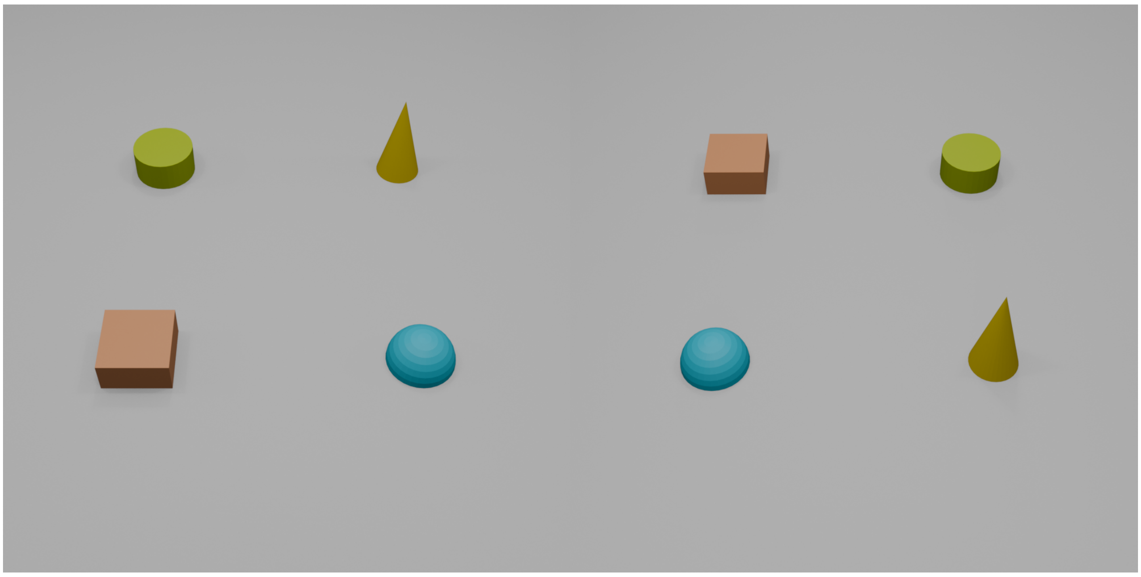


Figure 3: Overview of a Four-Oblique Image Frustum Projected over a Surface with Sightlines

The image subsets relate to the camera names. For this research, the following oblique naming convention is used:

- 402 representing the forward direction (corresponding to a 0 45 0/180 oriented camera) (402)
- 403 representing the right direction (corresponding to a 90 45 0/180 oriented camera) (403)
- 404 representing the backward direction (corresponding to a 180 45 0/180 oriented camera) (404)
- 405 representing the left direction (corresponding to a -90 45 0/180 oriented camera) (405)

This naming convention is based on the Miramap 2024 Rotterdam flight campaign.



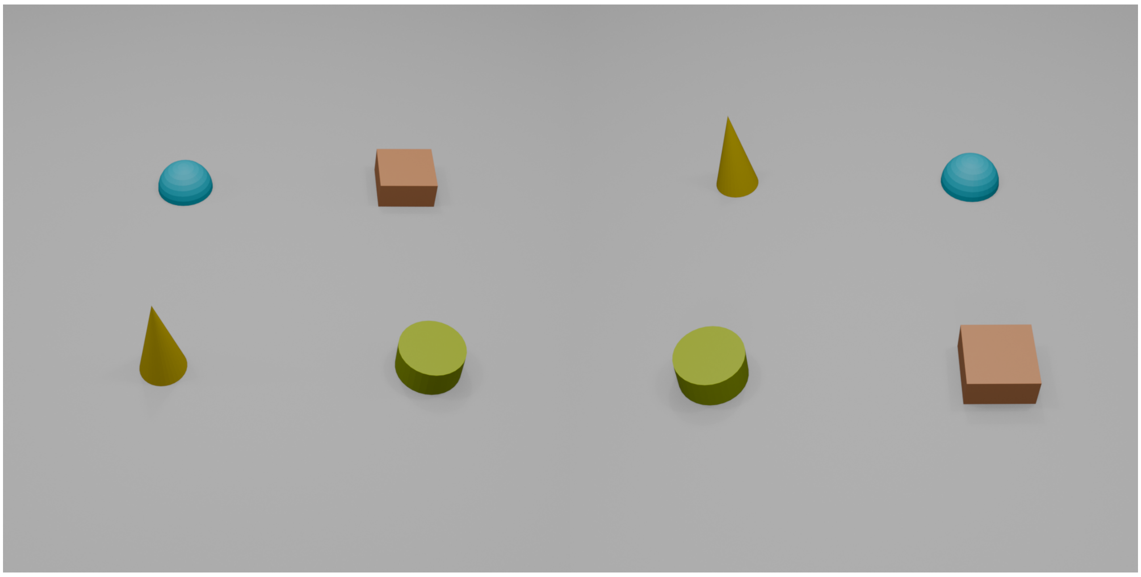
(a) Forward Oriented (402) Oblique Image Based on the Overview given in Figure 3

(b) Right Oriented (403) Oblique Image Based on the Overview given in Figure 3

The main advantage of oblique images, as mentioned before, comes from the angle at which the image is taken, allowing the user to see the sides or facades of buildings. Where nadir offers an almost 2D view and can thus be rarely used for 3D, an oblique image enables preservation activities as described in [Höhle, 2013]. It also finds its use in the creation and extraction of 3D city models, as described in [Haala et al., 2015].

Even though the base registrations are not in 3D as of writing, certain Dutch government organizations have already updated parts of the BGT to also include *z*-coordinates for some objects. In the case of this research, oblique images are most interesting because their adjustment is more complex compared to nadir. Furthermore, the correctness of adjustment within oblique imagery is not fully known [Haala et al., 2015]. Another issue is occlusion, which leads to a more challenging system construction Section 2.19.

An example of oblique imagery adjustment being applied is in land management systems such as the BGT. This is described in [Lemmens et al., 2007], which characterizes aerial oblique adjustment as being measurable. However, when compared to nadir images, it shows an accuracy of 90 Centimeters (CM). Modern results show better outcomes [Haala et al., 2015]. These results, however, do not yet mean the full requirements set forth by the HTW [Polman and ir. M.A. Salzmann, 1996].



(a) Backward-Oriented (404) Oblique Image Based on the Overview given in Figure 3

(b) Left-Oriented (405) Oblique Image Based on the Overview given in Figure 3

2.4 Flight Plan and Signal Plan

One of the first steps in the aerial imagery collection pipeline (see Section 2.1), according to the HTW, is the creation of a flight and signal plan [Polman and ir. M.A. Salzmann, 1996, p. 420]. The flight plan entails the planned route along which aerial images are captured to ensure proper coverage over the designated area. The flight plan used by Miramap in 2024 for the Miramap 2024 Rotterdam flight campaign is shown in Figure 6.

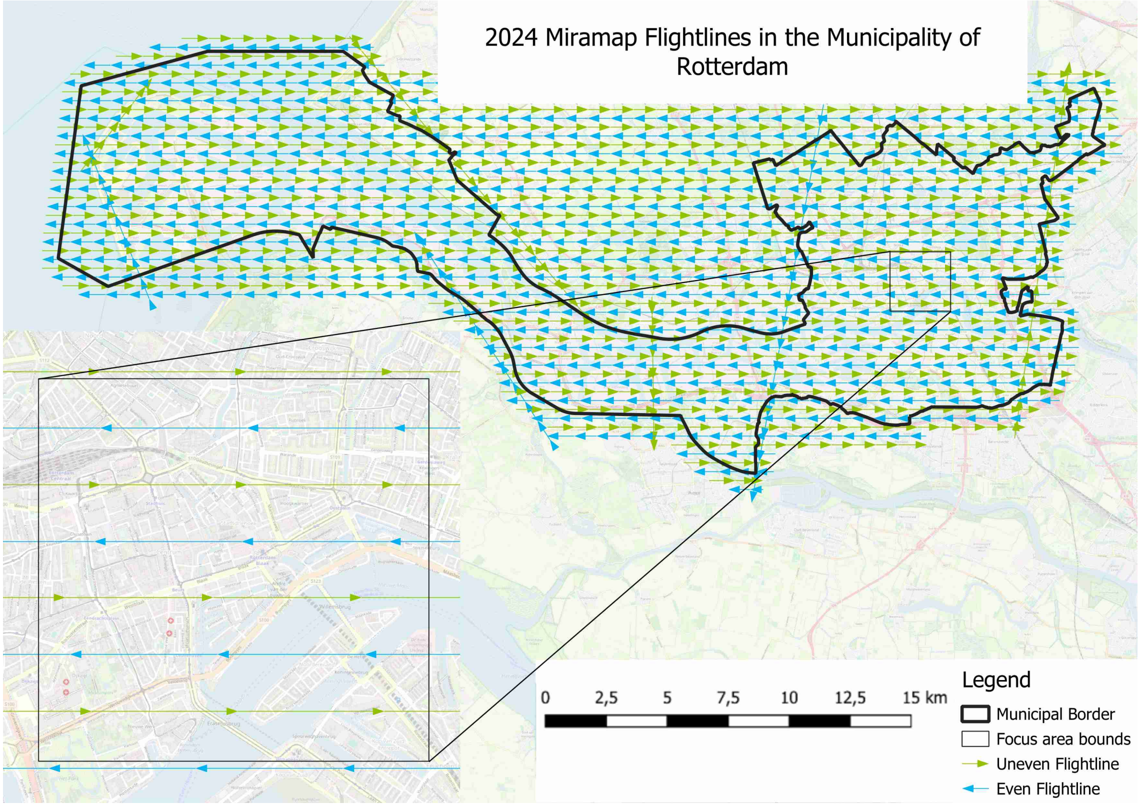
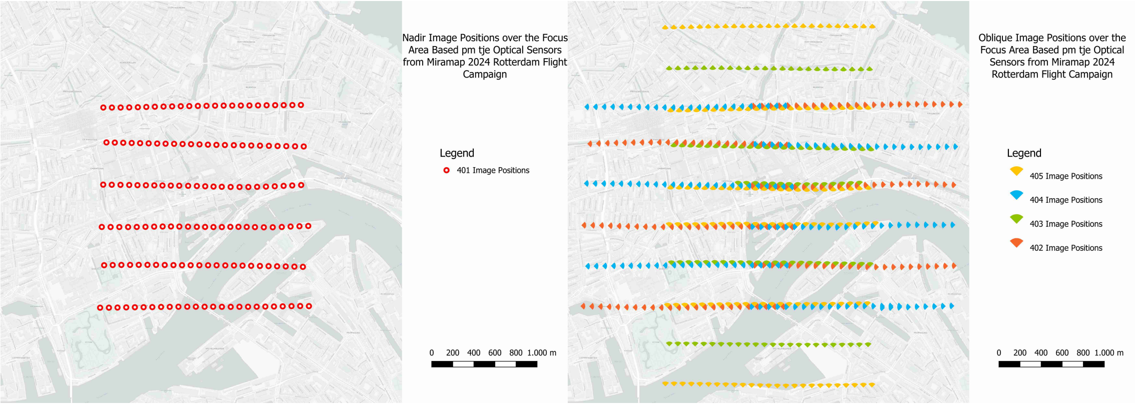


Figure 6: Miramap 2024 Rotterdam Flight Campaign Flight Plan for the Municipality of Rotterdam

In Figure 6, the even and odd numbered flight lines are shown, which affects the rotation of the physical camera (see Section 2.5) and the image center direction. The actual nadir and oblique image positions over the focus area are shown in Figure 7a and Figure 7b, respectively.


Nadir Image Positions over the Focus Area Based pm tje Optical Sensors from Miramap 2024 Rotterdam Flight Campaign
Legend
401 Image Positions
0 200 400 600 800 1.000 m
Oblique Image Positions over the Focus Area Based pm tje Optical Sensors from Miramap 2024 Rotterdam Flight Campaign
Legend
405 Image Positions
404 Image Positions
403 Image Positions
402 Image Positions
0 200 400 600 800 1.000 m

(a) Nadir Image Positions over the Focus Area

(b) Oblique Image Positions over the Focus Area

The signal plan refers to the positions of so-called GCP (see Section 2.10) and how they are used and maintained to allow for the real-world connection in second-phase adjustment (see Section 2.21).

2.5 Extrinsic Parameters of the Camera

The extrinsic camera parameters define the location of the camera and the angular orientation. The extrinsic parameters are paired with the intrinsic parameters (see Section Section 2.8). The extrinsic parameters consist of two sets of sub-parameters. The X, Y, and Z coordinates of a camera define its position in 3D space (For Dutch aerial imagery X and Y are defined by the Rijksdriehoeksstelsel (RD)-New coordinate reference system (EPSG:28992) which the Dutch Kadaster maintains, and Z which is defined by Normaal Amsterdams Peil (NAP)) as a relative height system maintained by Rijkswaterstaat (EPSG:5709). These coordinates specify where the camera is located in the environment:

- X represents the position along the horizontal axis (left-right)
- Y represents the position along the vertical axis (up-down)
- Z represents the depth axis (forward-backward)

The second set of sub-parameters of the optical center defines the rotation of the camera. The HTW describes these rotation angles as ω (Omega) ϕ (Phi) κ (Kappa) sometimes called $O\ P\ K$ for short [Polman and ir. M.A. Salzmann, 1996]. These correspond to $R_X\ R_Y\ R_Z$ rotation matrices. These matrices, from [Polman and ir. M.A. Salzmann, 1996, p. 460], are as follows:

$$\text{Omega rotation matrix: } R_x(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\omega) & -sin(\omega) \\ 0 & sin(\omega) & cos(\omega) \end{bmatrix}$$

$$\text{Phi rotation matrix: } R_y(\phi) = \begin{bmatrix} cos(\phi) & 0 & sin(\phi) \\ 0 & 1 & 0 \\ -sin(\phi) & 0 & cos(\phi) \end{bmatrix}$$

$$\text{Kappa rotation matrix: } R_z(\kappa) = \begin{bmatrix} cos(\kappa) & -sin(\kappa) & 0 \\ sin(\kappa) & cos(\kappa) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This results in the combined matrix given in Equation 1.

$$R_z(\kappa)R_y(\phi)R_z(\omega) =$$

$$\begin{bmatrix} cos(\kappa)cos(\phi) & -sin(\kappa)cos(\omega) + cos(\kappa)sin(\phi)sin(\omega) & sin(\phi)sin(\omega) + cos(\kappa)sin(\phi)cos(\omega) \\ sin(\kappa)cos(\phi) & cos(\kappa)cos(\omega) + sin(\kappa)sin(\phi)sin(\omega) & -cos(\kappa)sin(\omega) + sin(\kappa)sin(\phi)cos(\omega) \\ -sin(\phi) & cos(\phi)sin(\omega) & cos(\phi)cos(\omega) \end{bmatrix} \tag{1}$$

2.6 Global Navigational Satellite System (GNSS)-Inertial Motion Unit (IMU)

To determine the external position (see Section 2.5), as mentioned before, each plane is equipped with a GNSS-IMU [Kok et al., 2017]. The combination of these occurs through a Kalman filter [Kalman, 1960]. The GNSS provides absolute positioning and timing information by using signals from a constellation of satellites orbiting the Earth [Service, 2025]. Systems like GPS (USA), Galileo (EU), GLONASS (Russia), and BeiDou (China) transmit signals from satellites that a GNSS receiver onboard an aircraft can use to calculate its location on the globe. In aviation, GNSS is used for determining the aircraft's position in real time, supporting tasks like navigation, flight control, and geo-referencing.

However, GNSS signals can be degraded or temporarily unavailable due to atmospheric conditions [Misra and Enge, 2006], signal obstructions, or multipath effects [Groves, 2013], which can reduce their reliability in certain environments (see Section 2.7).

An Inertial Motion Unit (IMU) is a self-contained sensor system that measures linear acceleration and angular velocity using accelerometers and gyroscopes. Unlike GNSS, the IMU does not depend on external signals [Kok et al., 2017], which makes it highly valuable in situations where satellite signals are weak or blocked. It can track rapid motion and orientation changes with high frequency and precision, but it suffers from drift over time [Titterton and Weston, 2004]. This means that small measurement errors can accumulate, leading to increasingly inaccurate position estimates if used alone.

To overcome the limitations of both systems. GNSS and IMU data are often combined using a Kalman filter [Kalman, 1960], a mathematical algorithm that continuously estimates the most probable position, velocity, and orientation of the aircraft. The Kalman filter fuses the high-frequency, short-term accuracy of the IMU with the long-term stability of GNSS data, correcting for the IMU drift and filling in gaps during GNSS outages. This sensor fusion approach results in a navigation solution that is more robust and accurate when compared to its counterparts. It is widely used in manned and unmanned aircraft for precise flight control, aerial surveying, and sensor data geo-referencing.

2.7 Stochastic Errors

As mentioned in Chapter 1, different observations associated with aerial photogrammetry encoded within the full pipeline (see Section 2.1) are subject to stochastic noise. These stochastic errors can occur in the extrinsic parameters (see Section 2.5) due to GNSS-IMU (see Section 2.6) and the intrinsic parameters (see Section 2.8). These are also known as random errors [Orellana et al., 2021], meaning that a standard deviation of unpredictable variations in data or measurements is present. This arises from random and uncontrollable factors present during the flight. These errors occur due to inherent fluctuations in experimental conditions, limitations in measurement instruments, or unpredictable changes in the environment, and they affect the precision but not necessarily the accuracy of results.

Unlike systematic errors, stochastic errors do not follow a consistent deviation pattern and tend to average out over multiple observations. This is sometimes also referred to as Gaussian noise [Orellana et al., 2021]. In statistical models, they are typically represented as random variables with a mean of zero, capturing the noise or variability not explained by the deterministic part of the model [Orellana et al., 2021].

2.8 Intrinsic Parameters of the Camera

To allow for synthetic test case generation, the physical object of a camera needs to be modeled using a variety of parameters. This is important to mimic the detection and matching algorithms (see Section 2.15), perform the observation calculations (see Section 2.12), and determine the image plane and projected footprints (see Section 2.9). The intrinsic parameters are as follows:

- Sensor width and height (pixels) [dr. ir. M.G. Vosselman, 1995, p. 25]
- Pixel width and height in micrometers (μm)
- Focal length in MM [dr. ir. M.G. Vosselman, 1995, p. 12]
- Principle Point Average (PPA) (in MM)

The synthetic parameters used based on the Miramap 2024 Rotterdam flight campaign camera parameters are given in Section 3.2.

The sensor width and height refers to the number of pixels on the camera sensor, which for the type of aerial photography that falls within the scope are represented by R, G or B colour sensors [dr. ir. M.G. Vosselman, 1995, p. 25]. Each pixel has a width and height (often the same in modern cameras to create square pixels). The physical size of the sensor within the camera is the number of pixels times the width and height. Figure 8 illustrates the sensor-related parameters.

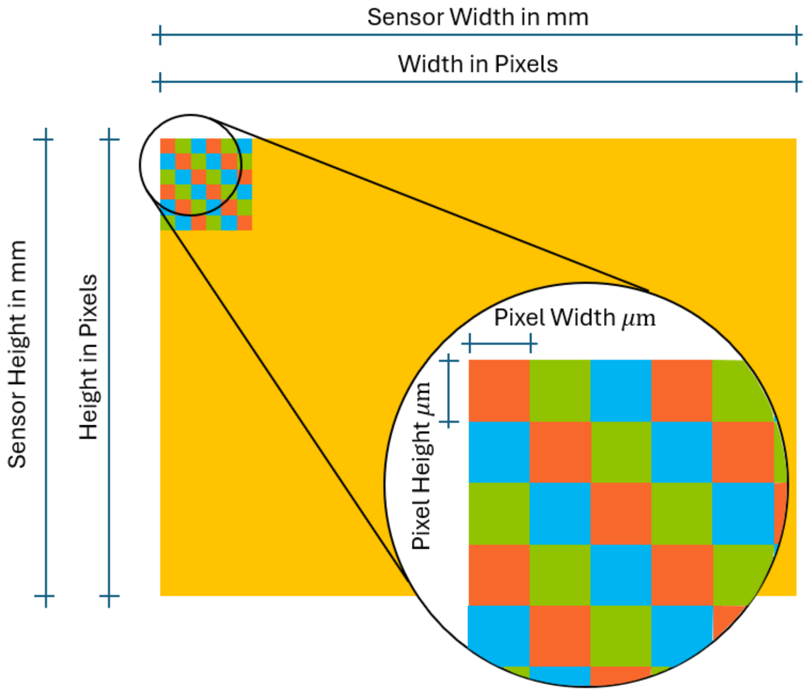


Figure 8: Sensor Related Parameters of a Camera Model

The focal distance [dr. ir. M.G. Vosselman, 1995, p. 12] defines the orthogonal distance between the sensor and the lens. Furthermore, the choice of focal distance defines the angle of the view for the camera and thus influences the footprint (see Section 2.9). Figure 9 illustrates the formation of the focal distance as it relates to a singular-lensed camera model.

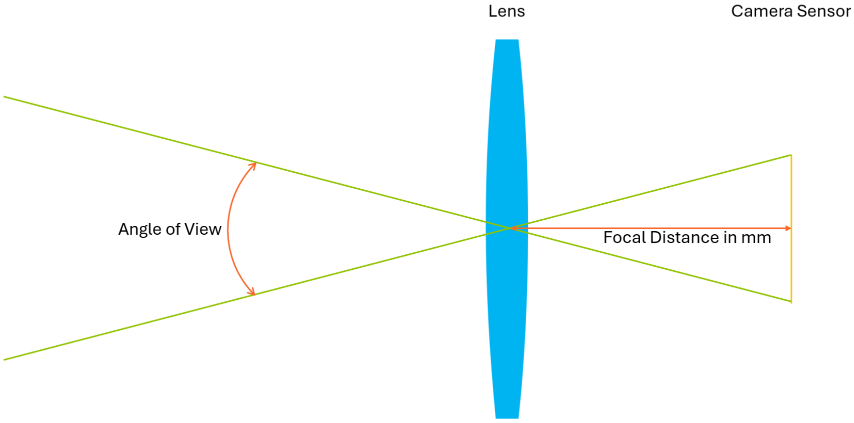


Figure 9: Schematic Overview of the Camera System Consisting of the Camera Sensor and Camera Lens, which Determines the Focal Distance in MM and the Angle of View in Degrees

The sensor direction [dr. ir. M.G. Vosselman, 1995, p. 26] refers to the rotation of the camera sensor and how the pixel coordinate axes are defined. The formulation to calculate these pixel coordinates is given in (see Section 2.12). The four types of available sensor directions are illustrated in Figure 10.

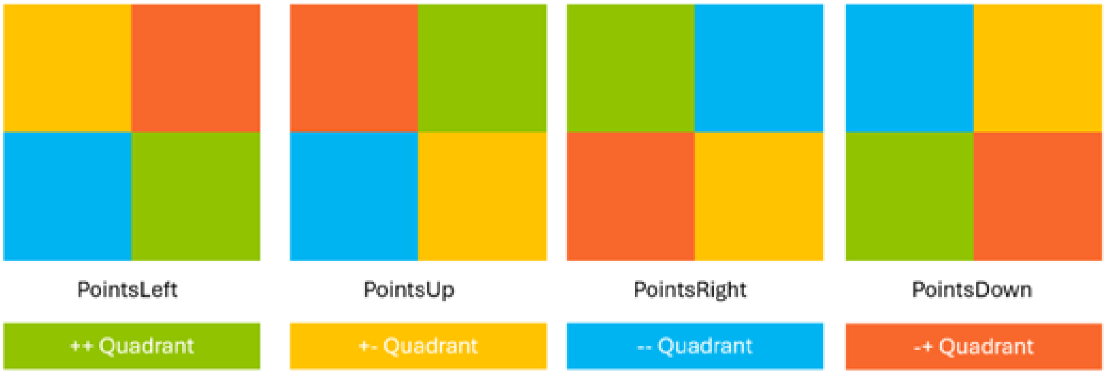


Figure 10: The Photo Axis Presets and their Quadrant Positions Available in the Geodelta Bundle Adjustment Software

The most basic camera model [dr. ir. M.G. Vosselman, 1995, p. 25 and 27] states that the camera sensor and lens distance, which defines the focal distance, should be orthogonal. This orthogonality is, however, not perfect, as such a slight offset might occur, which affects the distortion of a camera model. This Principle Point Average (PPA) [dr. ir. M.G. Vosselman, 1995, p. 37] can take place both over x and y and shall henceforth be referred to as PPAx and PPAy.

2.9 Image Plane and Image Footprint

The image plane is a theoretical plane projected by the camera system. The image plane lies orthogonal to the optical axis (also spanned by the Focal Distance (see Section 2.8) and is typically located in front of the sensor [Bracewell, 2003]. It can also be referred to as the positive plane [Vosselman, 1996, p. 13]. In photography and digital imaging, the image plane corresponds to the captured recording of the image [Vosselman, 1996, p. 23]. Precise alignment of the image plane is critical to ensure clarity and sharpness in captured images, as deviations can cause blurring or distortion [Bracewell, 2003].

The area captured on the image plane can be represented in world coordinates as a footprint. This footprint is based on the optical center parameters (see Section 2.5) and the camera model parameters (see Section 2.8). The shape of the footprint is dependent on the imaging angle: Nadir produces a roughly rectangular footprint, while oblique imagery creates a trapezoidal shape that extends away from the camera. The footprint shapes are illustrated in Figure 11.

Figure 11: The Real World Footprints Created by Nadir and Oblique Aerial Imagery

The projection of each image plane pixel on the ground is the GSD. The GSD is determined beforehand by municipalities based on the quality requirements of the aerial imagery application (see Section 2.1). This choice is made because GSD directly relates to idealization precision in the images. The GSD for nadir images is more consistent when compared to oblique images, due to the more uniform imaging angle.

2.10 Control Points

Aerial adjustment networks are geodetic networks [Polman and ir. M.A. Salzmann, 1996, p. 253] composed of cameras defined by their optical centers and the parameters encoded in the camera model (see Section 2.8), which are connected to sets of points representing 3D calculated points, preferably visible in multiple images.

These points are divided into two categories. The first category consists of GCP, hereafter referred to as control points [Polman and ir. M.A. Salzmann, 1996, p. 667]. These points are exclusively used in the second phase adjustment (See Section 2.21) [Polman and ir. M.A. Salzmann, 1996, p. 153] to transform the network uniformly, orthogonally, or affinely into a "global" coordinate system [Polman and ir. M.A. Salzmann, 1996, p. 158]. Control points are usually selected during the formulation of the signal plan (see Section 2.4) [Polman and ir. M.A. Salzmann, 1996]. Their selection is based on having a clear skyward sight line, ensuring visibility from multiple images to enable the connection of those images.

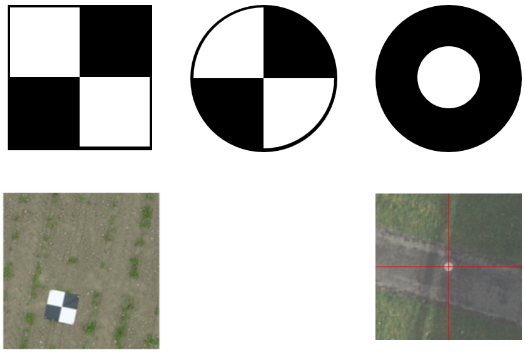


Figure 12: Ground Control Point Locations over the Focus Area

Figure 12 shows three types of control points commonly used throughout the Netherlands. Control points are based on the principle that the center of the marker represents the known coordinate. These coordinates are placed on stationary objects to allow for repeated use across temporal epochs Polman and ir. M.A. Salzmann [1996], and are measured using terrestrial methods such as a GNSS receiver or Total Station.

The focus area (see Section 1.5) did not include any existing control points, either from the municipal set or the BM5 network. Therefore, 9 points were manually selected to enable second-phase adjustment. This approach aligns with the fitting requirements proposed by [Polman and ir. M.A. Salzmann, 1996, p. 158]. The methodology and rationale for this selection are further discussed in Section 3 and Section 4. The selected points are visualized in Figure 13 and detailed in Table 1.

Name	X-coordinate in RD	Y-coordinate in RD	Z-coordinate in RD	X in mm	Y in mm	Z in mm
1001	92378.008	436562.188	5.343	0.03	0.03	0.05
1002	92367.062	437414.406	8.925	0.03	0.03	0.05
1003	92313.773	438194.688	-1.886	0.03	0.03	0.05
1004	93055.805	438186.188	-0.066	0.03	0.03	0.05
1005	93041.898	437389.031	1.258	0.03	0.03	0.05
1006	93042.039	436562.500	5.276	0.03	0.03	0.05
1007	93876.008	436530.781	3.878	0.03	0.03	0.05
1008	93841.297	437392.750	4.300	0.03	0.03	0.05
1009	93857.344	438168.906	4.968	0.03	0.03	0.05

Table 1: Ground Control Points Selected within the Focus Area

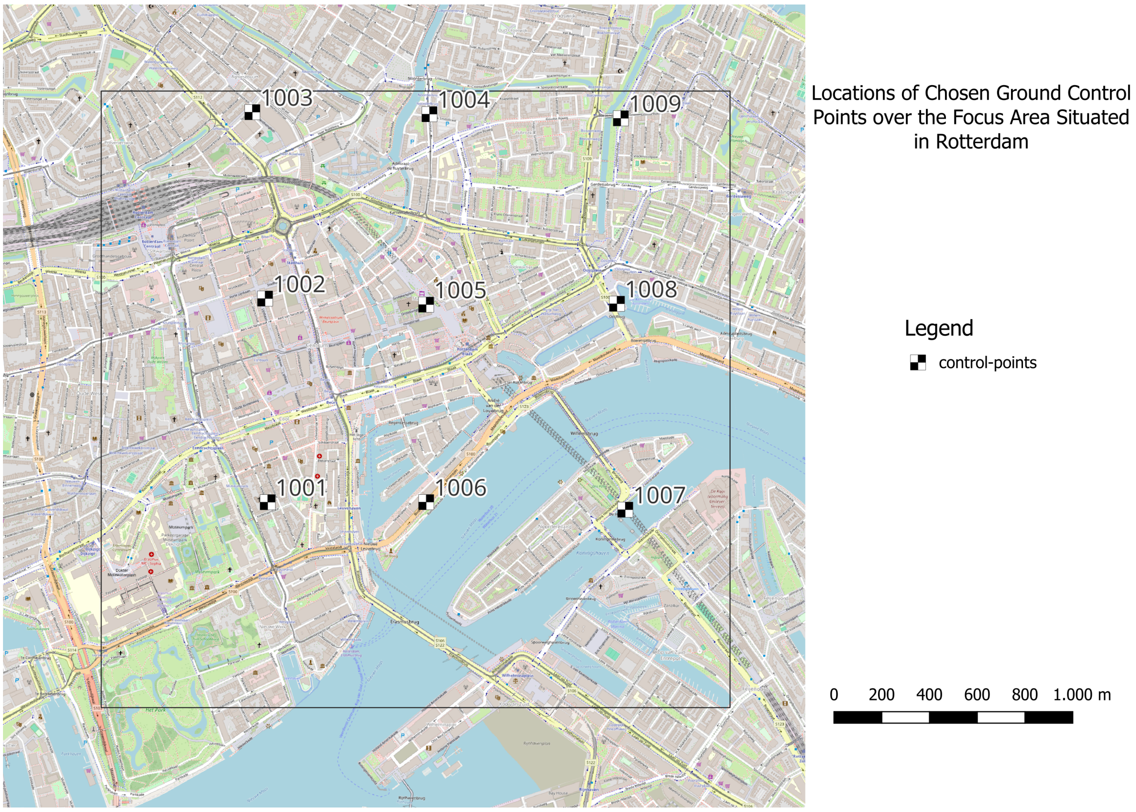

1003
1004
1009
1002
1005
1008
1001
1006
1007
Locations of Chosen Ground Control Points over the Focus Area Situated in Rotterdam
Legend
control-points
0 200 400 600 800 1.000 m

Figure 13: Ground Control Point Locations over the Focus Area

The standard deviations were derived from the values used in the Miramap 2024 Rotterdam flight campaign.

2.11 Object Points

The second set of points within aerial photography are the detected points (see Section 2.15), which are used to construct the basic adjustment network [Polman and ir. M.A. Salzmann, 1996, p. 667]. The world coordinates of these points are not known a priori; instead, they are estimated based on pixel coordinates using the 2D projection formula (see Section 2.12).

Due to the nature of the detection process itself (see Section 2.15), the identified points inherently contain a degree of uncertainty. This uncertainty arises from factors such as image noise, occlusions, or inaccuracies in feature matching.

To address this, the adjustment process (see Section 2.20) is designed to mitigate these uncertainties. It does so by refining the initially detected points using optimization techniques, often based on bundle adjustment or similar methods, which help to produce more consistent and accurate spatial relationships among the features.

For the synthetic adjustment method proposed in Section 3.2 and Section 3.3, the object points are selected from a predefined set of 3D−BAG vertices. The creation of this dataset is described in [Dukai et al., 2024]. The choice for 3D-BAG removes the uncertainty of detection, which is synthetically tackled in Section 3.4. Using predefined BAG vertices removes the uncertainty associated with feature detection, since the object locations are known with certainty.

This approach also ensures that each object point has a known ground truth coordinate, which makes the simulation process deterministic (repeatable) and the results replicable across multiple test cases (see Section 2.24).

2.12 Observation

The optical center (see Section 2.5) and camera model parameters (see Section 2.9) can be used to transform the 3D world point coordinates (see Section 2.10 and Section 2.11) into 2D image plane coordinates (see Section 2.9). These image plane coordinates act as observations [Teunissen, 2000, p. 42], which are used in both the first (see Section 2.20) and second (see Section 2.21) phases of adjustment. The calculated observations serve as known parameters to solve the non-linear systems of adjustment. The formulas provided in Equation 2 and Equation 4 are for expanded perspective cameras [Liangliang Nan, 2025, p. 6], allowing for the inclusion of all camera parameters (see Section 2.8).

2D projection formula:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f & \rho & c_x + ppa_x \\ 0 & f & c_y + ppa_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} P_X \\ P_Y \\ P_Z \\ 1 \end{bmatrix} \tag{2}$$

Where:

- r_{ij} refers to elements of the combined rotation matrix (see Section 2.5)
- t_i refers to elements of the translation vector (see Section 2.5)
- P_X P_Y P_Z represent 3D world coordinates (see Section 2.10 and Section 2.11)
- f focal length of the camera (see Section 2.8 and [Vosselman, 1996, p. 12])
- ppa_x and ppa_y refers PPA over their respective axes (see Section 2.8)
- c_x and c_y refer to the pixel coordinate origin. The thesis assumes the pixel coordinate origin is located at the center of the camera sensor (see Section 2.8)
- x and y refer to the position on the image plane in pixels (see Section 2.8)
- z refers to the depth relative to the image plane

To convert the image plane pixel-based positions to MM, which are at the camera sensor center. The x y z coordinates need to be converted to u and v, which are in MM. The respective formula for which is given in Equation 3.

$$\begin{aligned} u &= \frac{f \cdot x}{z} \\ v &= \frac{f \cdot y}{z} \end{aligned} \tag{3}$$

Where:

- u refers to the position along the horizontal axis of the image (left-right)
- v refers to the position along the vertical axis of the image (up-down)

To reverse the 2D projection and recover the coordinates in 3D, the 3D projection formula can be used.

$$\begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^T \left(\begin{bmatrix} P_Z \cdot \frac{(u-c_x)}{f} \\ P_Z \cdot \frac{(v-c_y)}{f} \\ Z \end{bmatrix} - \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \right) \tag{4}$$

In this 3D projection, P_Z is an estimated depth value. An extended method for determining accurate 3D coordinates involves resolving the uncertainty in P_Z using forward intersection (see Section 2.18).

2.13 3D Models

The representation of 3D models in digital systems is a foundational aspect of computer graphics, enabling a wide array of applications from gaming and animation to Computer Aided Design (CAD) and Augmented Reality (AR)/Virtual Reality (VR) systems. Over the decades, numerous 3D model file formats have been developed, each tailored to specific use cases, fidelity requirements, and platform constraints. This section surveys the most influential and widely adopted 3D data formats, highlighting their structural characteristics, encoding strategies, and suitability for different computational workflows.

The Wavefront Object File (OBJ) format, introduced in the 1990s by Wavefront Technologies, remains one of the most ubiquitous 3D model formats due to its simplicity and readability. It represents geometry using polygonal meshes, with support for vertex coordinates, normals, texture coordinates, and material definitions via companion (.mtl) files. However, OBJ lacks support for scene hierarchies, animation, or instancing, limiting its utility in dynamic or real-time contexts. Despite these limitations, its human-readable ASCII structure makes it a popular intermediate format in pipelines focused on static geometry [Bourke, 1999]. An example of a 3D model in OBJ format is the 3D-BAG [Dukai et al., 2024].

The Standard Tessellation Language (STL) format is a de facto standard in 3D printing due to its straightforward representation of triangulated surface meshes. Developed by 3D Systems in the 1980s, STL supports both American Standard Code for Information Interchange (ASCII) and binary encodings but does not include support for color, texture, or scene information. Its simplicity aids in physical manufacturing workflows but renders it unsuitable for modern applications requiring rich semantic or visual content [Choi and Cheung, 2008].

Collaborative Design Activity (COLLADA), an Extensible Markup Language (XML)-based schema developed by the Khronos Group, was designed to facilitate the exchange of digital assets among various graphics software. COLLADA supports complex scene hierarchies, kinematics, animations, shaders, and physics descriptions, making it suitable for interchanging full scene graphs. However, its verbose XML structure and slow parsing have led to performance concerns in real-time applications [Arnaud and Barnes, 2006]. Despite this, COLLADA remains influential, particularly in pipeline interoperability between tools such as Blender, Maya, and Unity.

Autodesk's Filmbox (FBX) format has become a de facto industry standard in animation, game development, and film production. FBX supports a wide array of features, including skeletal animations, blend shapes, lighting, cameras, and shaders. Its binary encoding ensures compact storage and fast parsing. However, FBX is proprietary, with limited public documentation, which can pose challenges for open-source integration and long-term archival [Kavan et al., 2010].

Developed by Pixar, Universal Scene Description (USD) is a high-performance framework and format for the interchange and augmentation of 3D scene data. USD supports complex composition, layering, instancing, time-sampled data (animation), and metadata. Unlike other formats that focus primarily on geometry, USD enables collaborative workflows where multiple authors can contribute non-destructively to the same scene [Studios, 2020].

3D Tiles is an open specification developed by Cesium for streaming and rendering massive heterogeneous 3D geospatial datasets. Designed with scalability and performance in mind, 3D Tiles enables the efficient delivery of detailed models such as photogrammetry, Building Information Model (BIM), and point clouds over the web. The format organizes data into a spatially subdivided hierarchy using a tile-based structure, allowing for Level of Detail (LOD) rendering and culling based on the viewer's perspective and distance. 3D Tiles are typically encoded in binary formats, which encapsulate geometry, textures, and metadata for high-speed transmission and real-time visualization. While powerful for geospatial visualization, the format is less suited to complex rigging, animation, or non-geospatial scenes, limiting its adoption in traditional entertainment pipelines [Cesium GS, Inc., 2017]. Nonetheless, 3D Tiles are increasingly pivotal in applications such as digital twins, smart cities, and urban planning, where streaming large-scale 3D environments is essential.

2.14 Point Distribution and Von Gruber Squares

The distribution of points over the images is required to comply with the HTW and is described using the point distribution visualization methodology known as Von Gruber squares [Polman and ir. M.A. Salzmann, 1996, p. 456]. These nine fictitious squares are meant to ensure that each section of an image contains a roughly equal number of observations. This equal distribution is due to the equal weighting of image observations in the Jacobian matrix (see Section 2.20). A higher number of observations in one square would lead to stronger testing in that region, potentially resulting in imbalances during adjustment.

For this thesis, a naming convention has been applied to the Von Gruber square sections, numbered as shown in Figure 14.

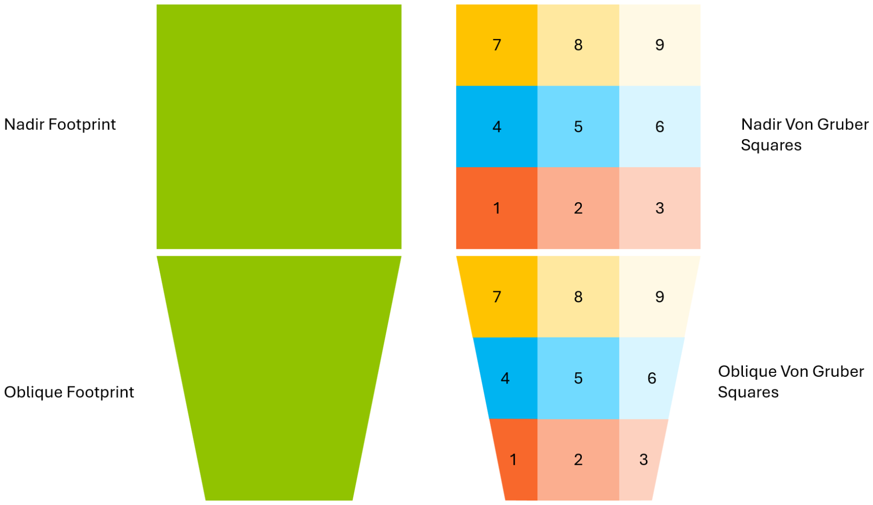


Figure 14: Von Gruber Square Names Used as a Naming Convention within the Thesis

According to the HTW, at least 12 object points equating to a minimum of 6 connections must be available in the overlapping Von Gruber squares between two images [Polman and ir. M.A. Salzmann, 1996, p. 456]. This requirement is based on a 60% cross overlap, typically between two adjacent flight lines. The connections required by the HTW are visualized in Figure 15. The required overlap for oblique is not given by the HTW.

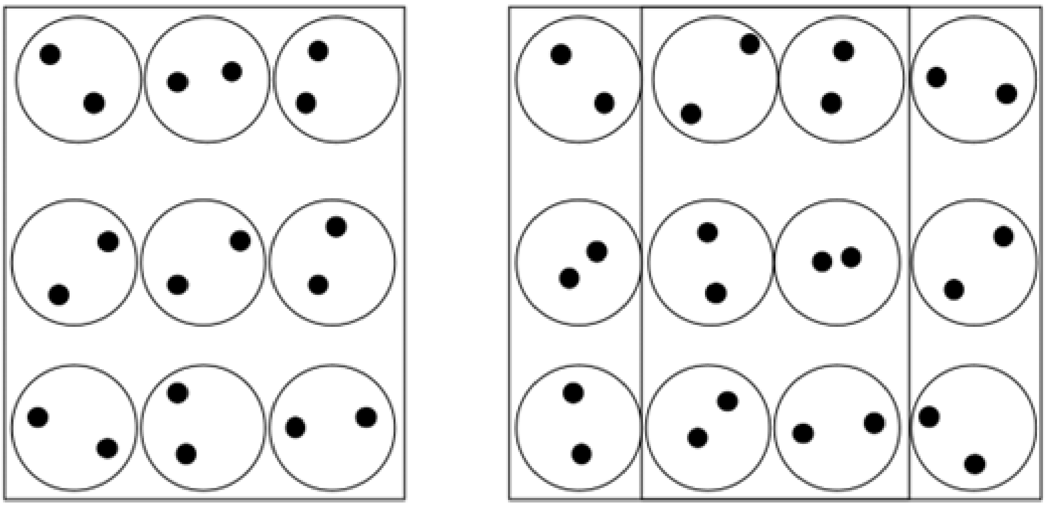


Figure 15: Indication of Point Connection Placements along Two Images with their Respective Von Gruber Squares based on [Polman and ir. M.A. Salzmann, 1996, p. 456]

2.15 Feature Extraction and Matching

In this thesis, two combinations of feature extraction and matching techniques are utilized to perform robust image correspondence: Scale Invariant Feature Transformation (SIFT) combined with LightGlue and DIScrete Keypoints (DISK) combined with LightGlue. These combinations were selected to represent both traditional and learning-based approaches to feature extraction. SIFT is a well-established, classical method known for its robustness to scale and rotation, while DISK is a more recent, deep learning-based technique designed for high-performance feature detection in complex scenes. LightGlue serves as the matching algorithm in both setups, providing an efficient and accurate neural matcher that can be integrated with various feature extractors. The performance and characteristics of these two pipelines are compared and analyzed throughout this work.

To ensure the Von Gruber squares are adequately filled with object points, suitable candidates must be extracted from the aerial imagery. This section describes two feature extraction methods and two feature matching algorithms employed in this research, along with their respective advantages and disadvantages.

The first extraction and matching method is known as SIFT, an object recognition algorithm from 1999 Lowe [1999], widely applied in feature extraction processes from aerial images (see Section 2.11). SIFT operates through a series of traditional image processing techniques, summarized in Figure 16. The method begins by down-scaling an evenly sized digital image into pyramid layers, each subjected to a Gaussian blur using different standard deviation (σ) levels (see Figure 17 and Figure 18).

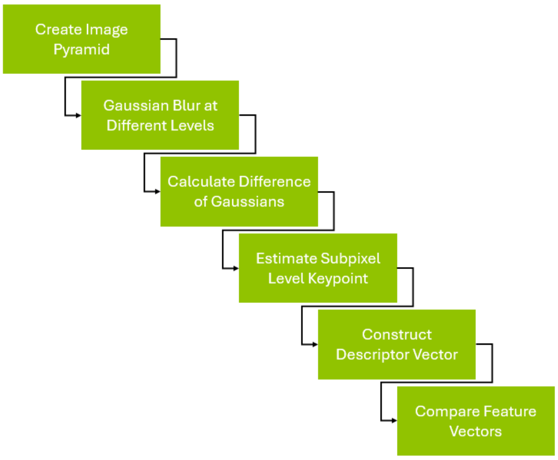

Create Image Pyramid
Gaussian Blur at Different Levels
Calculate Difference of Gaussians
Estimate Subpixel Level Keypoint
Construct Descriptor Vector
Compare Feature Vectors

Figure 16: SIFT workflow

Each pyramid layer is then blurred using an approximated Gaussian kernel, see Figure 17 with different levels of standard deviation σ, see Figure 18. The number of levels can be changed as a hyperparameter. Then, the gradient of the images is calculated. The levels result in highlighting points that are uniquely defined in their local area. In the case of a good object point, the point will appear at different pyramid and sigma blur levels.

The exact position of the object point is then determined based on the weight of the pyramid and how resistant the point is to the levels of blur, resulting in a sub-pixel object point. This means that the point is known in the image plane to a decimal level.

Each blurred layer highlights points uniquely defined in their local neighborhood. Object points that persist across multiple pyramid levels and σ values are selected and refined to sub-pixel accuracy. Each object point is then assigned a 128-dimensional descriptor vector encoding local image gradients. These descriptors, computed across the entire image, are matched using dot product similarity, resulting in potential object point correspondences [Stachniss, 2020, AI, 2023].

However, several disadvantages of SIFT are noted in the literature [Wu et al., 2013]. Its primary limitation in aerial applications, especially those involving oblique imagery, is its inability to match features reliably under large perspective changes or multi-axis rotations (see Section 2.19). While SIFT performs reasonably well under single-axis rotation, its effectiveness decreases significantly when matching between nadir and oblique images. Further limitations are reviewed in [Otero, 2015], many of which are addressed by more recent neural network-based methods.

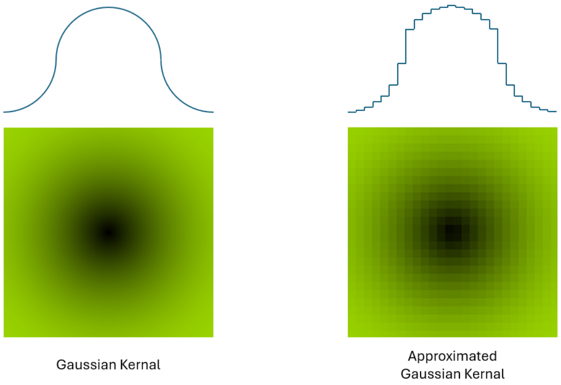


Figure 17: Gaussian Kernel and Its Approximated Form

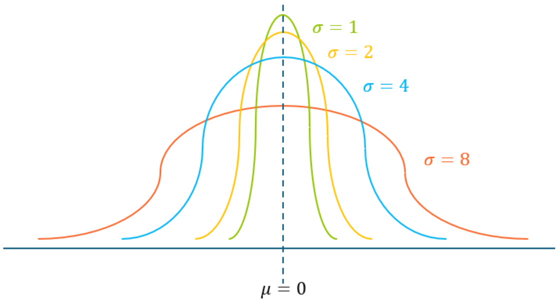


Figure 18: Different Levels of Sigma on a Gaussian Distribution

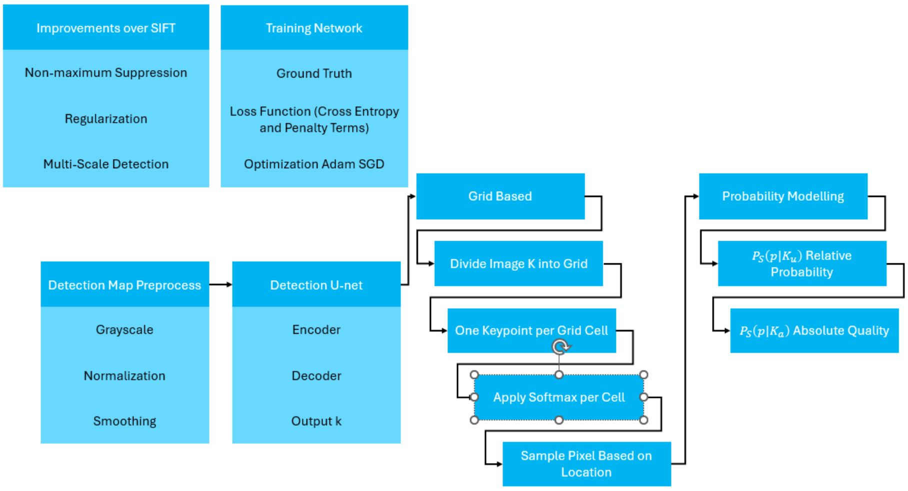


Figure 19: DISK Workflow

One such method is DISK, introduced in 2020 [Tyszkiewicz et al., 2020]. DISK combines convolutional image processing with a U-Net architecture to detect and describe object points. The method first resizes the input image to a square dimension divisible by 8 (using padding if necessary) and applies grayscale normalization and Gaussian filtering (see again Figure 17). The image is then divided into 8×8 cells, from which local maxima are selected as candidate object points. These are passed through a neural detection network trained using depth-masked data, enhancing robustness across varying perspectives. Each object point receives a confidence score used during matching.

Like SIFT, DISK generates a 128-dimensional descriptor for each detected point. Despite improved robustness, DISK also faces limitations, which have inspired various enhancements. For instance, [Gadelha and Carvalho, 2014] presents a version that accommodates varying lighting conditions. [Liu et al., 2021] incorporates temporal consistency into the data, while [Chen et al., 2023] employs $R\ G\ B\ A$ color channels instead of normalization to better capture object point information.

To complement SIFT and DISK, LightGlue can be used as a feature matching algorithm. LightGlue, a successor of Superglue [Sarlin et al., 2020], was proposed in [Lindenberger et al., 2023] and introduces a confidence-based mechanism for efficient and accurate descriptor matching. It constructs a graph based on the 128-dimensional descriptors and their spatial positions, enabling structure-based matching across image pairs. The graph construction process includes a depth parameter, which improves accuracy at the cost of computational performance.

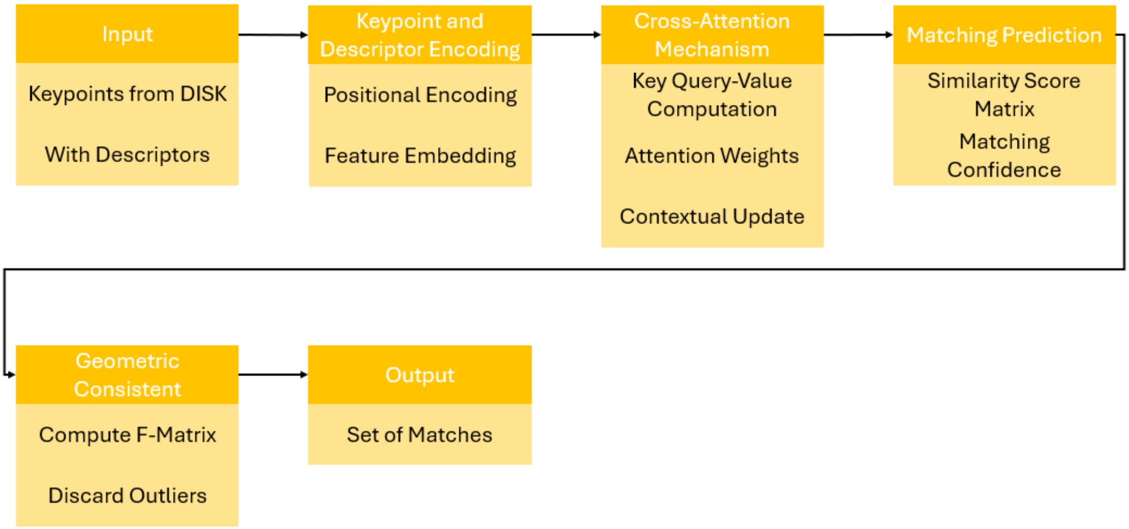


Figure 20: Methodological Workflow of LightGlue

LightGlue is specifically designed to handle affine transformations including translation, rotation, and shearing making it highly effective for aerial image adjustment, particularly when dealing with oblique angles. Unlike just purely using SIFT, LightGlue benefits from a spatially aware, pre-trained graph matching mechanism that resolves issues related to multi-axis perspective changes and occlusions.

2.16 SfM (COLMAP)

SfM is a technique used to reconstruct the 3D structure of a scene from a collection of 2D images taken from different viewpoints. SfM jointly estimates the camera poses and the 3D coordinates of scene points, relying on feature correspondences across multiple views. In this thesis, the SfM pipeline is implemented using COLMAP, a widely used and robust open-source photogrammetry and SfM library.

Mathematical Formulation

Given a set of N images $I_1, I_2, \ldots, I_N$, each associated with an unknown camera pose and a (partially unknown) set of 3D object points, the goal of SfM is to estimate:

- The camera intrinsics K_i for each image (if not known a priori (estimated))
- The camera extrinsics (rotation R_i and translation t_i) for each image
- The 3D positions of points in the scene $\mathbf{X}_j$, where j indexes the 3D points.

Let $\mathbf{x}_{ij}$ denote the 2D image observation of 3D object point $\mathbf{X}_j$ in image i. The basic SfM relationship is governed by 2D projection formula given in Section 2.12.

Once initial camera poses and 3D points are estimated (typically through incremental triangulation and pose estimation), a Bundle Adjustment step is applied to jointly refine all parameters by minimizing the reprojection error through Bundle Adjustment (see Section 2.20). This is a nonlinear least-squares optimization problem typically solved using algorithms like Levenberg-Marquardt.

The pipeline of Structure-From-Motion and Multi-View Stereo (COLMAP) follows an incremental SfM strategy:

- Feature Extraction and Matching: Keypoints are detected and matched across images using methods such as SIFT or DISK + LightGlue (see Section 2.15).
- Initial Pair Reconstruction: A seed image pair with sufficient parallax is selected to initialize the reconstruction.
- Incremental Triangulation: New images are registered by estimating their pose relative to the growing model, and new 3D points are triangulated.
- Global Optimization: Bundle adjustment is repeatedly applied to refine the entire model.

This iterative process continues until all images are registered or no more matches are available. The result is a sparse 3D point cloud and a set of calibrated camera poses.

2.17 Rendering

Rendering is the computational process of generating a 2D image from a 3D scene description. It lies at the core of computer graphics, encompassing both real-time rendering essential for interactive applications such as games and simulations and offline rendering used in film production, animation, and visual effects. Among the many tools and Application Programming Interface (API)s developed to facilitate rendering, Open Graphics Library (OpenGL) and Blender stand out as two of the most widely adopted systems, each targeting different ends of the rendering spectrum. This section reviews these tools in detail, comparing their architectures, capabilities, and roles in modern 3D content creation.

Open Graphics Library (OpenGL) is a cross-platform, low-level graphics API developed by Silicon Graphics Inc. in the early 1990s and now maintained by the Khronos Group. It provides a standardized interface to Graphics Processing Unit (GPU) hardware for rendering 2D and 3D graphics. OpenGL has become a foundational tool for real-time rendering in applications such as video games, CAD software, and simulation environments.

OpenGL follows a state-machine architecture, where rendering is controlled via a sequence of draw calls and buffer manipulations. It supports programmable shaders via the OpenGL Shading Language, enabling developers to implement custom lighting models, post-processing effects, and material behaviors. Since version 3.0, OpenGL has increasingly emphasized a programmable pipeline, phasing out older fixed-function features to allow more control and performance optimization [Shreiner et al., 2013].

OpenGL's strengths lie in its portability, extensibility, and deep integration into graphics hardware drivers. However, its steep learning curve and low-level nature demand significant expertise to use effectively. Furthermore, OpenGL does not prescribe a scene graph or asset management structure, requiring developers to build or integrate such abstractions independently. Despite these challenges, OpenGL remains a cornerstone of interactive graphics development, particularly in systems where real-time performance is essential.

Blender is an open-source 3D content creation suite that supports a full 3D pipeline consisting of modeling, animation, simulation, compositing, and rendering. Originating in the late 1990s and developed by the Blender Foundation, it has grown into a powerful alternative to commercial tools such as Autodesk Maya or Cinema 4D. While Blender can use OpenGL for its viewport rendering, it also includes dedicated offline render engines, notably Cycles (path tracing) [Lommel, 2018] and Eevee (real-time rasterization) [Foundation, 2020a].

Cycles is a physically-based path tracer, capable of generating photorealistic images by simulating global illumination, caustics, subsurface scattering, and volumetric scattering. It leverages Computer Processing Unit (CPU) and GPU compute acceleration, supporting features such as adaptive sampling, denoising, and physically-based shaders via the Principled Bidirectional Scattering Distribution Function (BSDF). In contrast, Eevee is a real-time engine designed for fast, high-quality previews, utilizing screen-space reflections, ambient occlusion, and light probes to approximate more computationally expensive lighting effects [Foundation, 2020b].

Blender's strength lies in its high-level abstraction and artist-friendly User Interface (UI), allowing users to produce complex scenes without writing code. The tight integration of modeling, animation, and rendering tools makes it an ideal environment for rapid prototyping and creative exploration. Blender's scripting capabilities (via Python) also enable automation and procedural content generation, bridging the gap between artistic workflows and programmatic control.

This thesis uses rendering for the creation of synthetic images.

2.18 Forward Intersection

As previously discussed, one of the main disadvantages of the 3D projection formula (see Section 2.12) is the necessity of estimating the object space height P_Z. This value is often unknown and introduces uncertainty when performing 3D measurements and 3D mapping in aerial imagery. This issue can be addressed through the use of the forward intersection method [Polman and ir. M.A. Salzmann, 1996, p. 88 and p. 104], which determines an optimal object point by intersecting measurement rays originating from multiple optical centers (see Section 2.5). Each of these rays is derived using the same 3D projection principles. In effect, forward intersection enables 3D measuring on 2D images. Figure 21 illustrates the concept of forward intersection using three camera centers.

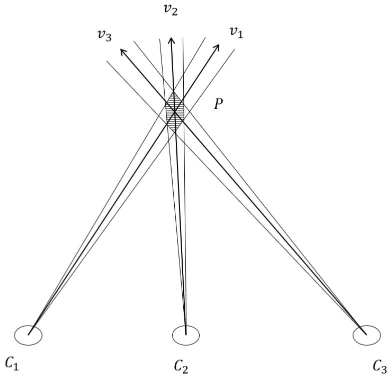


Figure 21: System of Forward Intersection based on Three Camera Centers

According to the HTW, each object point must be observed from at least three different images [Polman and ir. M.A. Salzmann, 1996, p. 447] to allow for robust spatial adjustment. Consequently, a forward intersection requires a minimum of three intersecting measurement rays, as depicted in Figure 21.

The forward intersection method requires the following inputs:

- A set of n camera centers C_i (see Section 2.8) where $0 \leq i \leq n$
- A set of n corresponding vectors v_i

Each measurement ray from the camera center can be expressed parametrically as:

$$r_i(t_i) = C_i + t_i v_i$$

The 3D object point P is calculated as the least-squares solution that minimizes the distance between the rays, given by:

$$P = (\sum_{i=1}^{n}(I - v_i v_i^T))^{-1}(\sum_{i=1}^{n}(I - v_i v_i^T)C_i) \tag{5}$$

Where:

- n is a set of cameras corresponding to the overlap
- $v_i v_i^T$ is the outer product of the direction vector
- $(I - v_i v_i^T)$ is the projection matrix onto the plane orthogonal to v_i

This formulation computes the point P that lies closest to all measurement rays in the least-squares sense, inherently accommodating small levels of noise or measurement inaccuracy. The resulting point P is considered the best estimate of the true 3D object position based on the given observations.

2.19 Occlusion

As previously discussed, a critical consideration in aerial photography, particularly in oblique imagery, is the phenomenon of occlusion, where elements in the built environment are partially or entirely obscured by other objects situated closer to the camera. This effect is especially prevalent in urban areas, where buildings and vertical structures can block the line of sight to features located behind them. As introduced in Section 2.2 and Section 2.3, occlusion can occur in both nadir and oblique imagery; however, it is inherently more pronounced in oblique views due to the increased perspective distortion and viewing angle. These concepts are illustrated in Figure 22.

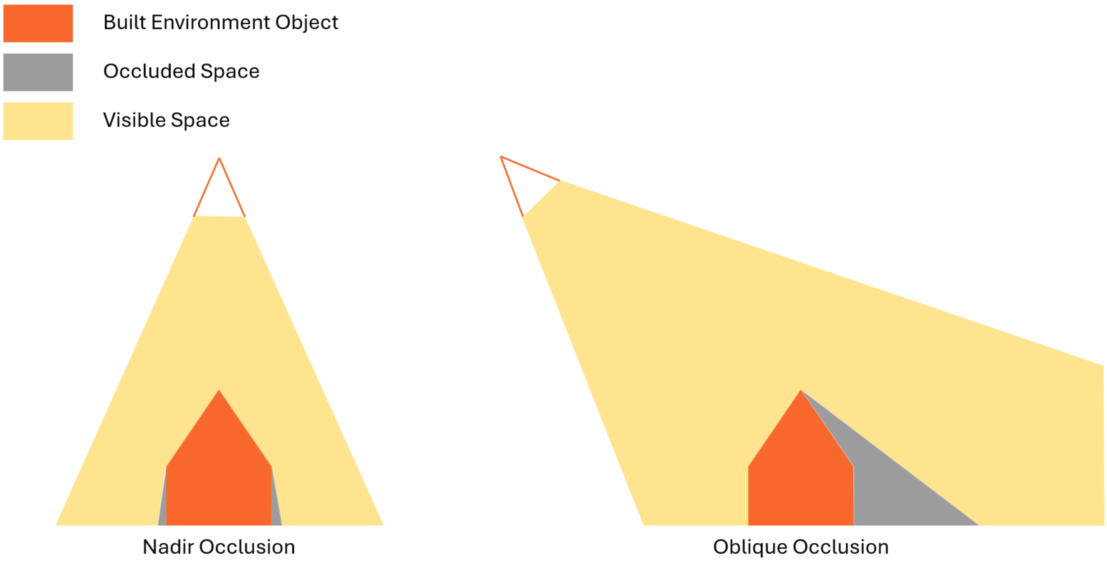


Figure 22: System of Forward Intersection based on Three Camera Centers

While the feature extraction algorithms (see Section 2.15) typically detect a similar number of object points in both nadir and oblique images, the number of successful matches differs significantly. This is because matching algorithms (also detailed in Section 2.15) rely on overlapping visible regions between image pairs. In oblique imagery, these overlapping regions are often reduced due to occlusion, resulting in fewer reliable connections between object points. This has a direct impact on the completeness and accuracy of image-based 3D reconstruction and adjustment processes.

2.20 First Phase Adjustment Aerial Adjustment

Free Network Adjustment [Polman and ir. M.A. Salzmann, 1996, p. 152] refers to the calculation of a local network, which, in the case of aerial adjustment, uses a set of optical centers (see Section 2.5) and observations (see Section 2.12) to compute the best-fitting system relative to an S-basis [Polman and ir. M.A. Salzmann, 1996, p. 43]. The S-basis defines the mathematical foundation introduced to eliminate the rank deficiency in the system of unknowns [Polman and ir. M.A. Salzmann, 1996, p. 668]. This rank deficiency, also known as the range defect, arises due to the lack of absolute scale or orientation and must be constrained to determine the size and orientation of the observation network. In an aerial system, the S-basis typically consists of two cameras and their associated camera centers. Aerial adjustment is performed by solving a non-linear least squares problem based on the Bundle Adjustment method proposed by [Triggs et al., 2000]. Bundle adjustment is used to adjust the following parameters:

- Camera extrinsic values (see Section 2.5)
- Camera intrinsic values (see Section 2.8)
- 3D object points (see Section 2.11)

These parameters each introduce their unknowns that need to be counterbalanced using observations. The excess number of observations (above zero) is referred to as Degrees of Freedom (DOF) [Polman and ir. M.A. Salzmann, 1996, p. 78], denoted by n, where $n \geq 0$. The least squares minimization is shown in Equation 6 from [Polman and ir. M.A. Salzmann, 1996, p. 92].

$$\min_{r,t,f,d,P} \sum_{i \, \in \, I} \sum_{j \, \in \, J} |x_{ij} - \pi(P_i, X_j)|^2 \tag{6}$$

Where:

- $\pi(C_i, X_j)$ is the forward intersection (see Section 2.18)
- x_{ij} is the observed 2D feature (see Section 2.12 and Section 2.14)
- $i \in I$ set of camera optical sensors containing intrinsic (see Section 2.8) and extrinsic (see Section 2.5) parameters
- $j \in J_i$ set of observed points by a camera i

The minimization parameters (unknowns) are:

- r is a vector encoding the ω, ϕ and κ (see Section 2.5) (3 unknowns per camera)
- t is a position vector (see Section 2.5) (3 unknowns per camera)
- f focal distance (1 unknown per calibration)
- d a set of distortion parameters (0 to n unknowns depending on the camera distortion model where n is the order of distortion per calibration)
- P a set of 3D object points (see Section 2.11) encoding within P_X P_Y P_Z vectors (3 unknowns per point)

The observations (known) are given as follows:

- u and v coordinates placed on the image plane (2 observations per detected point) calculated using (see Section 2.12)

Adjustment is carried out by applying B-model testing [Polman and ir. M.A. Salzmann, 1996, p. 114] and [Teunissen, 2000, p. 154] testing where $B = J^T J$. The formulation of J is given in Equation 7.

$$J = \begin{bmatrix} \frac{\partial r_{11}}{\partial P_1} & \cdots & \frac{\partial r_{I1}}{\partial P_1} & \frac{\partial r_{11}}{\partial X_1} & \cdots & \frac{\partial r_{11}}{\partial X_1} \\ \frac{\partial r_{12}}{\partial P_2} & \cdots & \frac{\partial r_{I2}}{\partial P_2} & \frac{\partial r_{11}}{\partial X_1} & \cdots & \frac{\partial r_{11}}{\partial X_1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_{ij}}{\partial P_I} & \cdots & \frac{\partial r_{IJ}}{\partial P_I} & \frac{\partial r_{11}}{\partial X_1} & \cdots & \frac{\partial r_{11}}{\partial X_1} \end{bmatrix} \tag{7}$$

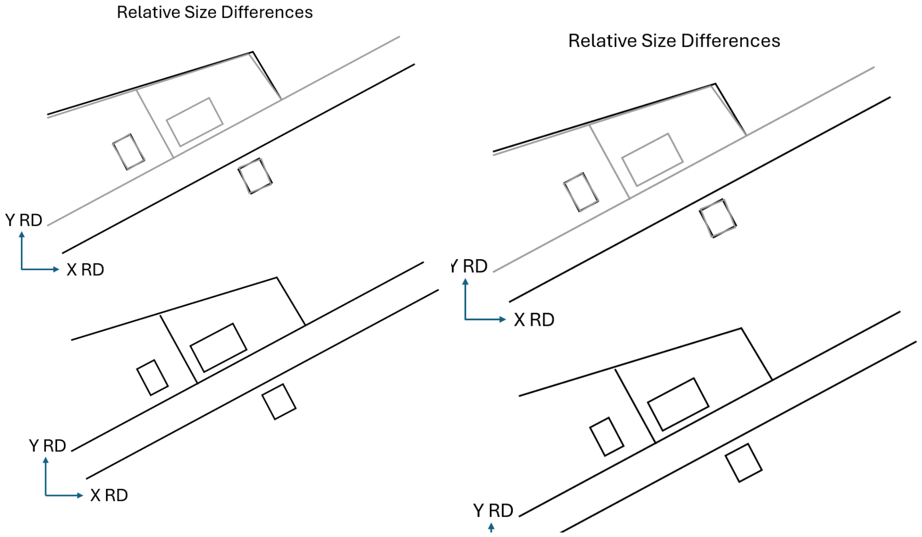
Where:

- r_{ij} is a 2D residual vector of $x_{ij} - \pi(P_i, X_j)$
- P_i is the camera model (see Section 2.8)
- X_J are the 3D points (see Section 2.10 and Section 2.11)

By iteratively minimizing the variables r,t,f,d, and P in the first phase adjustment, a best-fitting system can be established.

2.21 Second Phase Adjustment Aerial Adjustment

Second phase adjustment uses the initial system created during the first phase adjustment (see Section 2.20) and applies a system-wide transformation to make the local system referenced in a nationally referenced system (which in the case of the Netherlands is RD, NAP or a combination thereof, depending on the dimensionality of the measurement). This transformation is achieved through two mechanisms: coordinate transformation and relative size adjustment. This concept is visualized in Figure 23a (see Section 2.10). Coordinate transformation aligns the reconstructed field to fit within an existing coordinate reference system, while relative size adjustment incorporates an existing point field to refine the fitting accuracy, which is shown Figure 23b.



(a) Coordinate Transformation Retrieved from [Polman and ir. M.A. Salzmann, 1996, p. 152]

(b) Shape Changing Transformation Retrieved from [Polman and ir. M.A. Salzmann, 1996, p. 152]

Three forms of second phase adjustment are described in the HTW [Polman and ir. M.A. Salzmann, 1996].

- **Absolute Constrained Adjustment (Kleinste-kwadraten aansluiting)**
  This method is primarily used for quality control of homogeneous point fields (i.e., point fields of the same type and accuracy), and for testing the connection points between different networks or coordinate sets [Polman and ir. M.A. Salzmann, 1996, p. 160].
- **Pseudo Constrained Adjustment (Pseudo kleinste-kwadraten aansluiting)**
  Used when computing coordinates relative to an existing point field, whose coordinates are to remain unchanged [Polman and ir. M.A. Salzmann, 1996, p. 161].
- **Weighted Constrained Adjustment (Gewogen kleinste-kwadraten aansluiting)**
  Applied when one wants to preserve the internal geometry (the shape and relative positions) of both point sets as much as possible [Polman and ir. M.A. Salzmann, 1996, p. 162].

	Fitting Method			
	Coordinate Transformation		Relative Size Difference	
Connection Method	Free Points	Connection Points	Free Points	Connection Points
Absolute Constrained	Yes	Yes	Yes	Yes
Pseudo Constrained	No	No	Yes	Yes
Weighted Constrained	No	No	No	Yes

Table 2: The Correct fitting method is chosen based requirements outlined in the table from [Polman and ir. M.A. Salzmann, 1996, p. 161]

For aerial adjustment, the most logical method of constraint is the absolute constrained approach, as the reference field is not adjusted using aerial observations. Therefore, all further references to second-phase adjustment in this context refer to the absolute constrained method.

2.22 W-test and Data-Snooping

To ascertain the correctness of the relationship between observational quantities and unknowns, as encoded in the mathematical model constructed during the adjustment phases (see Section 2.20 and Section 2.21), a W-test can be applied. This is a one-dimensional, single-parameter statistical test used within the least squares adjustment [Polman and ir. M.A. Salzmann, 1996, p. 113]. The method aims to ensure the best-fitting model under the condition that modeling errors are detected, isolated, and removed. The objective of hypothesis testing in this context is to identify and eliminate erroneous observations, thereby providing a quality check within the adjustment process.

The geodetic testing procedure includes the W-test, also referred to as the waarnemingstoets or observation test [Teunissen, 2000, p. 35], [Teunissen, 2006, p. 62]. In addition, a global F-test (see Section 2.23) is used to evaluate the overall model fit.

The W-test evaluates a one-dimensional alternative hypothesis H_a that postulates an error in a single observation, in contrast to the null hypothesis H_0, which assumes all observations are free from noise [Polman and ir. M.A. Salzmann, 1996, p. 75]. The fundamental assumption is that only one observation may contain an error, while all others are correct. The test statistic w expresses how extreme the residual of that observation is, relative to its standard deviation. The critical value $k_\alpha = 3.29$ is commonly used, derived from the standard normal distribution, and corresponds to a significance level $\alpha = 0.001$ [Polman and ir. M.A. Salzmann, 1996, p. 76]. If $|w| > k_\alpha$, the observation is considered statistically suspicious and the null hypothesis H_0 is rejected.

The HTW formulation of H_0 and H_a is given in Equation 8 [Polman and ir. M.A. Salzmann, 1996, p. 114]:

$$\begin{aligned} H_0 :\ E\{\underline{y}\} &= Ax + a_0 \ ;\ D\{\underline{y}\} = Q_y \\ H_a :\ E\{\underline{y}\} &= \begin{bmatrix} B & C \end{bmatrix} \begin{bmatrix} x \\ \nabla \end{bmatrix} + a_0 \ ;\ D\{\underline{y}\} = Q_y \end{aligned} \tag{8}$$

Here, B is the design matrix used in the adjustment model (see Section 2.20). The significance of the hypothesized model error is computed using the HTW test statistic [Polman and ir. M.A. Salzmann, 1996, p. 228], as defined in Equation 9. The B-matrix is part of the B-model testing approach [Polman and ir. M.A. Salzmann, 1996, p. 114–115], which can also utilize data snooping [Polman and ir. M.A. Salzmann, 1996, p. 661]. Data-Snooping is a method for identifying and iteratively removing observations with the largest W-statistic until the system satisfies the F-test's critical value (see Section 2.23). However, in this thesis, data snooping is not used due to the added complexity it introduces to a deterministic adjustment process. Its use could reduce the impact of individual parameter estimates and obscure the analysis and interpretation of results (see Chapters 4 and Section 1.3). The W-test statistic is calculated as follows:

$$\underline{T}_q = \underline{\hat{r}}^T c (c^T Q_{\hat{r}} c)^{-1} c^T \underline{\hat{r}} \tag{9}$$

Where:

- $\underline{T}_q$ is the test statistic (toetsingsgrootheid),
- $\underline{\hat{r}}$ is the vector of reciprocal residuals,
- c is a vector defining the relation between observational quantities and a hypothesized single error,
- $Q_{\hat{r}}$ is the variance-covariance matrix of the reciprocal residuals.

This equation can be rewritten into the W-test.

$$\underline{w} \triangleq \frac{c^T \underline{\hat{r}}}{\sqrt{c^T Q_{\hat{r}} c}} \tag{10}$$

2.23 F-test

As mentioned previously, the W-test targets individual variables, whereas the F-test is designed to evaluate the overall fit of the adjustment model. The general test for the success of a measurement network adjustment is conducted using the F-test statistic $|f|$ [Polman and ir. M.A. Salzmann, 1996, p. 132]. This test evaluates whether the adjustment model, based on the observations filling the B-model (see Section 2.20), conforms to the assumptions of a normal distribution and correct variance-covariance structure. The test statistic (toetsingsgrootheid) of the general model evaluates how well the entire network fits the imposed functional and stochastic models.

The F-test is based on the a-posteriori variance factor $\underline{\hat{\sigma}}^2$ and is calculated as:

$$\underline{\hat{\sigma}}^2 = \frac{\underline{T}_b}{b} \tag{11}$$

Where:

- b is related to the DOF
- T_b is the test statistic associated with the total residuals
- $\underline{\hat{\sigma}}^2$ is the estimated a-posteriori variance factor

In case of aerial adjustment where the number of factors is relatively high a critical value of 1 is assigned as the estimated variance factor. The estimated variance factor should ideally be close to 1, indicating that the stochastic model (assumptions on variances and covariances) is correctly specified. If $\underline{\hat{\sigma}}^2$ significantly deviates from 1, the test may signal that the aerial model is either too lenient or too stringent [Polman and ir. M.A. Salzmann, 1996, p. 132].

The general model test is used to detect whether any errors are present in the functional or stochastic model without any a priori assumptions about the location or nature of those errors. It is assumed that the number of errors equals the number of constraints, i.e., $q = b$. The associated test statistic is typically denoted as T_b and follows a central chi-square distribution with b degrees of freedom:

$$\underline{\hat{\sigma}}^2 \sim \frac{\chi^2(b)}{b} \sim F(b, \infty) \tag{12}$$

In models based on conditional equations, the test statistic can also be computed using the closure terms vector l. For observation equations, it is calculated using the least-squares residuals vector g.

The test criterion for the general model test is:

$$T_b > k_\alpha \tag{13}$$

Where k_α is the critical value derived from the central chi-square distribution for b degrees of freedom and a specified significance level α. This threshold is determined using the B-method of testing. The general model test is thereby directly connected to lower-dimensional tests like the W-test and the point test.

However, as the number of degrees of freedom b increases, so does the unreliability threshold (onbetrouwbaarheidsdrempel), which increases the risk of incorrectly rejecting the null hypothesis. When the threshold becomes greater than 0.10, the risk of Type I errors becomes significant and must be accounted for.

The general model test is also known as the model test or general model test ("algemene modeltoets" or simply "modeltoets").

If specific errors cannot be traced using one-dimensional tests such as the W-test or point test, rejection of the null hypothesis in the general test may indicate a fault in the **stochastic model**. Possible causes include:

- Observational quantities are not normally distributed
- Variances or covariances of the observations have been incorrectly assumed

In practical applications, the a posteriori variance factor is sometimes used as a diagnostic indicator. Its value is related to the test statistic as follows:

$$\hat{\sigma}^2 = \frac{T_b}{b} \tag{14}$$

If, after removing modeling errors, the variance factor significantly deviates from 1, this suggests the stochastic model may be flawed:

- $\hat{\sigma}^2 > 1$: The assumed observational standard deviations may be too small
- $\hat{\sigma}^2 < 1$: The assumed observational standard deviations may be too large

The estimator of the a-posteriori variance factor follows the F-distribution:

$$\hat{\sigma}^2 \sim F(b, x) \tag{15}$$

In practice, this statistic is sometimes used in place of the test statistic T_b, leading the general model test to be referred to as the $\hat{\sigma}^2$-test or simply the F-test, about the distribution used.

2.24 Monte-Carlo simulation

Monte Carlo simulation is a statistical method used to assess the robustness, reliability, and uncertainty of geodetic and photogrammetric processes by repeatedly evaluating a model under varying input conditions. In the context of aerial adjustment networks, Monte Carlo methods are particularly useful for simulating the propagation of measurement noise through the network and evaluating the resulting variability in estimated object points.

The core principle of a Monte Carlo simulation involves generating a large number of random input variations based on known statistical properties, typically (see Section 3.2) assuming Gaussian noise, followed by running the entire adjustment pipeline (see Section 2.1) for each synthetic realization. This allows for a distribution of outcomes rather than a single deterministic result, providing insight into the precision and sensitivity of the system (see Section 2.20 and Section 2.21).

Each Monte Carlo run consists of the following steps:

- Generation of synthetic image observations by perturbing the projected point locations with Gaussian noise based on the Miramap 2024 Rotterdam campaign
- Calculation of the positional error (W-test (see Section 2.22)) of reconstructed object points concerning their known positions, as well as the F-test for the global fit.
- Aggregation and statistical analysis of results over N iterations (see Chapter 5).

By simulating a sufficiently large number of iterations, a stable estimate of the error distribution is obtained. This can then be visualized through histograms, confidence ellipses, or standard deviation maps, offering detailed insight into how uncertainties in image measurements and orientation parameters propagate to the 3D reconstruction (see Chapter 5).

This methodology allows for a direct comparison between different adjustment phases and different Case-Types, thereby serving as a basis for the evaluation in Section 4. It also ensures repeatability and reproducibility of results, which are essential for testing geodetic and aerial photogrammetry workflows under realistic, noisy conditions.

3 Methodology

Chapter 3 describes a methodological framework (see Section 3.1) in three phases that answer the main and sub-research questions defined in Chapter 1. As mentioned before, the objective of the research is to create an oblique framework using synthetic data (see Section 1.4) that can be applied to the detection and adjustment step of the aerial imagery collection pipeline (see Section 2.1). This synthetic data can be created for both nadir (see Section 2.2) and oblique (see Section 2.3) imagery. The created synthetic workflow can be used in a variety of applications, such as the testing of bundle adjustment algorithm additions [Wu et al., 2011], the testing of object point (see Section 2.11) detection algorithms, the testing of object point matching algorithms (see Section 2.15), testing of adjustment edge cases, testing of detection and matching edge cases. Finally, the outcomes of the research can also assist in situations that fall outside of the scope of this research but do fall within the aerial imagery collection pipeline (see Section 2.1), such as the creation of a and a Signal Plan (see Section 2.4).

This chapter gives an overview of the proposed novel methods, which includes the overall framework for oblique collection inspired by Chapter 7 of the HTW [Polman and ir. M.A. Salzmann, 1996, p. 417]. This framework is given at the start. As mentioned in Chapter Section 1, the subgoal of the research is to synthetically adjust and detect aerial geodetic networks (see Section 2.1).

3.1 Synthetic Data Oblique Framework

The principal aim of this research is to explore the feasibility of adapting the established nadir aerial image collection pipeline, as detailed by [Polman and ir. M.A. Salzmann, 1996, p. 420], into a system compatible with oblique aerial imagery. Unlike nadir images, which are captured with cameras pointing directly downward, oblique imagery introduces angled perspectives that result in increased complexity due to Occlusion, perspective distortion, and variable visibility of objects. Adapting the pipeline involves a systematic analysis of its components (see Chapter 2) to assess which parts can be translated to, or restructured for, oblique imagery processing.

In this context, the term pipeline is strictly defined as a complete, end-to-end workflow encompassing all the stages required for aerial image acquisition, preparation, adjustment, and final usage, specifically for nadir imagery. Since this research does not aim to provide a full implementation of such a pipeline for oblique images, the term framework is instead used to describe the proposed system. This framework represents an initial step toward appropriation, laying the groundwork for a future pipeline by examining the core components necessary for oblique imagery through the use of synthetic data.

The use of synthetic data is a deliberate methodological choice. Capturing real oblique aerial imagery involves high operational costs and introduces statistical dependencies that are difficult to isolate. By contrast, synthetic datasets provide a controlled environment in which variables can be isolated, modified, and studied without the logistical and analytical challenges that accompany real-world data collection. This enables more precise investigation into the effects of occlusion, visibility, detection accuracy, and adjustment reliability under repeatable conditions.

As discussed in Section 2.1, the conventional aerial imagery pipeline is composed of several key stages:

- **Signal Plan** (Section 2.4): Definition of the geo-spatial and temporal parameters governing image acquisition.
- **Flight Plan** (Section 2.4): Determination of aircraft trajectories, camera angles, and overlap rates to fulfill the signal plan.
- **Image Capture** (Section 2.3): Collection of raw image data with proper camera positioning, calibration, and orientation.
- **Object Point Detection** (Section 2.15): Extraction of image features that correspond to real-world ground points or structures.
- **First and Second Phase Adjustment** (Section 2.20, Section 2.21): Geometric correction through photogrammetric adjustment methods such as bundle adjustment.
- **Quality Inspection** (Section 2.23, Section 2.22): Statistical testing and validation of the photogrammetric adjustments.
- **Mapping and Maintenance**: Integration of final output into cadastral or municipal GIS systems.

Not all of these steps are equally suited for synthesis or simulation. For instance, the signal and flight planning stages are deeply interdependent with the physical parameters of image acquisition and adjustment. Their successful implementation relies on real-time navigation constraints, sensor calibration, and hardware specifications that are not trivial to simulate without also simulating a complete adjustment backend.

Similarly, the Mapping and Maintenance stage is largely institution-dependent, often bound to legacy systems like the HTW and existing municipal workflows. These systems are often not optimized for integration with modern computational or synthetic processes and would require separate institutional research.

As a result, this framework focuses on the components that are both essential and amenable to synthetic reproduction:

- Image Capture
- Object Point Detection
- Geometric Adjustment (First and Second Phase)
- Statistical Quality Inspection

These components can be reliably studied using synthetic datasets designed to simulate the conditions of real oblique imagery. In doing so, the framework allows for structured experimentation on issues like occlusion, feature detectability, and adjustment accuracy. Furthermore, these elements align with the research sub-questions and objectives outlined earlier in the thesis.

The synthesized framework is organized into three distinct phases. The general methodology of these phases is visualized in Figure 24, which presents a simplified overview of the methodology (each phase sub-section delves into an expanded methodology). Each phase is briefly introduced below, with a detailed breakdown provided in its respective sections.

Phase IA: Non-Obstructed Synthetic Adjustment (see Section 3.2) In the first sub-phase of the framework, synthetic object points are generated using 3D-BAG data, in conjunction with original external orientation parameters from the Rotterdam flight campaign. These object points are used to compute synthetic image observations (see Section 2.12). This scenario assumes no occlusion allowing for all features to be visible from all angles and is intended as a baseline for assessing the adjustment process under ideal conditions. Additionally, Monte Carlo simulation is used to examine statistical independence across test cases. The number of simulations per test case varies depending on computational complexity and runtime constraints. The use of Monte Carlo methods is explained further in Section 2.24.

Phase IB: Depth Map-Based Occlusion for Synthetic Adjustment (see Section 3.3) Phase IB extends the methodology of Phase IA by introducing occlusion, one of the major challenges unique to oblique imagery. Occlusion occurs when certain object points are not visible from specific viewpoints due to other structures obstructing the line of sight. To realistically simulate this effect, a depth map is rendered using OpenGL [Shreiner et al., 2016] (see Section 2.17). Depth maps are commonly used in 3D graphics applications to represent the distance of each visible pixel from the camera. In this framework, the depth map is used as a 3D viewshed to identify and filter out occluded points. This allows the pipeline to be stress-tested under realistic visibility constraints, and helps identify how occlusion affects adjustment and feature availability (see Section 2.19 and Section 2.11).

Phase II: Synthetic Computer Vision-Based Object Point Detection (see Section 3.4) The second phase of the framework shifts focus from adjustment to object point detection. Using rendering software such as Blender (Section 2.17) and geo-spatial 3D tiles from sources like Google Maps (Section 2.13) and OpenStreetMap, synthetic images are created that closely replicate real-world aerial imagery from the 2024 Miramap Rotterdam flight campaign. The original camera positions and orientations (Section 2.5) are used to render comparable images. These images are then augmented with exogenous effects such as varying lighting, weather, and cloud cover, based on real-world conditions that are otherwise difficult to replicate or control (see Section 1.1). The synthetic dataset allows testing of different object point detection and matching algorithms under diverse conditions (Section 2.15), assessing their robustness and reliability for use in an oblique image pipeline.

Phase III: Application to Real-World Data (see Section 3.5) The third and final phase of the framework reintroduces real data. Using actual aerial images from the Miramap 2024 Rotterdam campaign (provided by both Miramap and the municipality of Rotterdam), this phase validates the insights gained from the synthetic phases. While the same camera centers are reused, the image data now comes from real-world sources, making it possible to observe how theoretical findings translate into practice. This serves two purposes: first, to verify the conclusions drawn from synthetic data, and second, to highlight any real-world challenges or discrepancies that were not captured in the earlier phases.

It is important to note that Monte Carlo simulations are used exclusively in Phase I to assess statistical behavior under repeatable synthetic conditions. Phases II and III, being focused on feature detection and real-world application, respectively, are evaluated without simulation repetition.

Overall, this phased approach provides a structured methodology (see Figure 24) for the development and preliminary validation of an oblique aerial image collection framework. The progression from idealized conditions (Phase IA) to increasingly realistic simulations (Phase IB and II), and finally to real-world validation (Phase III), enables a thorough investigation of both technical and conceptual challenges. This structure allows the research to make targeted, measurable contributions to the broader goal of establishing an oblique-capable aerial imagery pipeline.

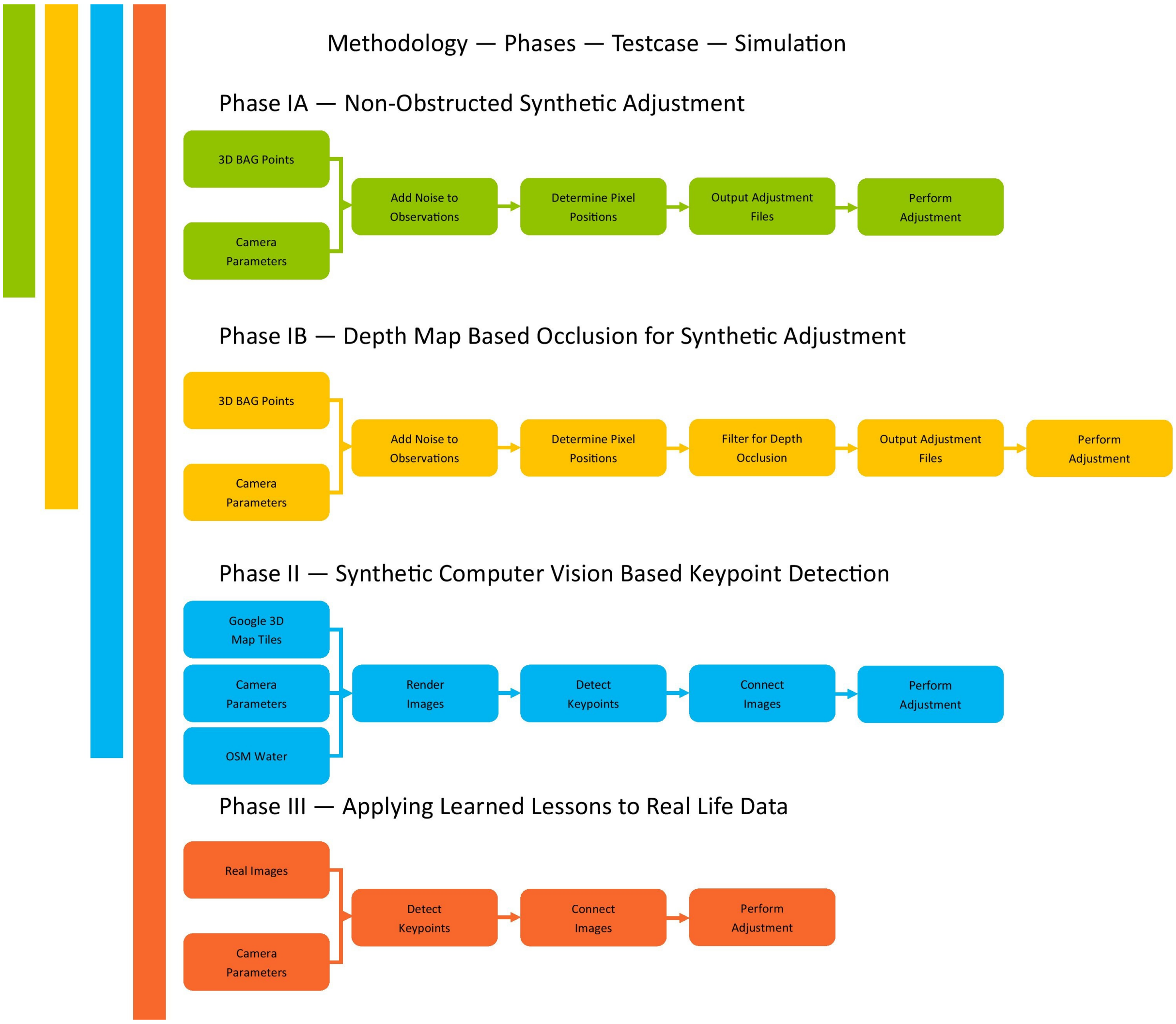

Methodology — Phases — Testcase — Simulation
Phase IA — Non-Obstructed Synthetic Adjustment
3D BAG Points
Camera Parameters
Add Noise to Observations
Determine Pixel Positions
Output Adjustment Files
Perform Adjustment
Phase IB — Depth Map Based Occlusion for Synthetic Adjustment
3D BAG Points
Camera Parameters
Add Noise to Observations
Determine Pixel Positions
Filter for Depth Occlusion
Output Adjustment Files
Perform Adjustment
Phase II — Synthetic Computer Vision Based Keypoint Detection
Google 3D Map Tiles
Camera Parameters
OSM Water
Render Images
Detect Keypoints
Connect Images
Perform Adjustment
Phase III — Applying Learned Lessons to Real Life Data
Real Images
Camera Parameters
Detect Keypoints
Connect Images
Perform Adjustment

Figure 24: Simplified form of the methodology showing the general steps for each phase

3.2 Phase IA: Non-Obstructed Synthetic Adjustment

As part of the broader framework outlined in Figure 24 and Section 3.1. The initial part of the methodology will look into the synthesisation of adjustment. This phase is called "Phase IA Non-Obstructed Synthetic Adjustment", which takes a point file and camera parameters to calculate the observations (see Section 2.12) for all points that fall within the camera footprint (see Section 2.9). As mentioned before (see Section 2.11), the chosen points are vertices from the 3D BAG OBJ files [Dukai et al., 2024] that have overlap with the focus area (see Section 1.5). Furthermore, a selection of 9 control points (see Section 2.10) was made for the inclusion of second-phase adjustment (see Section 2.21). For the camera parameters (see Section 2.8) and the optical centers (see Section 2.5), the Miramap 2024 Rotterdam flight plan (see Section 2.4) was used. Neither set of points has noise applied to it to allow for ground truth testing on repeatable simulations. The noise is only applied to the calculation of observations and outputted optical sensors (see Algorithm 3 and Algorithm 4). The added noise is done on an independent variable level, meaning that each test case only adds noise to a singular variable. To reduce uncertainty and variability, a Monte-Carlo simulation (see Section 2.24) is used, resulting in each test case containing 100 simulations. The implementation of phase IA is done in five steps, which are an expansion on the methodology shown in Figure 24. This phase is used to answer at least partially the following sub-questions:

- What is the mathematical pipeline for aerial adjustment theory?
- What hyperparameters are present in aerial adjustment, and what are their effects when changed in synthetic tests?

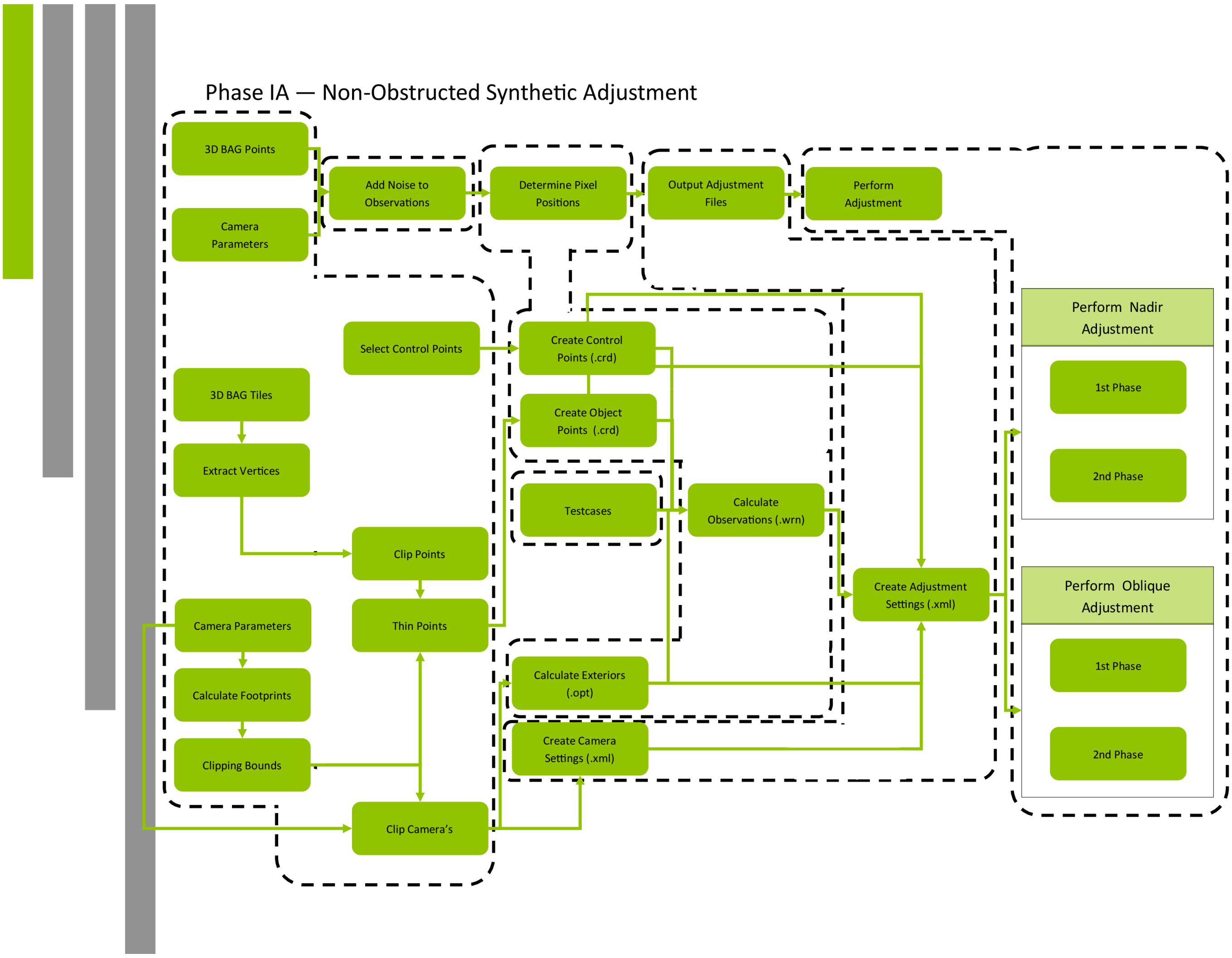

Phase IA — Non-Obstructed Synthetic Adjustment
3D BAG Points
Camera Parameters
Add Noise to Observations
Determine Pixel Positions
Output Adjustment Files
Perform Adjustment
Select Control Points
Create Control Points (.crd)
Create Object Points (.crd)
Testcases
Calculate Observations (.wrn)
Create Adjustment Settings (.xml)
3D BAG Tiles
Extract Vertices
Clip Points
Thin Points
Camera Parameters
Calculate Footprints
Clipping Bounds
Clip Camera's
Calculate Exteriors (.opt)
Create Camera Settings (.xml)
Perform Nadir Adjustment
1st Phase
2nd Phase
Perform Oblique Adjustment
1st Phase
2nd Phase

Figure 25: Phase IA: Non-Obstructed Synthetic Adjustment

The first step of the phase is reading and extracting from the input files. Which in the case of Phase IA are the most current version of 3D BAG [Dukai et al., 2024], which are based on AHN3 [Rijkswaterstaat, 2023] collected from 2014 to 2019 and AHN4 [Rijkswaterstaat, 2021] collected from 2020 to 2022. The other input files are the Miramap camera parameters from the 2024 collection campaign for the municipality of Rotterdam. Figure 25 shows the first part of the expanded methodology for phase IA, which relates to the selection of points and cameras.

As mentioned before, the chosen 3D BAG Tiles have overlap with the focus area (see Section 1.5). These tiles come in the form of singular OBJ files (see Section 2.13), which each hold around 2.000 objects (buildings). An OBJ file consists of different linked data types. For this phase, only the vertices marked with a "v" in the file are necessary. These vertices are placed in a vector, only maintaining their X Y Z. The process for this is shown in Algorithm 1.

Algorithm 1 Reading 3D BAG Tiles

Input: O (folder of i OBJ files), outerBounds
Output: X, Y, Z coordinates

```
foreach $o \in O$ do
    lines ← readlines($o$)  foreach line in lines do
        if $line[0] ==$ "v" and within(outerBounds, $X, Y, Z$) then
            return $X$, $Y$, $Z$
```

For the camera parameters, the optical center (.opt) file containing the name, X Y Z position, and ω ϕ κ rotation in degrees is used. Roughly 75.000 images were collected for the Miramap 2024 Rotterdam flight campaign. As such, the number of optical centers is reduced to only include the image of which the X Y Z position falls within the focus area with a 2 Kilometer (KM) buffer. For these remaining camera's a footprint is calculated. Thereafter, the number of optical centers is further reduced, only keeping optical centers for which the footprint center point falls within the focus area. The process for this is shown in Algorithm 2.

Algorithm 2 Reading and Clipping Optical Sensors

Input: O (optical sensor file with i cameras), outerBounds
Output: $name$, X, Y, Z, ω, ϕ, κ

```
foreach $c \in O$ do
    $name, X, Y, Z, \omega, \phi, \kappa$ ← readline($c$)  if within(outerBounds, $X, Y, Z$) then
        return $name$, $X$, $Y$, $Z$, $\omega$, $\phi$, $\kappa$
```

The determined footprints (see Section 2.9) are then further used to calculate the clipping bounds for the object points (see Section 2.11), which bounds the object points to the focus area (see Section 1.5). The bounded object points are then further reduced by thinning with a random selection. These thinned object points are given an identifier that is later used to link them to the determined pixel positions.

The final input file is the control points (see Section 2.10). As mentioned before, these are normally collected by the municipality using terrestrial methods; however, due to the relatively small size of the focus area (see Section 1.5), neither municipal points nor BM5 points are available within the focus area. The 9 control points selection was made from AHN4 [Rijkswaterstaat, 2021] (see Section 2.10) using Omnibase by Geodelta.

The second step of the expanded methodology (see Figure 25) for "Phase IA: Non-Obstructed Synthetic Adjustment" is "Add Noise to Observations", first shown in Figure 24. This noise is added in the form of thirteen independent test cases. These test cases are applied both to nadir and oblique adjustment. Each test case has a CaseType which is referenced in Algorithm 3 and Algorithm 4 as an identifier.

Description	CaseType
No noise simulation	NoneNoise
Simulate noise over X	XNoise
Simulate noise over Y	YNoise
Simulate noise over Z	ZNoise
Simulate noise over Omega ω	OmegaNoise
Simulate noise over Phi ϕ	PhiNoise
Simulate noise over Kappa κ	KappaNoise
Simulate noise over Principle Point Average (PPA)x	PPAxNoise
Simulate noise over Principle Point Average (PPA)y	PPAyNoise
Simulate noise in the Focal Distance	FocalNoise
Exclusively detect points in Von Gruber Squares 1 2 3	Gruber123Noise
Exclusively detect points in Von Gruber Squares 4 5 6	Gruber456Noise
Exclusively detect points in Von Gruber Squares 7 8 9	Gruber789Noise

Table 3: Description of the Different CaseTypes associated with Phase IA

This results in 1201 simulations, which are applied to four phases.

- first phase nadir adjustment (referred to as Phase1_Nadir)
- second phase nadir adjustment (referred to as Phase2_Nadir)
- first phase oblique adjustment (referred to as Phase1_Oblique)
- second phase oblique adjustment (referred to as Phase2_Oblique)

The total number of simulations for Phase IA is 4804.

As described before, the application of noise occurs in two functions, the first of which is noise on the extrinsic parameters (see Section 2.5). The process of which is shown in Algorithm 3. The concept is to loop over all cameras present in a set of cameras (see Section 1.5) and add a certain amount of individual noise to each associated variable. This noise follows a normal distribution (see Section 2.7) in which the mean μ is kept at zero. The choice for keeping the mean zero is done so that no bias is present, which is considered enough for the proof of concept in the framework. For the noise over X Y Z position which comes from the Real Time Kineticism (RTK) equipped GNSS-IMU plane, the standard deviation σ is kept at 0.05 meters. This value is $1/3$ of the maximum value described in [Polman and ir. M.A. Salzmann, 1996, p. 445–446].

In a similar sense, the standard deviation for the angles was chosen to be 1°. This value also corresponds to one-third of the maximum value described in [Polman and ir. M.A. Salzmann, 1996, p. 444]. The choice for these HTW values is made to leave some level of realism in the data. Only a singular set of standard deviations is used to reduce the number of potential tests and processing time.

Algorithm 3 Optical Sensor Noise Algorithm

```
Input: $C$ (set of $i$ cameras), CaseType
Output: $c_{name}$, $x$, $y$, $z$, $\omega$, $\phi$, $\kappa$
foreach $c \in C$ do
    $x \leftarrow c.x$  $y \leftarrow c.y$  $z \leftarrow c.z$  $\omega \leftarrow c.\omega$  $\phi \leftarrow c.\phi$  $\kappa \leftarrow c.\kappa$  if CaseType == XNoise then
        $x \leftarrow x + NormalDistribution(\mu, \sigma_{xyz})$
    if CaseType == YNoise then
        $y \leftarrow y + NormalDistribution(\mu, \sigma_{xyz})$
    if CaseType == ZNoise then
        $z \leftarrow z + NormalDistribution(\mu, \sigma_{xyz})$
    if CaseType == OmegaNoise then
        $\omega \leftarrow \omega + NormalDistribution(\mu, \sigma_{opk})$
    if CaseType == PhiNoise then
        $\phi \leftarrow \phi + NormalDistribution(\mu, \sigma_{opk})$
    if CaseType == KappaNoise then
        $\kappa \leftarrow \kappa + NormalDistribution(\mu, \sigma_{opk})$
    return $c_{name}, x, y, z, \omega, \phi, \kappa$
```

The second set of noise CaseType is applied in the calculation of the observations themselves (see Section 2.12). The process of which is shown in Algorithm 4. Unlike Algorithm 3, this is more focused on the synthetic simulation of the camera model (see Section 2.8). The function loops over all the cameras in a set and then all the points in a set (either object or control).

No parameters for focal distance are prescribed in the HTW; instead, it prescribes that the camera should be calibrated in the last year [Polman and ir. M.A. Salzmann, 1996, p. 443]. As such, the σ is set at 1 MM. This is seen as realistic for airplane-based aerial imagery collection. Similarly, no rules are set for the PPA; as such, a σ of 0.1 MM on the image plane is chosen. For the selection of Von Gruber squares, three cases are devised which relate to the lower third (1 2 3), middle section (4 5 6), or upper section (7 8 9) of the image as explained in Section 2.14.

Algorithm 4 Observation Noise Algorithm

```
Input: $C$ (set of $i$ cameras), $P$ (set of $j$ points), CaseType
Output: $u, v$
foreach $c \in C$ do
    $Rotation \leftarrow Rot(c.\kappa) \cdot Rot(c.\phi) \cdot Rot(c.\omega)$
    $translation \leftarrow (c.x, c.y, c.z)$
    $halfWidth \leftarrow c.sensorWidth/2$
    $halfHeight \leftarrow c.sensorHeight/2$
    if $CaseType == \texttt{FocalNoise}$ then
        $c.focal \leftarrow c.focal + noise$
    foreach $p \in P$ do
        $worldCoordinates \leftarrow (p.x, p.y, p.z)$
        $imageCoordinates \leftarrow Rotation \cdot (worldCoordinates - translation)$
        $u \leftarrow (c.focal \cdot imageCoordinates.x)/imageCoordinates.z$
        $v \leftarrow (c.focal \cdot imageCoordinates.y)/imageCoordinates.z$
        if $CaseType == \texttt{PPAxNoise}$ then
            $u \leftarrow u + noise$
        if $CaseType == \texttt{PPAyNoise}$ then
            $v \leftarrow v + noise$
        if $-halfWidth \leq u \leq halfWidth$ and $-halfHeight \leq v \leq halfHeight$ then
            if $CaseType == \texttt{Gruber123Noise}$ and $v > -(c.sensorHeight/6.0)$ then
                return $u, v$
            if $CaseType == \texttt{Gruber456Noise}$ and $v > (c.sensorHeight/6.0)$ then
                return $u, v$
            if $CaseType == \texttt{Gruber789Noise}$ and $v < (c.sensorHeight/6.0)$ then
                return $u, v$
            if $CaseType$ not in Gruber variants then
                return $u, v$
```

The third part of Phase IA is shown Figure 25, which deals with the calculation of the observation, which is already highlighted in Algorithm 4. The formula used is explained in Section 2.12.

The Geodelta Bundle adjustment program expects a dictionary in which the name of the camera is the same as in the extrinsic parameter file. This camera's name is used as the main key. it also needs to include a camera identifier (401, 402, 403, 404 and 405 see Section 2.2 and Section 2.3). The observation dictionary is filled with detected observations from the object (see Section 2.11) and, in case of second-phase adjustment, control points (see Section 2.10). The formula for the resulting u and v is given in Section 2.12.

The fourth step of Phase IA is the creation and writing of the adjustment files. This relates to the camera calibration and adjustment settings. Each simulation requires one adjustment settings file, which works as a linking agent between the different files created during steps 2 and 3. The number of camera calibrations depends on the phase of detection, in which nadir only has one camera, and oblique will always have four. These files are standardized by Geodelta for use in Bundle adjustment. For the camera settings first discussed in Section 2.8, the following parameters are used.

- Calibration mode (NoCorrection)
- Focal length (71 MM for nadir and 112 MM for oblique)
- Sensor width (40.0064 MM)
- Sensor height (53.3619 MM)
- Pixel width (3.76 μm)
- Pixel height (3.76 μm)
- PhotoAxis (PointsDown)

- Calibrated PPAx (0 MM)
- Calibrated PPAy (0 MM)

These values are based on the Miramap 2024 Rotterdam flight campaign. For the adjustment settings, the following settings are used:

- Conversion Model (InvertedSign further explained in Section 2.5)
- Extrinsic parameters (in Meter (M) and degrees) (see Section 2.5)
    - 401 (nadir adjustment)
    - 402 (Forward-facing camera in an oblique position)
    - 403 (Left-facing camera in an oblique position)
    - 404 (Backward-facing camera in an oblique position)
    - 405 (Right-facing camera in an oblique position)
- Points (in M)
    - Object Points (see Section 2.11)
    - Control Points (Only in case of second phase adjustment) (see Section 2.10)
- Observations (in MM) (see Section 2.12)
    - 401 (Nadir adjustment)
    - 402 (Oblique adjustment)
    - 403 (Oblique adjustment)
    - 404 (Oblique adjustment)
    - 405 (Oblique adjustment)
- Calibrations (see previous list)
    - 401 (nadir adjustment)
    - 402 (oblique adjustment)
    - 403 (oblique adjustment)
    - 404 (oblique adjustment)
    - 405 (oblique adjustment)
- Program Mode (Aerial)
- Calculation Type
    - Free Network (First phase adjustment) (see Section 2.20)
    - Absolute Constrained (Second phase adjustment) (see Section 2.21)
- Max Iterations (20 iterations)
- Standard Deviation Observations (4 μm) [Polman and ir. M.A. Salzmann, 1996, 446–447]
- Standard Deviation Control Points (0.03 meter, only second phase adjustment) [Polman and ir. M.A. Salzmann, 1996, 446–447]
- Standard Deviation GNSS XY (0.05 meter) [Polman and ir. M.A. Salzmann, 1996, 445–446]
- Standard Deviation GNSS Z (0.05 meter) [Polman and ir. M.A. Salzmann, 1996, 445–446]
- Standard Deviation IMU $\omega\phi$ (0.003 degrees) [Polman and ir. M.A. Salzmann, 1996, 443–445]
- Standard Deviation IMU κ (0.005 degrees) [Polman and ir. M.A. Salzmann, 1996, 443–445]

These values are also based on the Miramap 2024 Rotterdam flight campaign. The choice of values is made to reflect a realistic scenario. The fifth and final part of the methodology for Phase IA, as shown in Figure 25, is the running of the adjustments. As mentioned before, each CaseType contains 100 simulations, resulting in a total of 4804 simulations being run for Phase IA. For the accomplishment of multiple adjustments in succession, the Bundle Adjustment method has been extended to allow for batch processing. In case the max number of iterations (20) is exceeded, or the matrix becomes singular, the results in the simulation are deemed as failed.

3.3 Phase IB: Depth Map Based Occlusion for Synthetic Adjustment

The framework described in Section 3.1 separates the methodology for Phase I into two distinct sub-phases, Phase IA and Phase IB. This division allows specific sub-questions to be addressed using a simplified variant of the algorithm. As discussed in the related work, one of the primary challenges in oblique aerial image adjustment is occlusion (see Section 2.19). Occlusions reduce the number of connections between images taken from different flight lines and directions, directly impacting the reliability of feature matching (see Section 2.15).

To study this effect systematically, synthetic occlusion is simulated using OpenGL-based depth maps. The theoretical basis of occlusion is outlined in Section 2.19, and visualized at a 2D point level in Figure 26 and Figure 27.

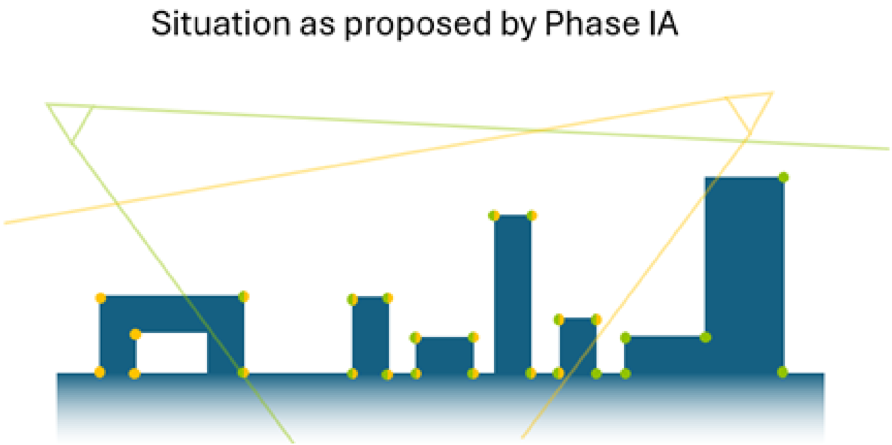


Figure 26: Schematic Situation of Two opposite Cameras Detecting All Points in Their Respective frustum Without Occlusion

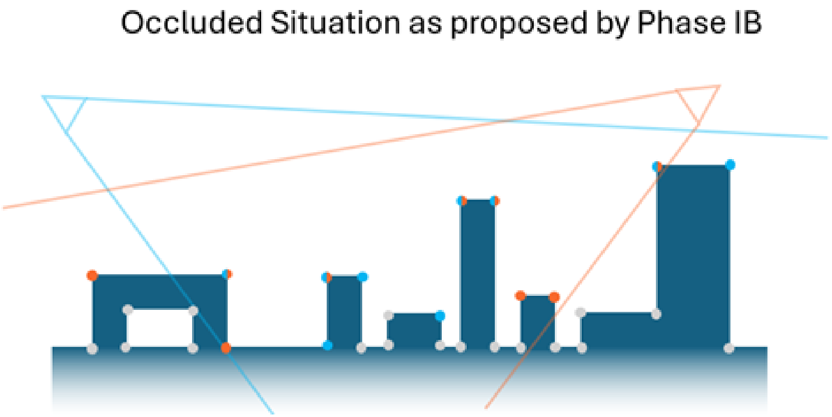


Figure 27: Schematic Situation of Two opposite Cameras Detecting Only Points in front of the Depth map due to Occlusion

In contrast to Phase IA, this phase introduces an occlusion-aware filtering mechanism using depth information. Since many steps are reused from Phase IA (see Section 3.2), only the new or extended steps are explained here.

This phase is specifically designed to answer the following sub-questions:

- What are the main differences between nadir and oblique image adjustment?
- How do the different parameters of oblique imagery influence the errors present in the final adjusted images?

Figure 28 illustrates the complete methodological workflow for Phase IB. Compared to Phase IA, two new components are introduced:

Building Extraction and OBJ Conversion: As an extension to Phase IA's Step 1, individual building objects are extracted from the 3D BAG tiles. This step outputs separate OBJ files for each building.

Synthetic Depth Rendering via OpenGL: These OBJ files are loaded into a custom OpenGL tool developed as part of this framework. A synthetic depth map is rendered from the camera's perspective.

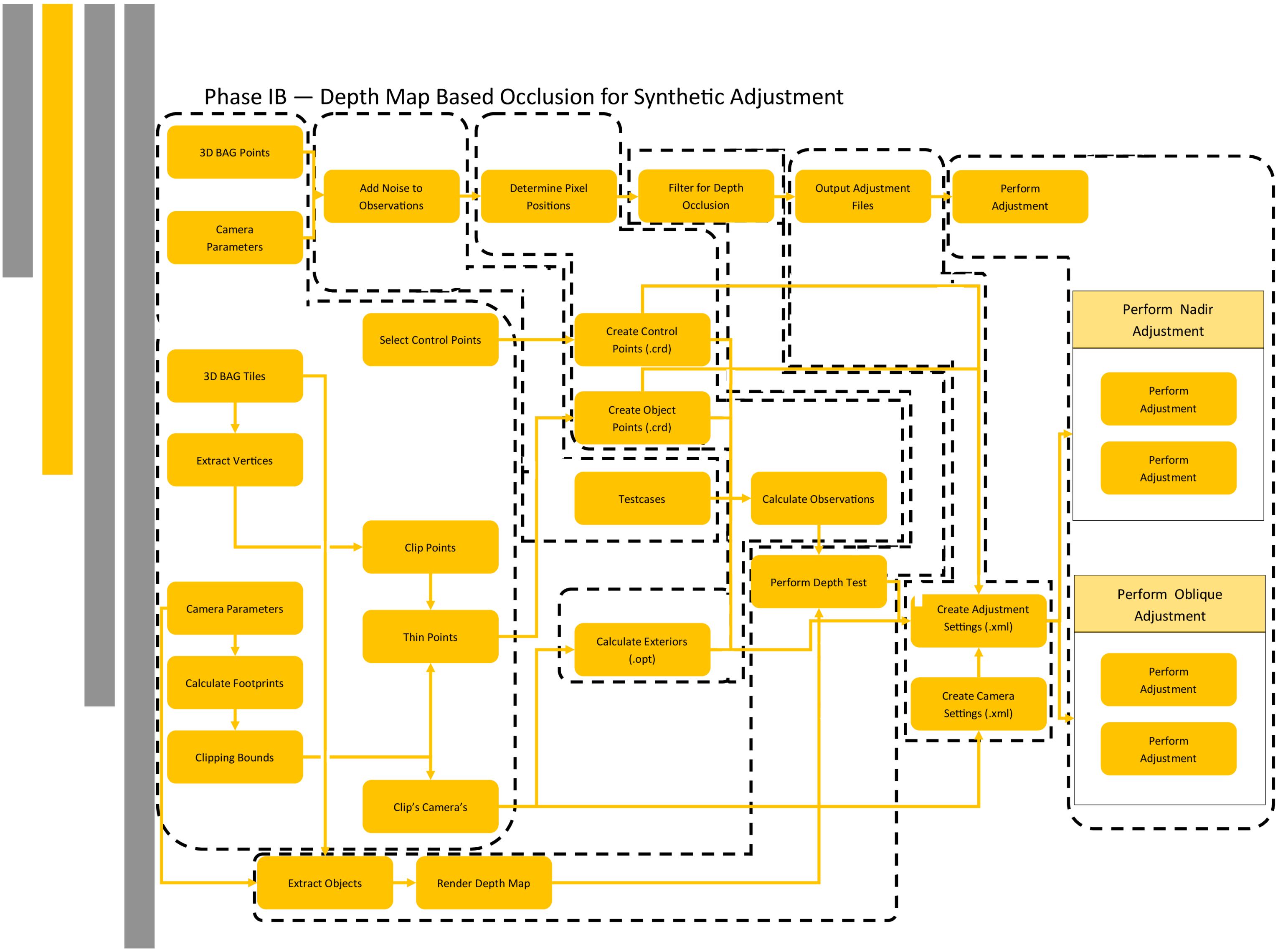


Figure 28: Phase IB: Depth Map Based Occlusion for Synthetic Adjustment

In computer graphics, a depth map is typically used to simulate shadows, rendered from the light's position and evaluated from the camera's perspective. In such cases, each pixel encodes the distance to the first surface it intersects, enabling shadow calculations. The novel adaptation in this study repurposes depth maps to identify and filter out occluded points rather than simulate lighting.

To visualize this process, Figure 29 presents a scene viewed from a standard camera, overlaid with a depth map generated using a full RGB rainbow color map. While OpenGL typically uses a single-channel (8-bit) grayscale texture for depth, a color-coded map was chosen here to facilitate visual interpretation. Each color encodes a unique depth value.

In the rendered scene:

- Vertices detected in the current frame are shown and serve as ground truth points, assumed to be noise-free
- The camera's extrinsic parameters (position and orientation) are used to both define the viewpoint and project the depth map onto the scene

A depth test is then executed as follows:

Algorithm 5 Depth-Based Occlusion Culling

```
Input: $P$ (set of 3D points), $D$ (depth map), $C$ (camera with known position and projection model)
Output: VisiblePoints
VisiblePoints $\leftarrow$ {}
foreach $p \in P$ do
    $(u, v) \leftarrow$ ProjectToImage(p, C)
    $d_p \leftarrow$ ComputeDepth(p, C.position)
    $d_{map} \leftarrow D[u, v]$
    if $d_p \leq d_{map}$ then
        VisiblePoints.append($p$)
return VisiblePoints
```

Figure 29 also shows the result of this filtering: occluded points are marked in red, while visible points are retained and used for the remaining phase, consistent with Phase IA Step 4. The test cases for Phase IB are the same as those used in Phase IA.

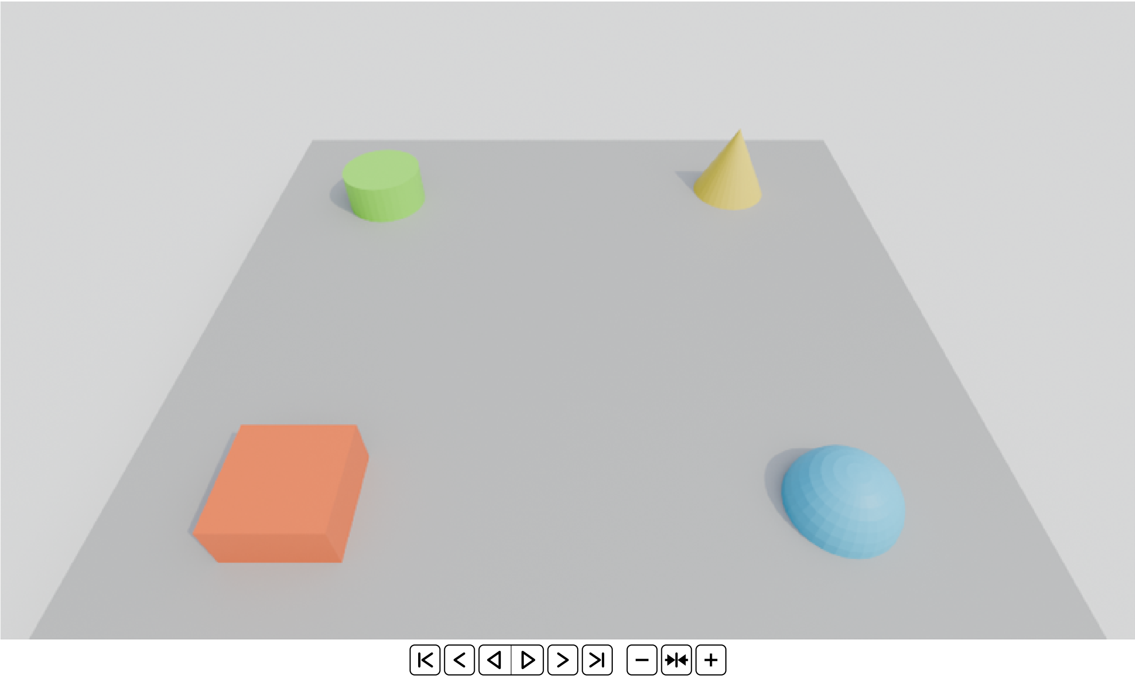


Figure 29: Simple overview of a depth map viewed from a normal camera with a basic mesh overlay (Best viewed in Adobe Acrobat)

3.4 Phase II: Synthetic Computer Vision Based Object Point Detection

The second phase proposed in this framework (see Section 3.1) introduces a novel method for testing and assessing object point detection and matching algorithms. This is achieved by synthesizing the images upon which detection is performed in the aerial collection pipeline (see Section 2.1). The method involves the synthetic creation of rendered images using Blender (see Section 2.17). A broad overview of this phase is provided in Figure 24.

Since this method is independent of the synthetic adjustment approach in Phase I, additional explanation is required, as none of the activities overlap between the two phases. The general idea behind the phase is to use 3D tiles from Google Maps and render scenes based on real camera parameters, including both the intrinsic (see Section 2.8) and extrinsic parameters (see Section 2.5). Rendering can also simulate various exogenous conditions such as sun, shade, rain, and clouds, alongside normal clear conditions (see Section 2.17).

Due to the significant processing time associated with each simulation, only one test case is performed per condition. This phase addresses the following sub-questions:

- What are the main differences between nadir and oblique image adjustment?
- How does the choice between object point extraction algorithms such as SIFT/LightGlue and DISK/LightGlue affect the reconstruction accuracy?
- How do simulated external factors such as sun position, wind, clouds, and foliage affect the adjustment?

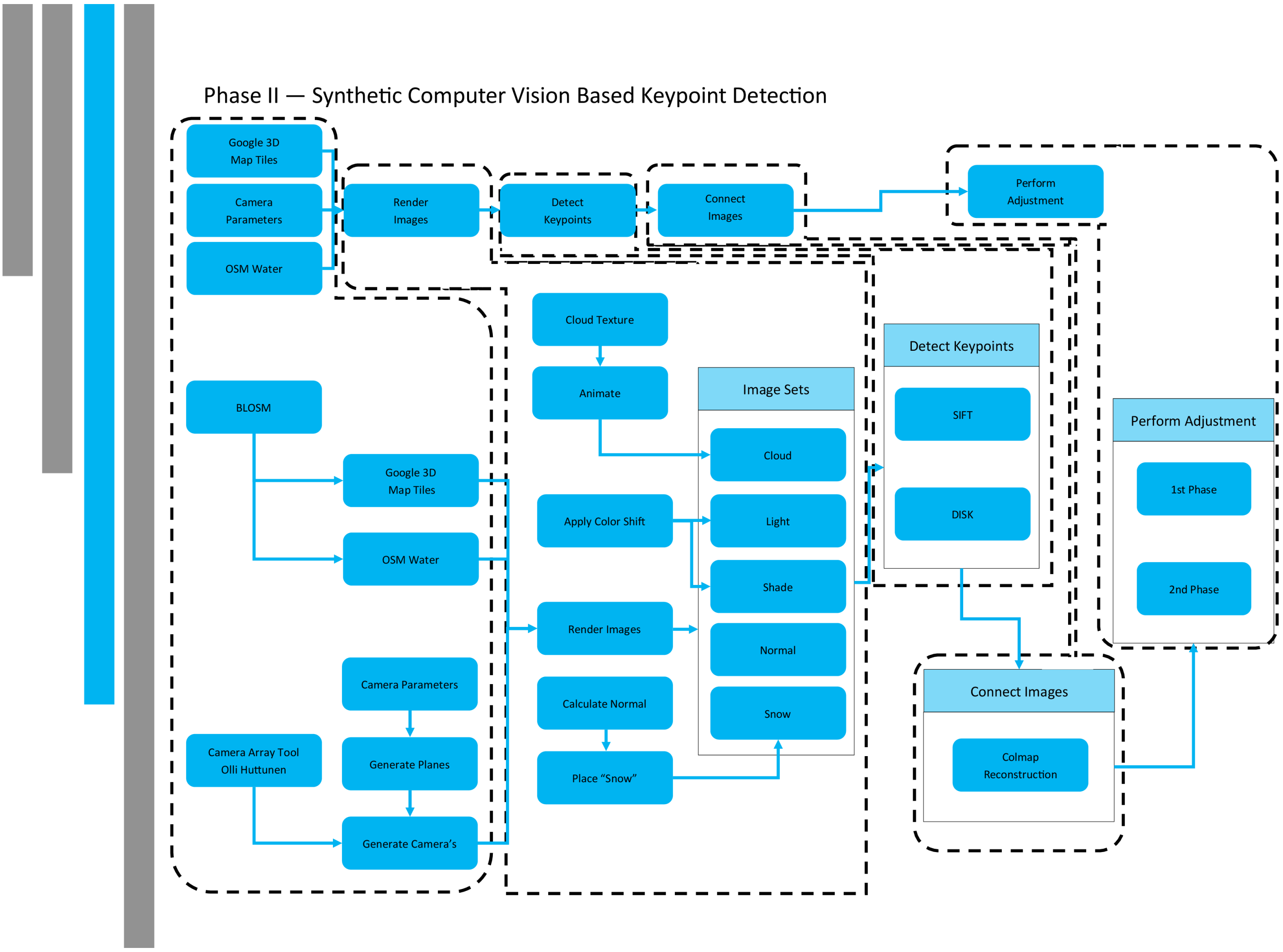


Figure 30: Phase II: Synthetic Computer Vision Based Object Point Detection

Camera	Image Set	Extraction Method
Nadir	Normal	DISK
		SIFT
	Light	DISK
		SIFT
	Shade	DISK
		SIFT
	Snow/Rain	DISK
		SIFT
	Clouds	DISK
		SIFT
Oblique	Normal	DISK
		SIFT
	Light	DISK
		SIFT
	Shade	DISK
		SIFT
	Snow/ Rain	DISK
		SIFT
	Clouds	DISK
		SIFT

Table 4: Testcases of Phase II

All test cases are shown in Table 4. Each exogenous condition is tested for both nadir and oblique views using both DISK and SIFT extraction methods. In total, five image sets are created.

Blender Open Street Map (BLOSM) [Prochitecture, 2025] is a Blender add-on developed by Prochitecture, which allows users to import various 3D tiles into Blender as triangulated meshes. To render (see Section 2.17) realistic-looking imagery, high-quality Google Maps 3D tiles are used. The specific area utilized (see Section 1.5) is shown in Figure 31.

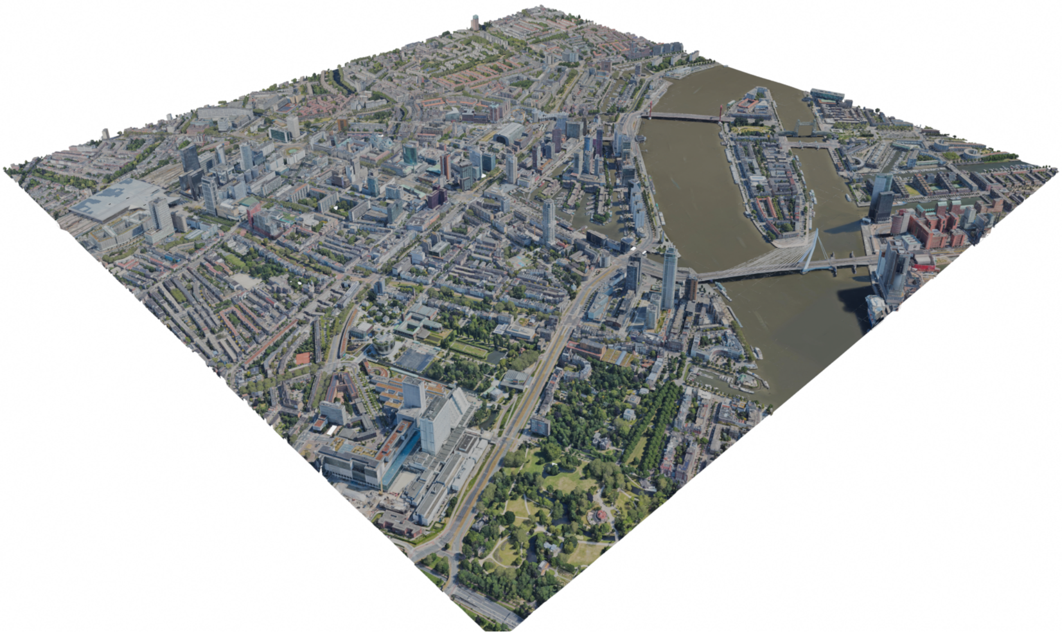


Figure 31: The used Google Maps 3D Tiles for the focus area

BLOSM also support the inclusion of OSM layers. This feature is used to overlay OSM water polygons, simulating unmatchable object points over water surfaces. Normally, water surfaces exhibit shifting textures that prevent stable object point detection. However, Google 3D tiles use Poisson reconstruction, which bakes textures into a static mesh (see Section 2.13); this results in the dynamic nature of water being lost.. To simulate this, a flat polygon with a solid detection-invariant color (e.g. magenta) is overlaid on top of the water surface. An example is shown in Figure 32.

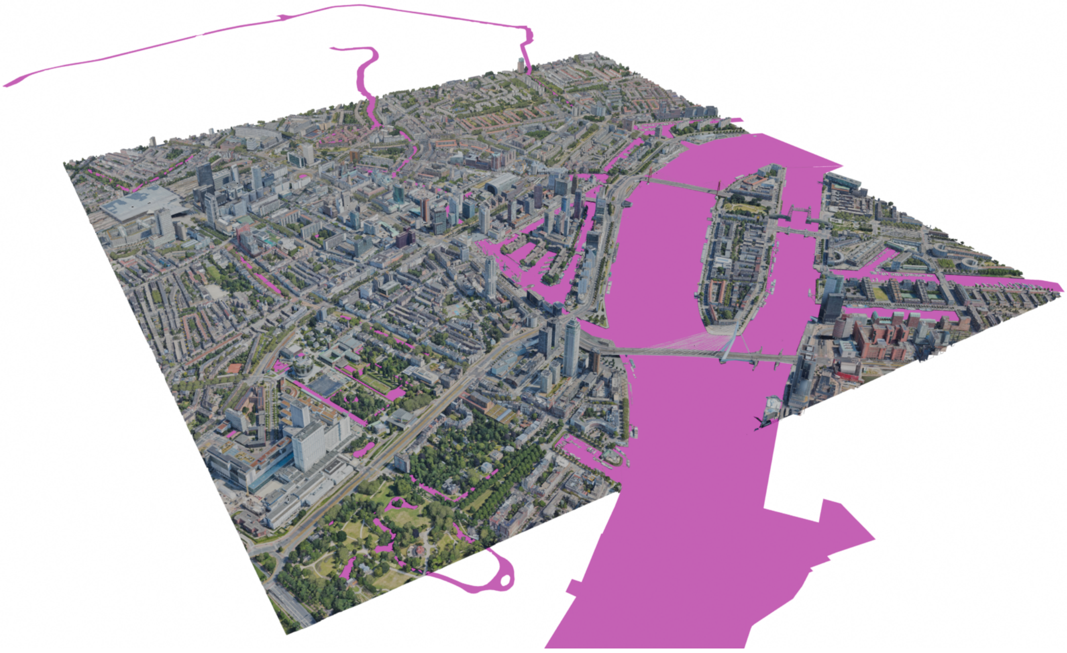


Figure 32: OSM Water overlaid on Google Maps 3D Tiles as a Mask with a Magenta Colour

To render the images, an array of cameras is generated in Blender. The position and rotation of these cameras are derived from the extrinsic parameters provided by Miramap (see Section 2.5). To allow Blender to place cameras at these positions with the necessary rotation, a Blender add-on called Camera Array Tool by [ToppiNappi, 2025] is used. This Blender add-on allows for the creation of multiple cameras and batch exports the rendered images. The scaling factors of 1.3 and 0.7 were chosen to produce visually perceptible yet moderate changes in brightness, effectively simulating overexposure and underexposure without causing clipping or excessive loss of detail in most images.

- Normal Image Set contains no alterations and serves as a baseline or ground truth.
- simulate overexposure by scaling the R, G, and B channels by factor of 1.3
- Shadow Image Set simulate underexposure by scaling the R, G, and B channels by factor of 0.7
- Cloud Image Set includes synthetic clouds to simulate occlusion (see Section 2.19). An animated cloud texture is used, slightly shifting between renders to mimic cloud movement. The texture is shown in Figure 33.
- Snow/Rain Image Set mimics puddling or moisture by randomly selecting triangles whose surface normals point upward and applying the detection-invariant magenta color

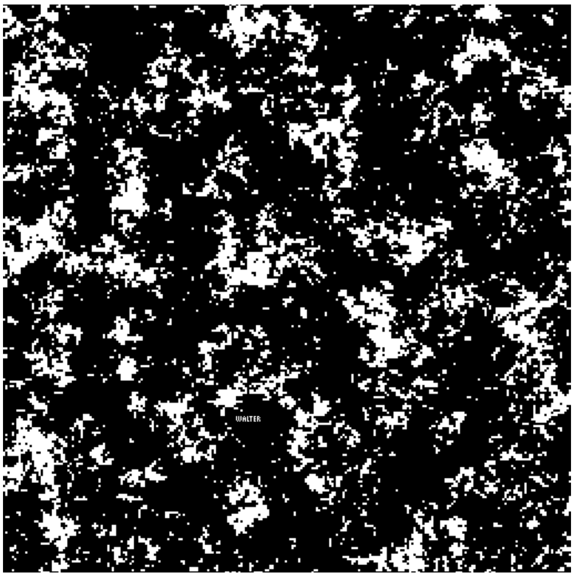


Figure 33: Animated Cloud Texture (Best Viewed in Adobe Acrobat)

With the rendered (see Section 2.17) images generated, the next step is the creation of a matching database. This database encodes image overlap using calculated footprints (see Section 2.9), which in turn determine candidate image pairs for object point matching (see Section 2.15).

SIFT operates by generating image pyramids and extracting object points using a Difference of Gaussians. These object points are described using 128-dimensional vectors for matching. DISK, on the other hand, downsamples the image, overlays an 8 × 8 grid, and selects the strongest object point in each cell (see Section 2.11). These points are similarly encoded with 128-dimensional descriptors. Both SIFT and DISK features are matched using the LightGlue matcher.

The next step is the reconstruction based on the LightGlue matches, performed entirely within COLMAP [Schönberger and Frahm, 2016]. COLMAP uses the database to perform structure-from-motion.

The final step is the adjustment, using outputs from the previous step. As in Phase I, these adjustments yield several data files, which are extracted and analyzed to generate the results shown in Section 4. Unlike Phase I, this step does not include Monte Carlo simulations due to the heavy computational load of rendering. The processing time of each test case depends on the rendering and feature extraction complexity, but it is not used as a quality metric, as it falls outside the scope of this research.

3.5 Phase III: Applying Learned Lessons to Real Life Data

Similar to how Phase IB is an extension of Phase IA, this final phase builds on both Phase IA, IB, and II. It applies the developed framework to real-life data from the Miramap flight for the Rotterdam 2024 flight campaign. The goal is to test the viability of the framework proposed in Section 3.1 by evaluating the applicability of earlier phases. Since most of the steps overlap with the previous phases, only the new activities are described here. The phase is called "Phase III: Applying Learned Lessons to Real Life Data".

This phase aims to answer the following sub-questions:

- What are the main differences between nadir and oblique image adjustment?
- How do the different parameters of oblique imagery influence the errors present in the final adjusted images?
- How does the choice between object point extraction algorithms such as SIFT/LightGlue and DISK/LightGlue affect the reconstruction accuracy?

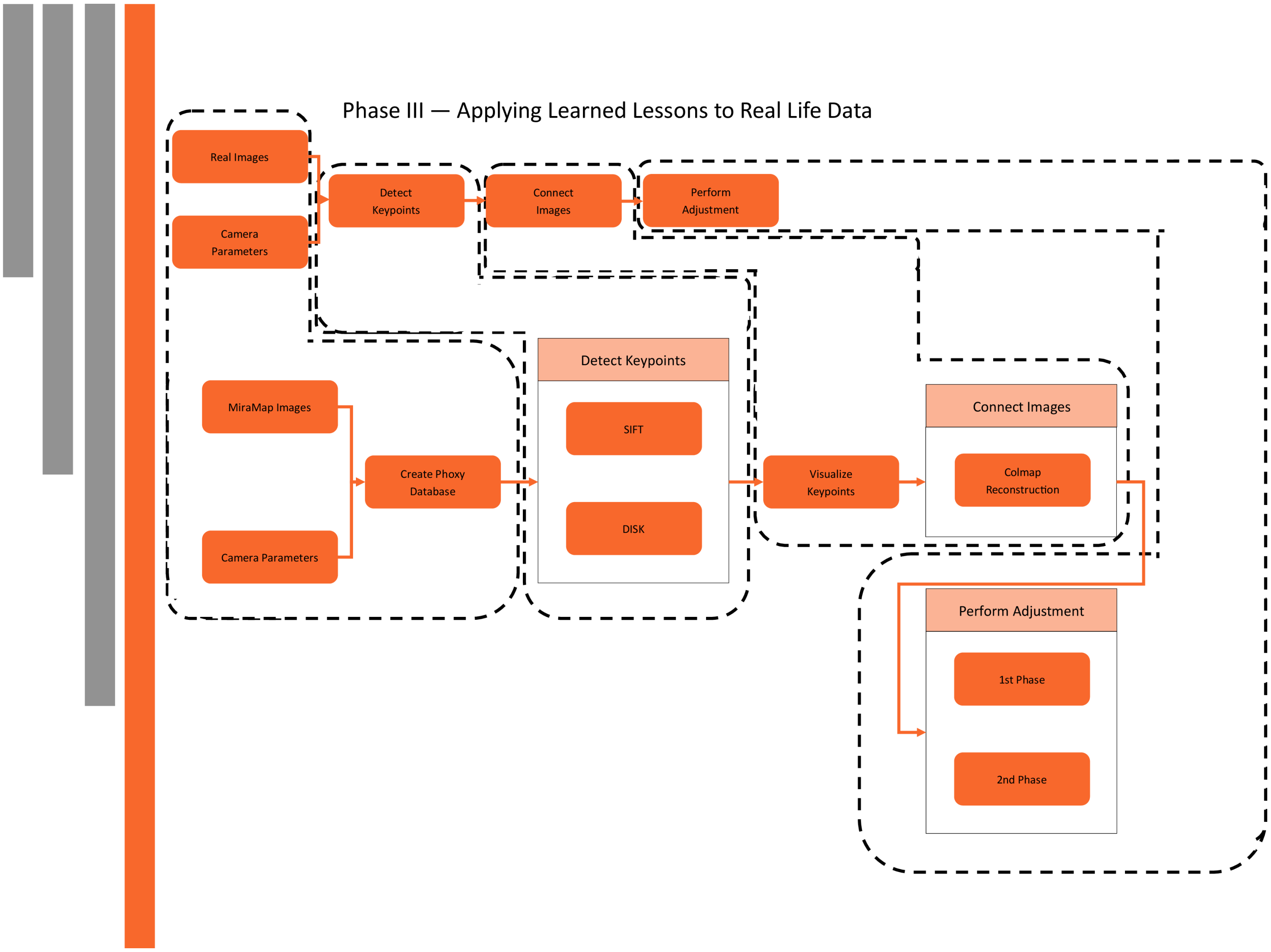


Figure 34: Phase III Applying Learned Lessons to Real Life Data

One of the key advantages of the multi-phase combination presented in this methodology is the ability to validate and transfer insights gained from synthetic data to real-world scenarios. By first experimenting with controlled, noise-free synthetic environments (see Section 3.2, Section 3.3, and Section 3.4), the methodology allows for a focused and detailed examination of the impact of various parameters on image adjustment and reconstruction quality. This sandbox-like setting offers the following benefits:

By applying these lessons to the 2024 Rotterdam dataset, this phase serves to validate the robustness and generalization of the proposed framework (see Section 3.1). The ability to replicate trends and behaviors observed in synthetic data (e.g., the influence of oblique angles, the impact of object point choice) strengthens confidence in the pipeline and its underlying assumptions. Moreover, discrepancies between synthetic and real data outcomes help identify areas where further refinement or additional modeling is needed (see Section 1.4), particularly concerning sensor noise, lens distortion, and urban scene complexity (Section 2.8).

In summary, bridging synthetic and real data enhances both the efficiency and reliability of the photogrammetric framework (see Section 3.1), ultimately contributing to more accurate and scalable oblique aerial mapping solutions.

The main addition in this phase, compared to previous ones, is the use of images from the 2024 Rotterdam campaign, collected by Miramap using a Leica CityMapper II. The relevant test cases for this phase are presented in Table 5.

Camera	Image Set	Extraction Method
Nadir	2024 Rotterdam Campaign	DISK
		SIFT
Oblique	2024 Rotterdam Campaign	DISK
		SIFT

Table 5: Testcases of Phase III

Miramap exported the adjustment with the following calibration parameters, which are consistent with those used in Phases I and II:

- Focal Length (71 MM for nadir and 112 MM for oblique)
- Sensor width (40.0064 MM)
- Sensor height (53.3619 MM)
- Pixel width (3.76 μm)
- Pixel height (3.76 μm)
- Calibrated PPAx (0 MM)
- Calibrated PPAy (0 MM)

Additionally, Miramap chose the following adjustment choices:

- Conversion Model (InvertedSign see Section 2.5)
- Image Direction (PointsUp see Section 2.12)

4 Results

Chapter 4 presents the results of the novel steps undertaken in the framework proposed in Chapter 3. The chapter is structured similarly to the previous one, meaning that each phase and sub-phase has its own corresponding results section. This chapter is partially used to answer the main and sub-questions defined in Chapter 1 by showing the process of completion, as well as providing short intermediate conclusions for each phase. The chapter shows the created synthetic data and highlights which of the proposed test cases were successful. Chapter 5 analyses the result from a purely adjustment-focused perspective.

4.1 Results of Phase IA: Non-Obstructed Synthetic Adjustment

As stated in Chapter 1, a sub-focus of the research is in the adjustment and detection steps of the framework (see Section 3.1), in which Phase IA is used to create synthetic adjustment based on 3D BAG vertices using the 2D projection formula (see Section 2.12). To illustrate this concept further, the perspective from both a nadir and oblique "camera" center is shown in Figure 35.

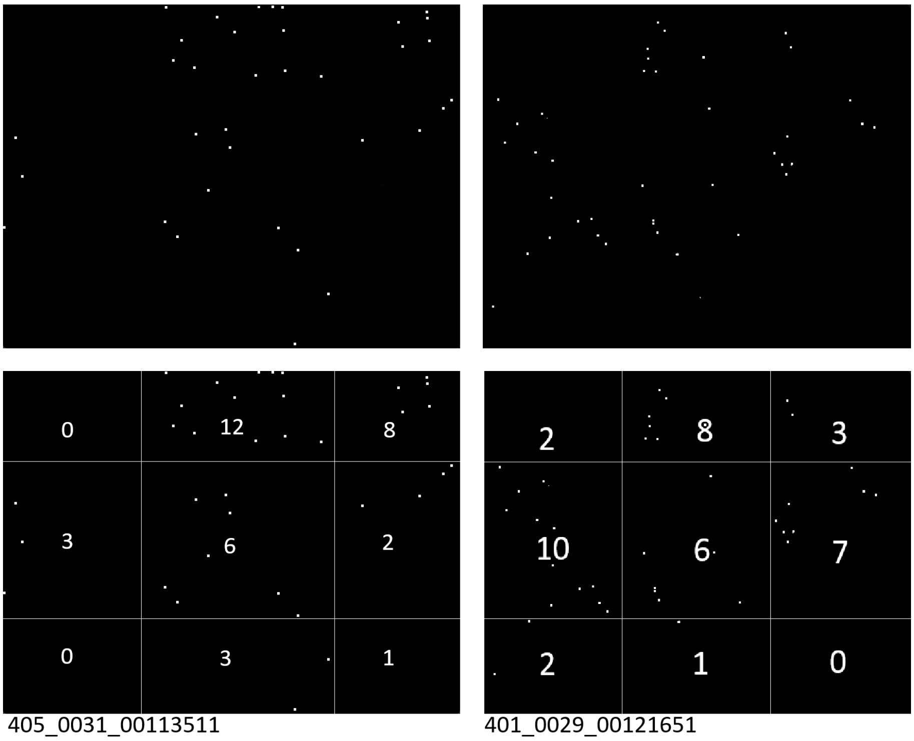


Figure 35: Gruber Square Point Count for both 405_0031_00113511 (30 points) and 401_0029_00121651 (39 points)

Despite detecting a similar number of projected points (39 for the nadir and 30 for the oblique image), their spatial distributions differ significantly. In particular, several Von Gruber squares remain unpopulated in both images, indicative of uneven coverage and potentially unstable image connections [Polman and ir. M.A. Salzmann, 1996, p. 434]. This lack of coverage can stem from an unfavorable distribution of 3D BAG buildings, which may limit how many features fall within view.

All projections are performed using the standardized PointsUp direction (see Section 2.8), and a rotation is applied to match the Miramap 2024 alignment during the observation calculation. According to the HTW guidelines, a minimum of 16 well-distributed points is required to establish a valid geometric connection between images [Polman and ir. M.A. Salzmann, 1996, p. 433]. However, many images analyzed in this study fall short of this threshold, particularly in foliage-covered zones or over water. Figure 36 provides an overview of non-occluded point counts for all cameras in the study area, with image centers marked by squares. The color-coded legend indicates compliance with HTW thresholds.

- Low counts near the Maas River are expected due to the non-existence of buildings
- Sparse detections within the foliage around the Euromast.

Both these elements are seen as realistic.

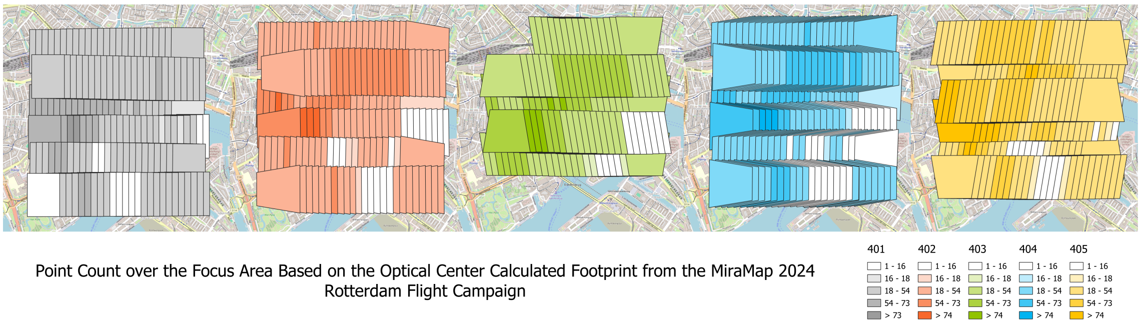


Figure 36: Non-Obstructed Point Count of Images within the Focus Area

As mentioned in Chapter 3, several CaseTypes have been defined and considered for the process of observation calculation. The NoneNoise CaseType corresponds to the perfect scenario in which no noise is added. The standard deviation values corresponding to each CaseType are defined in Section 3.2.

	Point Set	CaseType	Number of Simulations	Number of Points	Adjustment Phase
Phase IA: Non-Obstructed Synthetic Adjustment	3D BAG and Control Points (Second Phase Adjustment)	NoneNoise	1	1000	First and Second Phase Adjustment
		XNoise	100		
		YNoise			
		ZNoise			
		OmegaNoise			
		PhiNoise			
		KappaNoise			
		PPAxNoise			
		PPAyNoise			
		FocalNoise			
		Gruber123Noise			
		Gruber456Noise			
		Gruber789Noise			

Table 6: Overview of CaseType Configurations Used in Phase IA Simulations

All test cases associated with Phase IA: Non-Obstructed Synthetic Adjustment are shown in Table 6. The table also shows the basic hyperparameters present in the system setup. Such as the number of simulations being set at 100 for twelve of the thirteen CaseTypes for which noise is introduced either in the optical sensors or the observations. The algorithms for noise calculation of observations are shown in Section 3.2.

As mentioned before, a full analysis of the different test cases is given in Chapter 5. Table 7 provides a starting overview of the different average F-tests based on the simulation, where a critical or passing value is below 1. The concept of an F-test is explained in Section 2.23.

The results in Table 7 provide a clear overview of how different types of synthetic noise influence the adjustment process, as measured by the average F-test values. As expected, the NoneNoise case yields a result of zero across all test categories, confirming that the system performs correctly when no perturbations are introduced, thus validating the baseline behavior of the implementation.

	CaseType	Avg. F-test 1st Phase Nadir	Avg. F-test 2nd Phase Nadir	Avg F-test 1st Phase Oblique	Avg. F-test 2nd Phase Oblique
Phase IA: Non-Obstructed Synthetic Adjustment	NoneNoise	0.000	0.000	0.000	0.000
	XNoise	0.010	0.009	0.008	0.008
	YNoise	0.010	0.010	0.010	0.008
	ZNoise	0.011	0.009	0.005	0.005
	OmegaNoise	2.407	2.356	2.117	2.270
	PhiNoise	2.107	1.764	5.514	2.474
	KappaNoise	1.053	1.339	1.080	0.845
	PPAxNoise	282.099	280.432	73.582	74.639
	PPAyNoise	308.276	306.053	75.890	72.480
	FocalNoise	1.278	1.239	0.676	0.650
	Gruber123Noise	0.000	0.000	0.000	Failed
	Gruber456Noise	0.000	0.000	0.000	Failed
	Gruber789Noise	0.000	0.000	0.000	Failed

Table 7: Average F-Test Results for Various CaseTypes in Phase IA (Nadir and Oblique, First and Second Phases)

The noise applied to the extrinsic parameters (XNoise, YNoise, ZNoise, OmegaNoise, PhiNoise, KappaNoise) generally results in low to moderate F-test values, indicating that the adjustment system is relatively robust to small perturbations in pose-related parameters. Notably, rotation-related noise (especially PhiNoise and OmegaNoise) tends to have a slightly higher impact than translation, suggesting a greater sensitivity of the adjustment process to angular deviations.

In contrast, noise applied to the intrinsic and calibration parameters (PPAxNoise, PPAyNoise, FocalNoise) shows a much more significant effect on the F-test results. The particularly high values for PPAxNoise and PPAyNoise (exceeding 280 in the nadir phases and 70 in oblique ones) indicate that inaccuracies in principal point coordinates severely disrupt the adjustment process. Focal length noise also contributes noticeably, though to a lesser degree.

Interestingly, the Von Gruber noise cases yield zero F-test results in most scenarios, indicating that the system handles these test configurations well in the nadir setup. However, all three fail during the second phase oblique case, suggesting a consistent breakdown in handling these noise types in more geometrically complex configurations—an issue that warrants further investigation.

In summary, the adjustment system demonstrates solid resilience to pose-related noise but is highly sensitive to calibration noise, particularly to errors in principal point estimation. This implies that accurate internal calibration is critical for successful adjustment.

4.2 Results of Phase IB: Depth Map Based Occlusion for Synthetic Adjustment

The second part of the first phase builds upon the foundation established in Phase IA, extending its methodology by introducing depth testing to mimic occlusion. This extension is formally referred to as Phase IB: Depth Map Based Occlusion for Synthetic Adjustment, and is described in detail in Section 3.3. The primary innovation lies in integrating a depth map generated via a custom C++ OpenGL pipeline, allowing forward intersection calculations to consider visual occlusions due to 3D structures.

To illustrate this advancement, Figure 37 shows the rendered depth map with its corresponding 3D BAG building geometry. This building data is used to produce the depth map shown alongside it, where each pixel encodes the depth value from the camera center to the first intersecting surface in the 3D BAG dataset.

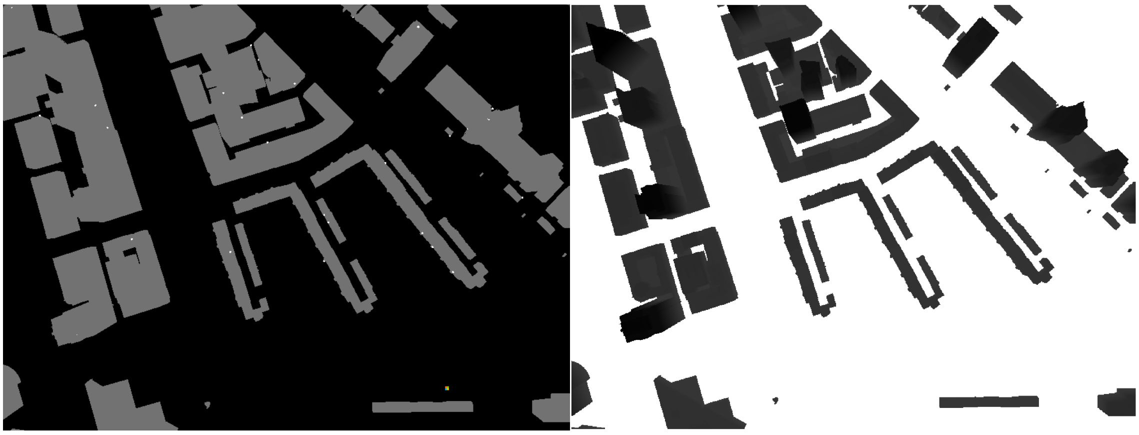


Figure 37: 3D BAG Geometry and Corresponding Depth Map for a Nadir View

A similar occlusion analysis is conducted for the oblique image 405_0031_0013511, shown in Figure 38. Again, the depth map visually demonstrates how urban geometry affects line-of-sight visibility in an oblique viewing configuration.

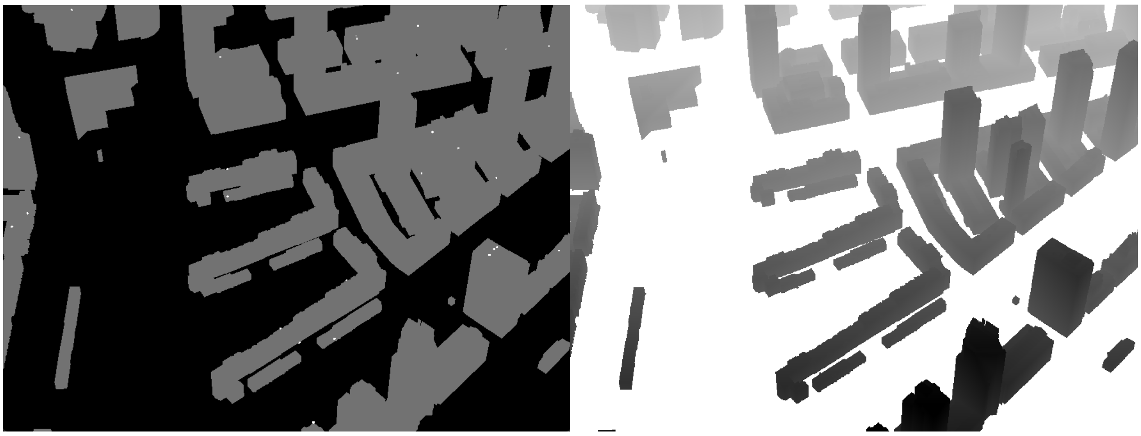


Figure 38: 3D BAG Geometry and Corresponding Depth Map for an Oblique View

Incorporating depth-based occlusion filtering significantly impacts the point visibility calculation. Figure 39 shows the resulting point counts for nadir and oblique images, respectively. When compared to Figure 36 from Phase IA, a key observation is the overall reduction in visible points per image, despite the total number of projected points being doubled. This confirms that occlusions (previously ignored in IA) exclude many previously accepted but invalid intersections.

Notably, nadir imagery seemingly suffers more severely under occlusion constraints this has to do with the case setup having many points occluded due to point density in 3D BAG, with a larger number of images falling below the 16-point threshold required by HTW standards. This could be attributed to the

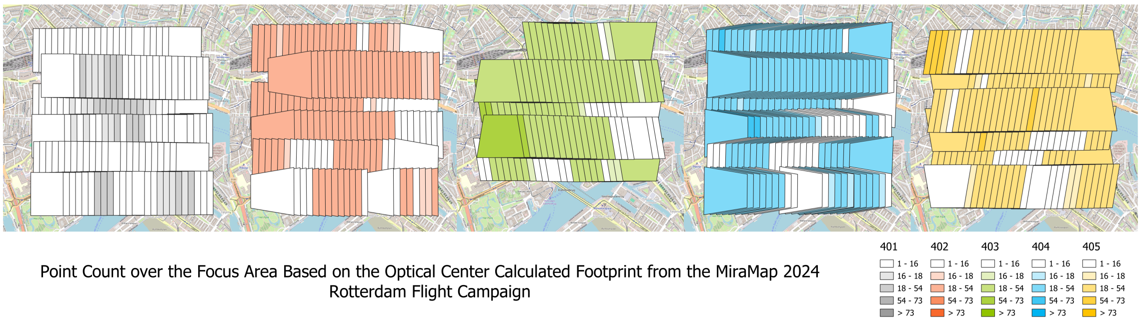

Point Count over the Focus Area Based on the Optical Center Calculated Footprint from the MiraMap 2024 Rotterdam Flight Campaign
401
402
403
404
405
1 - 16
16 - 18
18 - 54
54 - 73
> 73

Figure 39: Depth Map Point Count of Images within the Focus Area

vertical viewing direction's increased susceptibility to building blockage, whereas oblique views—though geometrically more complex—often maintain clearer lines of sight into urban canyons and facades.

To maintain comparable simulation conditions across both Phase IA and Phase IB, the same thirteen CaseType were tested again. The setup is summarized below.

	Point Set	CaseType	Number of Simulations	Number of Points	Adjustment Phase
Phase IB: Depth Map Based Occlusion for Synthetic Adjustment	3D BAG and Control Points (Second Phase Adjustment)	NoneNoise	1	1000	First and Second Phase Adjustment
		XNoise	100		
		YNoise			
		ZNoise			
		OmegaNoise			
		PhiNoise			
		KappaNoise			
		PPAxNoise			
		PPAyNoise			
		FocalNoise			
		Gruber123Noise			
		Gruber456Noise			
		Gruber789Noise			

Table 8: Overview of CaseType Configurations Used in Phase IB Simulations

After the 4800 simulations are generated and adjusted, an average F-test per CaseType can be calculated and presented in Table 14.

	CaseType	Avg. F-test 1st Phase Nadir	Avg. F-test 2nd Phase Nadir	Avg. F-test 1st Phase Oblique	Avg. F-test 2nd Phase Oblique
Phase IB: Depth Map Based Occlusion for Synthetic Adjustment	NoneNoise	0.000	0.000	0.000	0.000
	XNoise	0.034	0.033	0.011	0.015
	YNoise	0.044	0.039	0.015	0.013
	ZNoise	0.011	0.011	0.008	0.008
	OmegaNoise	6.513	9.113	2.744	3.064
	PhiNoise	6.665	6.220	3.732	3.421
	KappaNoise	1.310	1.735	1.119	1.229
	PPAxNoise	175.109	161.095	285.268	286.778
	PPAyNoise	366.599	341.614	296.492	297.105
	FocalNoise	0.992	1.216	0.627	0.572
	Gruber123Noise	0.000	0.000	0.000	0.000
	Gruber456Noise	0.000	0.000	0.000	0.000
	Gruber789Noise	0.000	0.000	0.000	0.000

Table 9: Average F-Test Results for Various CaseTypes in Phase IB (Nadir and Oblique, First and Second Phases)

The results in Table 14 present a comprehensive overview of how different synthetic noise types affect the adjustment process during Phase IB, using depth map–based occlusion for synthetic adjustment. The average F-test (see Section 2.23) values are reported across both nadir and oblique imaging configurations, as well as for the first and second phases of processing.

As in previous findings Table 7, the NoneNoise scenario produces zero F-test values across all configurations, reinforcing the correctness of the adjustment system in noise-free conditions and serving as a reliable baseline.

Noise introduced to the extrinsic parameters (XNoise, YNoise, ZNoise, OmegaNoise, PhiNoise, KappaNoise) (see Section 2.5) results in varying levels of impact. Among the translation noises, YNoise produces the highest F-test values ($\sim$ 0.044 nadir, $\sim$ 0.015 oblique), slightly more disruptive than XNoise and ZNoise. However, all translation-related noise types remain at relatively low F-test levels, indicating the system's robustness to minor positional perturbations.

By contrast, rotation-related noise—especially OmegaNoise and PhiNoise—yields significantly higher F-test values (up to 9.1), confirming that the adjustment process is more sensitive to rotational errors. This aligns with expectations, as angular deviations more directly affect image orientation and projection geometry. The values for KappaNoise, while non-zero, are considerably lower, suggesting a relatively lower sensitivity to this particular rotation axis in the context of the occlusion-adjusted process.

When it comes to intrinsic calibration parameters (see Section 2.8), the results demonstrate a substantial degradation in adjustment accuracy. Both PPAxNoise and PPAyNoise produce extremely high F-test values, with averages exceeding 175–366 in nadir and approximately 285–297 in oblique configurations. This substantial spike highlights the system's acute sensitivity to errors in principal point location, suggesting that even minor perturbations in the principal point significantly destabilize the solution. FocalNoise shows a moderate but noticeable impact, with values below 1.3 across all configurations, implying that focal length inaccuracies also affect results, albeit less drastically than PPA deviations.

Interestingly, the Gruber123Noise, Gruber456Noise, and Gruber789Noise cases all result in zero F-test values across the board. This indicates that the system either fully absorbs or disregards these synthetic noise scenarios during Phase IB, suggesting a high level of robustness (or potentially non-effectiveness) of these specific perturbations under the chosen occlusion strategy.

In summary, the adjustment system in Phase IB remains robust to small external orientation perturbations, particularly in translation. However, rotational and calibration noise, especially in the principal point, significantly impairs the adjustment process, regardless of whether the imaging geometry is nadir or oblique. These findings reinforce the importance of precise internal calibration in 3D reconstruction tasks using occlusion-based synthetic adjustment methods.

4.3 Results of Phase II: Synthetic Computer Vision Based Object Point Detection

The results from Phase II: Synthetic Computer Vision Based Object Point Detection comprise a total of five image sets (see Section 3.4). These sets were rendered using Blender (see Section 2.17), applying the BLOSM [Prochitecture, 2025] and Camera Array Tool [ToppiNappi, 2025] add-ons, with Google 3D Map Tiles, OSM Water, and Miramap 2024 camera parameters (see Section 2.17). Each set includes both nadir and oblique imagery. This section first provides a general overview of the rendered image sets, followed by a tabulated summary of test cases (Table 10). The adjustment analysis is discussed in Section 5.3.

	Rendered Image Set	Object Point Detection	Object Point Matching	Adjustment
Nadir / Oblique	Normal	DISK	LightGlue / COLMAP	Bundle
		SIFT		
	Bright	DISK		
		SIFT		
	Shade	DISK		
		SIFT		
	Snow / Rain	DISK		
		SIFT		
	Clouds	DISK		
		SIFT		

Table 10: Test Cases Phase II

Figure 40 visualizes the image overlap for the nadir set. According to the HTW, a minimum overlap of three images per point is required. The results indicate insufficient coverage near the image edges, typically mitigated by applying a spatial buffer in real campaigns. While HTW does not define overlap standards for oblique views, a minimum of three is used here for consistency. Overlap is highest in the left- and right-facing cameras; edge regions again suffer from lower coverage.

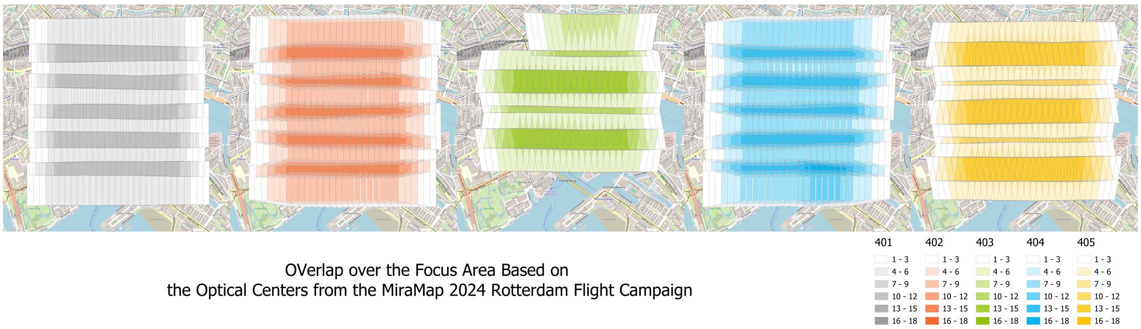


Figure 40: Nadir Image Overlap Map

Figure 41 shows that several images—especially around water bodies—contain too few detected object points to meet the HTW requirement of at least 16 object points per image.

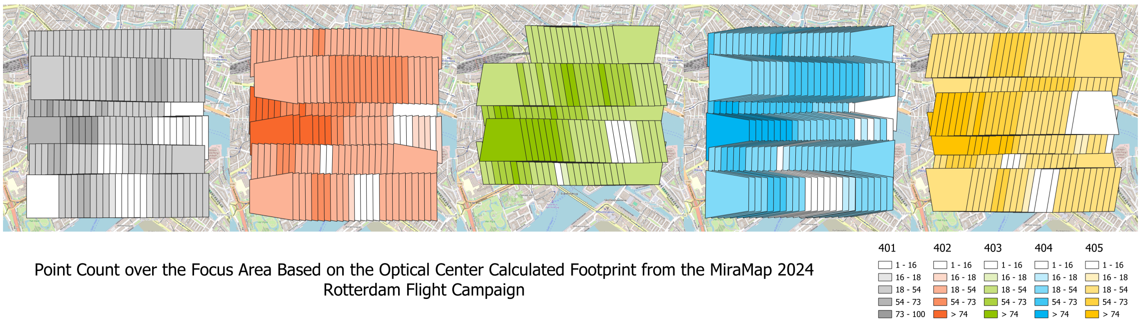
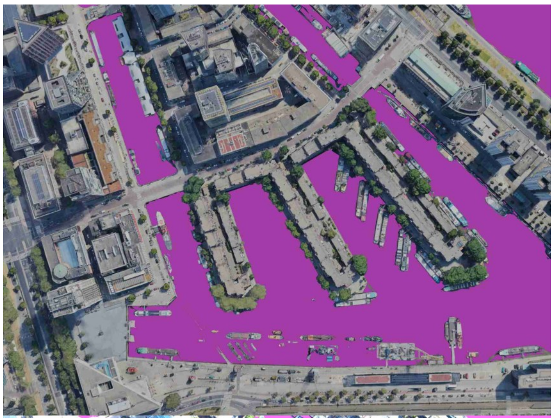
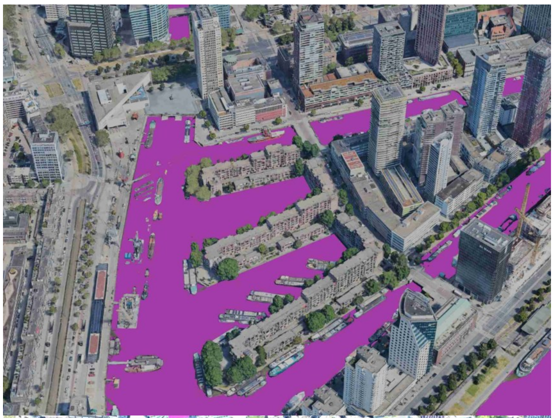


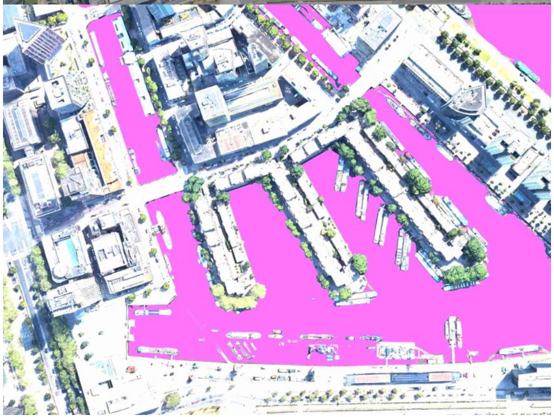
Figure 41: Object Point Count per Nadir Image



Rendered Image Nadir
401_0029_00121651
Effect “Normal”

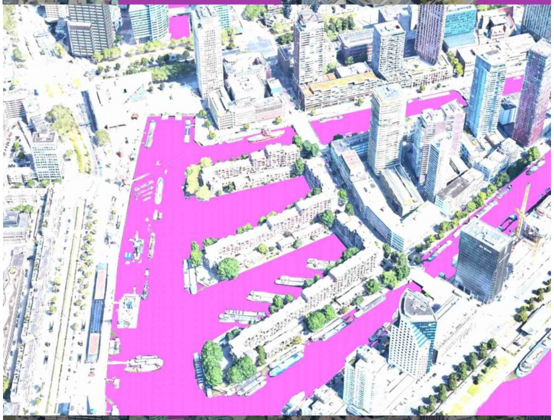
Effect “Normal”
405_0031_00113511
Rendered Image Oblique

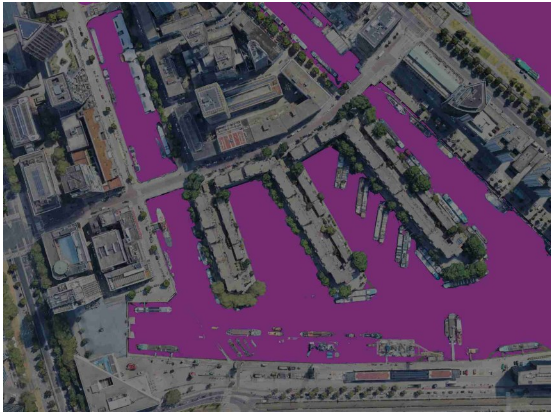




Rendered Image Nadir
401_0029_00121651
Effect “Bright”

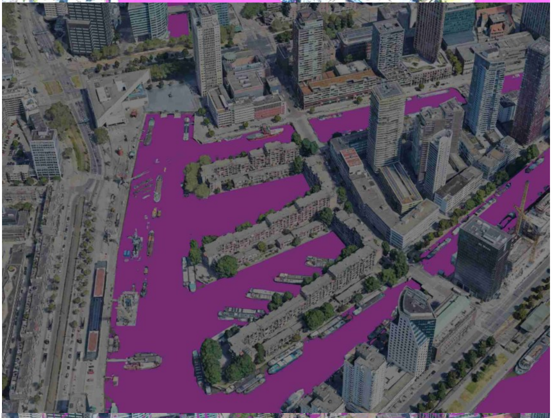
Effect “Bright”
405_0031_00113511
Rendered Image Oblique

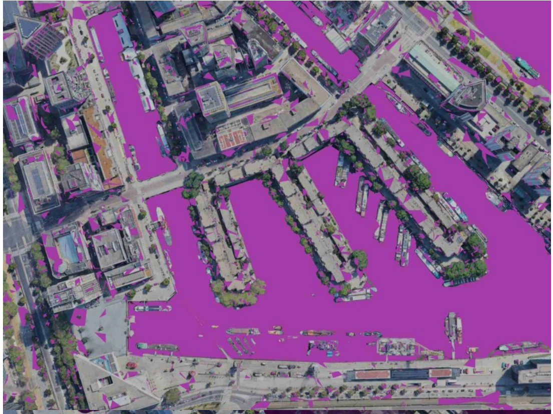




Rendered Image Nadir
401_0029_00121651
Effect “Shade”

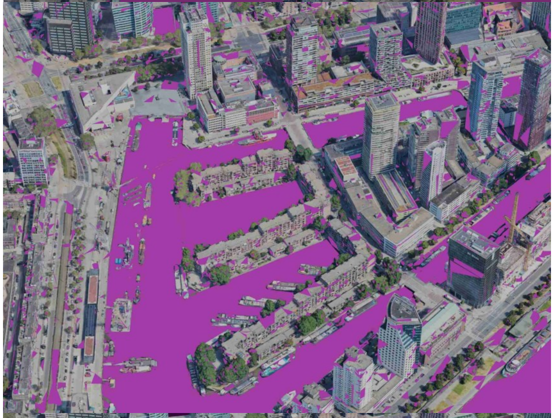
Effect “Shade”
405_0031_00113511
Rendered Image Oblique

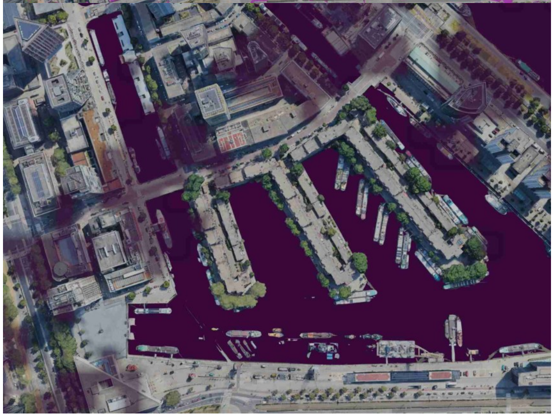




Rendered Image Nadir
401_0029_00121651
Effect “Snow/Rain”

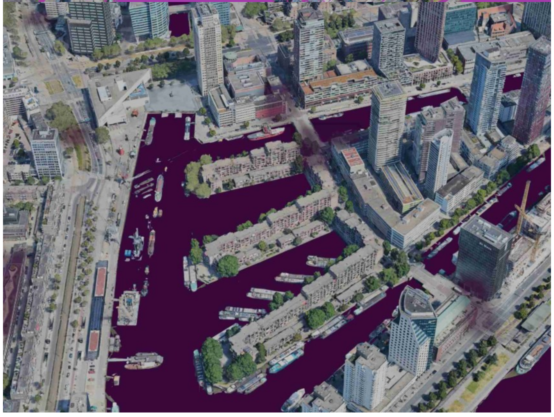
Effect “Snow/Rain”
405_0031_00113511
Rendered Image Oblique





Rendered Image Nadir
401_0029_00121651
Effect “Clouds”

Effect “Clouds”
405_0031_00113511
Rendered Image Oblique



Each image set was rendered under different environmental effects:

1. Normal Image Set: No exogenous effects, suitable for ground truth comparison.
2. Bright Image Set: Overexposed at 1.3× light intensity, less ideal for mapping.
3. Shadow Image Set: Underexposed at 0.7× light intensity, resulting in low contrast.
4. Snow/Rain Image Set: Includes puddles on 5% of upward-facing surfaces, simulating realistic precipitation.
5. Cloud Image Set: Semi-transparent cloud layers (alpha = 0.5), introducing light occlusion.

Each nadir dataset contains 146 images. The oblique datasets contain 552 images across four camera directions, using the same environmental conditions for consistency.

The adjustment process was conducted after rendering, detection, and matching, with results summarized in Table 15. Across both nadir and oblique image sets, SIFT-based configurations consistently achieved successful detection, matching, and adjustment. For nadir imagery, F-test values remained below 1.0, indicating acceptable reconstruction quality. The cloud condition yielded the highest F-test (0.447), likely due to partial occlusions, while bright and shade conditions reduced mapping usability because of diminished contrast. Snow and rain further lowered feature interpretability through occlusion of fine details. Oblique sets exhibited a similar trend in SIFT performance, although F-test values were generally higher—peaking at 1.103 in the cloud scenario—reflecting the added complexity introduced by perspective distortion and occlusions.

In contrast, DISK failed across all conditions for both nadir and oblique datasets. While feature detection succeeded, matching did not proceed reliably. For nadir scenes, some within-flight line matches were formed, but inter-flight line rotational differences undermined consistency, preventing global reconstruction. In oblique scenes, each camera system could reconstruct internally; however, weak connectivity between different view angles led to failure in global adjustment. These findings highlight SIFT's robustness across varying environmental and geometric conditions, while also underscoring DISK's limitations in cross-view feature matching under these challenging scenarios.

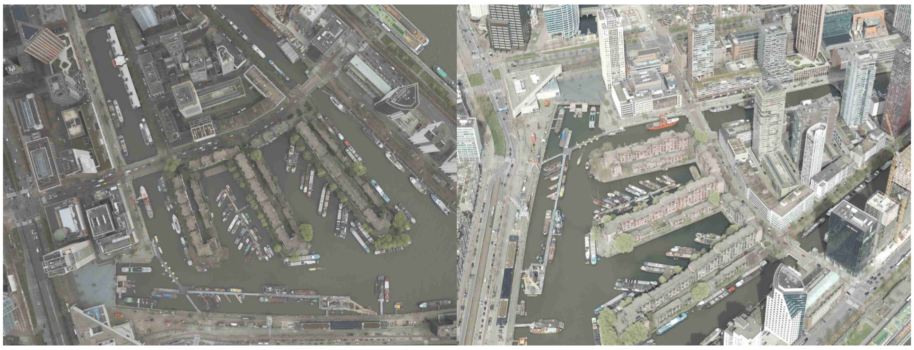
	Number of Images	Detection (S/F)/ Matching (S/F)/ Adjustment Phase (S/F) Nadir	F-Test Results Nadir	Detection (S/F)/ Matching (S/F)/ Adjustment Phase (S/F) Oblique	F-Test Results Oblique
DISK Normal	146 Nadir Images or 552 Oblique Images	S/F/-	-	S/F/-	-
SIFT Normal		S/S/S	0.429	S/S/S	0.830
DISK Bright		S/F/-	-	S/F/-	-
SIFT Bright		S/S/S	0.432	S/S/S	0.835
DISK Shade		S/F/-	-	S/F/-	-
SIFT Shade		S/S/S	0.434	S/S/S	0.838
DISK Snow / Rain		S/F/-	-	S/F/-	-
SIFT Snow / Rain		S/S/S	0.435	S/S/S	0.839
DISK Clouds		S/F/-	-	S/F/-	-
SIFT Clouds		S/S/S	0.447	S/S/S	1.103

Table 11: Results Phase II (S refers to a success and F refers to a failed step)

4.4 Results of Phase III: Applying Learned Lessons to Real Life Data

In Phase III, the lessons learned from earlier experiments were applied to real-world aerial imagery. These lessons are extracted from the intermediary conclusions. This phase evaluated the practical effectiveness of the refined workflows using actual image sets from the Miramap 2024 Image Campaign, provided by the Municipality of Rotterdam. The goal was to test the robustness of the detection, matching, and adjustment pipeline on real, unconstrained data.

These datasets were used to evaluate the performance of two object point detection methods DISK and SIFT combined with LightGlue/COLMAP for object point matching and a Bundle adjustment (see Section 2.16) backend for refinement.



(a) Example of Actual Nadir Image

(b) Example of Actual Oblique Image

Figure 42: Sample input images from the Miramap 2024 campaign used for real-world evaluation.

The following insights proved essential for improving the workflow's performance on real data:

- Rotate images in memory before processing to standardize orientation (Due to DISK not being rotationally invariant)
- Use realistic image content during training and testing (synthetic or idealized images were not representative)
- Increase the standard deviations in the camera model, especially for oblique imagery, to account for greater geometric variability

These adjustments significantly improved stability across all steps (object point detection, matching, and bundle adjustment) for both nadir and oblique datasets.

Evaluation Results Table 16 shows the success/failure status of each pipeline step (Detection, Matching, Adjustment), along with the F-test values, which reflect the relative quality of the geometric adjustment.

	Number of Images	Detection (S/F) Matching (S/F) Adjustment Phase (S/F) Nadir	F-test Results Nadir	Detection (S/F) Matching (S/F) Adjustment Phase (S/F) Oblique	F-test Results Oblique
DISK Real Data	146 Nadir Images or 552 Oblique Images	S/S/S	0.450	S/S/S	0.659
SIFT Real Data		S/S/S	0.460	S/S/S	0.830

Table 12: Pipeline Success and F-test Results for DISK and SIFT on Real Nadir and Oblique Imagery. (S refers to a success, and F refers to a failed step)

Both DISK and SIFT performed reliably on real-world data, achieving successful detection, matching, and adjustment across both nadir and oblique views. Interestingly, SIFT showed slightly better F-test results (see Section 2.23) on oblique imagery, while performance was comparable for nadir images. These results confirm that the adaptations made in Phase III significantly improved the pipeline's robustness and real-world applicability.

5 Analysis

Chapter 5 delves into the analysis of each of the workflow phases. The structure is the same as for Chapter 3 and Chapter 4 in that each phase has its dedicated subsection. This is the final chapter to show results. Chapter 6 delves into the discussion as to which parts could have been successful and the limitations of the research as it was currently implemented.

5.1 Analysis of Phase IA: Non-Obstructed Synthetic Adjustment

This section analyzes Phase IA, which represents the Non-Obstructed Synthetic Adjustment introduced in Section 3.2, with results presented in Section 4.1. It is the most basic phase described in Section 3.1 and is used to partially answer the following two sub-questions:

- What is the mathematical pipeline for aerial adjustment theory?
- What hyperparameters are present in aerial adjustment, and what are their effects when changed in synthetic tests?

As noted earlier (see Section 5.1), the F-test results (see Section 2.23) summarized in Table 13, provide an initial overview of the test performance. Rejected F-tests and failed cases are marked in Red. A failed test constitutes a CaseType in which all simulations became singular. The F-test critical threshold is based on the number of observations ($\sim$ 6.000) per simulation, leading to a roughly converged critical value (see Section 2.23) of 1.

The following per CaseType metrics are analyzed: the average W-test; the average X, Y, and Z differences (see Section 2.5); the average O, P, and K differences (also in Section 2.5); and the W-test (see Section 2.22) distribution of the u and v image plane coordinates resulting from the observation calculation (see Section 2.12). These are reported across all test cases for both nadir and oblique imagery.

	CaseType	Avg. F-test 1st Phase Nadir	Avg. F-test 2nd Phase Nadir	Avg F-test 1st Phase Oblique	Avg. F-test 2nd Phase Oblique
Phase IA: Non-Obstructed Synthetic Adjustment	NoneNoise	0.000	0.000	0.000	0.000
	XNoise	0.010	0.009	0.008	0.008
	YNoise	0.010	0.010	0.010	0.008
	ZNoise	0.011	0.009	0.005	0.005
	OmegaNoise	2.407	2.356	2.117	2.270
	PhiNoise	2.107	1.764	5.514	2.474
	KappaNoise	1.053	1.339	1.080	0.845
	PPAxNoise	282.099	280.432	73.582	74.639
	PPAyNoise	308.276	306.053	75.890	72.480
	FocalNoise	1.278	1.239	0.676	0.650
	Gruber123Noise	0.000	0.000	0.000	Failed
	Gruber456Noise	0.000	0.000	0.000	Failed
	Gruber789Noise	0.000	0.000	0.000	Failed

Table 13: Average F-Test Results for Various CaseTypes in Phase IA (Nadir and Oblique, First and Second Phases)

The ability to average the F-tests can only be done from roughly normalized distributions across all Monte Carlo-simulated CaseTypes. All CaseTypes, except NoneNoise, are simulated 100 times. These values have been averaged, which is only a valid procedure because the number of observations and conditions remains constant within each CaseType. To reduce visual clutter, only one F-test distribution histogram (OmegaNoise) is shown in Figure 43. Due to the changed number of observations, no F-test comparison between Phase IA and IB can be made.

The adjustment results across all adjustment steps and test sets as presented in Table 13 and **??**, **??**, and **??** consistently show that the "perfect" or baseline case achieves ideal adjustment, with F-test values at zero (see Section 2.23 and Section 2.22), confirming system stability under NoneNoise conditions. Across both nadir and oblique configurations, whether or not control points are included, the CaseTypes OmegaNoise, PhiNoise, PPAxNoise, and PPAyNoise consistently fail the F-test (see Section 2.23), indicating singularities in the B-model matrix and thus unsuccessful adjustments. While KappaNoise and FocalNoise approach the

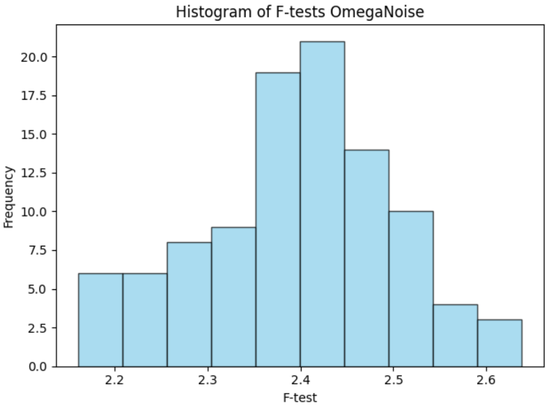


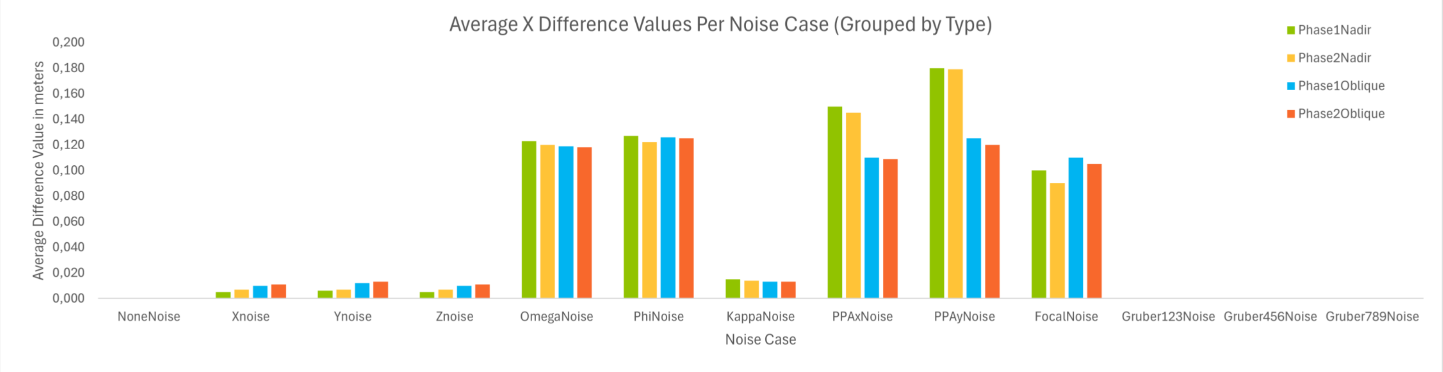
Figure 43: Histogram Distribution of OmegaNoise for Phase IA

critical rejection threshold; they often remain borderline and may not consistently fail with larger sample sizes.

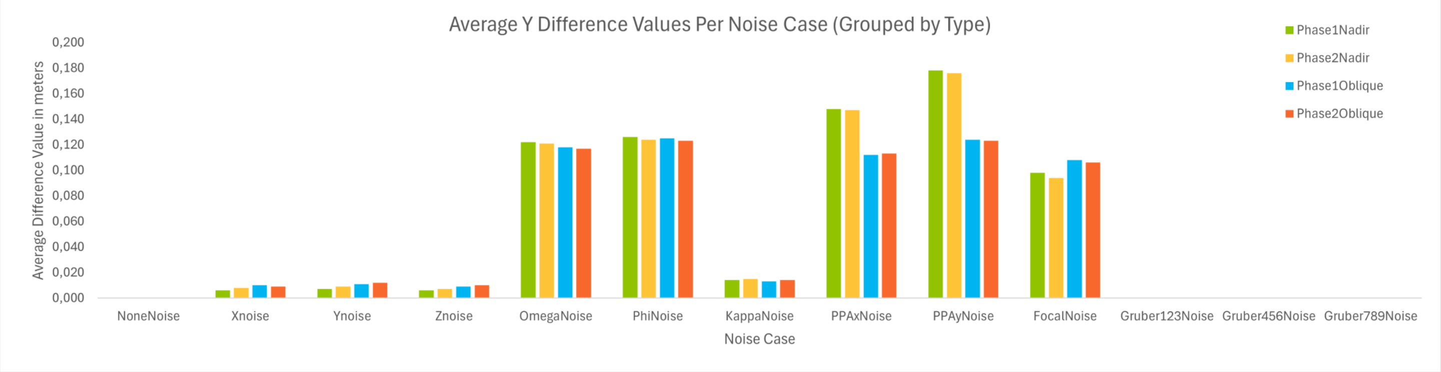
The similarity in F-test magnitudes between Omega and Phi, as well as between PPAx and PPAy, indicates that angular and principal point perturbations affect the adjustment process symmetrically and consistently across phases. Notably, non-obstructed synthetic oblique imagery appears more resilient to noise—especially in the PPAx and PPAy cases—likely due to an increased number of visible points improving system redundancy. However, all Von Gruber square-related noise cases (Gruber123Noise, Gruber456Noise, Gruber789Noise) consistently fail in the second-phase oblique adjustment with control points (see Table 13), due to singular B-model matrices (see Section 2.20), although these same cases succeeded in Phase IB nadir adjustments.

The subsequent analysis focuses on the spatial distribution of rejected W-tests (see Section 2.22) across the four result categories defined in Phase IA. As previously noted, the critical threshold for the W-test is 3.29; any test statistic exceeding this value indicates a statistically significant deviation in the residuals of the observation model.

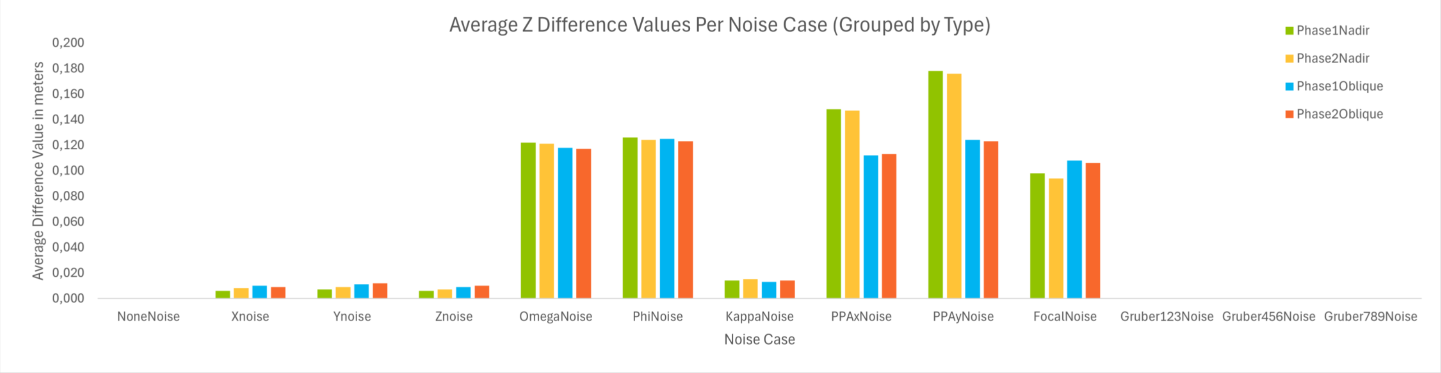
Rejected W-tests are visualized by projecting them directly onto the image plane, enabling a spatial assessment of where deviations cluster. This representation reveals patterns specific to different noise types and supports the evaluation of how individual perturbations impact image coordinate residuals. By analyzing these spatial distributions, it becomes possible to pinpoint areas or image regions more sensitive to error propagation, especially regarding angular and principal point noise.



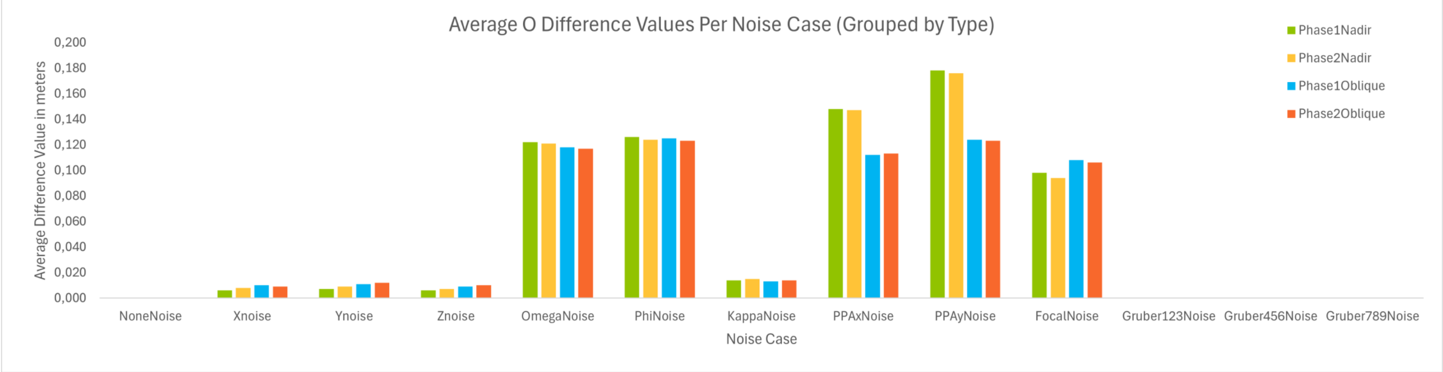
Average X Difference Values Per Noise Case (Grouped by Type)



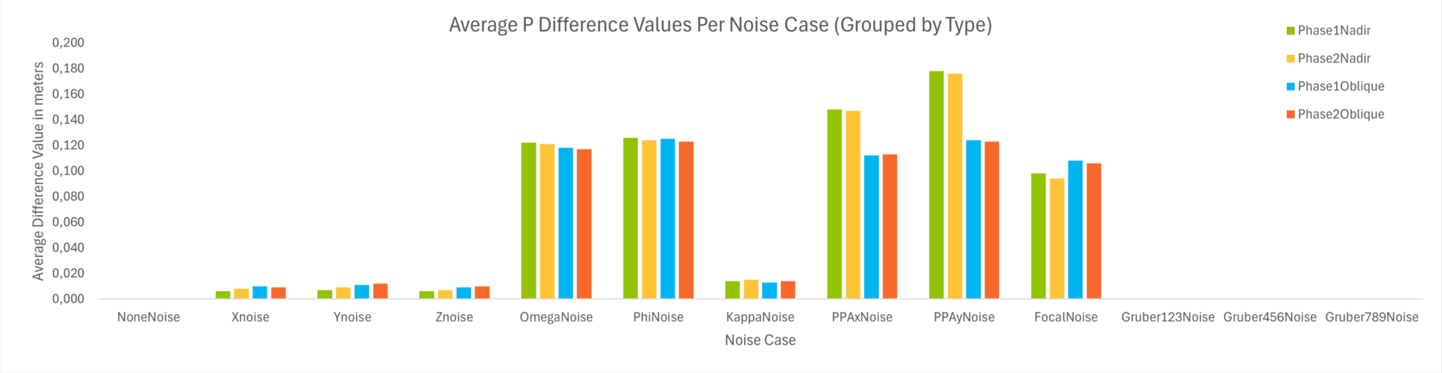
Average Y Difference Values Per Noise Case (Grouped by Type)



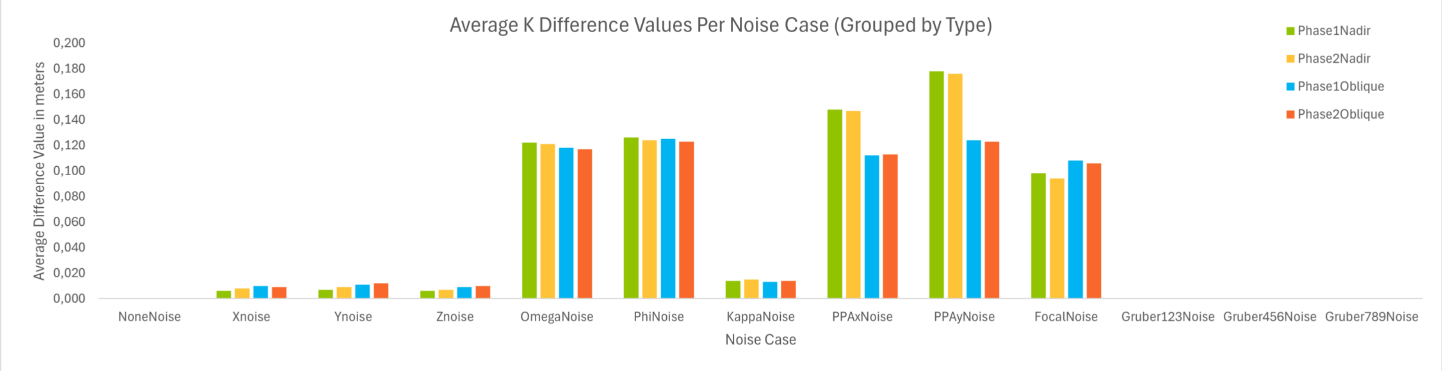
Average Z Difference Values Per Noise Case (Grouped by Type)



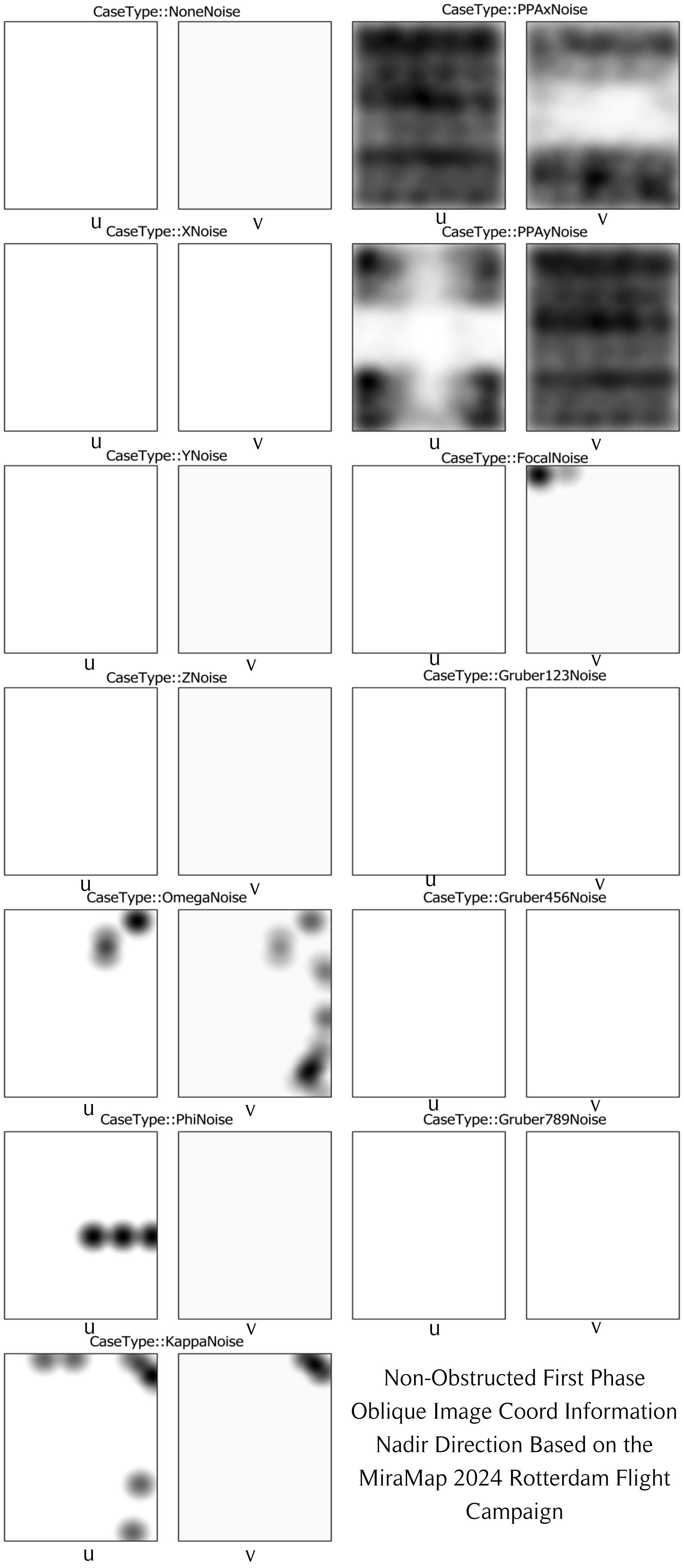
Average O Difference Values Per Noise Case (Grouped by Type)



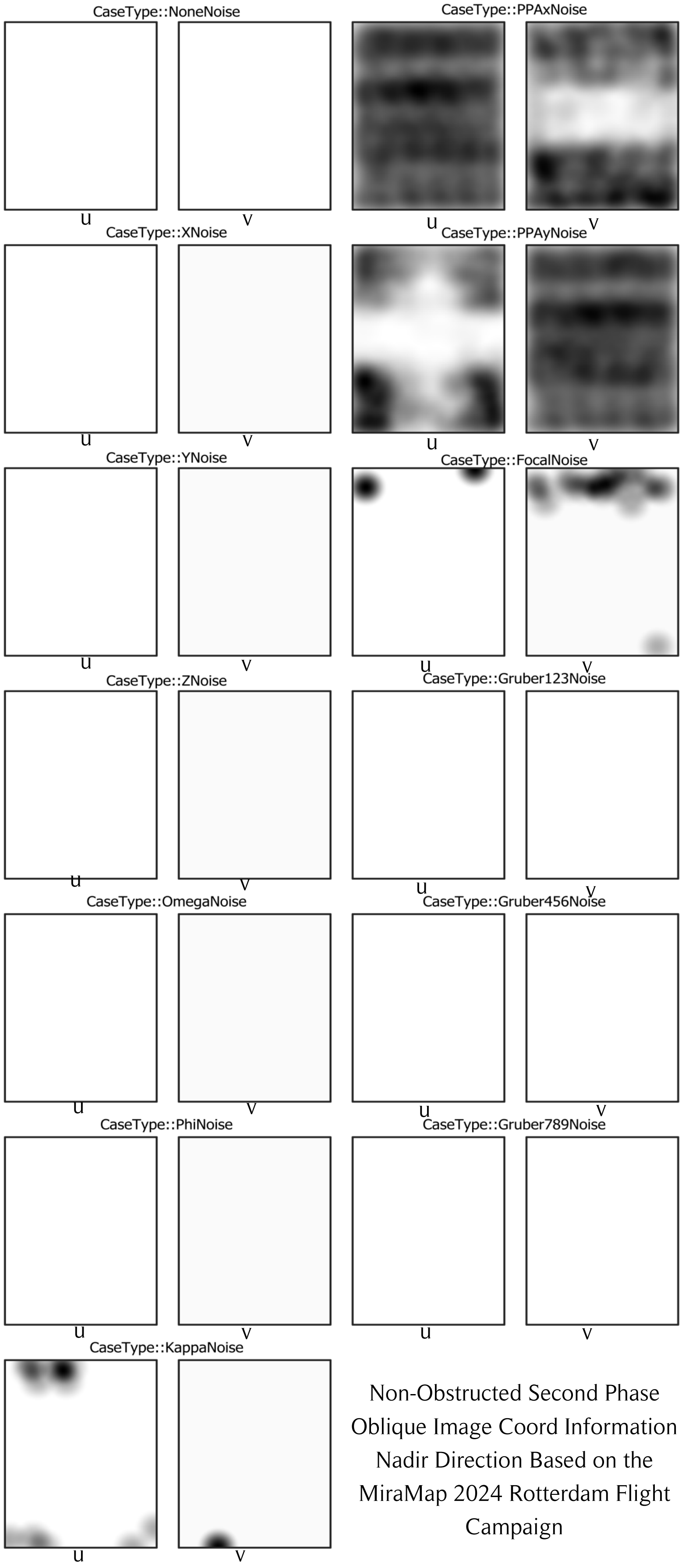
Average P Difference Values Per Noise Case (Grouped by Type)



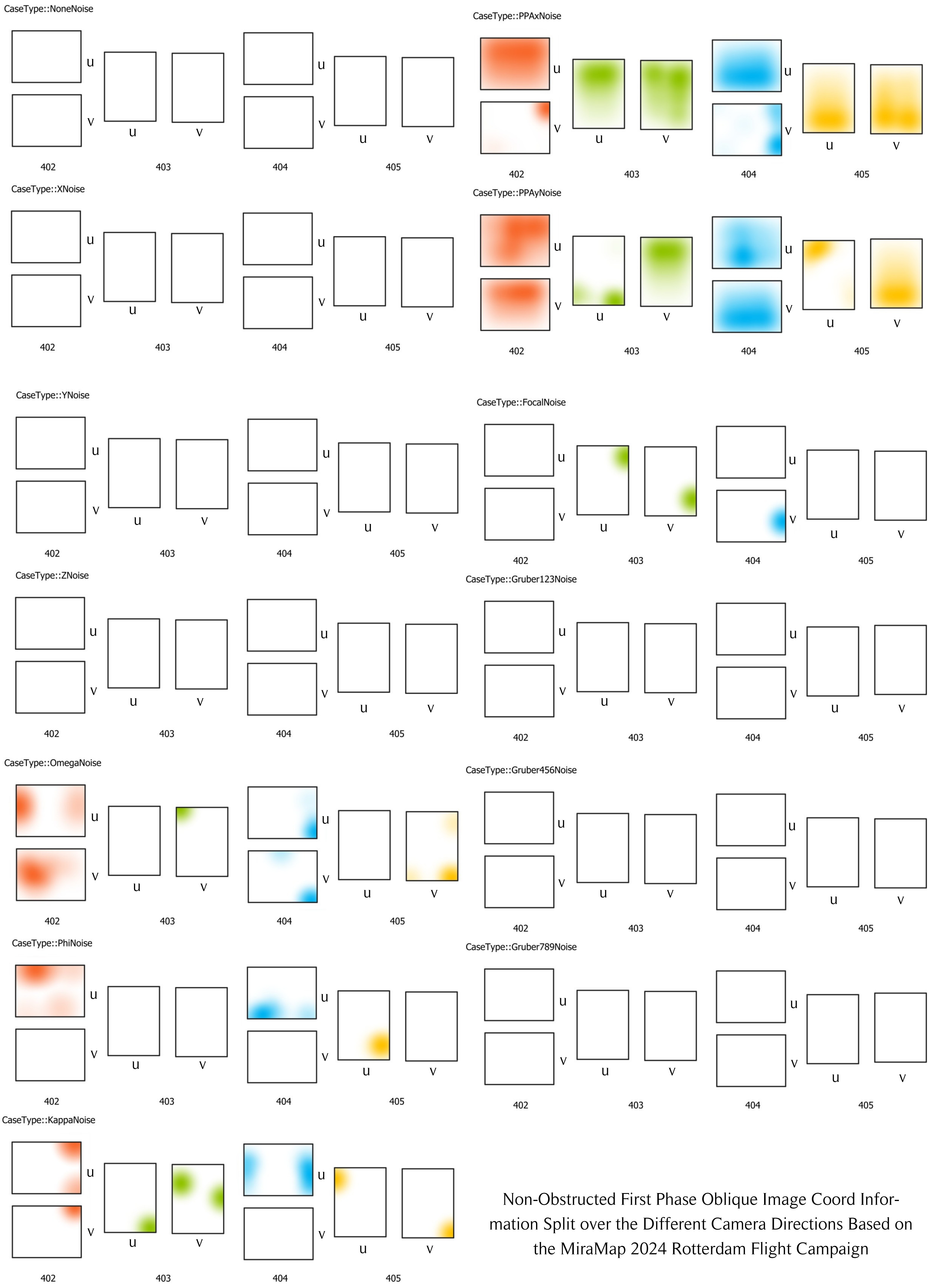
Average K Difference Values Per Noise Case (Grouped by Type)



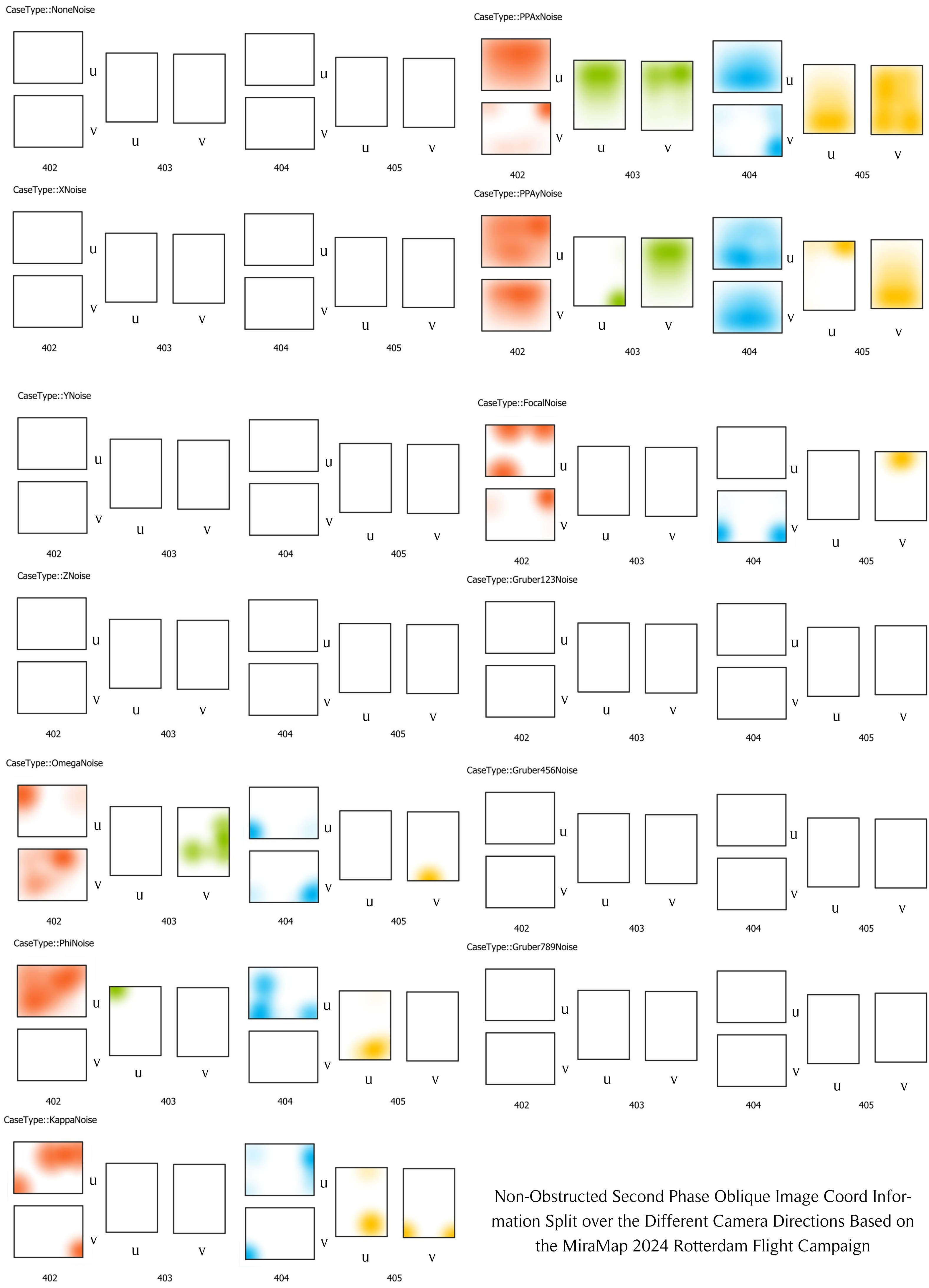
Non-Obstructed First Phase Oblique Image Coord Information Nadir Direction Based on the MiraMap 2024 Rotterdam Flight Campaign



Non-Obstructed Second Phase Oblique Image Coord Information Nadir Direction Based on the MiraMap 2024 Rotterdam Flight Campaign



Non-Obstructed First Phase Oblique Image Coord Information Split over the Different Camera Directions Based on the MiraMap 2024 Rotterdam Flight Campaign



Non-Obstructed Second Phase Oblique Image Coord Information Split over the Different Camera Directions Based on the MiraMap 2024 Rotterdam Flight Campaign

First and second Phase of Nadir Adjustment

The W-test results during the first (left) and second (right) phases of nadir adjustment reveal that different types of noise influence image coordinates in distinct ways. For NoneNoise, XNoise, YNoise, ZNoise, Gruber123Noise, Gruber456Noise, and Gruber789Noise, no rejections were observed—suggesting that these perturbations do not significantly impact the image observations.

However, OmegaNoise shows a clear pattern of influence, particularly on the u coordinates in Von Gruber squares 4, 5, and 6, where image observations remain mostly unaffected. In contrast, the v coordinates exhibit more widespread rejections, especially on the right side of the image.

PhiNoise selectively affects the u coordinates, leaving the v coordinates unchanged. KappaNoise introduces mild rejections in both u and v directions, particularly around the image periphery, reflecting the rotational nature of the kappa angle's influence.

PPAxNoise results in heavy rejection in u coordinates across nearly the entire image, whereas v coordinate rejections are more limited, predominantly in Von Gruber squares 4, 5, and 6. Conversely, PPAyNoise causes widespread rejection in v coordinates with a reversed pattern.

Finally, FocalNoise induces increasingly significant rejection toward the image edges in both u and v directions. This trend is attributed to the magnified distortion effects caused by focal length errors, which are more pronounced farther from the image center.

First Phase of Oblique Adjustment

In the first phase of oblique adjustment, W-test results highlight how noise sensitivity varies with camera orientation and internal geometry. NoneNoise, XNoise, YNoise, ZNoise, FocalNoise, and all three Gruber noise variants show no rejections, suggesting stable observation quality under these conditions.

By contrast, OmegaNoise and PhiNoise introduce significant rejections along the u or v axes. These effects stem from the alignment of the omega and phi angles with the horizontal and vertical axes of the image plane, respectively. The rotational influence of these angles causes specific image borders (e.g., top/bottom or sides) to exhibit increased sensitivity.

KappaNoise results in mild but consistent rejections at the image corners, reflecting its impact on overall image plane rotation. PPAxNoise produces widespread rejection in u coordinates, especially on the left and right image edges. Gaps in v coordinate observations—particularly in cameras 402 and 405—suggest reduced lateral observation robustness.

PPAyNoise primarily affects the v coordinates, especially in forward and backward viewing directions. In some cases (e.g., cameras 402 and 404), even the u coordinates show elevated rejection levels. These patterns are likely linked to reduced point density at higher elevations within the 3D BAG data, which diminishes image observation redundancy and robustness depending on viewing geometry.

Second Phase of Oblique Adjustment

During the second phase of oblique adjustment, the W-test results show that NoneNoise, XNoise, YNoise, ZNoise, KappaNoise, FocalNoise, and the three Gruber noise categories do not lead to any test rejections—indicating stable and resilient observations under these noise conditions.

In contrast, OmegaNoise and PhiNoise generate clear linear rejection patterns in either the u or v coordinates, depending on camera rotation. These patterns are due to each angle's association with specific image axes, which shift depending on camera orientation making certain regions (e.g., sides or top/bottom) more vulnerable to perturbations.

PPAxNoise triggers extensive rejection in u coordinates across most images, particularly along the lateral edges. Reduced point density away from the camera's central detection direction (at higher altitudes in the 3D BAG data) further compounds this vulnerability. In v coordinates, rejection is more uniform, but gaps appear in image centers for some cameras (notably 402 and 405).

PPAyNoise causes widespread v coordinate rejections, predominantly in the vertical viewing directions. Cameras 402 and 404 exhibit more u coordinate rejections compared to others, indicating that PPAyNoise impacts vertical camera geometries more strongly—again linked to point density distribution and 3D structural influences within the dataset.

5.2 Analysis of Phase IB: Depth Map Based Occlusion for Synthetic Adjustment

This section presents the analysis of Phase IB, which implements Depth Map-Based Occlusion for Synthetic Adjustment, as introduced in Section 3.3 and whose results are detailed in Section 4.2. Like Phase IA (Section 5.1), this phase uses the same set of CaseTypes, with the addition of a depth map to simulate occlusion. It builds upon Phase IA and is used to answer the following two sub-questions:

- What are the main differences between nadir and oblique image adjustment?
- How do the different parameters of oblique imagery influence the errors present in the final adjusted images?

As described in Section 4.2, the performance of Phase IB was initially evaluated using F-tests (Section 2.23), with the results summarized in Table 14. Rejected F-tests and failed simulations are highlighted in Red. A failed simulation is defined as a CaseType where all runs result in singular solutions. The F-test threshold is based on approximately 9.000 observations per simulation, producing a stable critical value near 1 (see Section 2.23).

To build upon the results from Phase IB, the analysis includes the average W-test statistic, the mean differences in external camera position (X, Y, Z) and orientation (O, P, K) parameters (see Section 2.5), and the spatial distribution of rejected W-tests over the image plane. These measures offer insight into how well the synthetic adjustment is performed under different perturbation scenarios for both nadir and oblique imagery.

As with Phase IA, each CaseType (except NoneNoise) was simulated 100 times using Monte Carlo simulations (see Section 2.24). Averaging of results is valid due to consistent numbers of observations and constraints across all simulations within each CaseType. To simplify visual interpretation, only a single histogram is presented for the OmegaNoise CaseType (Figure 44). Due to a difference in the number of observations ($\sim$ 6.000 for Phase IA and $\sim$ 9.000 for Phase IB), no comparison between the F-test results can be made between the two phases. To allow for F-test comparison, a setup needs to have roughly the same hyperparameters. Due to the inherent nature of the case setup, with the depth map testing leading to a reduced number of observations per camera center, which can only be counteracted with more points, leads to incomparable systems.

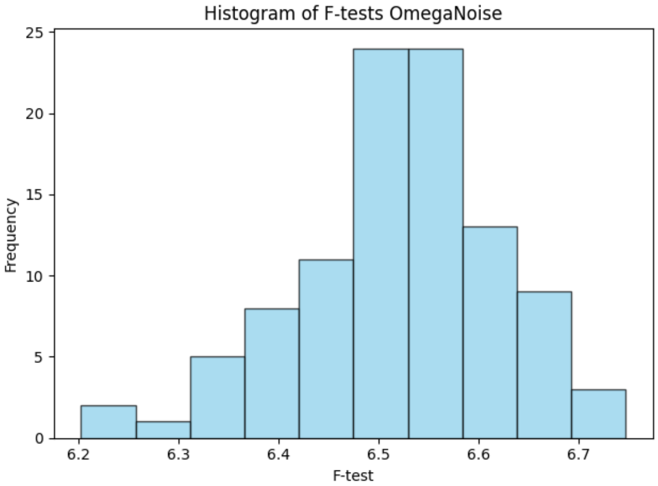


Figure 44: Histogram Distribution of OmegaNoise for Phase IB

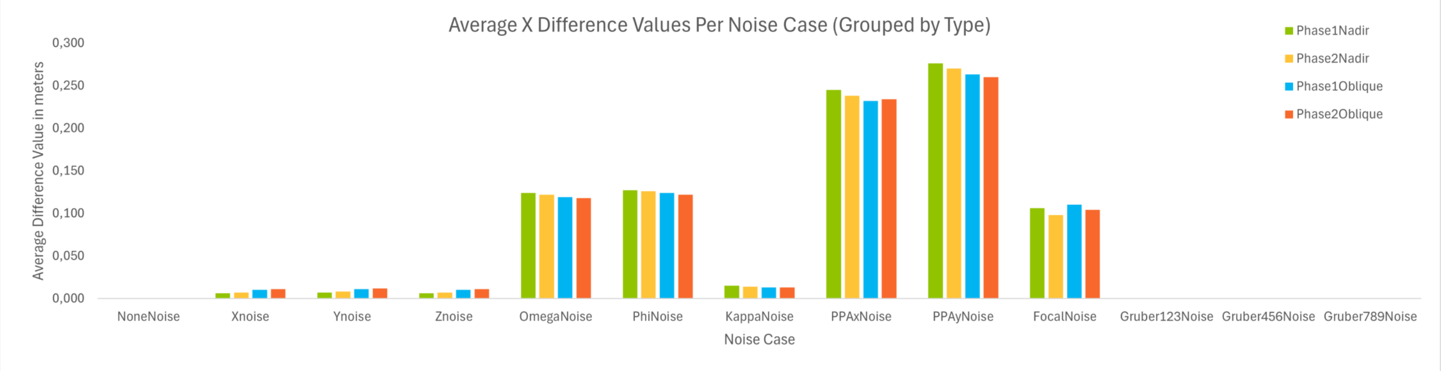
Table 14 provides an overview of the average F-test values across both phases (first and second) for nadir and oblique setups. Several trends are evident: rotational and principal point noise (PPAx, PPAy) consistently lead to F-test rejections, indicating unstable adjustment under these perturbations. In contrast, focal length noise (FocalNoise) only causes F-test failure in the first nadir phase, likely due to the absence of control points. Oblique imagery, having more observations, is more sensitive overall, especially in early phases, though its second-phase performance improves, particularly under focal length noise.

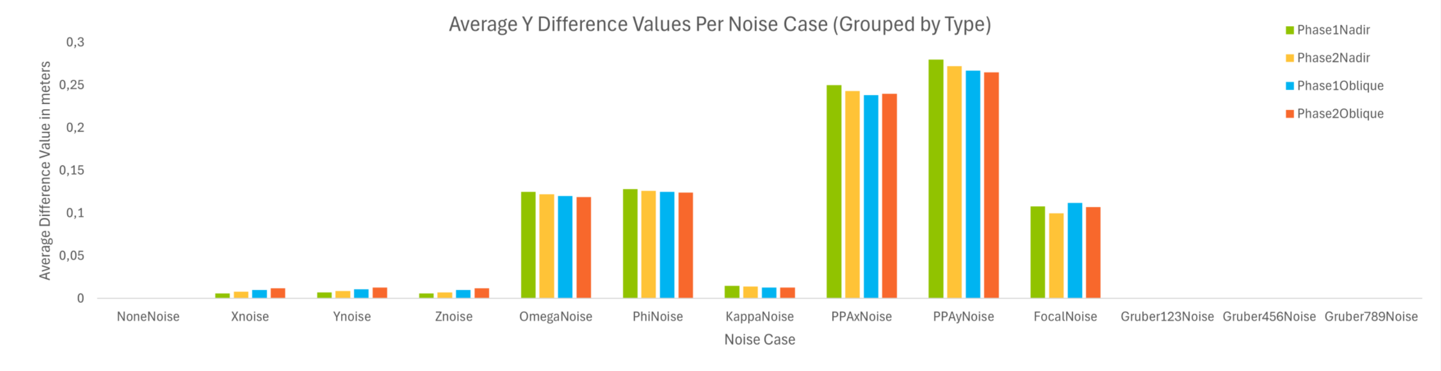
The provided bar charts further explore the effects of noise on external camera parameters. The position (X, Y, Z) and orientation (O, P, K) differences are generally more pronounced under rotational and principal point noise, consistent with F-test findings. These deviations highlight the types of perturbations most detrimental to adjustment accuracy in Phase IB.

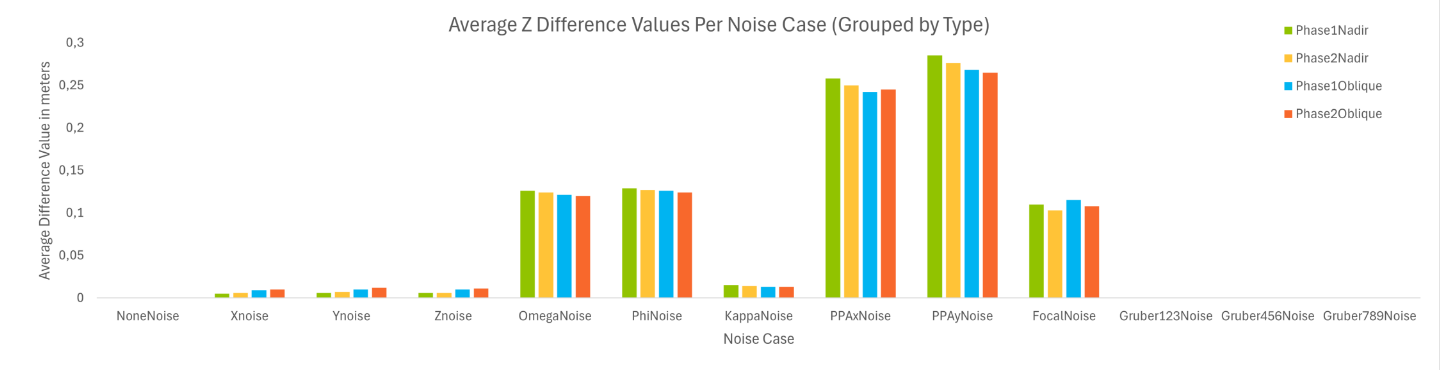
In addition to global metrics, local sensitivity is analyzed through the spatial distribution of rejected W-tests (Section 2.22), with a significance threshold set at 3.29. These are projected onto the image plane to identify regions of heightened instability or sensitivity to specific noise types. This visualization enables pattern recognition across the image space, revealing whether certain perturbations systematically affect particular regions of the imagery.

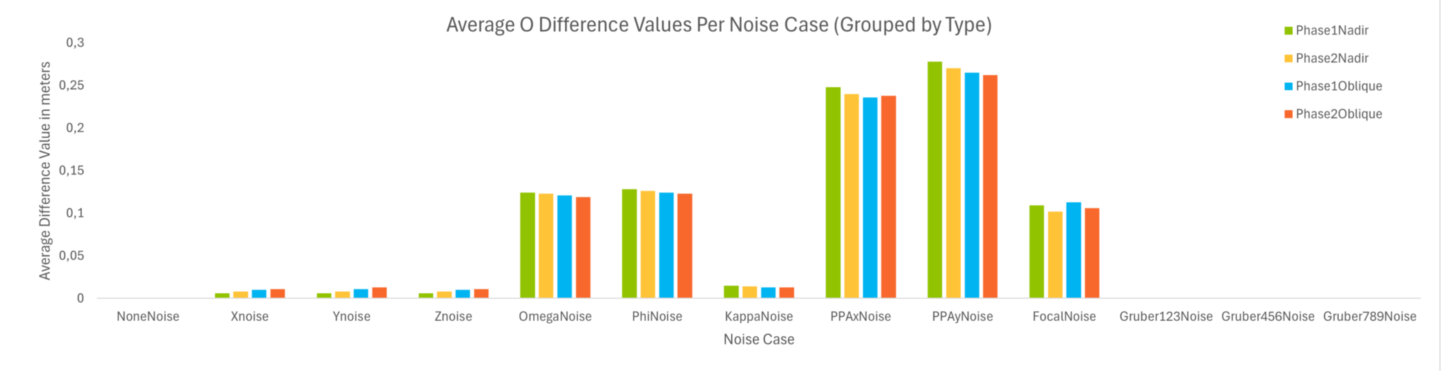
	CaseType	Avg. F-test 1st Phase Nadir	Avg. F-test 2nd Phase Nadir	Avg. F-test 1st Phase Oblique	Avg. F-test 2nd Phase Oblique
Phase IB: Depth Map **Based Occlusion for Synthetic Adjustment**	NoneNoise	0.000	0.000	0.000	0.000
	XNoise	0.034	0.033	0.011	0.015
	YNoise	0.044	0.039	0.015	0.013
	ZNoise	0.011	0.011	0.008	0.008
	OmegaNoise	6.513	9.113	2.744	3.064
	PhiNoise	6.665	6.220	3.732	3.421
	KappaNoise	1.310	1.735	1.119	1.229
	PPAxNoise	175.109	161.095	285.268	286.778
	PPAyNoise	366.599	341.614	296.492	297.105
	FocalNoise	1.216	0.992	0.627	0.572
	Gruber123Noise	0.000	0.000	0.000	0.000
	Gruber456Noise	0.000	0.000	0.000	0.000
	Gruber789Noise	0.000	0.000	0.000	0.000

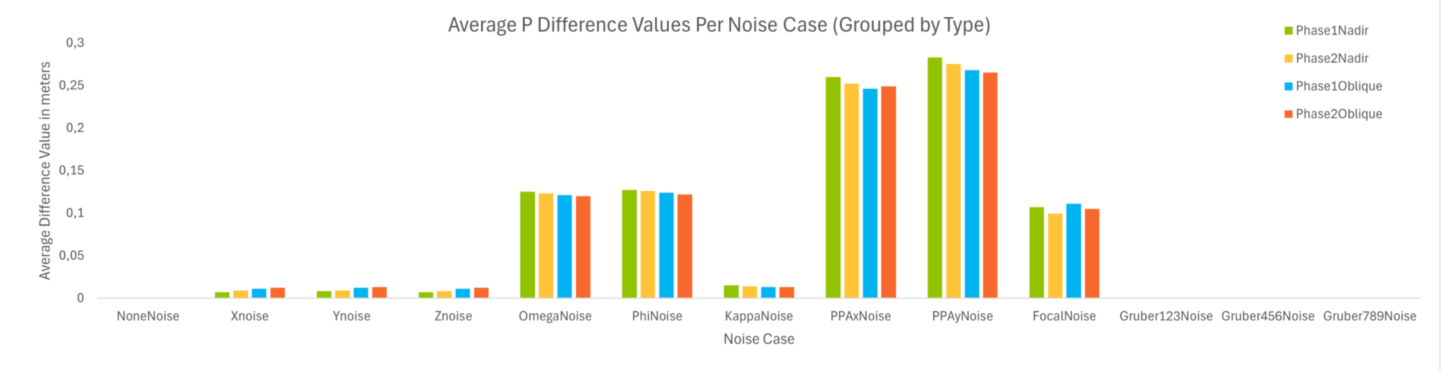
Table 14: Average F-Test Results for Various CaseTypes in Phase IB (Nadir and Oblique, First and Second Phases)

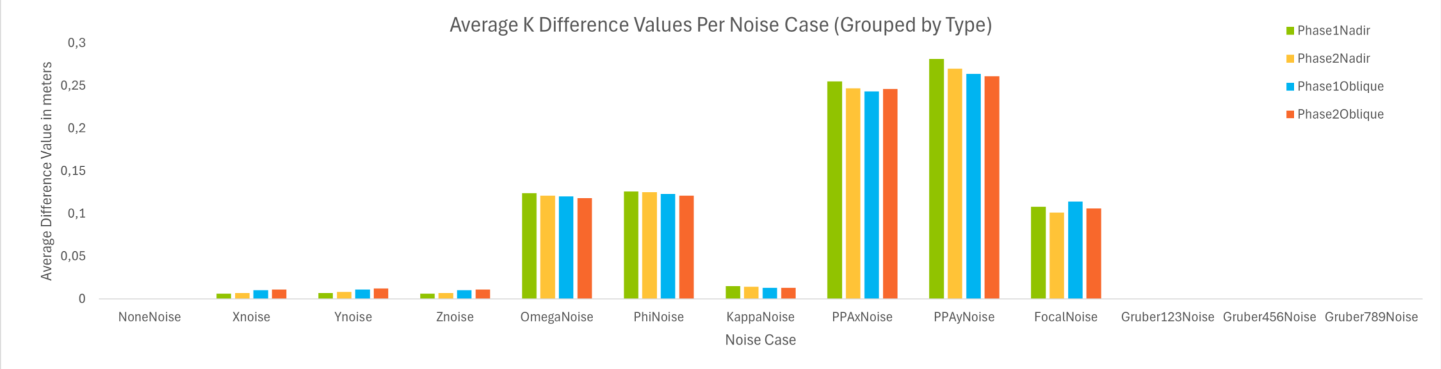

Average X Difference Values Per Noise Case (Grouped by Type)

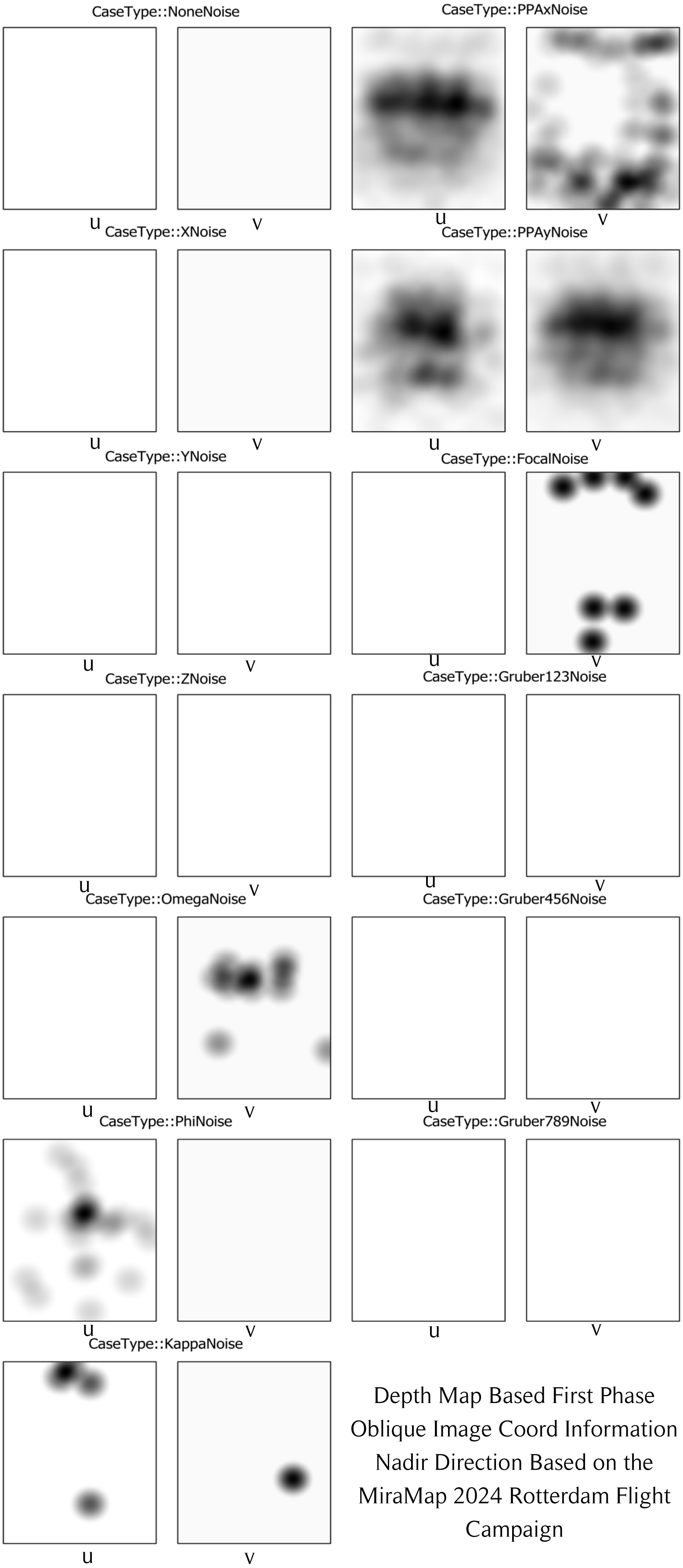

Average Y Difference Values Per Noise Case (Grouped by Type)


Average Z Difference Values Per Noise Case (Grouped by Type)

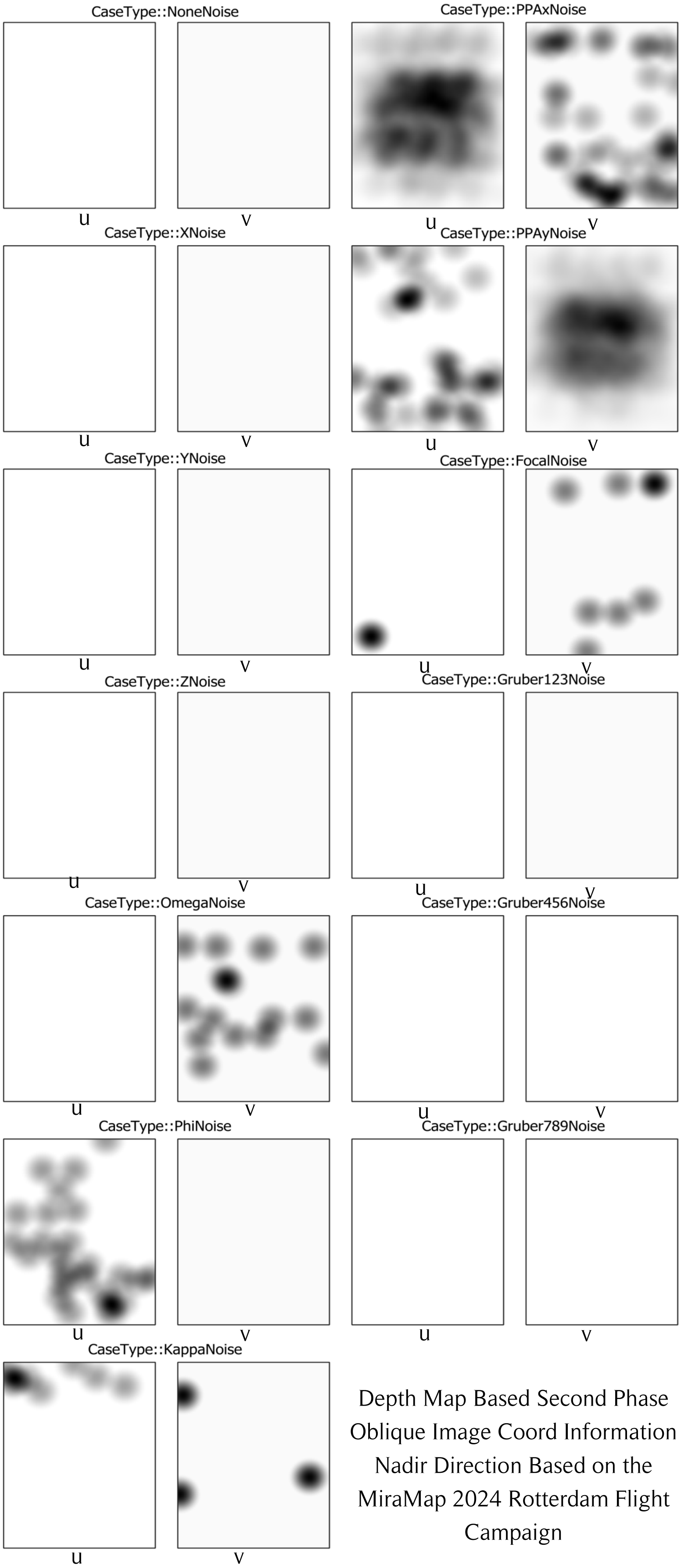

Average O Difference Values Per Noise Case (Grouped by Type)


Average P Difference Values Per Noise Case (Grouped by Type)

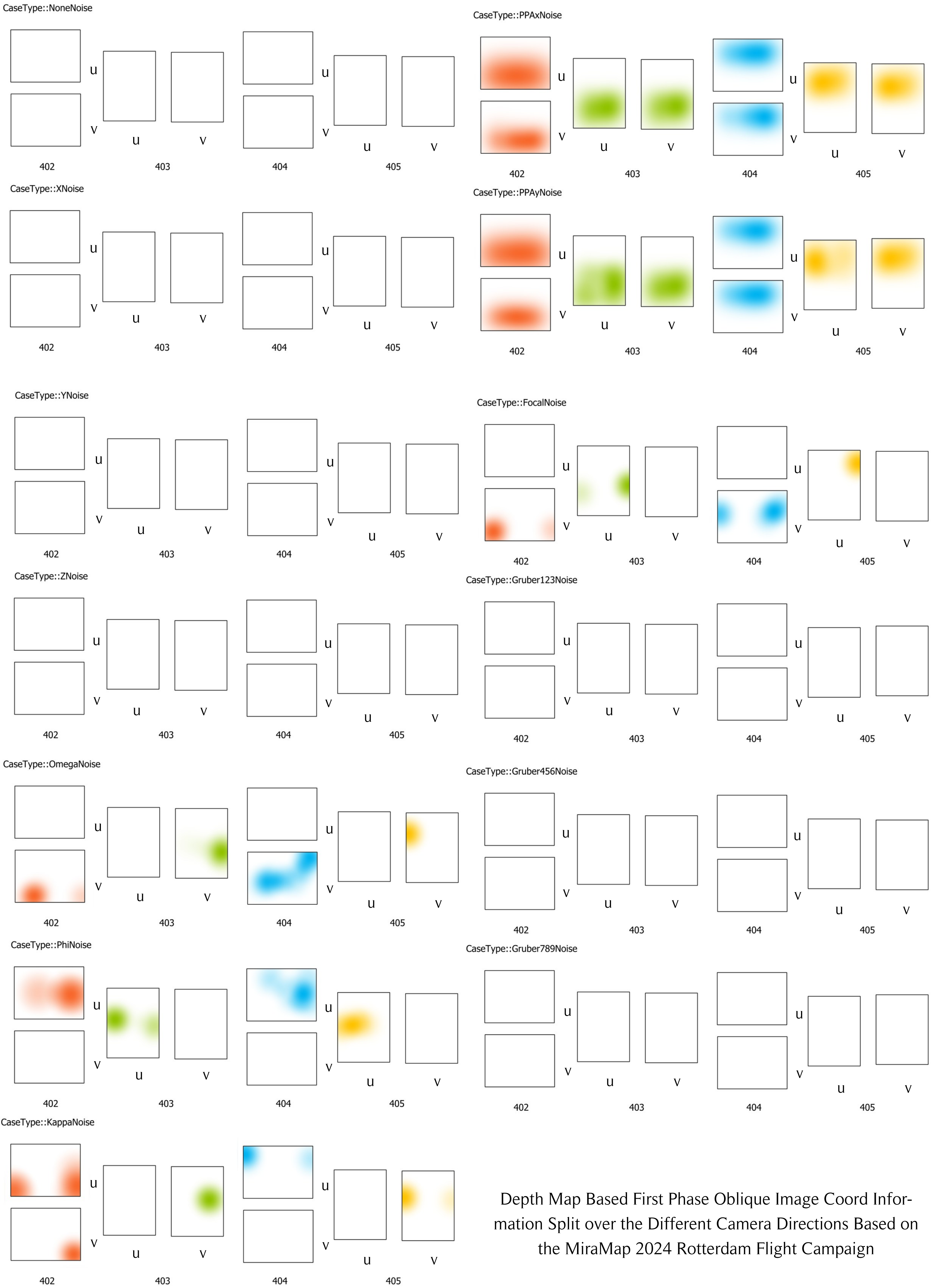

Average K Difference Values Per Noise Case (Grouped by Type)



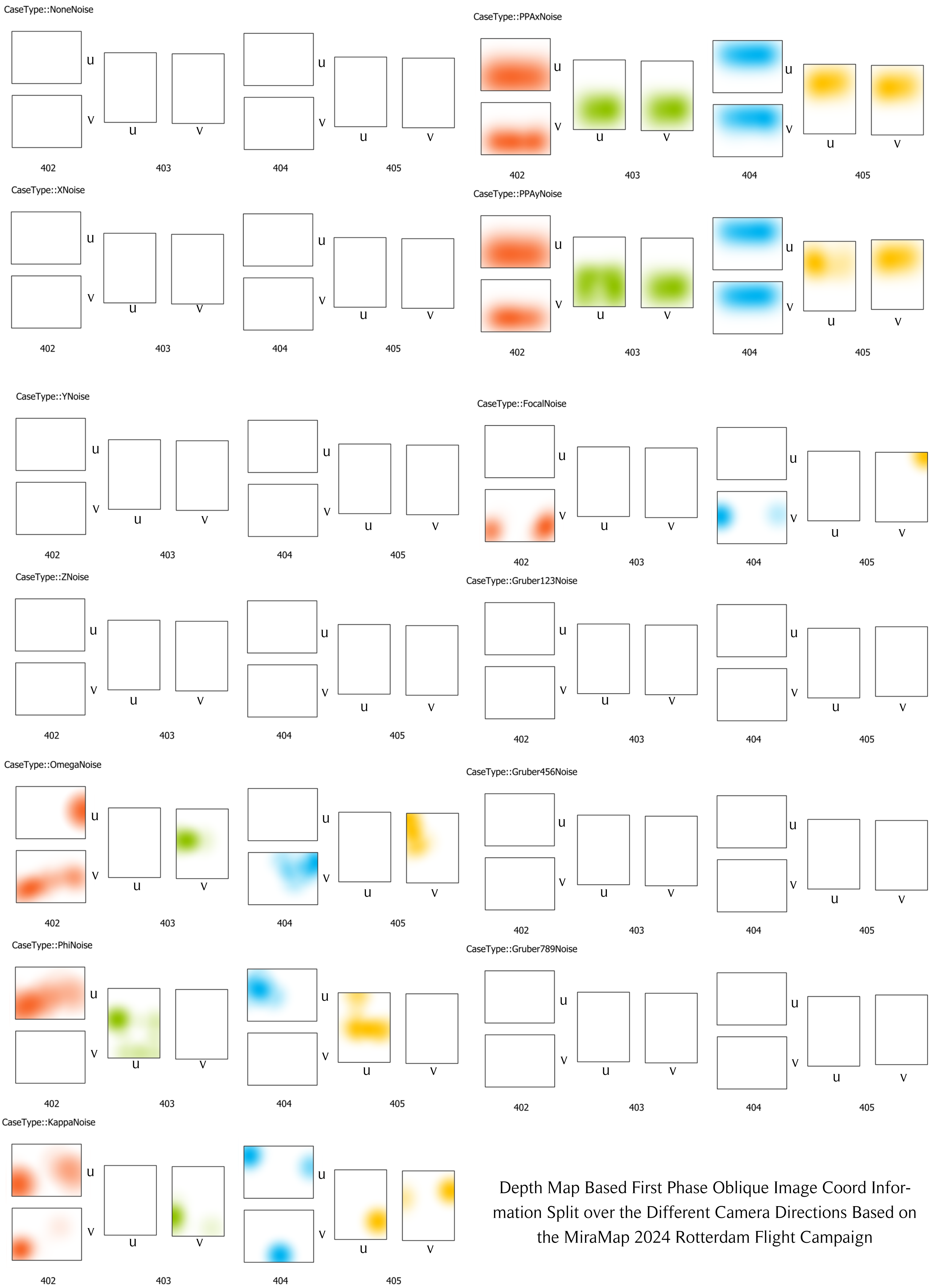
Depth Map Based First Phase Oblique Image Coord Information Nadir Direction Based on the MiraMap 2024 Rotterdam Flight Campaign



Depth Map Based Second Phase Oblique Image Coord Information Nadir Direction Based on the MiraMap 2024 Rotterdam Flight Campaign



Depth Map Based First Phase Oblique Image Coord Information Split over the Different Camera Directions Based on the MiraMap 2024 Rotterdam Flight Campaign



CaseType::NoneNoise

CaseType::PPAxNoise

CaseType::XNoise

CaseType::PPAyNoise

CaseType::YNoise

CaseType::FocalNoise

CaseType::ZNoise

CaseType::Gruber123Noise

CaseType::OmegaNoise

CaseType::Gruber456Noise

CaseType::PhiNoise

CaseType::Gruber789Noise

CaseType::KappaNoise

Depth Map Based First Phase Oblique Image Coord Information Split over the Different Camera Directions Based on the MiraMap 2024 Rotterdam Flight Campaign

Second Phase of Nadir Adjustment

For both the first (left) and second (right) phases of nadir observations, NoneNoise, XNoise, YNoise, ZNoise, and all GruberNoise categories (123, 456, 789) again yielded no W-test rejections, suggesting these noise types do not significantly impact the image coordinate observations.

In the first phase, OmegaNoise introduced rejections mainly in the v image coordinates, while PhiNoise led to a higher concentration of rejections in the u coordinates—exhibiting a similar directional behavior to OmegaNoise. KappaNoise produced rejections in u coordinates near the top and bottom of the image, while v rejections were more central, consistent with its rotational influence.

PPAxNoise caused widespread u coordinate rejections, reflecting its targeted influence on the horizontal axis. PPAyNoise, while primarily affecting the v coordinates, also showed some rejection in u coordinates, likely due to the portrait-oriented nature of image acquisition. FocalNoise was minimal in this phase, with only sparse rejections observed.

In the second phase, OmegaNoise affected only v coordinates, specifically within Von Gruber squares 4, 5, and 6, with u coordinates remaining stable. PhiNoise showed the inverse pattern, impacting only u coordinates. KappaNoise rejections were mild and mostly in the v coordinates, with a more dispersed spatial distribution.

PPAxNoise continued to show strong u coordinate rejections, especially in Von Gruber squares 4–6, while v rejections were confined to the image edges. PPAyNoise displayed the complementary pattern—more v coordinate rejections centrally, with some u coordinate influence at the edges. FocalNoise rejections increased toward the image periphery, suggesting the expected magnification of focal length errors with distance from the image center.

First Phase of Oblique Adjustment

In the first phase of oblique observations, NoneNoise, XNoise, YNoise, ZNoise, and all GruberNoise groups (123, 456, 789) again showed no W-test rejections—confirming the dataset's stability under these noise conditions.

OmegaNoise rejections were concentrated along the image center line, with the v coordinates being more significantly impacted. Interestingly, image directions 403 and 405 (side-viewing cameras) exhibited no u-coordinate noise, likely due to their alignment with the flight direction, rendering Omega's rotational effect ineffective on the u axis in these cases.

PhiNoise also showed rejection patterns along the image center but at a lower intensity than OmegaNoise, suggesting that the disturbances originate more from internal sensor perturbations than external geometric misalignments. KappaNoise rejections were primarily observed along the image sides, consistent with its rotational control over the image plane—especially near the periphery where its effects are amplified.

PPAxNoise showed frequent u coordinate rejections across multiple image directions. PPAyNoise, as the inverse of PPAxNoise, similarly exhibited u coordinate rejections, reinforcing the single-axis sensitivity observed previously during Phase IB. FocalNoise continued to have no rejections, supporting the robustness of focal length estimations under these configurations.

Second Phase of Oblique Adjustment

In the second phase of oblique adjustment, NoneNoise, XNoise, YNoise, ZNoise, and all GruberNoise variants (123, 456, 789) again showed no rejected W-tests, confirming the dataset's strong resistance to these types of perturbations.

OmegaNoise caused the highest number of rejections along the image's center line, with a strong impact on v coordinates. Notably, side-looking image directions (403 and 405) showed no u coordinate rejections—again reflecting the geometric alignment of these views with the flight path, which nullifies OmegaNoise's effect along that axis.

PhiNoise exhibited a similar center-line rejection pattern but with fewer instances than OmegaNoise, again indicating that internal sensor errors are more influential than external geometry for this noise type. KappaNoise rejections appeared mostly along the image edges, consistent with its effect on peripheral image rotation.

PPAxNoise produced widespread u coordinate rejections, while PPAyNoise, despite being its conceptual inverse, also showed elevated u coordinate rejections. This repeated pattern reinforces the earlier identification of a single-sided correlation in Phase IB, possibly linked to sensor layout or acquisition strategy. As in previous phases, FocalNoise showed no rejections, underlining its minimal influence on observation reliability in oblique setups.

5.3 Analysis of Phase II: Synthetic Computer Vision Based Object Point Detection

This phase of the research investigates the feasibility of using synthetic computer vision for object point detection by leveraging synthetic imagery generated in Blender (see Section 2.17 and Section 3.4). The analysis looks at both nadir and oblique. The first five synthetically rendered and adjusted nadir image sets (see Section 2.2); the second addresses the corresponding oblique adjusted image sets (see Section 2.3).

A total of 146 nadir images from 6 flight lines (see Section 2.4), all within the focus area (see Section 1.5), were used for this phase. For oblique imagery, 552 images were generated from 12 flight lines. Feature detection and matching were performed using both SIFT and DISK detectors (see Section 2.15). LightGlue was employed for feature matching, while the reconstruction was done using COLMAP (see Section 3.4). The results for each CaseType are summarized in Table 15, where steps that failed or produced overly high errors are marked in red. Detailed analysis and explanations of failure cases follow below.

	Number of Images	Detection (S/F)/ Matching (S/F)/ Adjustment Phase (S/F) Nadir	F-Test Results Nadir	Detection (S/F)/ Matching (S/F)/ Adjustment Phase (S/F) Oblique	F-Test Results Oblique
DISK Normal	146 Nadir Images or 552 Oblique Images	S/F/-	-	S/F/-	-
SIFT Normal		S/S/S	0.429	S/S/S	0.830
DISK Bright		S/F/-	-	S/F/-	-
SIFT Bright		S/S/S	0.432	S/S/S	0.835
DISK Shade		S/F/-	-	S/F/-	-
SIFT Shade		S/S/S	0.434	S/S/S	0.838
DISK Snow / Rain		S/F/-	-	S/F/-	-
SIFT Snow / Rain		S/S/S	0.435	S/S/S	0.839
DISK Clouds		S/F/-	-	S/F/-	-
SIFT Clouds		S/S/S	0.447	S/S/S	1.103

Table 15: Results of Phase II across Nadir and Oblique Image Sets

The results clearly show that SIFT performed reliably across all CaseTypes and in both nadir and oblique imagery. All SIFT-based pipelines succeeded through detection, matching, and adjustment phases, and the F-Test values (see Section 2.23) further affirm stable reconstructions.

In contrast, DISK consistently failed in the matching and adjustment phases, regardless of lighting or environmental condition (Bright, Shade, Snow/Rain, or Clouds). The likely reason stems from the inability of LightGlue to match features detected by DISK. As explained in Section 2.15, DISK is a CNN-based feature detector reliant on training data. Since it operates based on learned features from real-world datasets, it likely struggles with the synthetic image characteristics rendered in Blender (see Section 2.17). This issue is exacerbated in low-texture or simplified synthetic environments.

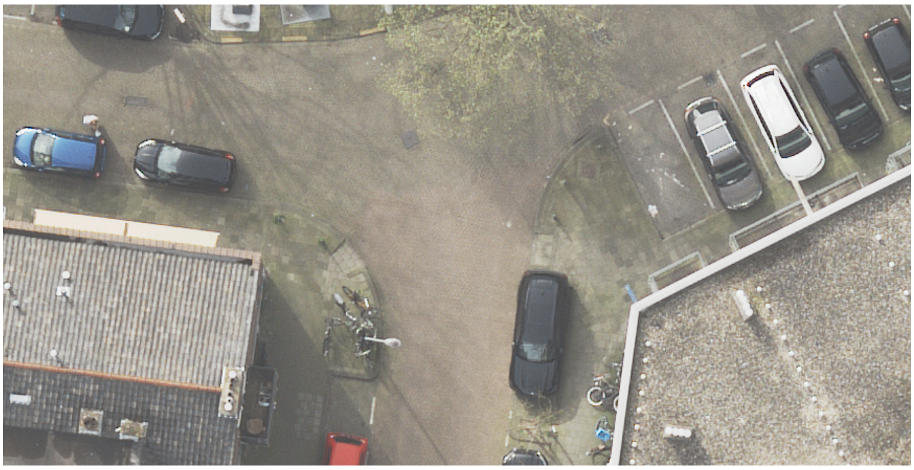


Figure 45: Real Rotterdam Image Snippet of 401_0026_00144400

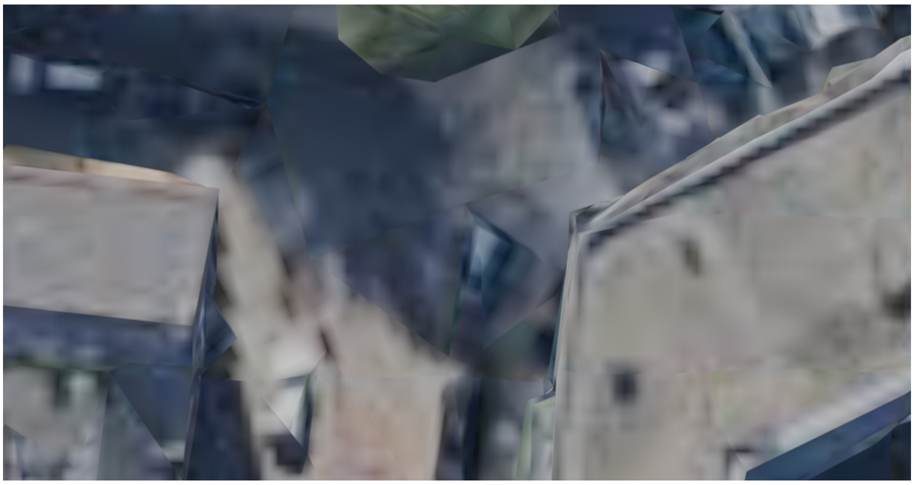


Figure 46: Rendered Rotterdam Image Snippet of 401_0026_00144400

Figure 45 and Figure 46 illustrate this discrepancy. Both images were captured at the same location using identical camera parameters. The real image exhibits more detailed and complex features, which are beneficial for CNN-based detectors like DISK. The rendered image lacks these qualities, suggesting that DISK may fail to compute meaningful descriptor vectors for LightGlue, subsequently causing COLMAP reconstruction to fail.

For the oblique imagery, while SIFT maintained full reconstruction capability across all CaseTypes, DISK again failed. Despite better overlap between images in the oblique sets (leading to some successful inter-image matches), the overall reconstructions were still incomplete. The partial success in matching did not translate to viable 3D reconstructions, as COLMAP generated intersecting point planes with singular point overlap, rather than cohesive models.

In conclusion, the failure of DISK in both nadir and oblique sets highlights a fundamental limitation in using synthetic imagery with CNN-based detectors that rely on real-world training data. The success of SIFT indicates that traditional object point detectors, not reliant on learned descriptors, are more robust in synthetic environments. This distinction is critical when considering synthetic data pipelines for tasks involving structure-from-motion or 3D reconstruction.

5.4 Analysis of Phase III: Applying Learned Lessons to Real Life Data

Phase III marks the transition from controlled, synthetic environments to the application of the developed pipeline on real-world data. This phase leverages the insights and intermediate conclusions from Phases I and II to evaluate performance under authentic, unconstrained conditions. Specifically, aerial imagery from the Miramap 2024 Campaign, acquired by the Municipality of Rotterdam, was used as the testbed to validate the end-to-end robustness of the proposed detection, matching, and adjustment workflow.

The primary objective in this phase was to assess how well the improvements and refinements introduced during synthetic testing generalized to real nadir and oblique datasets. The evaluation focused on two object point detectors—DISK and SIFT—in combination with LightGlue for object point matching, followed by bundle adjustment. COLMAP was used as a secondary benchmark for matching quality in some control runs.

Real imagery presents several challenges not encountered in synthetic environments, such as:

- Radiometric variations due to lighting, atmospheric conditions, or sensor differences. Due to the uniformity of the Google 3D tile color space
- Geometric inconsistencies, especially in oblique images where the rotation needs to be managed in a way to align the images in a consistent PointsUp manner (see Section 2.12)
- Stochastic (see Section 2.7) noise in camera parameters, particularly in focal length and principal point estimations, which are inherently less controlled than in synthetic setups

To mitigate these effects, several key lessons from earlier phases were applied:

- **Image orientation standardization:** Since DISK is not rotation invariant, all images were pre-rotated in memory to a consistent upright orientation, substantially improving object point repeatability.
- **Realistic content awareness:** Unlike synthetic datasets, real-world scenes exhibit much more heterogeneity. Detectors and matchers tuned on idealized imagery underperformed, highlighting the necessity of training and evaluating on real content.
- **Variance expansion in the camera model:** For oblique imagery in particular, increasing standard deviations in the intrinsic and extrinsic camera parameters proved critical for accommodating the higher geometric variability and ensuring convergence during bundle adjustment.

These refinements led to a significant increase in workflow stability and success rate, both in detection and subsequent adjustment stages.

Figure 42a and Figure 42b showcase representative examples of the nadir and oblique imagery used. Table 16 summarizes the stepwise success status—detection, matching, and adjustment—for both object point detectors and image types, alongside the F-test values that quantify the quality of geometric adjustment (see Section 2.23).

The F-test results fall well below the critical value (approximately 1), indicating successful adjustment across all configurations. This is notable given the scale and complexity of the input imagery consisting of 146 nadir images (see Section 2.2) and 552 oblique images (see Section 2.3)

- DISK achieved stable performance across both image types. The F-test value of 0.450 for nadir and 0.659 for oblique indicates that the adjustment successfully reconciled image geometry with the predicted observations. Notably, despite its lack of rotational invariance, the pre-rotation step enabled DISK to perform reliably.
- SIFT, by contrast, showed slightly worse performance in oblique imagery, with an F-test value of 0.830 compared to DISK with 0.659. This result reflects SIFT inherent invariance to scale and rotation, which is particularly beneficial in highly variable oblique conditions. The nadir results (F-test of 0.460) are similar to DISK, confirming both detectors' competence in planar, top-down scenarios.

The consistent success across all workflow stages, object point detection, matching, and bundle adjustment, demonstrates that the system, once tuned with realistic assumptions and noise handling, is viable for operational deployment. Several insights emerge from this:

SIFT remains a highly robust baseline, particularly in challenging oblique scenes where viewpoint variation is more extreme. Its rotation and scale invariance provide a clear advantage in such cases.

DISK, despite being a learning-based method, performs comparably well when rotational normalization is introduced. Its performance on nadir imagery is nearly identical to SIFT, suggesting its potential in scenarios with more predictable camera poses.

	Number of Images	Detection (S/F) Matching (S/F) Adjustment Phase (S/F) Nadir	F-test Results Nadir	Detection (S/F) Matching (S/F) Adjustment Phase (S/F) Oblique	F-test Results Oblique
DISK Real Data	146 Nadir Images or 552 Oblique Images	S/S/S	0.450	S/S/S	0.659
SIFT Real Data		S/S/S	0.460	S/S/S	0.830

Table 16: Pipeline Success and F-test Results for DISK and SIFT on Real Nadir and Oblique Imagery. (S refers to a success, and F refers to a failed step)

F-test values (see Section 2.23), while below the rejection threshold, are higher in oblique scenarios. This aligns with expectations, as these configurations involve more complex geometry, larger occlusion regions, and more variable imaging conditions. The higher F-test scores do not indicate failure but rather a closer proximity to the threshold, underscoring the importance of noise-aware parameter modeling in real-world workflows.

Phase III confirms the real-world feasibility of the synthetic workflow developed in earlier phases. The insights regarding image orientation, realistic scene content, and parameter flexibility translated directly into increased pipeline reliability. The framework now stands as a viable solution for large-scale aerial photogrammetry, with proven performance across multiple image geometries and two distinctly different object point detection paradigms.

6 Discussion and Limitations

Now that the adjustment and detection steps of the synthetic framework proposed in Section 3.1 have been explained (see Chapter 3), created (see Chapter 4), cases have been generated and the results have been analyzed (see Chapter 5) several discussion and extension points can be brought forward. This can be done based on the intermediate conclusions made at the end of the phases presented in Chapter 4 and Chapter 5. The chapter is split into two sections, the first (see Section 6.1) is the discussion of the overall implementation success of the two synthetic framework steps, what could be improved given more time, and what parts were successful. The second subsection (see Section 6.2) delves into possible extensions and limitations of the current proposed novel synthetic framework steps. These are brought forward as they show the current standings of the implementation that could be used as a reference for reproducibility as well as starting points of future research.

Both sections will again be referenced at the end of the thesis in Chapter 7 as possible future work. This will include the overall framework (see Section 3.1), the current focused steps (see 3) and applications of a synthetic photogrammetric collection framework not focused on the traditional aerial imagery collection pipeline laid out in the HTW from [Polman and ir. M.A. Salzmann, 1996] discussed in Section 2.1. These non-traditional extensions were first discussed in Chapter 1 as the relevance of the project extends further than mere oblique aerial adjustment.

6.1 Discussion

The overall goal of this thesis was to synthesize the steps of the obliquely appropriated aerial collection framework, as laid out for nadir in the HTW pipeline from [Polman and ir. M.A. Salzmann, 1996, p. 420]. A detailed explanation of the nadir pipeline is provided in Section 2.1, while the synthesized oblique framework is presented in Section 3.1. This synthesis led to the identification and selection of two critical steps for deeper exploration: adjustment (see Section 3.2 and Section 3.3) and detection (see Section 3.4). These steps were chosen due to the relatively high cost associated with using/ collecting real-world data (see Chapter 1), the novelty of the computational challenges they presented (see Chapter 3), and their potential broader impact on future research directions. As mentioned before, these future implications are further elaborated upon in Chapter 7.

Other elements of the pipeline (see Section 2.1) and possible components within the synthesized framework (see Section 3.1), such as the signal and flight planning stages (see Section 2.4), were not synthesized—primarily due to time constraints and the relatively lower cost associated with these steps in real-world applications. Consequently, they were assigned lower priority in terms of value and research focus within this thesis. Similarly, the mapping, maintenance, and reconnaissance activities, which represent the final step of the nadir pipeline, were also not further explored. This exclusion was partly due to their dependence on the successful completion of Phases IA, IB, and II. Furthermore, implementing this step was restricted by the significant variation in how municipalities across the Netherlands interpret and execute this process, indicating a need for future identification and standardization through user studies.

In addition to discussing the main goals and implemented components, one of the initial points of interest is the use of 3D-BAG vertices as the basis for object-level modeling and alignment. This choice influenced how synthetic data could be reliably generated and manipulated. Another choice could have been through the use of AHN points, which would have introduced a number of new opportunities and challenges. The main advantage related to AHN would be the more evenly distributed points across the whole framework. AHN would also allow for detection algorithm similarity by excluding the unclassified and water layers of the AHN. The challenges would arise in the choice of depth map mesh, as no full triangulated mesh exists that aligns with the AHN. Another possible problem could be due to certain facades having no points present due to occlusion, which could result in an uneven distribution of points.

Another relevant discussion concerns the decision to apply noise exclusively to the extrinsic (see Section 2.5) and intrinsic (see Section 2.8) camera parameters, while leaving the object points (see Section 2.11) unaffected. Conceptually, this introduces a critical distinction regarding which data elements were perturbed during the simulation. This would have affected the influence over the number of parameters that can be influenced and how these relate to the camera model. In case object point parameters were perturbed, only P_X P_Y P_Z (see Section 2.12) could have been changed. This excludes parameters such as PPA and O, which have been proven to be very insightful.

One of the primary accomplishments of this thesis is the development of a novel synthetic method for testing individual parameter-based adjustment. This included not only the simulation of errors but also the generation of test datasets that could be used in future work to evaluate techniques such as extensions to the Bundle Adjustment algorithm.

One of the notable achievements is the insight gained into the behavior of DISK, a CNN-based object point detector (see Section 2.15). The findings indicate that DISK performs poorly on unrealistic synthetic images, failing to reliably detect and match object points when compared to SIFT. This was particularly evident in Chapter 5, where synthetic images led to limited or failed detections. This problem might be alleviated either by improving the quality of synthetic ground-truth imagery (which would stray past the purpose of this ground-truth data, as it could potentially be very single-use-case specific) or by adopting more photorealistic rendering methods (such as Gaussian splatting; see Section 2.17) instead of comparatively visually simpler models like Google 3D Tiles (see Section 2.13). The results also showed a marked improvement when real-world imagery, properly rotated and aligned, was introduced in Phase III, confirming the importance of image realism for CNN-based methods.

The combination of synthetic adjustment and synthetic object point detection step for application on real-life data turned out to be significantly more challenging than initially expected. Many of the lessons learned in synthetic environments did not generalize well to real-world scenarios, illustrating the inherent complexity of such transitions. This either proves the inability of synthetic data to provide such conclusions or the necessity to improve the realism present in the synthetic data.

Furthermore, the large number of test cases created a bottleneck in both analysis and visualization. The sheer volume of data (especially the 832 distinct W-tests, based on combinations of CaseTypes, four adjustment phases, and the sub-phase structure of Phase I) rendered traditional visualization methods like spark lines, violin plots, and box plots impractical. As a result, a more general overview approach was adopted, serving as a proof of concept for the kinds of analytical depth that could be pursued in future research toward a comprehensive synthetic framework.

Another point of discussion relates to how the F-tests and W-tests were averaged and summarized to indicate overall trends. In the case of the F-tests, aggregation by CaseType is justifiable since they share a consistent experimental setup (see Section 5.1 and Section 5.2). However, this form of summarization, while offering a broad view, may obscure important nuances in the data and should be interpreted with care in any future application.

In summary, this thesis successfully defined and synthesized a subset of the oblique aerial imagery pipeline, implemented and analyzed novel methods for adjustment and detection, and uncovered key insights regarding synthetic data usage in computer vision pipelines. Although several challenges emerged—particularly with the generalization of synthetic methods to real-world applications—these challenges also reveal fertile ground for further research and development, especially in the creation of more robust synthetic-to-real transfer methods and visual analysis tools.

6.2 Extensions and Limitations

As discussed previously, the proposed novel methods outlined in Section 3 contribute various important insights into aerial adjustment and object point detection workflows. As demonstrated in the results presented in Section 6.1, these approaches offer promising directions and practical improvements for oblique adjustment processes. However, the current implementation is also subject to a number of limitations that must be acknowledged when interpreting the results. These limitations not only highlight constraints within the current methodology but also open avenues for further research, as discussed in Chapter 7.

Singular Standard Deviation and Absence of Bias
A primary limitation and possible extension is the use of a singular standard deviation value for each case type during Phases IA and IB. This choice was necessitated by processing time constraints associated with generating simulation data. The current setup uses the following standard deviations:

- X, Y, Z camera position deviations: 0.05 meters (approximately one-third of the limit proposed by the HTW)
- Omega, Phi, Kappa camera angle deviations: 1 degree (also one-third of the HTW limit)
- Principal point (x, y) deviations in pixel coordinates (u, v): 1 MM
- Camera focal length deviation: 1 MM
- Gruber square point simulations limited to: square 1–3, square 4–6, and square 7–9

An alternative strategy could have involved reducing the number of simulations per test case, thereby allowing the introduction of multiple standard deviation levels. While this would have introduced greater uncertainty (see Section 2.24), it would also have enabled a broader exploration of variance impacts. Additionally, all simulation data used a mean of zero to eliminate bias. While this helps isolate the effect of variance alone, it also removes the opportunity to observe the influence of biased input data, which may occur in real-world scenarios. In the end, this was not performed in this thesis due to the overall goal being to develop a framework.

Distance-Based Weighting
One of the possible extensions lies in the implementation of oblique distance-based weighting within Bundle Adjustment. The proposed method would have weights that are determined by the v-coordinate positions, in direct correlation with the distance from the camera, and consequently affect the GSD size. While this provides a form of adaptive precision, it introduces dependency on a single spatial metric, potentially overlooking more complex error propagation patterns.

Limited Detection Methods
The number of detection strategies could have been expanded, limiting their practical assessment. As a result, it is currently not possible to evaluate their full robustness, accuracy, or computational efficiency across varying scenes or conditions, as the current five similar image sets would need to be expanded. The use of DISK and SIFT merely scratches the surface of the search for the optimal implementation and was simply due to time constraints, not further explored.

Restricted Focus Area and Scope
The methodology has been evaluated exclusively in urban environments (see Section 1.5), which inherently contain repetitive structures, strong vertical features, and dense construction patterns. While this setting is relevant for many aerial imaging tasks, it poses challenges in generalizing findings to rural or agrarian areas, where terrain and visual features differ significantly. The method also currently only focuses on the Dutch data collection method, and no attempt has been made to analyze the methods of other countries. This limited scope is due to time constraints.

Singular Variable Analysis
Another key limitation is the use of singular-variable test cases. The current experimental setup includes the following isolated test scenarios:

- Normal
- XNoise
- YNoise
- ZNoise
- OmegaNoise
- PhiNoise
- KappaNoise
- PPAxNoise
- PPAyNoise
- FocalNoise
- Gruber123Noise
- Gruber456Noise
- Gruber789Noise

While this isolation simplifies analysis and attribution of effects, it limits the ability to study interactions between variables. A combined-variable test approach could reveal complex interdependencies and more closely mirror real-world uncertainties. Furthermore, the inclusion of Von Gruber noise types—though increasing result diversity—may introduce complexity without corresponding analytical value.

Singular Image Sets
Although five distinct image sets were created, each exogenous effect (e.g., shadowing, lighting variation, and perspective distortion) was applied to only one rendered image set (see Section 2.17). This reduces the statistical robustness of the analysis and limits the generalizability of findings. Multiple renderings under varying conditions could better capture the spectrum of real-world variability.

Baked-in Exogenous Effects
Finally, another significant limitation is that environmental effects such as shadows and lighting variations are inherently embedded (see Section 2.17) into the Google Maps

7 Future Work

As mentioned in Section 1.2, the completion of this research not only impacts the ability to adjust oblique aerial imagery to a standard that allows a municipality to use it in their geo processing, but also other types of image adjustment and aerial collection-related topics. Furthermore, Chapter 5 also highlights some of the limitations and interesting topics related to the proposed methods. Both Section 1.2 and Chapter 5 were used to form a basis for the future work. This chapter delves into recommendations and formulates possible research questions that could be of interest to extend the research or further research into other topics.

Gaussian Splatting for the Synthetic Object Point Detection

One of the main outcomes of the research related to the rendering (see Section 2.17) of synthetic 3D city models as was applied during the research on the 3D Tiles for Phase II was the relatively low number of triangles to represent the model which results in CNN-based image detectors like DISK not being able to create a reliable description vector. This results in LightGlue (see Section 2.15) not being able to reconstruct the adjustment system, which led to the problems in the result and analysis of Phase II.

As proposed in Section 6.1, this can either be resolved through the creation of ground truth data based on synthetic data. However, as discussed in Chapter Section 8 this would most likely not lead to good results. Instead, a recommendation could be made to test the CaseType setup of Phase II with a Gaussian Splatted 3D city model, which, as discussed in Section 2.13, would lead to a higher rendering time but also a more photo realistic view. Whether it is photorealistic enough is the question. The proposed framework is designed to accommodate such changes and could be further extended by other 3D models. Key research questions for this could be:

- How does Gaussian Splatting influence the density and quality of object points detected by CNN-based detectors like DISK when applied to low-detail 3D models?
- Can a photorealistic rendering of sparse 3D data using Gaussian splatting significantly improve the feature matching performance of LightGlue when compared to rendered Google 3D tiles or other 3D city models?
- How do different configurations (e.g. splat size, density, opacity) in Gaussian Splatting affect the stability of the object point detection?
- To what extent can Gaussian splatted models simulate real-world texture and lighting effects (High Dynamic Range) relevant to aerial perspective object point detection?

Creating a Dutch Aerial Photography Specific Ground Truth Dataset for DISK

Even though a synthetic ground truth data set is not necessary, it is theorized that the overall performance of DISK could be improved by creating a Dutch-specific aerial-oriented ground truth data set. The current ground truth data set is created by the Technical University of Munich based on roughly 100.000 images taken at ground level in and around Munich [Tyszkiewicz et al., 2020]. These images do not necessarily represent the Dutch situation (Difference in Dutch and South-German architecture), nor from an aerial point of view (street view vs aerial view). As such, certain elements within the Dutch built environment visible from an aerial perspective might be lacking in the original dataset. It could be interesting and valuable to improve both oblique and nadir adjustment processes. Key research questions for this could be:

- How do features in the Dutch built environment, as captured from aerial views, differ from those in the Munich-based ground truth dataset?
- Can the inclusion of oblique and nadir Dutch aerial images in a new ground truth dataset improve the robustness of feature detectors in local use cases?
- How does urban morphology (e.g. roof types, canals, agricultural structures) impact the repeatability and descriptiveness of object points in aerial photography?
- Would a region-specific dataset improve generalization or lead to overfitting in aerial feature detection tasks using learned descriptors?

The successful completion of this research would benefit all forms of CNN-based detectors that use the Munich dataset.

Distance-Based Weighting for Oblique

Another potential research extension, first mentioned in the limitations of Chapter 6, involves the incorporation of distance-based weighting for the Jacobian matrix (see Section 2.20). Due to the increasing GSD (see Section 2.9) in oblique imagery, it may be reasonable to assume that object points detected near the top of the image (farther from the camera) should be given less weight during the adjustment process. This contrasts with traditional approaches, which often assume nadir imagery with a relatively uniform GSD across the image. This concept was briefly explored in Phase IA and Phase IB (see Section 3.2) by excluding points in different Von Gruber Squares, although this primarily resulted in the system becoming disconnected. Nonetheless, it opens the door for further investigation into distance- or angle-based weighting strategies. Key research questions include:

- What are realistic weight values that can be applied to the Jacobian matrix as image coordinate v increases in the case of Oblique imagery?
- How does the GSD, as influenced by the viewing angle, affect the accuracy and reliability of object point detection?
- Can a continuous weighting function based on pixel height or estimated GSD improve the robustness of the adjustment process?
- How does perspective distortion in oblique imagery influence the recognizability of object points across differently angled image sets?
- Is there an optimal cutoff threshold beyond which object points should be down-weighted or discarded entirely in oblique views?

Adjustment of Terrestrial Panoramic Images TPI

While the current photo-based adjustment framework is focused primarily on aerial imagery, a possible extension could be its application to TPI. Panoramic images are often captured using 360° cameras or multi-camera rigs. TPI is increasingly used in urban mapping and street-level data acquisition. However, the adjustment and integration of these images into photogrammetric workflows pose unique challenges due to their spherical projection, varying optical centers, and frequent occlusions in dense environments. Key research questions include:

- How can a TPI system be synthetically modeled into a workflow?
- How can traditional photogrammetric adjustment models be adapted to handle spherical projection?
- What is the impact of varying camera baselines in 360° systems on bundle adjustment accuracy?
- How do different types of terrestrial environments (open streets vs. indoor corridors) affect adjustment stability?

Compare more Object Point Detection and Matching Algorithms

The performance of image-based adjustment systems is significantly influenced by the choice of object points detection and matching algorithms (see Section 5.4). In the current implementation, only a limited set of algorithms were evaluated, often based on legacy use or ease of implementation. However, with advancements in computer vision and the increasing complexity of aerial image acquisition (especially in oblique views), it could be seen as necessary to broaden the comparative scope. Key research questions include:

- How do modern object point detectors (SuperPoint, R2D2, D2-Net) compare to traditional methods (SIFT, ORB, SURF) in oblique aerial imagery?
- What is the impact of descriptor dimensionality on matching accuracy and processing time in large-scale image networks?
- Do certain algorithms perform better for nadir versus oblique images, and is there value in hybrid detection strategies?

8 Conclusion

This final chapter draws conclusions related to the assessment of the synthesized framework (see Section 3.1) and addresses the sub-questions based on the results presented in Chapter 4 and the analysis in Chapter 5. Since each sub-question may partially answer multiple phases, broader conclusions are given here. For phase-specific conclusions, readers are referred to the intermediate summaries at the end of each respective subsection in the aforementioned chapters. The chapter concludes by addressing the research sub-questions originally formulated in the introduction (see Section 1.3).

Before that, first a set of overarching conclusions is presented related to the synthesized workflow (see Section 2.1), here referred to as the framework (see Section 3.1). As noted earlier, the term framework is used to differentiate this experimental design from a full operational pipeline, as it focuses specifically on adjustment and detection, while deliberately excluding elements such as flight and signal planning, mapping, and maintenance.

From this, the first conclusion can be drawn. The developed framework proved insightful, achieving the primary aim of experimental research, exploring new novel methods that may advance the feasibility of oblique adjustment and support the creation of other synthetic frameworks. It can thus be said that the overall concept of the synthetic framework is successful. As specifically for the synthetic adjustment, it is possible to generate synthetic test cases using the mathematics involved to meaningfully investigate the independent statistical influence of key parameters (see Section 2.7).

One key accomplishment was the extended implementation of occlusion (see Section 2.19), which remains a major challenge in oblique adjustment. The successful application of computer-based graphics to Rendering occlusion (see Section 2.17) demonstrated that synthetic adjustment can realistically simulate more complex scenarios. This also led to more challenging adjustment conditions and a diverse range of results, showing how certain elements were either exaggerated or diminished when comparing unobstructed versus depth map-based cases.

In Phase II, the synthetic object point detection implementation was less successful. It was discovered that DISK is not well-suited to the image quality provided by Google 3D Map tiles. As discussed in Section 6.1, possible solutions for DIScrete Keypoints (DISK) include the creation of synthetic ground truth data or the adoption of more photorealistic image methods such as Gaussian splatting. However, synthetic ground truth data (see Section 2.15) could compromise generalization, potentially leading to overfitting, and photo-realistic synthetic images could provide more realistic analysis at the cost of higher quality input data being required, whilst still keeping the cost of collection low. Nevertheless, this phase was still highly instructive. For example, it confirmed the lack of rotational invariance in DISK, which implies that either descriptor vectors must be rotated or the images must be pre-rotated in memory for consistent performance.

The final phase of the research validated the broader concept through the successful application to real data from the municipality of Rotterdam. Adjustment was achieved in all case types, using both nadir and oblique imagery, and employing both SIFT and DISK. These results support the conclusion that synthetic adjustment can serve as a cost-effective method for research before real-world data collection, avoiding confounding variables inherent in non-controlled datasets. This success reinforces the potential value of synthetic signal and flight plan (see Section 2.4) generation and suggests that mapping and maintenance should be considered only after these earlier components have been fully explored. Only with such groundwork can a complete oblique aerial image collection pipeline be realized.

The second half of this chapter addresses each sub-question in turn.

What is the mathematical pipeline for aerial adjustment theory? (Phase IA)

The mathematical pipeline for Phase IA in aerial adjustment theory forms the foundational stage in photogrammetric block adjustment (see Section 2.20), where raw sensor observations are systematically transformed into a structured mathematical framework to enable precise geo-referencing of imagery. This phase is crucial in modern photogrammetry and plays a pivotal role in the accuracy of subsequent mapping and maintenance.

Unlike traditional terrestrial photogrammetry, aerial photogrammetry relies on processing large volumes of image data, making manual feature matching across images impractical. To address this, modern systems incorporate automated feature detection and matching as a preliminary step in the adjustment pipeline. Feature detectors such as SIFT and DISK are employed to identify salient object points in each image. These features are designed to be invariant to scale and, in the case of SIFT, also to rotation and partially to changes in illumination. After detection, descriptors are extracted and used to match features across overlapping images. This process establishes a network of corresponding 2D image points across multiple views, each set representing a single 3D object point (see Section 2.11). This automated approach replaces the manual point selection methods which was used historically, such as those described in the HTW and enables robust, scalable aerial triangulation suitable for high-throughput photogrammetric workflows.

After the object points have been detected and matched comes the use of the observation equation shown in Equation 16, which expresses the geometric relationship between the 3D object space and the 2D image space through the parameters of the imaging sensor. These parameters include the intrinsic parameters (see Section 2.8) and the extrinsic parameters (see Section 2.5). Mathematically, the observation equations are given by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f & \rho & c_x + ppa_x \\ 0 & f & c_y + ppa_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} P_X \\ P_Y \\ P_Z \\ 1 \end{bmatrix} \tag{16}$$

This observation formula is used in the formulation of the adjustment model. The object points correspondences serve as observations. The goal is to minimize the discrepancies (residuals) between observed image coordinates and those projected from the estimated 3D scene structure using the observation equation. This is typically achieved through a bundle block adjustment (see Section 2.20), where parameters such as r, t, f, d, and P are iteratively minimized.

To ensure convergence of the non-linear optimization, good initial approximations for the unknown parameters are required (see Section 2.20) is also referred to as an S-basis. These can come from either GNSS-IMU data or, if GCP are available, they are integrated as constraints in the adjustment model, helping anchor the solution in the desired coordinate system and improving geometric stability (see Section 2.21).

After adjustment, residuals are analyzed to assess the model's fit (see Section 2.23). High residuals might indicate incorrect matches, systematic errors, or miscalculated parameters (see Section 2.22). Error propagation through the covariance matrix provides estimates of the precision of the adjusted parameters, crucial for downstream tasks like base registration maintenance.

What hyperparameters are present in aerial adjustment, and what are their effects when changed in synthetic tests? (Phase IA) (Phase IB)

In the context of synthetic aerial adjustment, hyperparameters refer to user-defined settings that influence both the generation of simulated data and the subsequent adjustment process. Key hyperparameters include the number of 3D points, the number of images, and their overlap, and the assumed standard deviations for observations in the adjustment model. Increasing the number of points or image overlap generally improves geometric stability and accuracy, while varying standard deviations affects the weighting of observations and thus the confidence in the adjustment. Differences between first and second phase adjustment (e.g., using fixed vs. estimated interior/exterior parameters) also represent a hyperparameter choice, influencing convergence behavior and the robustness of the solution under simulated conditions (see Section 2.17).

What are the main differences between nadir and oblique image adjustment? (Phase IB) (Phase II) (Phase III)

The main differences between nadir and oblique image adjustment extend beyond the viewing angle and projection configuration (see Section 2.2 and Section 2.3), object point connectivity (see Section 2.15), and GSD noise sensitivity are also present. Oblique imagery typically provides higher overlap and multi-angle views, which increases redundancy and should, in theory, strengthen the adjustment network. However, oblique images also introduce greater variability in occlusions, which can reduce the number of object point connections across views, especially between images taken from significantly different angles. This challenge can be mitigated by increasing the number of tie points and images during synthetic scene creation; the stronger geometric diversity in oblique setups often leads to improved 3D reconstruction quality and better parameter recovery, albeit with higher computational complexity due to denser connectivity and larger system size.

How do the different parameters of oblique imagery influence the errors present in the final adjusted images? (Phase IA) (Phase IB)

A total of 13 case types for Phase IA and IB have been tested, each defined by specific parameter perturbations, and were tested for both nadir and oblique configurations, using both non-obstructed and depth map-based occlusion models. The tests include a series of synthetic perturbation types: XNoise, YNoise, ZNoise, OmegaNoise, PhiNoise, KappaNoise, PPAxNoise, PPAyNoise, FocalNoise, and Von Gruber square-based test cases labeled Gruber123Noise, Gruber456Noise, and Gruber789Noise, which represent the inclusion of observations from the numbered Von Gruber squares (see Section 2.14).

From the experiments, it was observed that perturbations in the cameras X, Y, and Z positions, as well as the grouped Gruber cases, did not produce significant deviations in the final adjustment error (see Chapter 4), suggesting that these parameters have a relatively low impact under the current test conditions. Rotation-based parameters (Omega, Phi, Kappa) showed moderate influence, indicating some sensitivity of the adjustment process to angular misalignment. However, the most impactful parameters were those related to the principal point, specifically PPAxNoise and PPAyNoise, which consistently resulted in higher error metrics across multiple case types.

It is important to note that these conclusions are based on a singular set of standard deviation values applied uniformly across all parameter tests. Therefore, while indicative, the analysis could be significantly refined by introducing a more comprehensive range of deviations and by exploring combined parameter interactions. An expanded independent variable analysis would provide deeper insight into the compound effects of parameter interplay and improve understanding of which aspects of oblique imagery most critically affect adjustment stability and accuracy (see Chapter 7).

How do simulated external factors such as sun position, wind, clouds, and foliage affect the adjustment? (Phase II)

Simulated external factors such as sun position, wind, clouds, and foliage generally had limited impact on both the detection and adjustment quality, though stronger effects did lead to slightly increased F-test values, indicating reduced system stability. Among the tested conditions, the only scenario that resulted in a failed adjustment was the clouded oblique case, where significant occlusion and feature loss disrupted object point matching and weakened the image connectivity. Overall, while these external influences introduce some variability, the SIFT detection remained largely resilient under most conditions tested in Phase II. The DISK detection failed in all test cases, and as such, it can be said that external factors can not be simulated.

How does the choice between object point extraction algorithms such as SIFT/LightGlue and DISK/LightGlue affect the reconstruction accuracy? (Phase II) (Phase III)

No definitive conclusion can be drawn regarding the performance difference between SIFT/LightGlue and DISK/LightGlue, as the synthetic imagery used in Phase II lacked the photometric detail and texture required for DISK/LightGlue to reliably detect and match object points. This limitation stems not from the algorithm itself, but from the quality and realism of the input data. In contrast, SIFT performed adequately under the same conditions due to its robustness in low-texture environments. However, when applied to real-world data in Phase III, both methods successfully contributed to accurate reconstructions, indicating that DISK remains a viable approach when provided with sufficiently rich visual input.

Based on this information, the main research question can be answered:

What parameters are key in the reconstruction accuracy of adjustment theory applied to RGB aerial images using synthetic test cases?

The central research question, whether synthetic data can meaningfully simulate and evaluate the performance of an aerial adjustment framework, particularly for oblique RGB imagery, can be answered in the affirmative. The findings demonstrate that synthetic data, when constructed with sufficient geometric, statistical, and visual fidelity, offers a powerful platform for probing the internal mechanics and sensitivity of adjustment theory in controlled conditions. This has notable implications for both research and operational development, as it enables cost-effective experimentation before engaging in resource-intensive real-world deployments.

The key parameters influencing reconstruction accuracy were systematically identified and isolated within a modular synthetic framework. Among these, the most impactful were found to be principal point offsets (PPAx and PPAy) and rotational misalignments (Omega, Phi, Kappa). Perturbations in these parameters consistently led to elevated residuals and degraded reconstruction accuracy, underscoring their critical role in stabilizing the bundle adjustment process. Conversely, other parameters—such as sensor position noise (X, Y, Z) and point distribution over Von Gruber squares—had relatively minor effects under the simulated conditions, suggesting a lower sensitivity to translational errors when overall geometric connectivity is preserved.

The research also highlighted the unique challenges introduced by oblique imagery, particularly concerning occlusion and viewing geometry diversity. These factors reduce the number of consistent tie points across views and complicate the adjustment process due to increased variation in parallax and surface visibility. Nonetheless, oblique images were also shown to provide stronger geometric baselines, which can enhance depth reconstruction accuracy if sufficient connectivity is maintained. The synthetic framework enabled a detailed investigation into these trade-offs, offering a level of analytical granularity not feasible with real-world data alone.

Another significant aspect of the study involved the simulation and evaluation of object point detection algorithms, specifically comparing SIFT/LightGlue and DISK/LightGlue combinations. The performance disparity under synthetic conditions revealed that traditional handcrafted detectors like SIFT maintain higher robustness in low-texture, low-photorealism scenarios, whereas learning-based detectors such as DIScrete Keypoints (DISK) are highly dependent on high-frequency visual content and lighting realism. This insight has practical implications for the selection of feature extractors in synthetic pipelines and suggests that the quality of synthetic imagery must be tailored to the capabilities of the intended detection algorithm.

Moreover, the influence of environmental factors, such as lighting changes, cloud cover, and vegetation movement, was also tested. These simulations revealed that while minor variations in scene conditions had little impact on adjustment outcomes, severe occlusions—such as those caused by clouds—could prevent convergence altogether by disrupting tie point networks. This further reinforces the value of synthetic testing environments, where such conditions can be systematically introduced and studied without the unpredictability and cost associated with physical data acquisition.

Ultimately, the study confirms that synthetic test cases are not only valid but instrumental in deconstructing the adjustment process into quantifiable, testable components. By enabling rigorous sensitivity analysis and controlled perturbation of key parameters, the synthetic adjustment framework provides a foundational tool for advancing both theoretical understanding and applied system development. While limitations remain—particularly in achieving full photorealism and realistic feature distributions—the framework demonstrated that synthetic data can approximate the essential geometric and photogrammetric behavior of real-world scenarios.

This paves the way for more strategic and informed development of future oblique aerial collection systems, where early design choices (e.g., flight planning, sensor calibration strategies, feature extractor selection) can be optimized through synthetic trials. The research thereby establishes a compelling case for integrating synthetic adjustment workflows into the early phases of aerial imaging system design and validation.

References

Kevin Wood Robotics AI. Opencv python sift feature detection (sift algorithm explained + code). Available at: https://www.youtube.com/watch?v=flFbNka62v8, 2023. URL https://www.youtube.com/watch?v=flFbNka62v8. Accessed: 2025-01-13.

Remi Arnaud and Mark C. Barnes. *COLLADA: Sailing the Gulf of 3D Digital Content Creation*. AK Peters/CRC Press, 2006. ISBN 9781568812925.

Paul Bourke. Object file format. Online, 1999. http://paulbourke.net/dataformats/obj/.

Ronald N. Bracewell. The image plane. In *Fourier Analysis and Imaging*, pages 20–110. Springer, 2003. doi: 10.1007/978-1-4419-8963-5_2. URL https://doi.org/10.1007/978-1-4419-8963-5_2.

Cesium GS, Inc. 3d tiles specification. https://github.com/CesiumGS/3d-tiles, 2017. Accessed: 2025-06-18.

Yiye Chen, Yunzhi Lin, Ruinian Xu, and Patricio A Vela. Keypoint-graspnet: Keypoint-based 6-degrees of freedom grasp generation from the monocular rgb-d input. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7988–7995. IEEE, 2023.

S. Choi and H. Cheung. A versatile stl file repair algorithm for 3d printing. *Rapid Prototyping Journal*, 14(5): 295–305, 2008. doi: 10.1108/13552540810907959.

Prof. dr. ir. M.G. Vosselman. *Fotogrammetrie I*. Faculteit der Geodesie Technische Universiteit Delft, 1995.

R. Dufour, J. van der Meer, L. van den Brink, and K. van den Berg. An updated approach to linked data publication at the dutch land registry. *ISPRS International Journal of Geo-Information*, 11(1):51, 2022. doi: 10.3390/ijgi11010051. URL https://www.mdpi.com/2220-9964/11/1/51.

Balázs Dukai, Ravi Peters, and Gina Stavropoulou. 3d registration of buildings and addresses (bag) v2024.02.28, 2024. URL https://4tu.edu.hpc.n-helix.com/datasets/c3d87f19-7d3b-4766-a915-9d9c986f1f84/1.

Blender Foundation. Eevee real-time render engine. https://www.blender.org/features/eevee/, 2020a. Accessed 2025-06-13.

Blender Foundation. Blender manual: Rendering. https://docs.blender.org/manual/en/latest/render/, 2020b. Accessed 2025-06-13.

Matheus A Gadelha and Bruno M Carvalho. Drink: Discrete robust invariant keypoints. In *2014 22nd International Conference on Pattern Recognition*, pages 821–826. IEEE, 2014.

Paul D Groves. Principles of gnss, inertial, and multisensor integrated navigation systems. *Artech House*, 2013.

Norbert Haala, Mathias Rothermel, and Stefan Cavegn. Extracting 3D urban models from oblique aerial images. In *2015 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4. IEEE, 2015.

Joachim Höhle. Oblique aerial images and their use in cultural heritage documentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40:349–354, 2013.

Joachim Höhle. Photogrammetric Measurements in Oblique Aerial Images. *Photogrammetrie - Fernerkundung - Geoinformation*, pages 7–14, 01 2008.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82 (1):35–45, 1960.

Ladislav Kavan, J. P. Lewis, Steven Collins, and Carol O'Sullivan. Practical animation transfer with correspondence-free segmentation. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 275–284, 2010.

Manon Kok, Jeroen D. Hol, and Thomas B. Schön. Using inertial sensors for position and orientation estimation. *Foundations and Trends in Signal Processing*, 11(1-2):1–153, 2017. doi: 10.1561/2000000094. URL https://doi.org/10.1561/2000000094.

Mathias Lemmens, Christiaan Lemmen, Martin Wubbe, et al. Pictometry: potentials for land administration. In *Proceedings of the 6th FIG regional conference 12-15 November 2007, San José, Costa Rica./Fredriksberg: International Federation of Surveyors (FIG) 13p*, 2007.

Nail Ibrahimli Liangliang Nan. Course notes 1: Camera models. 2025. URL https://3d.bk.tudelft.nl/courses/geo1016/handouts/01-camera_models.pdf,https://3d.bk.tudelft.nl/courses/geo1016/index.html.

Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed, 2023. URL https://arxiv.org/abs/2306.13643.

Xingyu Liu, Shun Iwase, and Kris M Kitani. Kdfnet: Learning keypoint distance field for 6d object pose estimation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4631–4638. IEEE, 2021.

Brecht Van Lommel. Cycles: A physically-based production renderer for blender. *Blender Foundation Technical Documentation*, 2018. https://www.blender.org/features/rendering/.

David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

Pratap Misra and Per Enge. Global positioning system: Signals, measurements, and performance. *Ganga-Jamuna Press*, 2006. Second edition.

Rafael Orellana, Rodrigo Carvajal, Pedro Escárate, and Juan C. Agüero. On the uncertainty identification for linear dynamic systems using stochastic embedding approach with gaussian mixture models. *Sensors*, 21(11):3837, 2021. doi: 10.3390/s21113837. URL https://www.mdpi.com/1424-8220/21/11/3837.

Ives Rey Otero. *Anatomy of the SIFT Method*. PhD thesis, École normale supérieure de Cachan-ENS Cachan, 2015.

Chao Pang, Jiang Wu, Jian Ding, Can Song, and Gui-Song Xia. Detecting building changes with off-nadir aerial images. *Science China Information Sciences*, 66(4):140306, 2023.

Ir. J. Polman and Dr. ir. M.A. Salzmann. *Handleiding voor de Technische Werkzaamheden van het Kadaster*, volume 2. Kadaster Nederland, 1996. ISBN 90-803078-1-5.

Prochitecture. Blender osm. https://prochitecture.gumroad.com/l/blender-osm, 2025. Accessed: 2025-05-22.

Rijkswaterstaat. Actueel hoogtebestand nederland 4 (ahn4), 2021. URL https://ahn.nl/ahn-4. Geraadpleegd op 22 mei 2025.

Rijkswaterstaat. Actueel hoogtebestand nederland 3 (ahn3), 2023. URL https://www.nationaalgeoregister.nl/geonetwork/srv/api/records/70d7af15-0bd8-4911-a069-148660491b66. Geraadpleegd op 22 mei 2025.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching with Graph Neural Networks, 2020. URL https://arxiv.org/abs/1911.11763.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

International GNSS Service. Igs gnss data and products, 2025. URL https://igs.org/data/.

Nicholas Shorter and Takis Kasparis. Automatic vegetation identification and building detection from a single nadir aerial image. *Remote Sensing*, 1(4):731–757, 2009.

Dave Shreiner, Graham Sellers, John Kessenich, and Bill Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3*. Addison-Wesley, 8th edition, 2013. ISBN 9780321773036.

Dave Shreiner, Graham Sellers, John Kessenich, and Bill Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Professional, 9th edition, 2016. ISBN 978-0134495491. URL https://www.opengl.org/.

Cyrill Stachniss. SIFT - 5 Minutes with Cyrill. Available at: https://www.youtube.com/watch?v=4AvTMVD9ig0&t=185s, 2020. URL https://www.youtube.com/watch?v=4AvTMVD9ig0&t=185s. Accessed: 2025-01-13.

Pixar Animation Studios. Universal scene description (usd). Online, 2020. https://openusd.org/.

Peter Teunissen. *Adjustment Theory An Introduction*. VSSD, first edition, 2000. URL https://textbooks.open.tudelft.nl/textbooks/catalog/book/95.

Peter Teunissen. *Testing Theory An Introduction*. Second edition, 2006. URL https://textbooks.open.tudelft.nl/textbooks/catalog/book/96.

David H. Titterton and John L. Weston. Strapdown inertial navigation technology. *The Institution of Engineering and Technology*, 2004.

ToppiNappi. Camera array tool for blender. https://toppinappi.gumroad.com/l/Camarray, 2025. URL https://toppinappi.gumroad.com/l/Camarray. Versie 3.2.0, beschikbaar op Gumroad.

Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis. In Reinhard Koch and Luc Van Gool, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer, 2000. doi: 10.1007/3-540-44480-7_21. URL https://doi.org/10.1007/3-540-44480-7_21.

Compton J. Tucker. Red and photographic infrared linear combinations for monitoring vegetation. *Remote Sensing of Environment*, 8(2):127–150, 1979. doi: 10.1016/0034-4257(79)90013-0.

Michal J. Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: learning local features with policy gradient. *CoRR*, abs/2006.13566, 2020. URL https://arxiv.org/abs/2006.13566.

M.G.V Vosselman. Fotogrammetrie I, 1 1996.

Jinwang Wang, Lingxuan Meng, Weijia Li, Wen Yang, Lei Yu, and Gui-Song Xia. Learning to extract building footprints from off-nadir aerial images. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):1294–1301, 2022.

Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore bundle adjustment. In *CVPR 2011*, pages 3057–3064, 2011. doi: 10.1109/CVPR.2011.5995552.

Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A Comparative Study of SIFT and its Variants. *Measurement science review*, 13(3):122–131, 2013.

Reproducibility self-assessment

Marks for each of the criteria

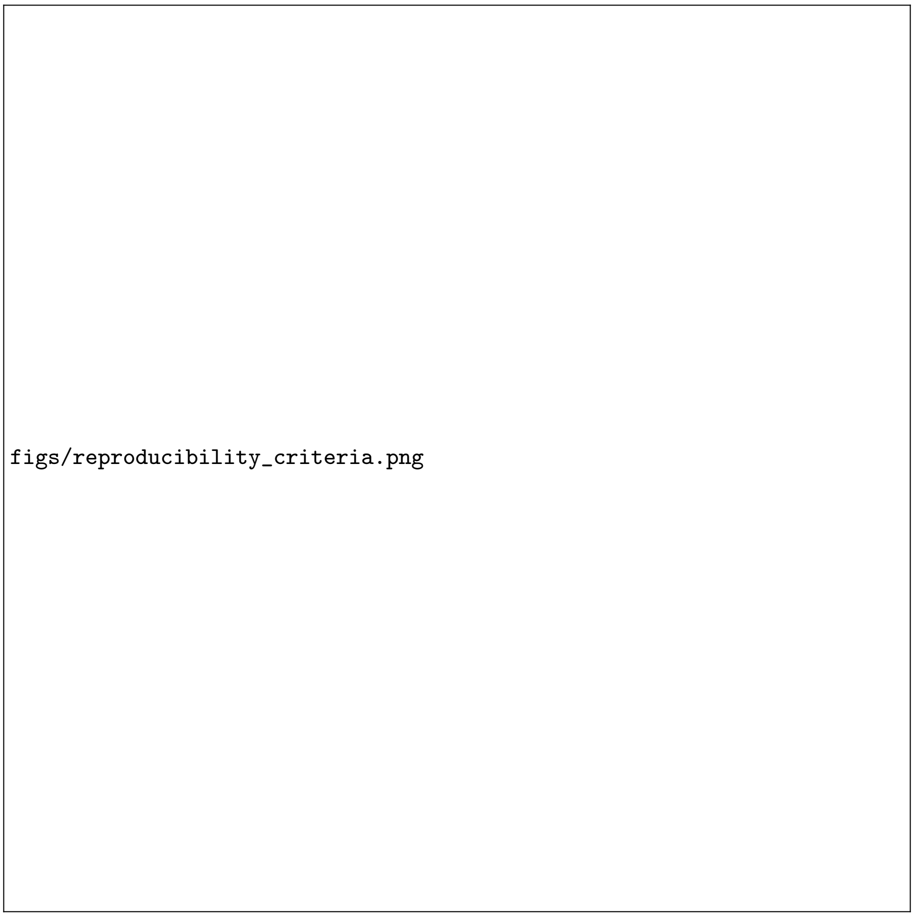


Figure 47: Reproducibility criteria to be assessed.

Grade/evaluate yourself for the 5 criteria (giving 0/1/2/3 for each):

1. input data
2. preprocessing
3. methods
4. computational environment
5. results

Self-reflection

Input data (Score: 2) — While the original data used in this research cannot be made publicly available due to privacy and contractual restrictions, a disclaimer is included in the thesis outlining how similar data may be requested. Specifically, data access may be pursued via Miramap and the municipality of Rotterdam. Though the process requires formal communication and approval, the data is not proprietary in the sense that it is completely inaccessible to others, which is why this category scores a 2.

Preprocessing (Score: 3) — The preprocessing steps involved in rendering, generating synthetic test cases, and preparing imagery are comprehensively explained in the thesis. Each step is described in detail, allowing others to follow the same procedures using either synthetic or similar datasets. Parameters, tools, and decision points are transparently documented, contributing to full reproducibility in this aspect.

Methods (Score: 1) — The core methods are well-described in both written and visual formats. Flowcharts and diagrams help in understanding the pipeline and adjustment process. However, due to the non-public nature of the codebase, some low-level implementation specifics are not accessible. This makes it difficult for an external party to reproduce the exact same pipeline without a degree of interpretation, hence the score of 1.

Computational environment (Score: 0) — Unfortunately, no code can be shared due to contractual obligations with Geodelta. The proprietary nature of both the input/output structure and internal processing logic, which may include company trade secrets, prevents disclosure of implementation details. As such, the computational environment—including software versions, dependencies, and scripts—cannot be replicated by external researchers.

Results (Score: 3) — All results generated during the project are made available and are described clearly. Each result is discussed in the context of its associated experiment or test case, and is supported by figures, tables, and textual explanation. This transparency allows others to interpret the findings and compare them against their own results should they follow a similar pipeline using available or synthetic data.

Glossary

3D Tiles A format developed by Google for streaming large-scale 3D geospatial datasets (see full explanation in Section 2.13). 29, 91

Absolute Constrained Adjustment An adjustment method that applies absolute constraints using control points. This method is primarily used for quality control of homogeneous point fields (i.e., point fields of the same type and accuracy), and for testing the connection points between different networks or coordinate sets(see full explanation in Section 2.21). 39

Adjustment Adjustment refers to the process of modifying measurements to best fit a model or reference frame (see full explanation in Section 2.20). 9, 38, 39

Aerial Imagery R G B images taken by a camera attached to an aerial vehicle. 9, 17

base registration Datasets collected and maintained by various Dutch governmental organizations. 13, 18

Blender An open-source 3D rendering program that allows for high customization of camera parameters and supports loading of different datasets (see full explanation in Section 2.17). 35

Bright Image Set An image set with increased brightness applied as an exogenous effect. 68

Bundle Adjustment An algorithm that iteratively adjusts observations to better fit a model, based on standard adjustment theory. It is commonly used in aerial imagery with a large number of observations (see full explanation in Section 2.16 and Section 2.20). 34, 50

CaseType A classification category defining different synthetic test scenarios used to analyze the impact of various types of noise or perturbation in the image adjustment process. 48, 49, 64, 91

Cloud Image Set An image set with animated cloud coverage rendered to simulate real atmospheric effects. 56, 68

COLMAP An open-source program used to apply SfM (Structure from Motion) to a collected image set (see full explanation in Section 2.16). 9, 34

Computer Vision A field of study focused on enabling machines to interpret and understand visual information from the world (see full explanation in Section 2.17). 9, 54

Control Points 3D field points with known coordinates, often collected using terrestrial methods, that are used to scale the adjusted observations correctly (see full explanation in Section 2.10). 9, 26

Data-Snooping special form of error detection, in which each observation is tested to see whether it contains a model error. 9, 40

Depth Map A two-dimensional representation where each pixel value encodes the distance from the camera to the corresponding point in the scene. Used to derive depth information for 3D reconstruction, rendering, or object detection purposes (see full explanation in **??**). 9, 51

DIScrete Keypoints (DISK) A deep-learning-based keypoint detection and matching algorithm (see full explanation in Section 2.15). 31, 93, 96

Extrinsic Parameters The camera's position and orientation in space, typically encoded in a transformation matrix (see full explanation in Section 2.5). 9, 22

F-test Also known in Dutch as the variantiefactor, this test evaluates the global fit of a network. It is based on the a-posteriori variance factor from an F-distribution, which depends on the DOF (degrees of freedom) (see full explanation in Section 2.23). 9, 41, 71

First Phase Adjustment Initial free network adjustment phase in aerial triangulation (see full explanation in Section 2.20). 9, 38

Flight Line A 2D path followed by the plane during the collection campaign to ensure proper coverage of the photographed object. 103

Flight Plan A 2D map detailing the predefined Flight Line used during the collection campaign (see full explanation in Section 2.4). 9, 21

Focal Distance Also called "brandpuntsafstand" in Dutch, this is the distance between the camera sensor and the lens, measured in MM (millimeters) (see full explanation in Section 2.8). 25

FocalNoise Artificial noise added to the focal length of the camera, affecting the scale and internal calibration of the imagery. 48, 49, 61

Forward Intersection A geometric method used to derive the 3D location of a point from two or more images with known camera parameters (see full explanation in Section 2.18). 9, 36

framework The synthetic version of the aerial collection pipeline created in this thesis (see full explanation in Section 3.1). 13, 14, 43, 93

Free Network Adjustment An adjustment without any external constraints, often used in the first phase of aerial adjustment (see full explanation in Section 2.20). 38

Global Navigational Satellite System (GNSS) A satellite-based system used to determine precise positioning (see full explanation in Section 2.6). 9, 22

Gruber123Noise Only detecting points within Von Gruber square 1 2 3 (see Section 2.14). 48, 49, 61

Gruber456Noise Only detecting points within Von Gruber square 4 5 6 (see Section 2.14). 48, 49, 61

Gruber789Noise Only detecting points within Von Gruber square 7 8 9 (see Section 2.14). 48, 49, 61

Image Footprint The area on the ground that is visible within a single image frame (see full explanation in Section 2.9). 9, 25

Image Plane A virtual geometric plane located in front of the camera that defines the frustum size and footprint based on focal length and image resolution (see full explanation in Section 2.9). 9, 25

Inertial Motion Unit (IMU) A device that uses accelerometers and gyroscopes to track orientation and motion (see full explanation in Section 2.6). 9, 22

Intrinsic Parameters Camera-specific parameters, such as focal distance and principal point average, used to model the camera in a synthetic environment (see full explanation in Section 2.8). 9, 23

Kappa The κ rotation angle of the plane, represented by the R_z component (see full explanation in Section 2.5). 22, 48, 96

KappaNoise Artificial noise added to the kappa rotation parameter (rotation around the Z-axis), affecting the yaw or heading of the camera. 48, 61

LightGlue An object point matching algorithm used for image matching and structure-from-motion workflows. 31, 57

Monte-Carlo simulation A statistical method using repeated random sampling to model uncertainty and variability (see full explanation in Section 2.24). 9, 42

nadir Nadir aerial imagery is taken at a 0-degree angle along the zenith axis, directed away from the plane towards the photographed mass (see full explanation in Section 2.2). 9, 13–16, 18, 20, 21, 43

NoneNoise A control test case in which no artificial noise is added to the data; used as a baseline for evaluating the effects of other noise types. 48, 61

Normal Image Set An image set with no exogenous effects applied. 56, 68

Object Points 3D points detected using feature extractors, used to link images according to the Von Gruber rules from the HTW (see full explanation in Section 2.11). 9, 27

oblique Oblique aerial imagery is taken at a 45-degree angle in relation to the zenith axis, directed away from the plane towards the photographed mass. It is often taken in multiple directions relative to the sides of the plane (Forward, Backward, Left, Right) (see full explanation in Section 2.3). 9, 19–21, 37, 43

observation A stochastic variable created through measurement, with an associated observational quantity (see full explanation in Section 2.12). 9, 23, 28

Occlusion A situation where an object is hidden or blocked from view in an image (see full explanation in Section 2.19). 9, 37, 43

Omega The ω rotation angle of the plane, represented by the R_x component (see full explanation in Section 2.5). 22, 48, 96

OmegaNoise Artificial noise added to the omega rotation parameter (rotation around the X-axis), affecting the pitch of the camera. 48, 61

Open Graphics Library (OpenGL) OpenGL is an open-source set of instructions that can be used in various programming languages for rendering software and video games (see full explanation in Section 2.17). 35

Phase IA The first and most basic form of the tests associated with the synthetic adjustment the full name of which is Phase IA: Non-Obstructed Synthetic Adjustment and color coded as Green. 9, 46, 60, 64, 70, 92

Phase IB An expansion upon Phase IA through the inclusion of a OpenGL based depth map the full name of which is and Phase IB: Depth Map Based Occlusion for Synthetic Adjustment and color coded as Yellow. 9, 51, 63, 64, 71, 77, 92

Phase II The phase dedicated to the synthetic rendering and object point detection of images the full name of which is Phase II: Synthetic Computer Vision Based Object Point Detection and color coded as Blue. 9, 54, 66, 84, 91, 93

Phase III The final phase dedicated to the use of real life data based on the lessons learned from earlier phases the full name of which is Phase III: Applying Learned Lessons to Real Life Data and color coded as Red. 9, 58, 69, 86

Phi The ϕ rotation angle of the plane, represented by the R_y component (see full explanation in Section 2.5). 22, 48, 96

PhiNoise Artificial noise added to the phi rotation parameter (rotation around the Y-axis), affecting the roll of the camera. 48, 61

pipeline The pipeline describes the steps and choices laid out by the HTW (see full explanation in Section 2.1). 9, 13, 17, 43

PPAxNoise Artificial noise applied to the x-coordinate of the principal point of autocollimation (PPA), affecting the internal geometry of the image sensor. 48, 49, 61

PPAyNoise Artificial noise applied to the y-coordinate of the principal point of autocollimation (PPA), simulating misalignment of the optical axis. 48, 49, 61

Principle Point Average (PPA) A camera parameter that defines the average principal point across images (see full explanation in Section 2.8). 25, 48

Pseudo Constrained Adjustment A hybrid adjustment method with weighted or partial constraints. Used when computing coordinates relative to an existing point field, whose coordinates are to remain unchanged (see full explanation in Section 2.21). 39

Rendering The process of generating photorealistic or non-photorealistic images from 3D models (see full explanation in Section 2.17). 9, 35, 93

S-basis The S-basis in Dutch also referred to as the schrankingsbasis defines the mathematical foundation introduced to eliminate the rank deficiency in the system of unknowns [Polman and ir. M.A. Salzmann, 1996, p. 668]. 38

Scale Invariant Feature Transformation (SIFT) A feature extraction algorithm robust to scale and rotation variations (see full explanation in Section 2.15). 31

Second Phase Adjustment A second phase of adjustment that introduces control points and constraints (see full explanation in Section 2.21). 9, 39

Shadow Image Set An image set with darkened areas simulating shadow as an exogenous effect. 56, 68

Signal Plan A document listing the control points used (see Section 2.10) for the second phase of adjustment (see Section 2.21). See full explanation in Section 2.4. 9, 21, 43

Snow/Rain Image Set An image set with synthetic rain or snow, using detection-invariant colors to disrupt object point detection. 56, 68

Stochastic Errors Random errors that arise from the probabilistic nature of measurement processes (see full explanation in Section 2.7). 9, 23

Synthetic Data Data that is either rendered or mathematically defined, used to enable controlled parameter testing. 9, 15, 43

Von Gruber Squares Spatial distribution rules for object points to ensure sufficient overlap and coverage (see full explanation in Section 2.14). 9, 30, 48, 92

W-test Also known in Dutch as the waarnemingstest, this is a one-dimensional test used to determine the impact of a single variable's error through alternative testing (see full explanation in Section 2.22). 9, 40

Weighted Constrained Adjustment An adjustment method where constraints are applied with varying degrees of confidence. Applied when one wants to preserve the internal geometry (the shape and relative positions) of both point sets as much as possible (see full explanation in Section 2.21). 39

XNoise Artificial noise applied to the X-coordinate of the camera position, simulating lateral displacement errors along the East-West axis. 48, 61

YNoise Artificial noise applied to the Y-coordinate of the camera position, simulating displacement errors along the North-South axis. 48, 61

ZNoise Artificial noise applied to the Z-coordinate of the camera position, simulating altitude or elevation errors. 48, 61

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.