

Cost Optimization of Volumetric Surveillance for Sky Monitoring

Towards Flying Object Detection and Positioning

Hiba Alqaysi

Main supervisor: Mattias O'Nils

Co-supervisors: Najeem Lawal

Faisal Z. Qureshi

Faculty of Science, Technology and Media

Doctoral Thesis in Electronics

Mid Sweden University

Sundsvall, 2022-01-12

Akademisk avhandling som med tillstånd av Mittuniversitetet i Sundsvall framläggs till offentlig för avläggande av licentiatexamen i elektronik **den 12 Januari 2022**, klockan **9:00** i sal **C312**, Mittuniversitetet Sundsvall. Seminariet kommer att hållas på engelska.

Cost Optimization of Volumetric Surveillance for Sky Monitoring

Towards Flying Object Detection and Positioning

©Hiba Alqaysi, 2022

Printed by Mid Sweden University, Sundsvall

ISSN 1652-893X

ISBN 978-91-89341-36-4

Faculty of Science, Technology and Media

Mid Sweden University, Holmgatan 10, SE-851 70 Sundsvall, Sweden

Phone: +46 (0)10 142 80 00

Mid Sweden University Doctoral Thesis 358

To my family.

ABSTRACT

Unlike surface surveillance, volumetric monitoring deals with three-dimensional target space and moving objects within it. In sky monitoring, objects fly within outdoor and often remote volumes, such as wind farms and airport runways. Therefore, multiple cameras should be implemented to monitor these volumes and analyze flying activities.

Due to that, challenges in designing and deploying volumetric surveillance systems for these applications arise. These include configuring the multi-camera node placement, coverage, cost, and the system's ability to detect and position flying objects.

The research in this dissertation focuses on three aspects to optimize volumetric surveillance systems in sky monitoring applications. First, the node placement and coverage should be considered in accordance with the monitoring constraints. Also, the node architecture should be configured to minimize the design cost and maximize the coverage. Last, the system should detect small flying objects with good accuracy.

Placing the multi-camera nodes in a hexagonal pattern while allowing overlap between adjacent nodes optimizes the placement. The inclusion of monitoring constraints like monitoring altitude and detection pixel resolution influences the node design. Furthermore, presented results show that modeling the multi-camera nodes as a cylinder rather than a hemisphere minimizes the cost of each node. The design exploration in this thesis provides a method to minimize the node cost based on defined design constraints. It also maximizes the coverage in terms of the number of square meters per dollar.

Surveillance systems for sky monitoring should be able to detect and position flying objects. Therefore, two new annotated datasets were introduced that can be used for developing in-flight birds detection methods. The datasets were collected by Mid Sweden University at two locations in Denmark. A YOLOv4-based model for birds detection in 4k grayscale videos captured in wind farms is developed. The model overcomes the problem of detecting small objects in dynamic background,

Abstract

and it improves detection accuracy through tiling and temporal information incorporation, compared to the standard YOLOv4 and background subtraction.

SAMMANFATTNING

Till skillnad från videoövervakning av ytor, behandlar volymetrisk övervakning föremål som rör sig inom en tredimensionell volym. I till exempel vindkraftsparker och flygplatser finns ett behov av övervakning av objekt i luften. För kunna övervaka dessa områden behövs flera kameror och analysmetoder för att kunna detektera flygaktiviteter över hela volymerna. Utifrån behovet att gå från ytbaserad övervakning till volymetrisk övervakning kommer ett antal utmaningar med optimering av dessa övervakningssystem. Dessa inkluderar konfiguration av nodplacering av flera kameror, övervakningstäckning, kostnad och systemets förmåga att upptäcka och positionera flygande objekt.

Den här avhandlingen studerar tre delområden för volymetriska övervakningssystem. Först studeras nodens placering och täckning i enlighet med krav för övervakningen. Dessutom bör nodarkitekturen konfigureras för att minimera kostnad och maximera täckningen. Sist bör systemet upptäcka små flygande föremål med god noggrannhet.

Avhandlingen visar att placeringen av multikameranoderna i ett hexagonalt mönster utifrån krav på övervakningen, så som maximal höjd och nödvändig pixelupplösning, ger effektivaste placeringen av kameranoder. Vidare presenteras resultat som visar att modellera kameranoder som en cylinder snarare än en halvsfär minimera kostnaden för varje nod, vilket minimerar kostnaden för täckningen av ett visst område.

Den tredje delen av avhandlingen studerar metoder för att kunna upptäcka flygande föremål. En YOLOv4-baserad modell för att detektera fåglar i högupplöst videomaterial har utvecklats utifrån två annoterade datamängder insamlade på två olika vindkraftsparker i Danmark. Modellen övervinser problemet med att upptäcka små föremål i en dynamisk bakgrund. Den förbättrar detekteringsnoggrannheten jämfört med modeller baserad på standard YOLOv4 och traditionell bildbehandling, genom att dela upp bilden i delbilder samt inkludera temporal information i modellen.

CONTENTS

ABSTRACT	v
SAMMANFATTNING	vii
CONTENTS	viii
LIST OF FIGURES	x
LIST OF TABLES	xi
LIST OF PAPERS	xiii
1 INTRODUCTION	1
1.1 Problem Formulation/Motivation	2
1.2 Research Objectives	3
1.3 Dissertation Outline	7
1.4 Authors Contributions	7
2 THEORY	9
2.1 Volumetric Surveillance	9
2.2 Camera Calibration	12
2.3 Object Positioning	12
2.4 Object Detection	13
3 METHODOLOGY	19
3.1 Node Placement	19
3.2 Node Configuration	22
3.3 Object Detection	25
4 RESULTS AND DISCUSSION	31
4.1 Coverage Evaluation and Node Placement	31
4.2 Node Design Exploration and Cost Optimization . . .	33
4.3 Object Detection	37

Contents

5 CONCLUSION & FUTURE WORK	41
BIBLIOGRAPHY	47

LIST OF FIGURES

1.1	Grid and dome topology	2
1.2	Multi-camera dome coverage	4
2.1	Covered volume of a camera.	10
2.2	Intersecting FoVs between multi-camera domes.	11
3.1	Flying altitude.	20
3.2	Hexagonal placement of camera domes.	21
3.3	Multi-camera dome architecture.	23
3.4	Coverage of multi-camera domes in hexagonal placement.	24
3.5	Optimized cylindrical dome design	26
3.6	Example frames from Skagen and Klim datasets	27
3.7	Bird size statistics of Skagen and Klim datasets.	27
3.8	Three YOLOv4 based models	28
4.1	Pixel resolution plot for six bird paths	32
4.2	Solution space exploration for dome design	35
4.3	Coverage cost for eight configurations	36
4.4	Results of background subtraction for bird detection	37
4.5	Size and number of ground truth and true detected objects using Model 3	39

LIST OF TABLES

- 1.1 Authors contribution. 8
- 3.1 Training parameters for Model 1, 2 and 3. 29
- 4.1 Coverage evaluation of configuration 2 33
- 4.2 Sample results for a 3-layers dome design. 34
- 4.3 Node cost and coverage comparison 36
- 4.4 Accuracy comparison for for Model 1, 2 and 3 38
- 4.5 Testing accuracy using computer vision and deep learning 39

LIST OF PAPERS

This thesis is based on the following publications, herein referred to by their Roman numerals:

PAPER I

Evaluating Coverage Effectiveness of Multi-Camera Domes Placement for Volumetric Surveillance
H. Alqaysi, N. Lawal, I. Fedorov, and M. O’Nils, In *Proceedings of the 11th International Conference on Distributed Smart Cameras*. Stanford,CA, USA: Association for Computing Machinery, 2017
55

PAPER II

Full Coverage Optimization for Multi Camera Dome Placement in Volumetric Monitoring
H. Alqaysi, N. Lawal, I. Fedorov, and M. O’Nils, In *Proceedings of the 12th International Conference on Distributed Smart Cameras*. Eindhoven, Netherlands: Association for Computing Machinery, 2018 63

PAPER III

Design Exploration of Multi-Camera Dome
H. Alqaysi, N. Lawal, I. Fedorov, B. Thörnberg, and M. O’Nils, In *Proceedings of the 13th International Conference on Distributed Smart Cameras*. Trento, Italy: Association for Computing Machinery, 2019 71

PAPER IV

Cost Optimized Design of Multi-Camera Dome for Volumetric Surveillance
H. Alqaysi, N. Lawal, I. Fedorov, B. Thörnberg, and M. O’Nils, In *IEEE Sensors Journal*, vol. 21 , pp. 3730–3737, 2021 79

PAPER V

A Temporal Boosted YOLO-based Model for Birds Detection
around Wind Farms

H. Alqaysi, I. Fedorov, F. Qureshi, and M. O’Nils, In *Journal of
Imaging*, vol. 7, 2021 89

Publication by the author that is not included in the thesis:

Two-Layer 3-D Reconstruction Method and Calibration for Multicamera-
Based Volumetric Positioning and Characterization

I. Fedorov, B. Thörnberg, H. Alqaysi, N. Lawal, and M. O’Nils, In *IEEE
Transactions on Instrumentation and Measurement*, vol. 70 , pp. 1-9, 2021

ACKNOWLEDGEMENTS

I am extremely thankful to my supervisor Prof. Mattias O'Nil for his valuable, ever encouraging, and motivating guidance throughout the past years. I am grateful that he took me on as a student and continued to have faith in me. This work would not have been done without the support of many people. My gratitude is extended to Dr. Najeem Lawal whose constant support since the first step of this research enabled me to develop an understanding of the subject. Many thanks to Prof. Faisal Qureshi for his valuable feedback and encouragement. Also, I would like to thank Dr. Igor Fedorov who was always ready to help with any questions I had.

Many thanks to all my colleagues and the staff members in the Electronics department at Mid Sweden University for their kind support during my PhD study. I extend my thanks to all of my friends, near and far, who endured this long journey with me, always offering support and love. I am also thankful for my fellow PhD students for their kindness, especially Enkeleda and Cristian for always being there for me whenever I needed them.

الشكر موصول الى أحباب قلبي ماما و بابا. لو لا دعمكم الدائم و حبكم
الغير مشروط لما أكملت الطريق. ممتنة لأنكم علمتوني عدم الاستسلام و
شكراً لأنكم دائماً كنتم مشجعي الأول.
كذلك أحبائي أخوتي علي و حسن و أختي الجميلة هالة، شكراً لوجودكم
الى جانبي في كل الأوقات. شكراً لأنكم دائماً تجعلوني أبتسم في الأوقات
الصعبة.

INTRODUCTION

Advancement in smart cameras, image processing, and machine learning have improved surveillance systems drastically. However, this has led to new challenges in optimizing the design and implementation of these systems. Smart cameras are designed to perform specific, high speed, and accuracy tasks, such as machine vision and surveillance, [42]. In addition to capturing an image of the scene, smart cameras perform computer vision and image processing tasks, such as object detection, tracking, and segmentation. This is done through built-in image processing algorithms in the camera processing unite to extract information from images for later decision making, [39].

For complex vision applications, such as volumetric surveillance, multiple cameras should be integrated to monitor a volume. The target space, in this case, is three-dimensional with moving objects. The objects are then detected or positioned depending on the application requirement. Unlike area surveillance, where objects are bounded to a surface, objects move within three-dimensional (3D) space in volumetric surveillance. One specific volumetric surveillance problem is sky monitoring, such as wind farms, airport runways, or border control surveillance.

Deployment and operating costs of monitoring systems in large volumes can be challenging if not properly investigated at the design stage. Issues like the node architecture, pixel resolution at a given distance, and coverage should be well defined during that stage. Multi-camera systems in volumetric surveillance provide coverage across a wide area, ensuring the object's visibility for detecting and tracking. Deployment of multiple cameras system may include grid or dome topology, [24]. The grid topology, shown in Fig 1.1 (left) divides the monitored volume into segments covered by fixed nodes with a specified field of view (FoV). The drawback of this placement is that it requires a high number of nodes and consequently high deployment and maintenance costs.

On the other hand, the dome topology, Fig 1.1 (right), groups

Chapter 1. Introduction

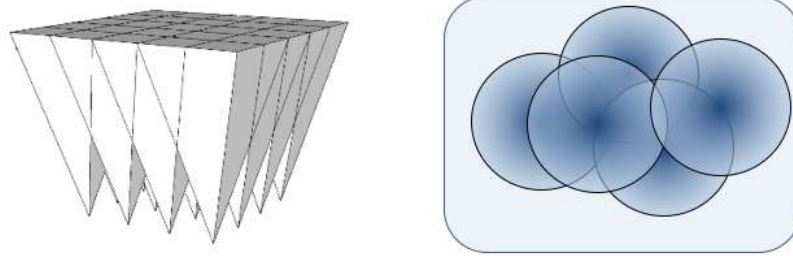


Figure 1.1: (left) Grid and, (right) Dome topology.

number of cameras such that their field of views forms a hemisphere around its center. This design reduces the number of visual sensor nodes required to monitor a given volume and reduces implementation and maintenance costs.

To develop a vision system for sky monitoring, flying objects should be detected and localized. Therefore, in order to cost optimize a volumetric surveillance system, three main aspects should be considered. The multi-camera node architecture, its placement, and the system's ability to detect and position flying objects within the volume should be investigated.

1.1 Problem Formulation/Motivation

Traditional sky surveillance systems like radar systems [29] [40] are expensive and complex to deploy in large open volumes like in sky monitoring. For example, researchers in [45] concluded that radar systems designed to monitor polar activities in the higher altitude of the sky require constant changes in operating frequencies and elevation angles in order to maintain constant detection of the polar activity. Furthermore, the use of four radar systems to monitor birds' nightly migration patterns in southern Sweden was investigated in [33]. The BirdScan radar was classified as a low altitude system to be used in applications such as collision risks with man-made structures. The maximum detection range for small birds under short-pulse of this radar does not exceed 800 meters. The data collected from

1.2. Research Objectives

radar systems are often inconclusive when flying object size becomes smaller and when many objects interact.

Another challenge in sky monitoring is that the monitored volume is often large and remote. For example, in onshore wind farms, the distance between turbines is normally five times the rotor diameters in order to avoid the turbulence effect [49]. So for a 500 kW wind turbine, the distance among them in a farm is 250 meters and 410 meters for a 2.5 MW turbine. Clearly, wind farm facilities require a massive piece of land. Surveillance systems implemented in confined spaces are not scalable for such application. Therefore, arranging multiple cameras in one node that covers a 2π radian around its center, Fig. 1.2, is one way to address reducing the deployment cost over the large volume.

Furthermore, wind farms installations often interfere with birds' migration paths where it can lead to their collision with the turbine blades [43]. It is estimated that up to 500,000 birds collide with wind turbines per year in the U.S. [48]. Therefore, there is a need to capture the impact of wind farms on birds' habitats and their migration patterns. The surveillance system's ability to correctly detect and position flying birds around turbines can influence the design of collision avoidance systems in wind farms, [2] [3]. Also, accurate birds characterization and counting are required in order to predict birds migration changes [1].

Therefore, surveillance systems for sky monitoring applications should consider the challenges of having big and often remote volumes. The background in these applications is mostly dynamic in terms of clouds movement and illumination variation. Also, the system requirements of pixel resolution and its ability to detect small flying objects should be considered.

1.2 Research Objectives

The main goal of this dissertation is to optimize volumetric surveillance systems to enable sky monitoring. This includes optimizing the multi-camera node placement, configuration, and object detection. This leads to the following three research questions and their contributions:

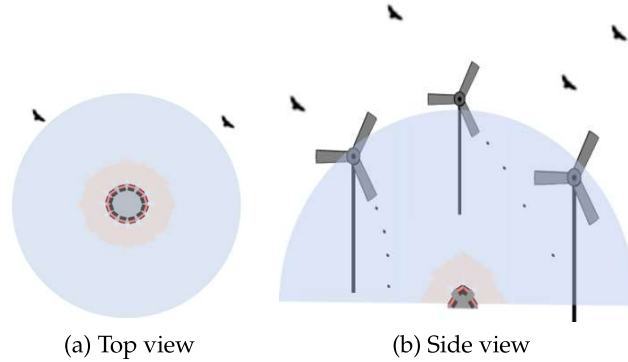


Figure 1.2: Multi-camera dome coverage.

1.2.1 Optimizing Multi-camera Dome Placement and Coverage Evaluation

- **R.Q.1** How can the multi-camera dome placement be optimized to ensure full coverage while fulfilling detection requirements of objects flying within the volume?

Optimize Multi-camera Dome Placement Most of the work conducted in the literature to optimize camera coverage and placement considers limited indoor areas. Monitoring a volume includes freely moving objects at a specific altitude within it. This makes many of the camera placement and coverage strategies in area monitoring inapplicable for such applications. For example, extending optimal 2D placement, such as the art gallery and sphere packing, leads to NP-Hard complexity in three-dimensional deployment [36].

Contribution: In [Paper2], an optimized placement method of the multi-camera domes in 3D environments is developed. The goal is to reduce the number of nodes implemented while achieving full coverage at a specified flying altitude. Objects flying within the hemisphere are detected when the requirement for pixel resolution is fulfilled. The domes are modeled in the volume by adapting the hexagonal packing of circles. The evaluation method in [Paper1] is used to assess different placement configurations.

1.2.2. Configuring the Multi-camera Dome Design

Evaluate Multi-camera Dome Coverage Node placement in designing monitoring systems determines the size and shape of its coverage. In volumetric surveillance, three-dimensional coverage of the space is required. This means the altitude at which objects are flying should be considered. A method to measure node coverage effectiveness in volumetric surveillance is necessary.

Contribution: The developed method in [Paper1] measures the degree to which a system fulfills its monitoring objectives. The analysis is based on three criteria, detection, positioning, and pixel resolution to detect the object. The method is tested using GPS data of flying birds, and it can be used to evaluate dome placement algorithms for volumetric monitoring systems.

1.2.2 Configuring the Multi-camera Dome Design

- **R.Q.2** From a set of camera sensors and objective lenses, how to configure the design of the multi-camera dome, subjected to the defined monitoring constraints? and how can its cost be optimized?

Design Exploration of the Multi-camera Dome The choice of cameras sensor and lens and their deployment affects the design cost, the accuracy of the monitoring system, and the ability to position objects. The node design with respect to camera sensors and lenses and the trade-offs between design constraints and monitoring objectives should be investigated. The defined set of constraints includes the volume to be monitored, the number of cameras that can be integrated in a node, and pixel resolution at a given distance.

Contribution: A design exploration method was developed in [Paper3] for optimizing the cost of a multi-camera dome. The design exploration minimizes a defined cost function and is subjected to constraints, such as pixel resolution and the number of cameras in the node. Analyzing trade-offs among the design constraints allow, for example, increasing coverage for a given cost.

Cost Optimization of the Multi-camera Dome In sky surveillance, objects fly within the 3D volume. The flying altitude should be considered when

Chapter 1. Introduction

the node design is configured. Volumes above the defined flying altitude are not required to be covered, which counts as a coverage overhead. Also, the altitude constraints the overlapping areas between adjacent domes. Therefore the multi-camera dome can be redesigned.

Contribution: In [Paper4] a cost-optimized design for the multi-camera dome that maximizes its coverage is developed. The proposed design is cost-optimized per node and provides more coverage as compared to the hemispherical multi-camera dome. The comparison is made based on the cost per node and number of covered square meters per dollar.

1.2.3 Improve Flying Objects Detection Accuracy

- **R.Q.3** For the problem of sky monitoring, how to detect small flying objects?

Object Detection Object detection in sky surveillance can be challenging due to having small flying objects in a large volume and constantly changing background, which requires high-resolution frames. For example, detecting birds flying in wind farms to prevent their collision with the wind turbines. Object detection methods are either traditional computer vision-based or deep learning-based.

Contribution: The performance of traditional computer vision algorithm against deep neural networks for the problem of detecting flying birds around wind turbines was investigated in [Paper5]. A YOLOv4 based model that uses tiling and temporal features of 4k video frames was proposed. The model's validation and testing accuracy were improved compared to the standard YOLOv4 and background subtraction. Since deep learning requires big datasets for training, validation, and testing, two datasets collected by Mid Sweden University in Denmark for that purpose were introduced in [Datasets].

Moreover, the deep learning-based detection model can be used to build a multiple object tracker and construct object trajectory in 3D. This will be further investigated as future work.

1.3 Dissertation Outline

The next chapters are organized as follows. Chapter 2 has the implemented literature investigated and studied in depth. The theoretical aspect of volumetric surveillance, multi-camera dome, and camera calibration is discussed. In addition, deep learning-based object detection is reviewed. Chapter 3 introduces the methodology used to investigate each of the research questions in terms of optimizing the multi-camera node placement and configuration. It also presents the datasets used to develop the object detection models. Chapter 4 explains and discusses the research findings in relation to the research objectives. Last, chapter 5 summarises the conclusion of this work and the next step forward to continue as future work.

1.4 Authors Contributions

The following table is the authors contribution for each publication included in the thesis:

H. A. - Hiba Alqaysi
N. L. - Najeem Lawal
I. F. - Igor Fedorov
B. T. - Benny Thunberg
F. Z. Q. - Faisal Qureshi
M. O'N. - Mattias O'Nils

Contribution	Paper I	Paper II	Paper III	Paper IV	Paper V
Conceptualizing the idea	M. O'N and H. A.	M. O'N. and H. A.	M. O'N. and H. A.	M. O'N. and H. A.	M. O'N. and H. A.
Method development	H. A.	H. A.	H. A.	H. A.	H. A.
Implementation	H. A.	H. A.	H. A. and N. L.	H. A. and N. L.	H. A.
Data acquisition	H. A.	H. A.	H. A.	H. A.	H. A. and I. F.
Results analysis	H. A.	H. A.	H. A.	H. A.	H. A.
Writing the original draft	H. A.	H. A.	H. A.	H. A.	H. A.
Editing and revising the manuscript	H. A., N. L., and M. O'N.	H. A., N. L., and M. O'N.	H. A., N. L., B. T., and M. O'N.	H. A., N. L., B. T., and M. O'N.	H. A., F. Z. Q., and M. O'N.
Supervised the work	N. L. and M. O'N.	N. L. and M. O'N.	N. L. and M. O'N.	N. L. and M. O'N.	F. Z. Q. and M. O'N.

Table 1.1: Authors contribution.

THEORY

This chapter introduces the theoretical base for the work done in the thesis. The concept of volumetric surveillance is presented, with relation to the multi-camera dome. Overlapping between domes means the possibility to position objects, therefore, camera calibration and object positioning is discussed. Lastly, object detection using deep learning methods is reviewed.

2.1 Volumetric Surveillance

Triple integrals can be used to calculate a volume [46]. If it is assumed that a camera is placed at the origin of a spherical coordinate system, as shown in Fig. 2.1, and that its coverage radius is r , where the pixel resolution requirement, δ , is met. Then the maximum (ideal) volume, V , the camera monitors can be calculated using triple integrals as:

$$V = \frac{2r^3}{3} FoV_h \cdot \cos \frac{\pi - FoV_v}{2} \quad (2.1)$$

where FoV_h and FoV_v are the camera's horizontal and vertical angles of view, respectively. The coverage radius, r , that fulfills the minimum resolution constraint, δ , pixels per meter of a volume at that distance can be expressed as:

$$r = \frac{P_h}{\delta \cdot FoV_h} \quad (2.2)$$

where P_h is the number of pixels horizontally in the camera sensor. Therefore, the maximum number of square meters surveilled at height h for a specific camera sensor and objective lens combination will be:

$$A(h) = \frac{2P_h^3}{3h \cdot \delta^3 \cdot FoV_h^2} \cdot \cos \frac{\pi - FoV_v}{2} \quad (2.3)$$

This is an ideal volume calculation that in order to fully monitor a volume it requires multiple cameras to hoover in the middle, which is unrealistic for most applications. However, it sets a baseline when analyzing methods for implementing volumetric surveillance.

Chapter 2. Theory

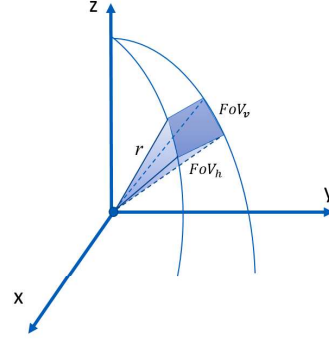


Figure 2.1: Covered volume of a camera.

2.1.1 Multi-camera Dome

A multi-camera dome composes of several cameras that are integrated into one node and are not necessarily of the same type. The cameras are arranged such that a hemisphere around the dome center is monitored. Each camera is a combination of a sensor and an objective lens. Because each camera-lens combination forms a discrete FoV, multiple cameras are arranged to cover a 2π radian view, Fig. 2.2. Cameras' FoV are set up with minimum overlaps to ensure full coverage and avoid the blind spots in the covered volume. The multi-camera dome is introduced in the literature, [25], [7], as an efficient implementation for volumetric surveillance.

The dome architecture results in two cost reduction aspects, compared to the grid deployment of cameras :

- 1) Multiple cameras are combined in one node to surveil a volume with a minimum overlap between them.
- 2) The cost of deployment, C , is reduced since several cameras are integrated into one node.

$$C = c_d + \sum_{i=1}^n c_c$$

Where c_d is the deployment cost of one node and the sum of c_c is the cost of n camera sensors that compose the node. The deployment cost

2.1.1. Multi-camera Dome

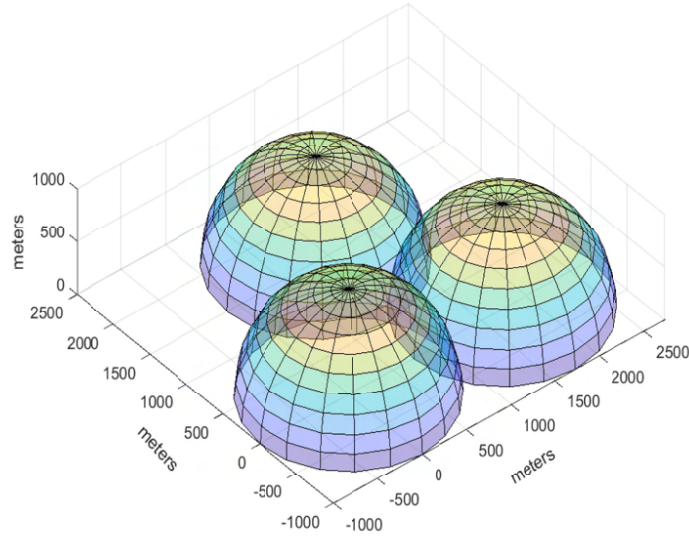


Figure 2.2: Intersecting FoVs between multi-camera domes.

is the same for all the cameras in a node instead of having to deploy each camera separately.

The dome capacity, DC , as defined in [25], is obtained from the dome coverage radius, r , multiplied by the minimum pixel resolution, p , required for detection at that distance. For example, if the coverage radius of the dome is 500 meters and the required minimum detection resolution is 10 pixels at 500 meters, then the dome capacity is 5000 pixels. Therefore it can be defined as:

$$DC = r \times p \quad (2.4)$$

where r is the coverage radius of the dome and p is the pixel resolution required for detection.

Fully monitoring large open volumes requires implementing more than one camera dome. When the coverage of adjacent nodes overlaps as in Fig. 2.2, then it is possible to extract objects' position in 3D. The cameras however should be calibrated in order to do that.

Chapter 2. Theory

2.2 Camera Calibration

The process of estimating the camera parameters is called camera calibration [17]. These parameters are extrinsic, intrinsic, and distortion coefficients. To estimate the camera parameters, 3D world points and their corresponding 2D image points should be known. The correspondences can be obtained using multiple images of a calibration pattern, such as a checkerboard, [34].

Extrinsic parameters define the location and orientation of the camera in the 3D world in terms of rotation and translation. Intrinsics are the internal parameters of the camera that are used to map between camera coordinates and pixel coordinates in the image frame, and vice versa. They include the focal length in the x- and y-axis and the principal point coordinates in pixels. The ideal pinhole camera model in theory does not have a lens, so it means there is no lens distortion. Therefore the parameters for the pinhole camera model are represented in the following matrix:

$$P = K[R|t] \quad (2.5)$$

where R is the rotation matrix and t is the translation vector, the two combined represent the extrinsic matrix; and K is described as:

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

where f_x and f_y are the focal length of the camera in the x-axis and the y-axis respectively, (c_x, c_y) is coordinates of the camera's principal point. K is called camera calibration matrix.

2.3 Object Positioning

There are three coordinate systems in the forward projection model that transforms a 3D world point into a 2D image point. The coordinates are world, camera, and image. The 3D world points are transformed to 3D camera coordinates using the extrinsic parameters. That is rotation and translation. Then camera coordinates are mapped

2.4. Object Detection

into the image 2D plane using the intrinsic parameters. The transformation between 3D world points and 2D image points is calculated using the following projection matrix:

$$x = PX \quad (2.7)$$

The 3D position of an object can be obtained through back-projection of the image points to the world, given the cameras are calibrated. A point is back-projected to the world as a ray. The intersection of the two rays corresponding to the same point from the two camera frames represents the object position in 3D, [Paper6].

2.4 Object Detection

A challenging object detection problem is sky surveillance where flying objects in a volume should be accurately detected. This may include monitoring airports runways, wind farms, or drones activities. Object sizes are normally small in a large open volume and the background changes in terms of illumination and clouds movement. Object detection combines classifying (assigning class labels) and localizing (identifying location) objects of interest in an image [53]. For decades, the main tool for the problem of object detection has been the traditional computer vision algorithms. These well-known algorithms include background subtraction, thresholding, salient detection, etc. Such algorithms rely on handcrafted features, such as corners, shapes, and colors which have to be identified by a human. Therefore the performance of computer vision algorithms depends on how accurately the features were identified and extracted.

As deep learning and the use of Convolutional Neural Networks (CNNs) emerged, [26], different architectures were introduced for object detection. Models were able to learn more complex features and predict objects with higher accuracy. Deep learning replaced handcrafted features with efficient algorithms for unsupervised or supervised feature learning and extraction [9], [44], [5], as opposed to doing that manually. However, it requires massive datasets in order to train, validate, and test models effectively and high computational power.

Chapter 2. Theory

2.4.1 Convolutional Neural Networks

The architecture of CNNs is different than the standard Neural Network. In Neural Networks, an input vector goes through a series of hidden layers [13]. Every layer consists of a set of neurons, and is fully connected to all neurons in the next layer. These layers are called fully connected layers. The last fully-connected layer in the network is the one that makes the predictions.

Convolutional Neural Networks are different in the way neurons are connected. The neurons in one layer are not connected to all the neurons in the next layer, which decreases the number of parameters in the layer. These layers are called convolutional layers, and a CNN may have one or more convolutional layers [13]. For image analysis such as feature extraction, CNNs have proven big success when it comes to accuracy. Images are passed to the network as a matrix of width, height, and depth so its spatial structure and pixel connectivity are preserved [27]. The depth of an image is the color channels. For example, an RGB image has a depth of 3, and a greyscale image has a depth of 1. Layers in CNN can be organised in 3 dimensions: *width*, *height*, and *depth*.

The building block of CNN is the convolutional layers [28]. These layers detect patterns in an image by applying convolution of a learnable filter (kernel) with the image pixels. When the input image is an RGB image with three channels, the filters should also be 3D. Convolution is the dot product of the filter and the corresponding spatial pixels. As a result, the network learns low-level features such as edges and lines in early layers then a high-level representation of the object such as faces in deeper layers. This reduces the tasks of developing feature extractors.

2.4.2 Deep Learning based Object Detectors

Deep learning-based object detectors can be one of two categories: one- or two-stage detectors [22]. Two-stage detectors generate predictions based on region proposals from selective search in the first stage. The selective search creates candidate bounding boxes that are then fed to a CNN for feature extraction and classification as a second stage

2.4.2. Deep Learning based Object Detectors

[11], [16]. Region-based Convolutional Neural Network detectors such as R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN are two-stage models [52]. Models from the R-CNN family have high accuracy, for example, Faster R-CNN and R-FCN achieved mean Average Precision (mAP) of 73.8% and 77.6% on the PASCAL VOC 2012 dataset, respectively [21], [8]. However, they are time-consuming because of the large number of region proposals. In the first R-CNN, 2000 region proposals per image are created and forwarded to the CNN [15].

On the other hand, single-stage detectors such as YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), and RetinaNet treat the object detection as a regression problem. This means the algorithm learns the bounding boxes coordinates and class probability in an input image directly without region proposals. Thus, they are time efficient and can be used for real-time. YOLOv3 achieves an accuracy of 57.9% on PASCAL VOC 2012 for resolution of 544×544 , [38], using Darknet-53 backbone, [10]. While SSD is faster than YOLO, it is more suitable for detecting large objects and it performs badly in detecting small objects. The multi-scale detection that was presented in YOLOv3 improves its accuracy in detecting small objects. There is always a trade-off between accuracy and speed in all these detectors, the right detector depends on the application requirements.

Small Objects Detection In object detection literature, objects are considered small when their mean relative overlap, which is the overlap area between the object bounding box and the image, is between 0.08% to 0.58%, respectively [6]. When objects are in the range of 10% - 20% of the image, then SSD gives better results than different versions of YOLO [35]. If objects are less than 10% of the image, then YOLO performs better than SSD.

Single-stage detectors can run in real-time as they are much faster in inference compared to two-stage detectors. If the application target is a balance of accuracy and speed, YOLO is the choice [32]. YOLOv2 does not work well with small objects, on the other hand, YOLOv3 develops a deeper network with 53 layers (Darknet-53) which improves detecting small objects.

Chapter 2. Theory

YOLO The YOLO family models (v1,2,3) are end-to-end deep learning models developed by J. Redmon *et al.* for object detection [37]. This architecture divides an input image into $s \times s$ grid cells and applies a single forward pass to the neural network. Each grid cell predicts B bounding boxes and their confidence score C , which represents the probability of that box contains an object. Next, each cell also predicts a class probability of the object. Multiple bounding boxes are predicted for the same object, they get thresholded by the confidence score. Each cell is responsible for detecting one object in the image, where its center falls into. Each bounding box consists of the 5 components: $(x, y, w, h, confidence)$ where (x,y) coordinates represent the center of the box, (w,h) are its height and width. The output prediction is a tensor as: $S \times S \times (B \times 5 + C)$.

Unlike YOLOv2 which predicts the output at the last layer, YOLOv3 predicts boxes at 3 different scales in order to detect objects of various sizes. This is done by downsampling the dimensions of the input image by 32, 16, and 8 respectively. For example, an input image of size 416×416 gets downsampled into grids of 13×13 , 26×26 , and 52×52 .

YOLOv4 improves the accuracy by 10% compared to YOLOv3, using the MS COCO dataset, [4]. YOLOv4 comprises of a backbone network, a neck network, and a head network. It uses the novel CSPDarknet-53 as the backbone network which enhances the learning capability. Darknet-53 is 53 convolutional layers deep and uses Spatial Pyramid Pooling (SPP) for feature extraction [19]. Path Aggregation Network (PANet) serves as the neck network [30]. It collects features from different layers of the backbone network. YOLOv3 is used as the head network for YOLOv4.

YOLO detectors deploy the concept of anchor boxes. These boxes are estimated using K-mean clustering. The ground truth bounding boxes in the training dataset is classified into nine clusters. Each detection scale is assigned three anchor sizes for large, medium, and small objects. Every grid cell predicts three boxes using these three anchors. This means having a total of 10,647 predictions for each input image $((13 \times 13 \times 3) + (26 \times 26 \times 3) + (52 \times 52 \times 3))$.

The predicted boxes are thresholded first using the confidence threshold, C . Then, the intersection over union (IoU) with the ground

2.4.3. Detection Evaluation

truth bounding box is calculated. If the IoU is greater than the IoU threshold, the anchor box should detect the object for the learning process otherwise it does not learn from that example. Therefore, anchors are important parameters to tweak based on the object sizes in the annotated training dataset.

2.4.3 Detection Evaluation

To determine whether a prediction is correct or not, Intersection over Union (IoU) is used. It is defined as the ratio of the overlap between the predicted and ground truth bounding box to the union of the two boxes. The objectness (confidence) score is the network's confidence that an object exists in the given box. A prediction is a True Positive (TP) if its objectness score is greater than or equal to some confidence threshold, the predicted class matches the class of the ground truth, and the IoU with ground truth is greater than or equal to the defined IoU threshold. A prediction is a False Positive (FP) if either of the latter two conditions is not true.

The most common evaluation metric used for the deep learning model is mean Average Precision (mAP) [20]. Specifically, the Pascal VOC 2010–2012 [14], which samples a curve at all precision and recall values. The mAP is then calculated as the exact area under the precision-recall curve (AUC).

Precision is the percentage of TF among all predictions, and recall is the percentage of TF among the ground truths. The mAP metric in the Pascal VOC 2010–2012 interpolates all the points to calculate the AUC of the precision-recall curve. By default, the IoU threshold in this calculation is 0.5. Mathematically,

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

where FP and FN are false positives and false negatives, respectively.

METHODOLOGY

This chapter discusses the implemented methods used to conduct the research. This includes the dome placement and coverage in a volume. Also, a design exploration method that minimizes its cost is introduced. The placement and monitoring requirements influence changes to the node design. The coverage can be modeled as a cylinder instead of a hemisphere to achieve maximum coverage per cost. Moreover, in-flight objects detection methods are investigated and implemented using two new annotated datasets.

3.1 Node Placement

The work conducted to optimize the multi-camera dome placement included optimizing its placement in order to achieve full coverage, [Paper2]. In addition, developing a method to evaluate node placement in terms of detection, positioning, and maximum pixel resolution captured to detect flying objects within the volume [Paper1].

3.1.1 Placement Optimization

The combined FOVs of the cameras in the dome form a hemisphere whose base is a circle. Domes are assumed to have the same coverage radius, r . Packing circles in 2D could be in a square or hexagonal pattern. Hexagonal tiling is the optimal form and it is 12% more dense [41], [12]. However, circles placed in this pattern do not intersect, which means having areas that are not monitored. In order to cover these areas, domes coverage should overlap. Arranging circles in a *thinnest hexagonal pattern* allow overlap to fully cover a Euclidean 2D plane and minimize the overlapping size, [23].

This placement provides full coverage on the ground level. However, in 3D volume, intersection points of every three adjacent non-collinear domes are blind spots with zero coverage at the flying altitude as shown in Fig. 3.2a. Therefore, domes should be re-positioned in order to provide full coverage at the defined altitude. Fig. 3.1 shows

Chapter 3. Methodology

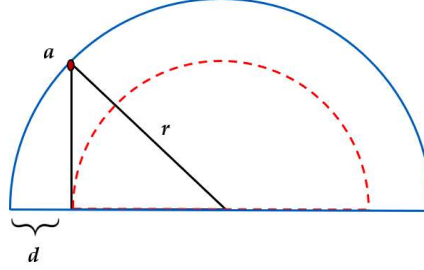


Figure 3.1: Flying altitude.

that the flying altitude a can be used to calculate the distance by which does centers should be re-positioned, d .

In the thinnest hexagonal placement, every three adjacent non-collinear domes form an equilateral triangle. Its side, s , is the distance between the centers of the two adjacent domes, Fig. 3.2a. That distance equals to $2\sqrt{(r^2) - (h^2)}$ and its altitude, t , equals to $\frac{\sqrt{3}}{2} \times s$.

In order to cover the blind spots while keeping the thinnest hexagonal pattern, the altitude t should be reduced by d in Fig. 3.1. Therefore, the new triangle side, s' , which represents the distance between domes center is equal to:

$$s' = s - 2d/\sqrt{3} \quad (3.1)$$

Re-positioning the domes ensures full coverage of the volume at altitude a , including intersection points as shown in Fig. 3.2b. It also changes the minimum overlapping height, h , between adjacent domes that ensure 100% detection at flying altitude a . This height can be calculated as the apothem of the equilateral triangle, which equals to:

$$h = t/3 \quad (3.2)$$

3.1.2 Placement Evaluation

The optimized placement method discussed in Sec. 3.1.1 ensures 100% detection for objects flying within the defined altitude when their pixel resolution is equal to or greater than the required detection

3.1.2. Placement Evaluation

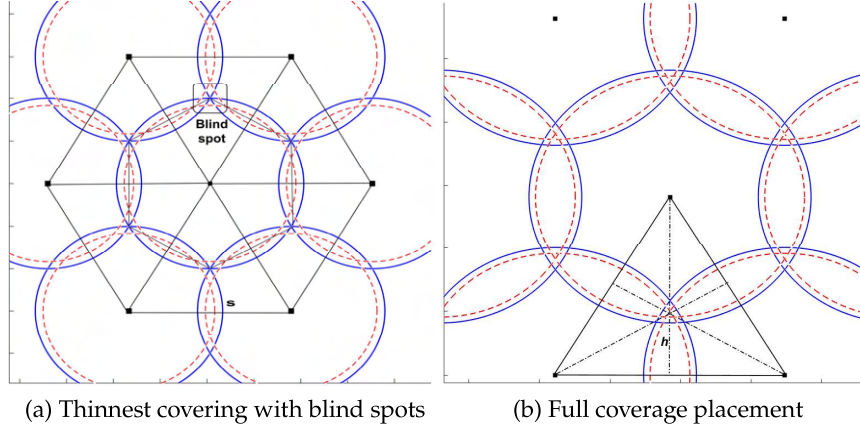


Figure 3.2: Hexagonal placement of camera domes.

resolution. Overlapping coverage between camera domes allows extracting object stereoscopic information such as size and position. These criteria should be measured in order to assess the multi-camera placement.

The method developed in [Paper1] evaluates the coverage effectiveness of each dome placement in volumetric surveillance. Three monitoring objectives are defined and measured using GPS trajectories of six birds over an area of 9 km^2 . The tracks are obtained from the work in [18]. The defined criteria are detection and positioning percentage of the flying object; as well as the maximum pixel resolution captured to detect it.

The required minimum detection resolution and the dome coverage radius define the dome capacity as in Eq. 2.4. Each track composes of a number of samples representing the coordinates in longitude and latitude, and the altitude is assumed to be 200 meters for all tracks. These coordinates are converted to a Euclidean distance-vector corresponding to each track. The dome capacity is then used to calculate the pixel resolution of each sample in the distance vector.

Detection percentage, is calculated for each tack when its pixel resolution is higher than the required detection resolution. The defined resolution is 3 pixels per meter (pix/m). Whenever this resolution is fulfilled for the object flying over any of the domes, it

Chapter 3. Methodology

means the object is detected. The placement method in 3.1.1 ensures 100% detection for the six used trajectories over the defined 9 km^2 volume. Also, when the object is detected in two domes, its position can be extracted. The *positioning percentage* measures that when the birds fly within overlapping areas. Moreover, the pixel resolution to detect the object changes as it flies closer or further to the dome center. The *maximum pixel resolution* gives an indication of that.

3.2 Node Configuration

Optimizing the surveillance system includes the node design configuration at its core. To achieve that, two factors are considered; the cost of the multi-camera dome, [Paper3], and its design, [Paper4].

3.2.1 Node Design Exploration

The multi-camera dome is assumed to consist of multiple layers vertically, and each layer consists of a number of cameras at the horizontal level as in Fig. 3.3. The design exploration in [Paper3] constructs a cost-efficient camera dome, subject to monitoring constraints, from a set of camera sensors and lenses combination. The combined horizontal angles of the cameras in one layer should cover a minimum of 360° . The vertical angles of all the layers combined should cover a minimum of 90° to ensure monitoring a 2π radian view. Cameras in one layer are identical; however, it is not necessary that all layers have similar cameras.

For the design exploration, a number of constraints are defined. They include the dome coverage radius, r , the minimum pixel resolution at that distance, and the number of layers in the node. The method searches through available sensors and lenses to combine them such that the diameter of the covering lens must be equal to or larger than the sensor size. Then, an exhaustive search algorithm is performed using the short-listed combinations. Each sensor-lens pair should fulfill pixel resolution constraints at the specified dome radius. The solution with the minimum cost is chosen as optimal. The search for an optimal solution can be constrained by cost, the number of cameras, or other parameters. The optimal solution has

3.2.1. Node Design Exploration

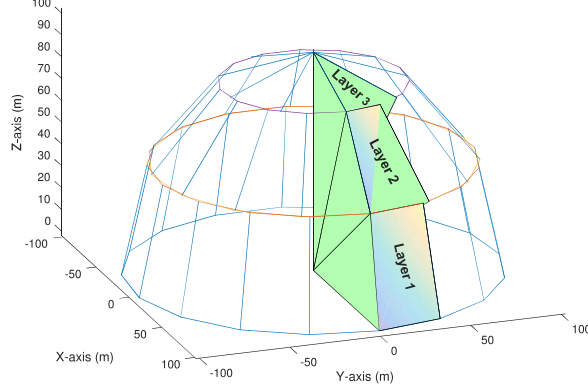


Figure 3.3: Multi-camera dome architecture.

the following properties, the angle covered by all the layers vertically, the total number of cameras in the node, the number of cameras in each layer, and the solution cost.

The method estimates the cost of each member, S_{ijn} , of the solution space, S , subject to the defined design constraints. The objective function for the design exploration can be defined as minimizing the following cost function:

$$\begin{aligned}
 & \underset{\tilde{C}_{ijn}}{\text{minimize}} && f(S) \\
 & \text{subject to} && d_i^c \leq d_j^o, \\
 & && M_{ijn} \leq M, \\
 & && N \leq L, \\
 & && \delta_{rijn} \geq \delta, \\
 & && A_{ijn} \geq \frac{\pi}{2} + \varepsilon \cdot (N - 1)
 \end{aligned} \tag{3.3}$$

where d_i^c and d_j^o are the diameter of the sensor and the covering lens, respectively, M_{ijn} is the number of cameras required by the solution. A_{ij} is the total vertical angle covered by all layers, N , and δ_{rijn} is the pixel resolution in the optical plane of the FoV for the combination of camera i and lens j at a distance r from the dome for each layer n in the solution.

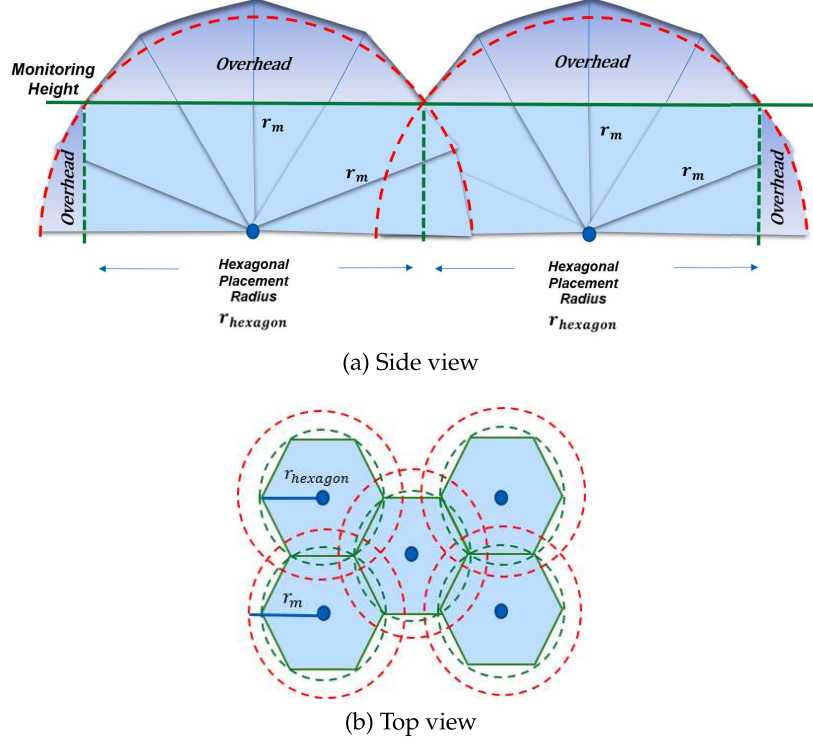


Figure 3.4: Coverage of multi-camera domes in hexagonal placement.

3.2.2 Node Cost Optimization

The placement method in Sec. 3.1.1 motivates the inclusion of the placement in the node design. In the hexagonal placement, domes overlap in order to achieve full coverage. Volumes above the monitoring height are not required to be monitored. This results in coverage overhead as shown in Fig. 3.4a. Coverage of each dome without overlapping volumes in the hexagonal pattern can be modeled as a hexagon that has a coverage radius of $r_{hexagon}$, Fig. 3.4b.

Taking these points into consideration leads to node design changes to minimize coverage overhead in the hexagonal placement. Therefore, in [Paper4] the hemispherical design of the node is optimized accordingly. The monitored volume for each dome is modeled as a cylinder instead of a hemisphere, Fig. 3.5b. Non-overlapping

3.3. Object Detection

volumes have a minimum coverage radius equals the monitoring height. Overlapping volumes coverage radius is bounded by the cylinder radius, r_c . The resulting cylindrical dome is then divided into layers, and each of them has its coverage radius. Monitoring requirements should be met at that distance. At the top layer, cameras are stacked with multiple overlaps at the horizontal level. This also causes coverage overhead. Therefore, only one camera can be mounted at the top layer, Fig. 3.5a. That layer can be considered as a half layer when arranging the layers vertically from the bottom up to cover a minimum of 90° . The minimum coverage radius of each layer is defined based on the monitoring height, h .

$$r_L = \begin{cases} r_c / \cos(c_A), & \text{if } x \ \& \ y < h. \\ r_m, & \text{if } x > h \ \& \ y < h. \\ h / \sin(p_A), & \text{if } x \ \& \ y > h. \end{cases} \quad (3.4)$$

where x and y are height of the current and previous layer respectively, as shown in Fig. 3.5b for layer 2. c_A is the covered vertical angle of the current and previous layers. p_A is the vertical angle of view of the previous layer. The vertical angle of view for each layer is calculated as:

$$v_A = 2 \arctan(d_v / 2f) \quad (3.5)$$

where d_v is the sensor's vertical dimension and f is the lens focal length. Each layer's height is calculated as:

$$h_L = |r_m \cdot \sin(c_A)| \quad (3.6)$$

3.3 Object Detection

For the problem of sky monitoring in wind farms, object detection is an essential surveillance task. To investigate methods for in-flight birds detection, the performance of state of the art deep learning YOLOv4, [4], is compared to computer vision-based detection using background subtraction. An improved YOLO-based model is developed in [Paper5], and it achieves detection accuracy on testing sets of 90% and 92% using the datasets in [Datasets].

Chapter 3. Methodology

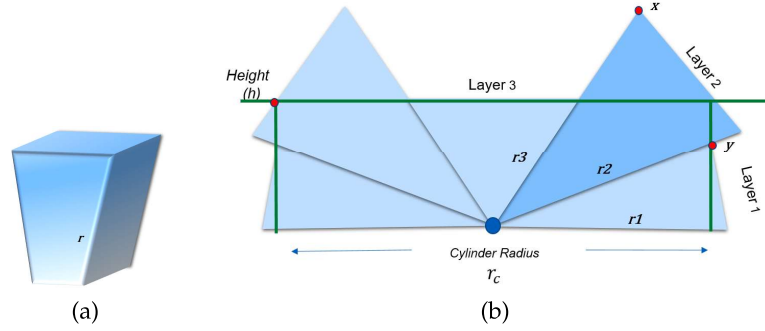


Figure 3.5: (a) Camera's field of view in 3D, (b) Optimized cylindrical dome design with one camera on the top layer.

3.3.1 Deep Learning based Object Detection

Dataset Acquisition Object detection using deep learning requires annotated datasets for training, validation, and testing. There were no available datasets for in-flight birds that could be used for this study. Therefore, datasets were collected by Mid Sweden University in two wind farms in Denmark over a few months in 2017 and 2018. The datasets were manually annotated using the open-source annotation tool *LabelImg*, [47].

The Skagen dataset was collected at the *Skagen Grey Lighthouse Center of Migratory Birds* using a pair of wide-angle monochrome cameras fixed inside rigid boxes. The cameras were connected to Nvidia Jetson TX2 edge-computing device recording 4K images at 5 *fps*. This dataset has a relatively stationary background in terms of clouds movement and illumination, shown in Fig. 3.6 (left). The majority of objects sizes in the Skagen dataset are less than 200 pixels, Fig. 3.7a, which are considered very small. This dataset was used mainly to test the deep learning models rather than for training.

The second annotated dataset is Kilm, and it was collected at *Klim Fjordeholme* using the same camera setup; except here the cameras were mounted on tripods. This dataset exhibits a more dynamic background of moving clouds and variant illumination, Fig. 3.6 (middle). Objects sizes vary in this dataset as shown in Fig. 3.7b,

3.3.1. Deep Learning based Object Detection



Figure 3.6: (left) Frames from Skagen dataset, (middle) from Klim dataset, and (right) examples of birds from both datasets with the corresponding pixels count for each box.

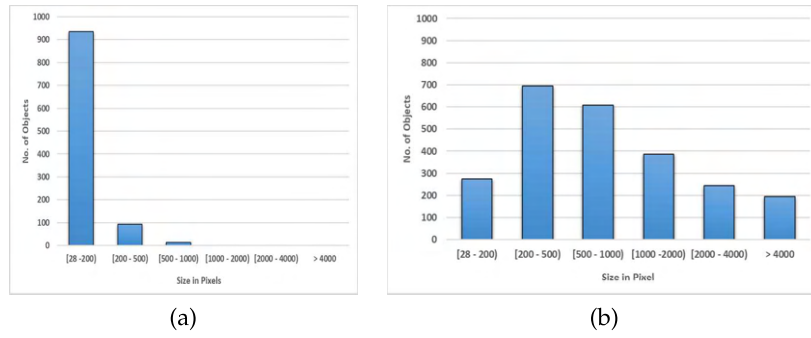


Figure 3.7: Bird size statistics of (a) Skagen and (b) Klim datasets.

which makes it a good option for training and validation.

YOLOv4-based Models Model 1 takes the full 4K frame as input and pass it to YOLOv4 network. The input frames get resized to the network width and height of 1024×1024 . The model is trained on the Klim dataset. Objects in this dataset are small objects as shown in Fig. 3.7. Therefore, Model 1 performed poorly detecting small birds and had a high rate of False Negatives. This happens because frames get resized into the specified width and height of the network architecture during training, 1024×1024 . When objects are already small and the frame gets resized, pixels information get lost, and as a result, the model does not learn from these examples and consequently does not detect

Chapter 3. Methodology

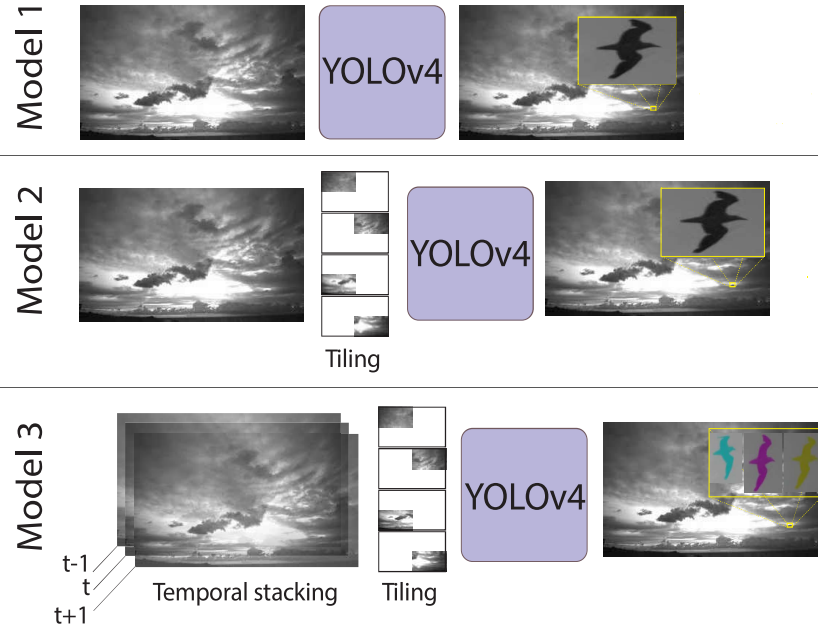


Figure 3.8: Three YOLOv4 based models, each trained on the Klim dataset, for birds detection around wind farms.

them during inference. In Fig. 3.8, this model is referred to as Model 1.

To address the problem of small objects in 4k resolution, Model 2 splits the input frames into four frames of 1920×1080 before feeding them to the network, Fig. 3.8. The frames are then resized to the network width and height of 1024×1024 . This model was trained also using the Klim dataset and it detects small birds better than Model 1. It is assumed that birds at the tiling boundaries are rare and that they do not stay long at these areas of the frame.

Frames in the Klim dataset that is used for training are extracted from a video sequence. This means that temporal information from successive frames is connected. Incorporating temporal in addition to the spatial information of objects across multiple frames allows the network to learn objects more accurately and differentiate them from noise since objects will appear in successive frames. Frames in both datasets are 24-bit grayscale, such that each pixel has three channels

3.3.1. Deep Learning based Object Detection

Table 3.1: Training parameters for Model 1, 2 and 3.

	Model 1	Model 2	Model 3
burn in	100	400	400
steps	(1600, 1800)	(1400, 1700)	(1400, 1700)
scales	(0.1, 0.1)	(0.1, 0.01)	(0.1, 0.01)
anchors	(4,5) (6,7) (9,9) (8,14) (14,12) (15,20) (24,25) (32,46) (74,121)	(5,8) (7,11) (11,13) (13,18) (20,18) (20,29) (33,27) (50,47) (60,98)	(5,8) (7,11) (11,14) (15,17) (21,20) (19,31) (32,27) (48,46) (61,97)

R, G, and B. For each frame at time t , Model 3 uses R and B channels to store pixel values of the object in the previous and next frame at time $t - 1$ and $t + 1$, respectively, before feeding it to the model for training. Model 3 also uses tiling similar to Model 2. Therefore, given 3 4K images, four $1920 \times 1080 \times 3$ tiles are constructed. Each of them is then resized to $1024 \times 1024 \times 3$ and passed into the YOLOv4 network. Utilizing temporal features in Model 3 is shown in 3.8.

Training Parameters Transfer learning was used to train the three models using the custom Klim dataset. Training starts using YOLOv4 weights pre-trained on the MS COCO dataset. The used weights file is yolov4.conv.137, which freezes the wights up to convolutional layer number 137, one layer before the first YOLO layer, and train the rest of the layers using the custom dataset. The models were trained using Google Colab Pro cloud services, alternating between Tesla P100-PCIE and V100-SXM2 16GB GPUs

The network configuration includes number of hyperparameters. A grid search is performed to find optimal values for the following parameters: batch size, subdivision, network resolution, learning rates, and anchors. These parameters are set set as: batch size: 64, sub-division: 32, height and width: 1024, momentum: 0.949, learning rate: 0.001, decay: 0.0005, and max batch: 2000. As the dataset has only one class, the number of filters before each of the three YOLO layers is set to 18. The hyperparameters for each model are defined in Table 3.1.

RESULTS AND DISCUSSION

4.1 Coverage Evaluation and Node Placement

The placement method in [Paper2] contributes to R.Q.1, which is optimizing the multi-camera dome placement. The method is based on placing the domes in a hexagonal thinnest covering. Then, it is optimized as presented in Sec. 3.1.1. The monitoring objectives of full coverage and 100% detection of flying objects when their altitude is less than or equal to the predefined altitude and their pixel resolution is equal to or greater than the defined detection resolution are satisfied.

To evaluate the placement, the defined method in [Paper1] measures pixel resolution, detection, and positioning percentage. The method is tested using tracks of flying birds using GPS-data loggers, fixed on the birds' backs. Track samples are the coordinates of bird locations. The corresponding pixel resolution vector for samples in each track is calculated using the dome capacity from Eq. 2.4. The detection percentage for each track represents the number of samples that have pixel resolution equal to or higher than p . This gives a measure of how good a specific combination of node placement and camera dome configuration is for detection. The positioning vector for each track indicates if a sample point is detected by more than one dome. This means that the bird is flying within coverage overlapping areas. The pixel resolution varies at each distance from the dome's center and when the bird is at the same point but at different altitudes. Therefore, the method calculates the pixel resolution captured for the samples in each track.

Fig.4.1a shows the pixel resolution plot for six tracks when their samples are detected by one dome. The defined minimum resolution is $3 \text{ pix}/m$ and the dome radius is 1 km. It shows that it is possible to have pixel resolution up to $25 \text{ pix}/m$ for Path 6 and that many paths can be analyzed simultaneously. Fig. 4.1b shows the pixel resolution plot of Path 3 when it is within the coverage of randomly placed 3 domes of the same capacity. Where the plots overlap, is where the

Chapter 4. Results and Discussion

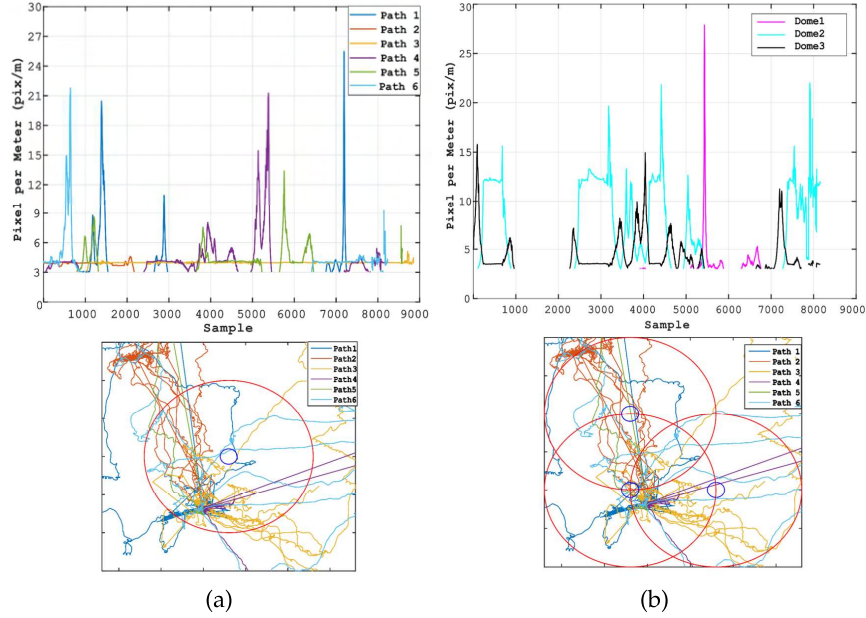


Figure 4.1: (a) Pixel resolution plot for six bird paths, covered by one camera dome of 3000 pixels capacity, (b) Pixel resolution plot of Path 3, covered by 3 domes of the same capacity.

object can be positioned.

The optimized hexagonal placement method in Sec. 3.1.1 was introduced to address R.Q.1. Different configurations are tested using that placement and the evaluation method. Configuration 1 has a dome radius of 1 km, minimum pixel resolution is 3 *pix/m*, flying altitude of 200 m, and the area required to be monitored is $3 \times 3 \text{ km}^2$. Seven domes should be placed in a hexagonal pattern to ensure full coverage. Distance between domes centers calculated using Eq. 3.1 is equal to 1937 m. Moreover, to ensure 100% detection of objects flying at 200 m, the domes should intersect at a minimum height of 559 m.

In configuration 2 has the dome radius is 550 m, and the rest of the monitoring constraints are similar to configuration 1. Twenty domes should be deployed to fully cover the area. Distance between domes centers in the hexagonal pattern is equal to 980 m. Domes should intersect at a minimum height of 283 m, calculated using Eq. 3.2.

4.2. Node Design Exploration and Cost Optimization

Table 4.1: Coverage evaluation of configuration 2, detection is 100% at both altitudes.

Track	200 m altitude		100 m altitude	
	Positioning (%)	Max. Pixel Resolution (pix/m)	Positioning (%)	Max. Pixel Resolution (pix/m)
1	33.14	29.98	69.66	59.88
2	23.81	29.13	36.03	57.05
3	28.34	29.90	39.03	59.26
4	21.73	20.18	97.71	24.83
5	9.65	19.76	57.97	24.06
6	25.05	29.99	38.04	59.94

Table 4.1, shows the evaluation results of the placement from configuration 2 when birds are flying at 200 and 100 m respectively. Both pixel resolution and positioning percentage are higher for the lower altitude, this is because birds stay longer in the overlapping regions between domes. This motivates changes to the design of the node itself as it is discussed in the next section.

4.2 Node Design Exploration and Cost Optimization

To configure the design of the multi-camera dome, the design exploration in [Paper3] that addresses the first part of R.Q.2 is presented. Given the constraints of the number of layers in a node, pixel resolution, and dome radius, the solution can be optimized for the cost or number of cameras. The pixel resolution multiplied by the dome radius indicates the dome capacity. An example of design exploration for a 3-layers dome is presented in Table 4.2. The design is constrained to 3 layers, maximum of 40 cameras, and is explored for dome capacity between 5000 and 9000 to cover 2π radian view. The trend in the table is that larger node capacity generates more costly solutions. This is because a large capacity requires a higher number of cameras. On the other hand, there is no correlation between the number of cameras and cost. A higher number of cameras can in some cases generate lower node cost, for example when the capacity is 5400 in the table. The table also shows that it is not possible to implement 9000 pixels capacity dome with less than 40 cameras.

Chapter 4. Results and Discussion

Table 4.2: Sample results for a 3-layers dome design.

Capacity (Pixels)	Cost (\$)	Camera selected per layer			Number of cameras per layer			Total cameras
		layer 1	layer 2	layer 3	layer 1	layer 2	layer 3	
5000	3114	25	25	25	7	6	3	16
5200	3392	36	23	23	10	6	3	19
5400	3586	36	23	23	11	6	3	20
5600	3660	36	25	45	11	6	2	19
5800	3698	25	25	25	8	7	4	19
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
6000	4845	34	34	49	16	12	2	30
7000	6916	58	58	58	14	12	7	33
8000	7964	58	58	58	16	14	8	38
9000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

The dome capacity can be used to explore the impact of design constraints triad-offs on node cost. For example, Fig. 4.2 shows that it is possible to design a dome of 3000 pixels capacity at a cost of US\$ 2,000 from the available camera-lens list. This solution requires 10 cameras organized into 2 layers, Fig. 4.2b. This means that a dome with pixel resolution p , of 30, 50, 60 or 100 *pixels/m* and radius, r , of 100, 60, 50 or 30 m respectively can be designed with US\$ 2,000. This makes it possible to maximize the capacity of a monitoring system for a given budget. The figure also shows that for a camera dome with capacity range between 3500 to 6000 pixels, it is possible to design a 3-layers node with the same cost and number of cameras.

As it was introduced in Sec. 3.2.2, the hemispherical dome design can be cost-optimized if its coverage is modeled as a cylinder, Fig. 3.5b. In addition to the node cost, the coverage cost is also investigated. That is the number of square meters covered per dollar ($m^2/\$$) when placing the node in the hexagonal pattern. This cost analysis addresses the second part of R.Q.2.

Using the placement method in [Paper2] and the design exploration in [Paper3] to estimate the node cost, node, and coverage cost comparison for the hemispherical and cylindrical designs is conducted. The node cost of multiple configurations is calculated assuming a required detection resolution of 10 *pix/m* and testing monitoring

4.2. Node Design Exploration and Cost Optimization

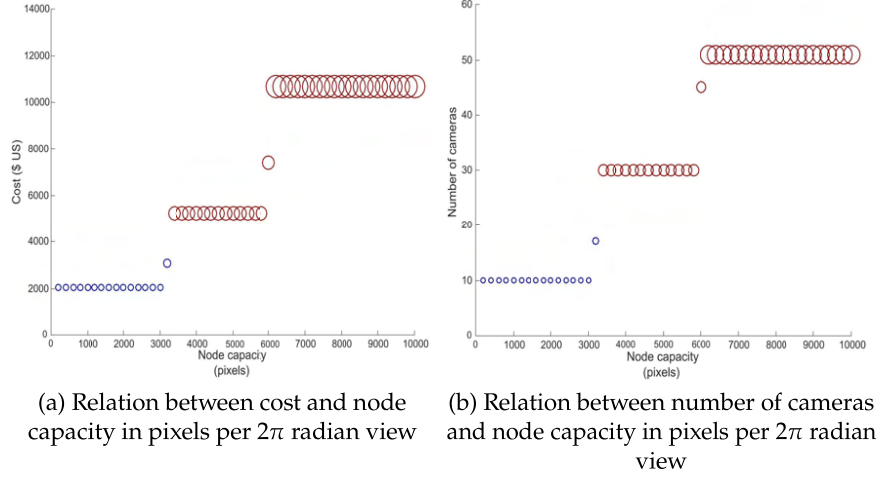


Figure 4.2: Unconstrained solution space exploration, bubbles size represent the number of layers.

radii of 100, 200, 300, 400, and 500 m at different heights. In Table 4.3, the optimized cost per node, $\$_n$, is calculated along with the radius, r_m , that provides this cost. The configurations are for both hemispherical and cylindrical designs with 1.5, 2, 2.5, and 3 layers. Results in the table show that the new cylindrical design is cheaper per node than the hemispherical at all heights when the number of layers is 2.5 and 3. Whereas when the node 1.5 and 2 layers, the hemisphere design is cheaper but has less monitoring radius than the cylindrical. Moreover, the concept of half a layer cuts down the number of cameras on the top layer and hence reduces the cost of the node. For example, the cost of a 2.5 layers node is less compared to a 3 layers node in both designs.

When nodes are placed in the hexagonal pattern, Fig. 3.4b, the radius that provides full coverage, $r_{hexagon}$, is different than the defined node radius. That radius is used to measure the coverage cost, $C_{coverage}$. The coverage cost for the node is calculated as:

$$C_{coverage} = (r_{hexagon}^2 \pi) / \$_n \quad (4.1)$$

where $\$_n$ is the optimized cost of the node and $r_{hexagon}$ is its placement radius in the hexagonal placement. Fig. 4.3 illustrates coverage cost

Chapter 4. Results and Discussion

Table 4.3: Node cost and monitoring radius for maximum coverage.

Layers		Height									
		50		100		200		300		400	
		$\$_H$	r_M	$\$_H$	r_M	$\$_H$	r_M	$\$_H$	r_M	$\$_H$	r_M
Hemisphere design	1.5	1404	300	1404	300	1404	300	20975	400	NaN	NaN
	2	2056	300	2056	300	2056	300	28487	500	28487	500
	2.5	3975	500	3975	500	3975	500	3975	500	3975	500
	3	5227	500	5227	500	5227	500	5227	500	5227	500
Cylinder design	1.5	3083	500	3905	500	1400	300	20975	400	NaN	NaN
	2	2949	500	2949	500	3363	500	22666	400	28487	500
	2.5	3402	500	3402	500	3402	500	3975	500	3975	500
	3	3502	500	3502	500	3920	500	5147	500	5227	500

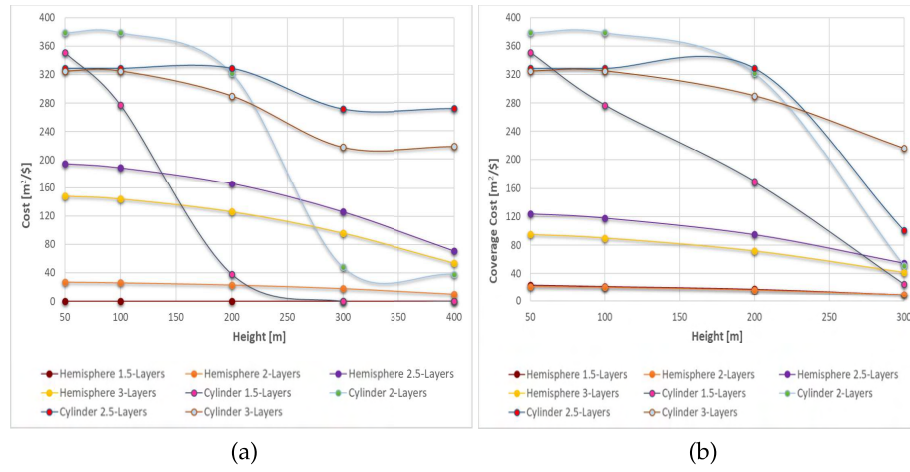


Figure 4.3: Coverage cost for eight configurations when monitoring radius is (a) 500 m and (b) 400 m.

for the eight configurations in Table 4.3, when the monitoring radius is 500 and 400 m, respectively. High coverage cost indicates a higher number of square meters covered per dollar. The figure shows that at 200 m height, 2 layers cylindrical node is the optimal design to implement for both radii. For monitoring heights between 200 and 400 m, the optimal solution is a cylindrical node of 2.5 layers when the radius is 500 m. whereas when it is 400 m the optimal configuration is a cylindrical node of 2.5 layers for a height range of 200-235 m, and 3 layers cylindrical node for 236-300 m. Moreover, 2.5 layers node is the optimal solution at all heights for the hemispherical dome.

4.3. Object Detection

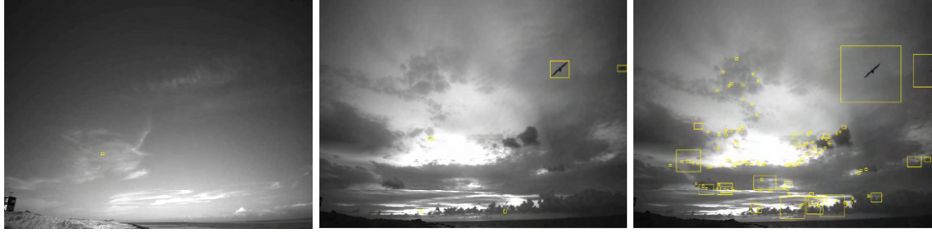


Figure 4.4: Results of background subtraction for bird detection. (left) Detection result for a frame from Skagen dataset, (middle) ground truth bird bounding boxes of a frame from Klim dataset, and (right) its detection result.

4.3 Object Detection

For the problem of in-flight birds detection, deep learning-based models are compared against background subtraction in [Paper5] using two custom datasets, [Datasets], to address R.Q.3. First, the background subtraction is tested using the Skagen dataset. It achieved precision and recall values of 84% and 60%, respectively. However, that method performed very badly on the Klim dataset. Skagen dataset exhibits a relatively static background, whereas the Klim dataset exhibits a highly dynamic background due to clouds movement and changes in illumination, as shown in Fig. 3.6. This suggests that background subtraction-based bird detection is unable to deal with settings where birds are viewed against a dynamic background. Fig. 4.4 highlights the failure of the background subtraction method on a frame from the Klim dataset with a large number of false positives, Fig. 4.4 (right). Note that this model does well on the frame from the Skagen dataset, Fig. 4.4 (left).

Model 1 is trained using 4K frames from the Klim dataset. The dataset includes 1607 objects for training, 340 for validation, and 357 for testing. This model achieved a mAP score of 83.7% on the validation set. The default objectness score (confidence score) that all predicted boxes are thresholded by is 0.25. Objectness score is the network's confidence that an object exists in a given box. The testing mAP score for Model 1 is 82.9% on the unseen frames.

Model 2 splits the 4K images into four 1920×1080 . These images are subsequently resized to 1024×1024 and fed into YOLOv4. For

Chapter 4. Results and Discussion

Table 4.4: AP_{50} and mAP scores for Model 1, 2 and 3 on the Klim dataset. Here M1, M2, and M3 refer to Models 1, 2 and 3, respectively.

	Training/Validation/Test set	AP_{50} (%) (Validation)	mAP(%) (Validation)	AP_{50} (%) (Test)	mAP(%) (Test)
M1	1607/340/357	83.6	83.7	82.4	82.9
M2	1120/266/231	82.5	82.8	88.4	88.7
M3	1108/258/223	77.9	78.1	89.6	90.1

Model 2, 1120 objects were used for training, 260 for validation, and 231 for testing. This model achieved a mAP score of 88.7% on the unseen testing data. Model 3 used temporal information stacking across frames to improve bird detection accuracy. Model 3 achieved mAP score of 90% on the test data. This is 2% improvement over Model 2 and 8% improvement over Model 1. Results for Model 1, 2, and 3 are summarized in Table 4.4. The table also shows the accuracy using COCO's AP_{50} , which is very close to the mAP results of VOC 2010-2012 for the three models.

Model 3 achieves better mAP values as compared to those achieved by the other three approaches: background subtraction-based method, Model 1, and Model 2. For the Klim test dataset of 330 birds, Model 3 was able to correctly detect 315 birds. Furthermore, for 156 birds between the sizes of 36 to 200 pixels, 145 were correctly detected. In other words, Model 3 was able to correctly detect 93% of the birds in the smallest category on the Klim test dataset, Fig. 4.5(right). In addition, Model 3 only missed eight birds in the Skagen test dataset, Fig. 4.5(left).

Table 4.5 summarizes testing mAP values for background subtraction-based method, Model 1, and Model 3 from both Skagen and Klim datasets. Note that neither Model 1 nor Model 3 were trained on the Skagen dataset. Background subtraction based bird detection method achieved mAP value of 71% on the Skagen dataset; however, this method performed poorly on the Klim dataset. Model 1 achieved mAP value of 60% on Skagen and 82% on the Klim dataset. This model was not trained on the Skagen dataset. Model 3 performed the best out of the three. This model achieved mAP values of 92% and 90% on Skagen and Klim testing datasets, respectively. The model

4.3. Object Detection

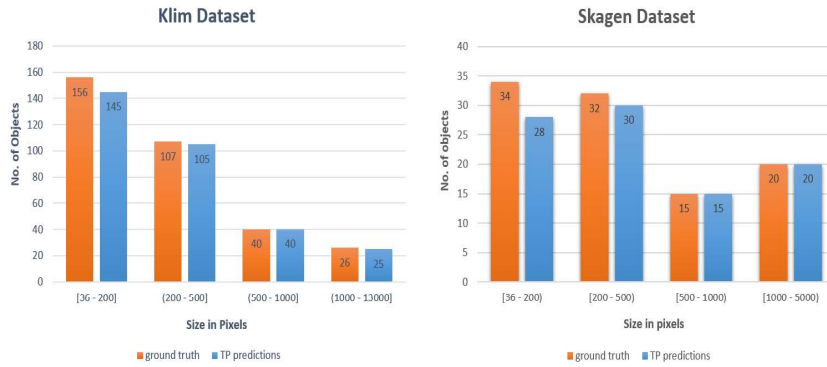


Figure 4.5: Size and number of ground truth and true detected objects using Model 3 for Klim and Skagen testing datasets.

Table 4.5: Testing accuracy using computer vision and deep learning.

mAP (%)	Klim	Skagen
Background subtraction	<1%	71%
Model 1	82	60
Model 3	90	92

was never trained on the Skagen dataset and yet it achieved a very good testing result of 92%.

CONCLUSION & FUTURE WORK

In order to optimize volumetric surveillance for sky monitoring, different aspects should be considered. Challenges in sky monitoring applications include that the volume is open and often remote, such as wind farms and airport runways. Also, objects fly in unpredicted patterns with complex backgrounds. Therefore, extending 2D area surveillance algorithms to 3D volume surveillance is inapplicable. Arranging multiple cameras in one node reduces the deployment and maintenance cost. This node architecture covers 2π radian hemisphere and is known as the multi-camera dome.

The inclusion of monitoring constraints like monitoring height and pixel resolution influences the node design. As a result, the node coverage is modeled as a cylinder rather than a hemisphere. This also optimizes the coverage cost in terms of the number of square meters per dollar. The design exploration in this thesis provides a method to minimize the node cost based on the defined design constraints. In addition, the proposed CNN-based model improves flying object detection compared to the standard YOLOv4 and background subtraction. This is achieved through tiling and temporal information incorporation.

The work in this thesis presents an optimized implementation of volumetric surveillance for sky monitoring. This includes constructing a cost-efficient multi-camera node, its placement in a volume, and detecting small flying objects. As will be explained in Sec. 5.0.2, object positioning can be simplified using the object detection model.

5.0.1 Environmental and Ethical Impact

The conducted research aims to optimize surveillance systems in sky monitoring. The considered application in the thesis and the published papers is monitoring volumes around wind turbines in wind farms. According to the Swedish Wind Energy Association, it is projected that wind power generation will increase to 90 TWh in 2040. This is part of the Swedish Energy Association plan of renewable

Chapter 5. Conclusion & Future Work

power expansion that wind power will be the source for over half of the electricity consumption in Sweden [51].

As presented in Ch. 1, it is estimated that up to 500,000 birds collide with wind turbines per year. With the expansion of these facilities, their impact on birds' habitat and their migration patterns should be investigated. Deploying a surveillance system that is able to detect, track, and count birds around wind turbines could be used in collision avoidance. This reduces the fatality rate of birds' collisions with the turbine blades. Also, it can be used when deciding appropriate locations for wind farms.

The measurements in [Paper6] and [Paper5] were conducted in two places in Denmark. The prototypes were deployed in *Klim Fjerritslev*, a wind farm of *Vattenfall* power company, [50], and in *Skagen Grey Lighthouse* at the Center for Migratory Birds in Denmark. The experiments were conducted after obtaining the proper authorizations in both sites. Hours of raw video footage were collected in these remote sites. However, if a person passed within the FoV of the cameras, that was removed from the footage by any chance.

5.0.2 Future Work

The CNN-based object detector can be used to obtain object trajectory through implementing a multiple object tracker (MOT), [31], on top of it. The 2D image points of the track are mapped to camera coordinates using the matrix in Eq. 2.6. Then the points of each track can be back-projected into the world as a surface. When the surfaces back-projected from two calibrated cameras intersect, the intersection curve represents the object trajectory in 3D. These measurements will be investigated in the future.

5.0.2. Future Work

ACRONYMS

2D	two-dimensional.
3D	three-dimensional.
CNNs	Convolutional Neural Networks.
FN	False Negative.
FoV	Field of View.
FP	False Positive.
fps	frame per second.
GPU	Graphical Processing Unit.
IoU	Intersection over Union.
km	kilometers.
kw	kilowatt.
m	meters.
mAP	mean Average Precision.
MS COCO	Microsoft Common Objects in Context).
pix/m	pixels per meter.
RGB	Red Green Blue image.
TP	True Positive.
YOLO	You Only Look Once.

BIBLIOGRAPHY

- [1] H. G. Akçay et al. "Automated Bird Counting with Deep Learning for Regional Bird Distribution Mapping". In: *Animals* 10.7 (2020). DOI: 10.3390/ani10071207.
- [Datasets] H. Alqaysi. *Annotated birds dataset for object detection using deep learning, Skagen & Klim*. Swedish National Data Service, 2021. DOI: <https://doi.org/10.5878/x1cm-pq40>.
- [Paper5] H. Alqaysi, I. Fedorov, F. Z. Qureshi, and M. O'Nils. "A Temporal Boosted YOLO-Based Model for Birds Detection around Wind Farms". In: *Journal of Imaging* 7.11 (2021). DOI: 10.3390/jimaging7110227.
- [Paper2] H. Alqaysi, N. Lawal, I. Fedorov, and M. O'Nils. "Full Coverage Optimization for Multi Camera Dome Placement in Volumetric Monitoring". In: *Proceedings of the 12th International Conference on Distributed Smart Cameras*. ICDSC '18. Eindhoven, Netherlands: Association for Computing Machinery, 2018. DOI: 10.1145/3243394.3243690.
- [Paper3] H. Alqaysi, N. Lawal, I. Fedorov, B. Thörnberg, and M. O'Nils. "Design Exploration of Multi-Camera Dome". In: *Proceedings of the 13th International Conference on Distributed Smart Cameras*. ICDSC 2019. Trento, Italy: Association for Computing Machinery, 2019. DOI: 10.1145/3349801.3349808.
- [Paper4] H. Alqaysi, N. Lawal, I. Fedorov, B. Thörnberg, and M. O'Nils. "Cost Optimized Design of Multi-Camera Dome for Volumetric Surveillance". In: *IEEE Sensors Journal* 21.3 (2021), pp. 3730–3737. DOI: 10.1109/JSEN.2020.3025359.
- [Paper1] H. Alqaysi, N. Lawal, I. Fedorov, and M. O'Nils. "Evaluating Coverage Effectiveness of Multi-Camera Domes Placement for Volumetric Surveillance". In: *Proceedings*

Bibliography

- of the 11th International Conference on Distributed Smart Cameras*. ICDSC 2017. Stanford, CA, USA: Association for Computing Machinery, 2017, pp. 49–54. doi: 10.1145/3131885.3131916.
- [2] J. B. Bishop, H. McKay, D. P. Parrott, and J. S. Allan. "Review of international research literature regarding the effectiveness of auditory bird scaring techniques and potential alternatives." In: 2003.
 - [3] B. F. Blackwell et al. "Exploiting avian vision with aircraft lighting to reduce bird strikes". In: *Journal of Applied Ecology* 49.4 (2012), pp. 758–766. doi: <https://doi.org/10.1111/j.1365-2664.2012.02165.x>.
 - [4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: abs/2004.10934 (2020).
 - [5] N. K. Chauhan and K. Singh. "A Review on Conventional Machine Learning vs Deep Learning". In: *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. 2018, pp. 347–352. doi: 10.1109/GUCON.2018.8675097.
 - [6] C. Chen, M.-Y. Liu, O. Tuzel, and J. Xiao. "R-CNN for small object detection". In: *Asian conference on computer vision*. Springer. 2016, pp. 214–230.
 - [7] O. Cogal et al. "A new omni-directional multi-camera system for high resolution surveillance". In: *Mobile Multimedia/Image Processing, Security, and Applications 2014*. Ed. by S. S. Agaian, S. A. Jassim, and E. Y. Du. Vol. 9120. International Society for Optics and Photonics. SPIE, 2014, pp. 179–187. doi: 10.1117/12.2049698.
 - [8] J. Dai, Y. Li, K. He, and J. Sun. "R-FCN: Object Detection via Region-Based Fully Convolutional Networks". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Barcelona, Spain: Curran Associates Inc., 2016, pp. 379–387.

- [9] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar. "A survey of deep learning and its applications: a new paradigm to machine learning". In: *Archives of Computational Methods in Engineering* 27.4 (2020), pp. 1071–1092.
- [10] *Darknet: Open Source Neural Networks in C*. <https://pjreddie.com/darknet/>. (Accessed on 11/23/2021).
- [11] L. Du, R. Zhang, and X. Wang. "Overview of two-stage object detection algorithms". In: 1544 (May 2020), p. 012033. doi: 10.1088/1742-6596/1544/1/012033.
- [12] J. Dziewierz, S. Ramadas, A. Gachagan, R. O’Leary, and G. Hayward. "A 2D Ultrasonic Array design incorporating Hexagonal-shaped Elements and Triangular-cut Piezocomposite Substructure for NDE applications". In: *2009 IEEE International Ultrasonics Symposium*. IEEE. 2009, pp. 422–425.
- [13] F. Emmert-Streib, Z. Yang, H. Feng, S. Tripathi, and M. Dehmer. "An Introductory Review of Deep Learning for Prediction Models With Big Data". In: *Frontiers in Artificial Intelligence* 3 (2020). doi: 10.3389/frai.2020.00004.
- [14] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (2010), pp. 303–338. doi: 10.1007/s11263-009-0275-4.
- [Paper6] I. Fedorov, B. Thörnberg, H. Alqaysi, N. Lawal, and M. O’Nils. "A Two-Layer 3-D Reconstruction Method and Calibration for Multicamera-Based Volumetric Positioning and Characterization". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–9. doi: 10.1109/TIM.2020.3023202.
- [15] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *CoRR* abs/1311.2524 (2013). arXiv: 1311.2524.

Bibliography

- [16] *Guide to Two-stage Object Detection: R-CNN, FPN, Mask R-CNN*. <https://medium.com/codex/a-guide-to-two-stage-object-detection-r-cnn-fpn-mask-r-cnn-and-more-54c2e168438c>. (accessed October 22, 2021).
- [17] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518, 2004.
- [18] J. Hernández-Pliego, C. Rodríguez, and J. Bustamante. "Why Do Kestrels Soar?" In: *PLOS ONE* 10.12 (Dec. 2015), pp. 1–24. doi: 10.1371/journal.pone.0145402.
- [19] Z. Huang et al. "Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection". In: *Information Sciences* 522 (2020), pp. 241–258. doi: 10.1016/j.ins.2020.02.067.
- [20] J. Hui. *mAP (mean Average Precision) for Object Detection*. <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>. accessed August 29, 2021.
- [21] J. Hui. *Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and YOLOv3)*. <https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359>. accessed November 2, 2021.
- [22] L. Jiao et al. "A Survey of Deep Learning-based Object Detection". In: *CoRR abs/1907.09408* (2019). arXiv: 1907.09408.
- [23] R. Kershner. "The number of circles covering a set". In: *American Journal of mathematics* 61.3 (1939), pp. 665–671.
- [24] N. Lawal, M. O'Nils, and M. Imran. "Design Exploration of a Multi-Camera Dome for Sky Monitoring". In: *ICDSC '16*. Paris, France: Association for Computing Machinery, 2016, pp. 14–18. doi: 10.1145/2967413.2967419.

- [25] N. Lawal, M. O’Nils, and M. Imran. “Design Exploration of a Multi-camera Dome for Sky Monitoring”. In: *Proceedings of the 10th International Conference on Distributed Smart Camera*. ICDSC ’16. Paris, France: ACM, 2016, pp. 14–18. doi: 10.1145/2967413.2967419.
- [26] Y. LeCun et al. “Backpropagation Applied to Handwritten Zip Code Recognition”. In: *Neural Computation* 1.4 (1989), pp. 541–551. doi: 10.1162/neco.1989.1.4.541.
- [27] Y. LeCun, Y. Bengio, et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995).
- [28] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444. doi: <https://doi.org/10.1038/nature14539>.
- [29] F. Liechti et al. “Cross-calibration of different radar systems for monitoring nocturnal bird migration across Europe and the Near East”. In: *Ecography* 42.5 (2019), pp. 887–898. doi: <https://doi.org/10.1111/ecog.04041>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/ecog.04041>.
- [30] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. “Path Aggregation Network for Instance Segmentation”. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018*. IEEE Computer Society, 2018, pp. 8759–8768. doi: 10.1109/CVPR.2018.00913.
- [31] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. “MOT16: A benchmark for multi-object tracking”. In: *arXiv preprint arXiv:1603.00831* (2016).
- [32] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le. “An evaluation of deep learning methods for small object detection”. In: *Journal of Electrical and Computer Engineering* 2020 (2020).

Bibliography

- [33] C. Nilsson et al. "Field validation of radar systems for monitoring bird migration". In: *Journal of Applied Ecology* 55.6 (2018), pp. 2552–2564. doi: <https://doi.org/10.1111/1365-2664.13174>. eprint: <https://besjournals.onlinelibrary.wiley.com/doi/pdf/10.1111/1365-2664.13174>.
- [34] *OpenCV: Camera Calibration*. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html. (Accessed on 11/23/2021).
- [35] P. Pham, D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le. "Evaluation of Deep Models for Real-Time Small Object Detection". In: *Neural Information Processing*. Ed. by D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy. Cham: Springer International Publishing, 2017, pp. 516–526.
- [36] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme. "Sensor network configuration and the curse of dimensionality". In: *Proc. Third Workshop on Embedded Networked Sensors (EmNets 2006)*, Cambridge, MA, USA. Citeseer. 2006.
- [37] *Publications*. <https://pjreddie.com/publications/>. accessed November 2, 2021.
- [38] J. Redmon and A. Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).
- [39] B. Rinner and W. Wolf. "An introduction to distributed smart cameras". In: *Proceedings of the IEEE* 96.10 (2008), pp. 1565–1575.
- [40] H. SCHMALJOHANN, F. LIECHTI, E. BÄCHLER, T. STEURI, and B. BRUDERER. "Quantification of bird migration by radar – a detection probability problem". In: *Ibis* 150.2 (2008), pp. 342–355. doi: <https://doi.org/10.1111/j.1474-919X.2007.00797.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1474-919X.2007.00797.x>.

- [41] T. D. Schneider and V. Jejjala. "Restriction enzymes use a 24 dimensional coding space to recognize 6 base long DNA sequences". In: *PLOS ONE* 14.10 (Oct. 2019), pp. 1–25. doi: 10.1371/journal.pone.0222419.
- [42] Y. Shi and S. Lichman. "Smart cameras: a review". In: *Proceedings of. Citeseer*. 2005, pp. 95–100.
- [43] K. Sinclair and E. DeGeorge. "Wind Energy Industry Eagle Detection and Deterrents: Research Gaps and Solutions Workshop Summary Report". In: (Apr. 2016). doi: 10.2172/1248080.
- [44] C. Szegedy, A. Toshev, and D. Erhan. "Deep Neural Networks for Object Detection". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., 2013, pp. 2553–2561.
- [45] T. Thayaparan, J. Marchioni, A. Kelsall, and R. Riddolls. "Improved Frequency Monitoring System for Sky-Wave Over-the-Horizon Radar in Canada". In: *IEEE Geoscience and Remote Sensing Letters* 17.4 (2020), pp. 606–610. doi: 10.1109/LGRS.2019.2928172.
- [46] *Triple integrals in spherical coordinates (article) | Khan Academy*. <https://www.khanacademy.org/math/multivariable-calculus/integrating-multivariable-functions/x786f2022:polar-spherical-cylindrical-coordinates/a/triple-integrals-in-spherical-coordinates>. (Accessed on 12/02/2021).
- [47] Tzutalin. *labelImg*. <https://github.com/tzutalin/labelImg>. 2015.
- [48] U.S. Fish and Wildlife Service—Migratory Bird Program *Conserving America's Birds*. <https://www.fws.gov/birds/bird-enthusiasts/threats-to-birds/collisions/wind-turbines.php>. accessed October 17, 2021.

Bibliography

- [49] *ULocation/Size/No. of Wind Turbines*. <https://www.renewablesfirst.co.uk/windpower/community-windpower/location-size-no-of-wind-turbines/>. accessed October 17, 2021.
- [50] *Wind power - Vattenfall*. <https://group.vattenfall.com/what-we-do/our-energy-sources/wind-power>. (Accessed on 11/30/2021).
- [51] *wind power combating climate change: 100 percent renewable electricity by 2040*. https://swedishwindenergy.com/wp-content/uploads/2019/10/Svensk_Vindenergi_ROADMAP_2040_rev_ENG-1.pdf. (Accessed on 12/03/2021).
- [52] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu. "Object Detection With Deep Learning: A Review". In: *IEEE Transactions on Neural Networks and Learning Systems* 30.11 (2019), pp. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.
- [53] Z. Zou, Z. Shi, Y. Guo, and J. Ye. "Object Detection in 20 Years: A Survey". In: *CoRR abs/1905.05055* (2019). arXiv: 1905.05055.