MoniSys: Sensing, Communication, and Video Analysis in UAV Monitoring System

Dissertation

zur Erlangung des Doktorgrades
Dr. rer. nat.
der Mathematisch-Naturwissenschaftlichen Fakultäten
der Georg-August-Universität zu Göttingen

im PhD Programme in Computer Science (PCS)
der Georg-August University School of Science (GAUSS)

vorgelegt von

Weijun Wang
aus Nanjing, Jiangsu, China

Göttingen
im August 2022

Betreuungsausschuss:	Prof. Dr. Xiaoming Fu, Georg-August-Universität Göttingen
	Prof. Dr. Marcus Baum, Georg-August-Universität Göttingen
	Prof. Dr. Guihai Chen, Nanjing University
Prüfungskommission:	
Referent:	Prof. Dr. Xiaoming Fu, Georg-August-Universität Göttingen
Korreferenten:	Prof. Dr. Falko Dressler, Technische Universität Berlin
	Prof. Dr. Robert Schober, Friedrich-Alexander-Universität Erlangen-Nürnberg
Weitere Mitglieder der Prüfungskommission:	Prof. Dr. Yunxin Liu, Tsinghua University
	Prof. Dr. Marcus Baum, Georg-August-Universität Göttingen
	Prof. Dr. Dieter Hogrefe, Georg-August-Universität Göttingen

Tag der mündlichen Prüfung: 07.09.2022

Abstract

Over the past 20 years, visual data has taken over every aspect of our lives, and camera deployment has experienced an unprecedented increase. For example, in the USA and UK, there is one camera for every 8 people used for diverse applications, like surveillance and public safety. However, in many extreme conditions, pre-deploying cameras are not feasible. Fortunately, the development of UAVs let these agile, flexible, and powerful devices make up the limitation of pre-deployed cameras. While high-resolution visual data offers rich information about the sensing environment, it causes significant challenges to the data analysis. Advances in computer vision present an excellent opportunity to process and analyze this massive amount of data; however, they have come at the expense of compute and network costs. For analysis, computing-resource-limited UAVs need to transmit sensing data to a computing-resource-rich server (on the ground). This distributed architecture posits several network-level resource management challenges: to ensure optimal UAV trajectories for sensing visual data; and to address the mobility impact and fair data delivery in multi-UAV access networks; and to provide low-latency, high-accuracy, and low-bandwidth-cost analysis.

We begin by presenting a general algorithm design schema for the *waypoint planning problem* to generate waypoints (*i.e.*, UAVs hovering and sensing points) achieving quality bounded sensing data. This schema includes three steps: discretization divides the entire solution space into subspaces; then dominating set extraction find out all the optimal solutions in every subspace; at last, transform waypoint planning into submodular optimization problem and propose an approximate algorithm. We apply this schema to three scenarios and verify the performance of our method provides 1.6× gain in sensing data quality.

Next, in order to address communication challenges, specifically the mobility impact and fairness among multiple UAV-server streamings, we develop *VSiM* - an easy-deployment and high-compatibility end-to-end solution to fairness in multiple mobile video streaming applications with a shared bottleneck bandwidth. It is pluggable to the server directly without caring and modifying any existing protocols or components. VSiM consists of three key techniques: dynamic and fair bandwidth allocation by incorporating mobile profile and QoE-related information; quick buffer filling for clients with lower playback time according to the requirement of the buffer-sensitive clients; adaptiveness to heterogeneous wireless network environments, like varied mobility patterns and topologies of base stations (BSes). It improves more than 40% on min QoE, which equals resolution improvement of viewing quality from 720p to 1080p) compared to state-of-the-art solutions.

Finally, in order to achieve low-latency, high-accuracy, and low-bandwidth-cost analysis, we present AccDecoder - a new decoder that derives important video content from bits stream and enhances them by super-resolution (SR) model. SR model achieves low-bandwidth-cost by allowing UAVs to transmit low-resolution data and enhance them into high-resolution getting high-accuracy. AccDecoder performs low-latency by analyzing bits stream in compressed-video-space, then selecting and enhancing a small part of data. AccDecoder preliminary opens the original decoder and reveals handy video codec information has potential room to accelerate analysis more than 4× speed.

This dissertation combines abstract mathematical models to describe and derive UAV Monitoring System (MoniSys) behaviors to design theory-level algorithms and develop system-level implementations for a set of working. The proposed key solutions have been implemented on DASH, PyTorch, H.264, and QUIC, four open-source codes, and our code is also public released on GitHub.

Acknowledgements

It is with great pleasure and gratitude that, I would like to convey, my sincerest acknowledgement and appreciation to everyone who generously contributed their invaluable time, support and guidance in my pursuit of PhD.

First and foremost, I would like to sincerely thank my PhD advisers: Prof. Dr. Xiaoming Fu, Prof. Dr. Macus Baum, and Prof. Dr. Guihai Chen, whose continuous support, encouragement and guidance along with their expertise were vital to author my PhD thesis.

I am extremely grateful to Prof. Dr. Xiaoming Fu for giving me the opportunity to pursue PhD under your expert guidance. I sincerely thank you for your invaluable time, effort and guidance during my PhD. I am grateful for the freedom and the multitude of opportunities that you let me explore for my research. I am also thankful for your generosity in funding my PhD. I am deeply grateful for all the encouragement and support that I have received from you throughout my PhD.

I am also obliged to my thesis defense committee members: Prof. Dr. Falko Dressler, Prof. Dr. Robert Schober, Prof. Dr. Yunxin Liu, Prof. Dr. Marcus Baum, and Prof. Dr. Dieter Hogrefe whose review comments and suggestions have greatly improved this thesis.

I would also like to thank all my collaborators also good friends, whose expertise and efforts have helped me at various stages of my PhD, especially (now Prof.) Dr. Yali Yuan, Dr. Tingting Yuan, Dr Bangbang Ren, Dr. Sripriya S. Adhatarao, Dr. Jiaquan Zhang, Dr. Yachao Shao, M.Sc. Fabian Wölk, M.Sc. Zhengze Li, M.Sc. Xin Gao, M.Sc. Yi Li, M.Sc. Yue Zhang, M.Sc. Cong Li.

I wish to convey my deepest and heartfelt gratitude to my families including Miss Woo Heo-Jeong. Thanks for your giving without expectation of return, accompanying without a grumble, guiding without a censure, and patience without a doubt. I would also like to thank anyone and everyone who directly and indirectly helped me to pursue my PhD.

Contents

List of Figures

List of Tables

List of Abbreviations

PANDA	*Waypoints **P**lanning of unmanned **A**erial vehicles achievi**N**g 3D **D**irectional cover**A**ge*
VISIT	*Waypoint plan of Unmanned Aerial **V**eh**I**cles for ani**S**otropic mon**I**toring **T**asks*
WiPlan	***W**aypo**i**nts **Plan**ning for Adjustable Multi-camera UAVs*
QoM	*Quality of Monitoring*
VSiM	***V**ideo **S**treaming **in** **M**obile Environment*
GP	*Gaussian Process*
GPS	*Global Positioning System*
MDS	*Monitoring Dominating Set*
UAV	*Unmanned Aerial Vehicle*
OPT	*Optimal Solution*
WSN	*Wireless Sensor Networks*
DCS	*Dominating Coverage Set*
MIMO	*Multi-Input Multi-Output*
SWIPT	*Simultaneous Wireless Information and Power Transfer*
CNN	*Convolutional Neural Networks*
RAC	*Rotation Around Camera*
RAO	*Rotation Around Objects*
PPI	*Pixels Per Inch*

RA	*Representative Arrangement*
AOV	*Angle of View*
RAs	*Representative Arrangements*
RAE	*Representative Arrangements Extraction*
RL	*Representative Locations*
RLE	*Representative Locations Extraction*
RS	*Representative Subarrangements*
RSE	*Representative Subarrangements Extraction*
VSiM	*Video Streaming in Mobile Environment*
DASH	*Dynamic Adaptive Streaming over HTTP*
QoE	*Quality of Experience*
BS	*Base Station*
ABR	*Adaptive Bitrate*
QUIC	*Quick UDP Internet Connections*
SNR	*Signal Noise Ratio*
SDFR	*Slow Degrade Fast Recovery*
WebRTC	*Web Real-Time Communications*
SR	*Super Resolution*
GAN	*Generative Adversarial Network*
CV	*Computer Vision*
QP	*Quantization Parameter*
VAP	*Video Analytics Pipeline*
VAP	*Video Analytics Pipeline*
DNN	*Deep Neural Network*

MB	*Macro Blcok*
Bbox	*Bounding box*
DRL	*Deep Reinforcement Learning*
MDP	*Markov Decision Process*

Chapter 1

Introduction

Monitoring with a camera sensor network has attracted great attention in recent years as it provides detailed environment data by retrieving rich information in the form of images and videos [1, 2]. It has found a wide range of applications, such as surveillance, traffic monitoring, crowd protection, disaster management, *etc.* For some temporary situations (*e.g.*, assembly, concerts, matches, and outdoor speeches) and sensing holes caused by sensors failure, fast establishing or recovering a stationary camera sensor network in advance may cost too much time and money, and may be inconvenient, even impossible. Fortunately, the development of Unmanned Aerial Vehicle (UAV) technology in the past few years [3–6] offers a promising way to address this issue.

1.1. UAV Monitoring System

The UAV monitoring system (called MoniSys in this dissertation) consists of low-cost and agile UAVs, and a computation-resource-rich server (*e.g.*, edge or cloud) dynamically senses high-quality images and video data and then delivers them to sever for real-time and reliable analysis. Specifically, as the big picture depicted in Fig. 1.1, MoniSys first receives the Points of interest (PoIs) [1] in the target monitoring area. It plans the waypoints of UAVs and schedules UAVs for sensing PoIs' visual data [2]; after that, UAVs transmit sensing data to sever over wireless connection; at last, sever

[1] How to generate PoIs from satellite images is out of the scope of MoniSys. For example, in disaster response, DLR (Deutsches Zentrum für Luft- und Raumfahrt) processes these images and provides PoIs' locations for MoniSys.

[2] Compared to the coarse-grained satellite data, fine-grained UAV sensing data contains richer detailed information. For example, *DJI Phantom 4* UAV can fly at $72\,km/h$, rise at $6\,m/s$, swerve at $250\,^{\circ}/s$, and provide $2K$ real-time images and videos [7].

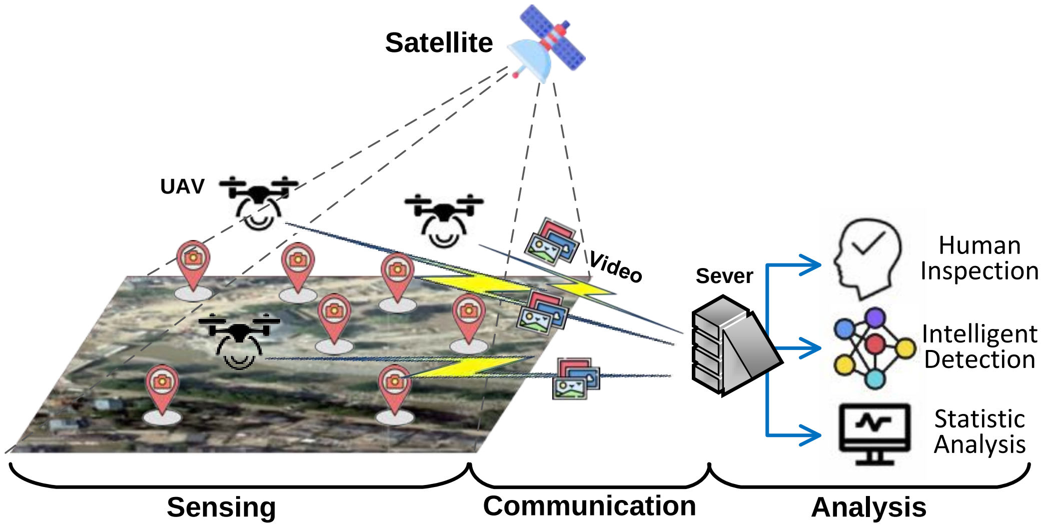


Figure 1.1.: **The big picture of UAV Monitoring System. After satellite provides the locations of Objects (also called Points of Interest), the UAV flies to capture fine-grained photos for following analysis.**

analyzes received data for various purpose. However, to improve the performance of the entire MoniSys, individually and/or jointly optimizing the performance of each phase (*e.g.*, sensing, communication, and analysis) is very important.

1.2. High Level Research Problems

The UAV monitoring system is plagued with sensing, communication, and analysis issues. Fig. 1.2 presents some of the critical high-level research problems associated with MoniSys, which are briefly discussed below:

(P1) *How to efficiently sense (capture) the monitoring data?* Sensing (or visual data collection) is the very beginning of the UAV monitoring system. Capturing high-quality data and streaming it to the ground server (in a real-time or retrospective way) is dramatically significant for the following analysis (See Fig. 1.1 the big picture). Three natural questions occur to capture high-quality data: 1. How to measure the data high-quality or not? 2. Where to capture high-quality data? 3. How to efficiently capture these data?

(P2) *How to fairly and efficiently deliver monitoring data from multiple UAVs to sever?* In the communication phase, multiple UAVs directly/indirectly connect to

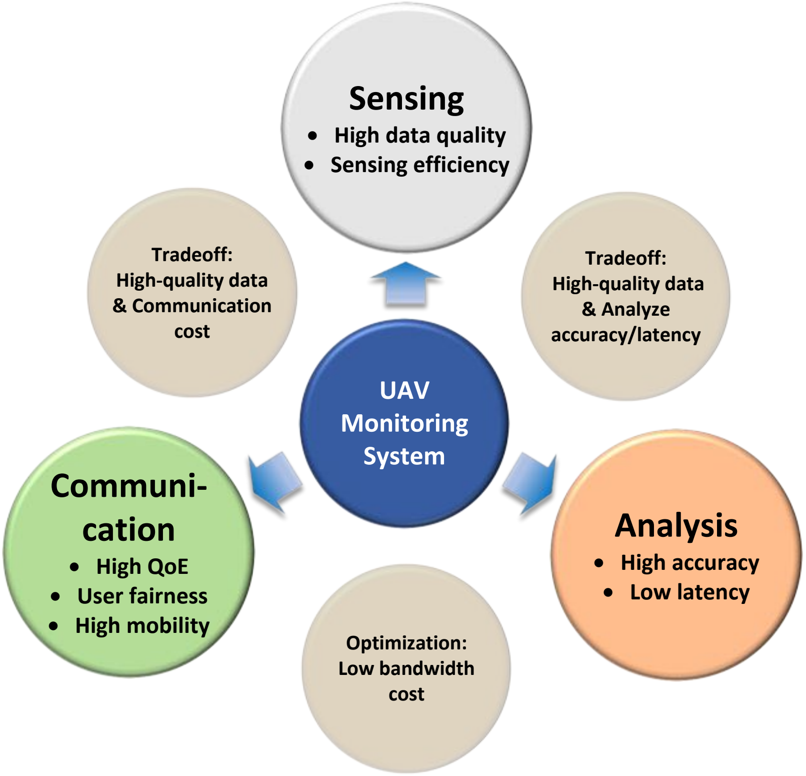


Figure 1.2.: **High-level Research Problems associated with MoniSys.**

sever and deliver a massive volume of video data (*e.g.*, $2K$ video streaming). It is easy to suffer congestion in such network conditions without careful bandwidth management and protocol. Holistic fairness and individual high user quality of experience (QoE)[3] are crucial in communication phase design. Moreover, high mobility is the most prominent characteristic of the UAV system. It is important to explore how mobility impacts communication, QoE, and fairness in MoniSys.

(P3) *How to accurately and efficiently analyze the monitoring data?* Advances in computer vision, especially with Deep Neural Networks (DNNs) presents a great opportunity to process and analyze video data with accuracy beating human beings. In this context, P3 focus on accurate and efficient video analytics (called Video Analytics Pipeline, VAP) with DNN-based methods. Traditional VAPs pursue high accuracy (the distance between analysis results and ground truths) and low end-to-end latency (from sensing data captured to analysis results).

[3] Here, QoE indicates not only human user but also NN-based (Neural Network based) applications in analysis phase.

Joint considering three phases in MoniSys leads to new optimization/design space (brown bubbles in Fig. 1.2).

Joint considering Communication and Analysis. Under distributed architecture in MoniSys, low bandwidth cost is the third optimization dimension except for accuracy and latency (bottom brown bubble in Fig. 1.2). This additional constance inspires compression method which prunes invalid video with on-UAV lightweight DNN or feedback from server to reduce bandwidth cost. Other methods let UAVs compress and stream the captured video in low resolution; the server recovers the high-resolution frames from the low-resolution stream via the super-resolution model. Uneven video codec (*e.g.*, assign a high quality to the foreground but low quality to the background) also fit tackle this issue.

Joint considering Sensing, Communication, and Analysis. High-quality data brings high accuracy but high bandwidth cost (left top brown bubble in Fig. 1.2). For example, compared to low-resolution video (*e.g.*, 180p), the accuracy of high-resolution video (*e.g.*, 720p) is much higher but 5× bandwidth cost (see details in Sec. 12.2.1 in Part III). At the same time, high-quality data brings high inference latency (right top brown bubble in Fig. 1.2). Take the semantic segmentation task as an instance, it needs to label each pixel one class which causes much higher latency on 720p video than 180p video.

Hence, it is necessary to both jointly and individually address these problems with a careful system and algorithms design to improve the performance of the UAV monitoring system.

1.3. Research Goals

In this dissertation, we intend to discern and address a few of the problems in MoniSys outlined in Sec. 1.2. We particularly seek to develop MoniSys to achieve three goals:

(G1) High-quality sensing. Sensing high-quality data needs us to understand how the quality of sensing data impacts the analytics performance, namely, building reasonable sensing models to quantify data quality. Efficiency is also essential for energy-limited UAVs. We seek to design an approximate algorithm carefully planning the minimum sensing points (called waypoints) to capture the most number of high-quality data for given PoIs is our goal.

(G2) QoE-optimized and fairness communication. We seek to build a video streaming system based on the Dynamic Adaptive Streaming over HTTP (DASH) framework to preliminary study the mobility effects under scaling users' contention environment. We also seek the QoE-fairness (not the bandwidth-fairness) mechanism among users to allocate the bandwidth of the bottleneck.

(G3) Accurate and fast analysis. We seek to design a video analytics pipeline that jointly considers the bandwidth, analysis accuracy, and end-to-end latency; in particular, this VAP requires high accuracy, low bandwidth cost, and low latency. High accuracy needs the VAP understands which data is "important" for the performance of analysis, low bandwidth cost needs the VAP only to transmit "important data". In contrast, low latency requires the VAP only to analyze "important" data.

Overall, to reach the above goals for MoniSys, we seek to design a general algorithm design schema for the waypoint planning problem (illustrated in Sec. 2.1) in the sensing phase, a QoE-fairness video streaming system (illustrated in Sec. 8.1) in communication phase, and a video-enhancement-participate[4] video analytics pipeline for analysis but holistic thinking all tradeoffs.

1.4. Dissertation Contributions

This section states the key challenges in addressing the problems in Sec. 1.2, and the contributions of this dissertation to achiecing above research goals.

1.4.1. Sensing phase

Recall that three natural questions occur to sense high-quality data: 1. How to measure the data high-quality or not? 2. Where to capture high-quality data? 3. How to efficiently capture these data? The last two questions correspond to two consecutive classic problems: the waypoint selection problem and the trajectory plan problem. In this dissertation, we only consider the *waypoint planning problem*, because more and more companies, including DJI [8], Skydio [9], and Parrot [10], provide waypoint flying mode [11] to free UAV pilots from complicated operations. In this mode, pilots only need to set appropriate waypoints on a map (usually 2D). UAVs can automatically plan the trajectory, adjust flying height for obstacle avoidance, and cruise via each waypoint. The formulation of the waypoint planning problem and its related works are proposed in Chap. 2. Here, we point out the

[4] Video-enhancement-participate means the VAPs leverage Super-Resolution tech or Generative Adversarial Network to enhance received data into higher-quality one.

general challenges of the waypoint planning problem and the specific challenges for each case in Part I.

- Challenge 1: *How to build models to quantify the quality of monitoring data?* Sensing models are extensive among different monitoring targets and sensing UAVs. Sec. 2.2 comprehensively survey the current models. Physical models of the target are classified into objects, barriers, and areas, while the sensing models of UAVs are omni-directional and directional. Building models case by case is one key also challenge for sensing.
- Challenge 2: *Facing continuous free 2D/3D space, how to plan waypoints based on the quality model?* For instance, the position of waypoints in 2D space is the whole free (x, y) coordinates; moreover, UAVs equipped with cameras often only sense one specific direction which is also continuous. Another critical challenge is finding optimal waypoints in one continuous, non-convex solution space.
- Challenge 3: *How to design an approximate algorithm with a performance guarantee?* This challenge (or called goal) is from challenge 2. Rather than a heuristic algorithm without a performance guarantee, we prefer to propose an approximate algorithm to bound the performance gap of sensing quality to the optimal one.

Including general challenges, in part I, we also consider three different problems, PANDA (Chap. 4), VISIT (Chap. 5), and WiPlan (Chap. 6). Each of them imports additional specific challenges of their own corresponding scenarios.

P1. PANDA problem inherits the general challenges but also *imports additional challenges of discrete nonlinear function* from the quality model. In short, as illustrated in figure in Fig. 1.3, targets are modeled as cones, and the sensing UAV is modeled as a straight pyramid.; the angular constraints for both camera and object models and the constraint of pitching angle of the camera in the objective function.

P2. Besides the common challenge (infinite and continuous solution space), *the additional challenge in VISIT is the anisotropic Quality of Monitoring (QoM) Model.* As shown in Fig. 1.3, the QoM is anisotropic and continuous; the quality of monitoring a human face or vehicle license plate drops a lot when the sensing direction is far away from the target's frontal view. The colored surface is the image of function $QoM(distance, viewed - angle)$, and the grey-scale figure is the projection of QoM function image in the αOd plane (solid lines in grey-scale figure are QoM contour). Given an increasing distance between object and UAV and an increasing angle between viewed direction and the facing direction of the object, QoM is decreasing monotonously [12, 13] (for details refer to Sec. 5.1.1).

P3. Simple Case with Two Waypoints Plan of WiPlan is illustrated in Fig. 1.3. Black hollow circles denote a set of objects with known locations to be monitored. Solid red and blue circles, *i.e.* u_1 and u_2, denote 2 waypoints. Red and blue dotted-line sectors and solid-line arrows indicate the 3 cameras' monitoring regions and directions. In this case, UAV pilot try to maximize the 4 objects' overall monitoring utility by planning two waypoints' locations, cameras' directions and focal lengths. *Besides the normal challenges in the waypoint planning problem, WiPlan involves two tightly coupled NP-hard problems.* One of the NP-hard problems is the location determination problem. Ignore cameras' directions and fix their focal lengths; UAV is equipped with unadjustable 360° cameras. The WiPlan problem is then slacked to deploy a set of unit disks to maximize the monitoring utility for all objects, which is NP-hard [14]. The other one is the cameras' directions and focal lengths scheduling problem. Suppose the locations of all waypoints are determined. Then WiPlan is slacked to the problem of scheduling cameras' focal lengths and directions to maximize the monitoring utility for all objects, which are also NP-hard [15]. Further, the decision variables impacting each other (the shared location of multiple cameras and the tradeoff between each camera's field of view and QoM), hence, are tightly coupled.

Contributions. To the above challenges, this dissertation proposes one algorithm design scheme and then addresses three waypoint planning problems facing different scenarios with different sensing and target models.

General Algorithm Design Schema (Chap. 3). In Chap. 3, we propose a general algorithm design schema for the waypoint planning problem. It consists of three steps: discretization, Dominating Coverage Set (DCS) extraction, and approximate algorithm. The core idea is to transform the original continuous non-linear solution space into a discrete one. Take a view of waypoint planning problem from geometry. Step 1 discretization, from the perspective of targets, naturally (in PANDA) or leverages piecewise constant function to approximate original function (in VISIT and WiPlan) to divide the entire solution space into cells. Step 2 uses the idea of "Divide and Conquer", namely, extracts DCS in each cell to reduce the infinite solution space to a limited one without performance loss. As a result (*i.e.*, Step 3), we transform an optimization problem with continuous solution space into a maximizing monotone submodular problem.

PANDA (Chap. 4) is the first study considering the 3D waypoint planning problem in a directional coverage scenario, to the best of our knowledge. The target is modeled as a sphere-based cone (Sec. 4.1.2) and sensing model is modeled as a straight pyramid (Sec. 4.1.1). We take $0-1$ binary function as monitoring quality; 0 indicate the target is not sensed and 1 indicates the target is sensed (See o_1 and

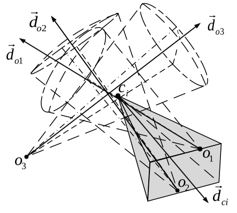
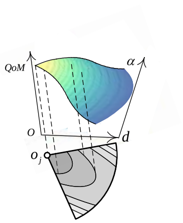
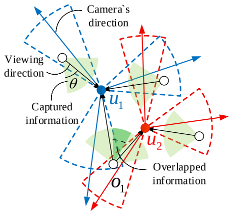
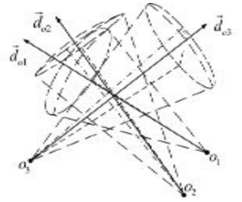
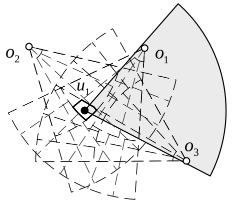
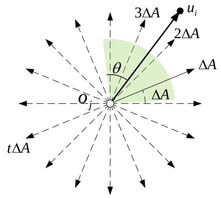
	PANDA	VISIT	WiPlan
Model	3D directional coverage model.  $\vec{d}_{o1}$ $\vec{d}_{o2}$ $\vec{d}_{o3}$ o_3 o_1 o_2 $\vec{d}_{ci}$	Anisotropic directional coverage model.  	Adjustable multi-camera coverage model.  Camera`s direction Viewing direction θ Captured information u_1 u_2 o_1 Overlapped information
Additional Challenges	Discrete 0-1 value model.	Non-convex & anisotropic model.	Two tightly coupled NP-hard problems.
Algorithm Design Scheme	Step 1: Objects' models naturally divide the whole space into cells (subspace).  $\vec{d}_{o2}$ $\vec{d}_{o1}$ $\vec{d}_{o3}$ o_3 o_1 o_2	Step 1: Approximate quality of monitoring (QoM) into piecewise constant value.  	Step 1: The same with VISIT but import additional aspects combining tech.  $3\Delta A$ u_i $2\Delta A$ ΔA θ ΔA o_j $t\Delta A$
	Step 2: Dominating Coverage Sets (DCSs) Extraction in 3D.	Step 2: Monitoring Dominating Sets (MDS) Extraction in 2D.	Step 2: Representative Locations (RLs) Extraction and Subarrangment Extraction.
	Step 3: Transform to a submodularity problem S.T. uniform matroid constraints, then propose a 1-1/e approximate algorithm.	Step 3: Transform to a submodular problem S.T. partitional matroid constraints, then propose a 1-1/e-ε approximate algorithm.	Step 3: Transform to a two-level submodular problem S.T. uniform matroid constraints, then propose a $1/2e^2$-1/e-ε approximate algorithm.

Figure 1.3.: **How general algorithm design schema applies to three different sensing problems.**

o_2 are sensed in Fig. 1.3). Utilizing the algorithm frame, PANDA is reformulated problem as maximizing a monotone submodular function subject to a matroid constraint and present a greedy algorithm with $1 - 1/e$ approximation ratio to address this problem.

VISIT (Chap. 5) is the first waypoint planning study importing anisotropic quality and driven by computer vision tasks. Take face recognition as an example; the frontal face offers higher accuracy than other viewing directions. Therefore, VISIT replaces the $0-1$ function in PANDA with the entropy of a Gaussian random variable in our sensing quality model and establishes the conditional covariance matrix to quantify the monitoring utility with a reduction of variance. We still follow the general schema in algorithm design but expand step 1 to approximate the QoM as a piecewise constant function of distance and angle. By doing so, the monitoring region is divided into many subareas in which the distance between real QoM and approximate QoM is bounded by ϵ and the approximated QoM at any point in each subarea becomes constant. Finally, we present a greedy algorithm with $1 - 1/e - \epsilon$ approximation ratio.

WiPlan (Chap. 6) is the first work considering the novel multi-camera model and exploring the fundamental challenges caused by this model. WiPlan figure out that the fundamental challenge from multi-camera model is that it involves two coupling NP-hard problems which making it much more complicated than in previous work (for details in Sec. 6.3). To tackle this challenge, we present a *two-level greedy algorithm* with $\frac{1}{2} + \frac{1}{2e^2} - \frac{1}{e} - \epsilon$ approximate ratio. Specifically, we propose a greedy algorithm that adds arrangements in order of non-increasing marginal benefit into the final arrangements set. However, the multi-camera model leads to computing the marginal benefit of each representative arrangement (RA) (evaluate each RA's marginal benefit by adding it into the currently selected arrangements set) itself is a hard problem. To this end, we relax the evaluation to use a greedy algorithm with polynomial-time and constant approximation ratio for marginal benefit computing. Finally, we prove the submodularity of the transformed combinatorial optimization problem and bound the approximation ratio of our two-level greedy algorithm.

1.4.2. Communication phase

Before deploying the video streaming system on real UAV monitoring system, in Part II (Chapter 8 and Chapter 9), we study how is the impact of mobility to video streaming on Dynamic Adaptive Streaming over HTTP (DASH) framework in a mobile environment. To fairness and efficiency, pose a more stringent requirement on efficient bandwidth allocation in mobile networks where multiple users may share a bottleneck link. This provides an opportunity to optimize multiple user's experiences jointly, but users often suffer short connection durations and frequent handoffs because of their high mobility. Here, we propose an end-to-end scheme, VSiM, for supporting QoE-fairness video streaming applications for mobile users in heterogeneous wireless networks.

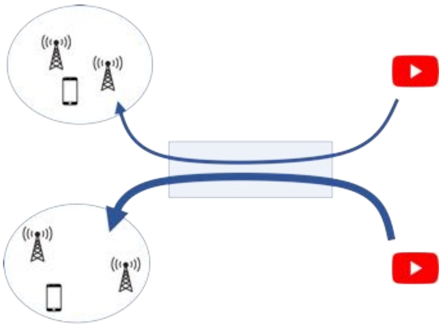


Figure 1.4.: **Bandwidth contention in multi-user environment.**

We face several challenges:

- Challenge 1: *How to profile the mobility impact and use the profile to maximize clients' QoE fairness in a mobile network?* Most existing works [16–23] depend on off-the-shelf mechanisms to ensure the network performance, like QoE, by dividing bandwidth evenly among multiple clients' connections, which neglects the knowledge of clients. Clients with the same bandwidth may experience different viewing experiences in mobile video streaming applications because of clients' mobility profiles, e.g., speed, direction, and acceleration. For instance, fast-moving clients may suffer more frequent handoffs [24,25], which causes the rebuffering and reduces clients' QoE.
- Challenge 2: *How to satisfy the buffer requirement of buffer-sensitive clients due to their mobility?* Because of the movement, after a period of time, some clients may be more sensitive to the playback buffer size [26]. Besides, clients may not receive the complete chunk with the requested bitrate due to the short stay time in one BS.
- Challenge 3: *How to ensure our system's robustness to support the heterogeneous mobile wireless network environment?* Mobile wireless network environment is heterogeneous due to the varied topologies and number of BSes, as well as the diverse clients' mobility patterns. The existing bandwidth allocation approaches to QoE improvement for video streaming applications [18–23] did not consider the robustness of the model. However, It is critical and valuable to build a model which can be adapted to various scenarios in the real world.

Contributions. We design **VSiM**, an easy-deployment and high-compatibility end-to-end solution to the QoE fairness problem in mobile video streaming applications with a shared bottleneck bandwidth. To the best of our knowledge, *VSiM is the first work towards QoE-fairness in mobile video streaming applications.* To tackle these challenges in Sec. 1.4.2, clients of VSiM inherit the excellent performance of the Dynamic Adaptive Streaming over HTTP (DASH) framework; VSiM deployed on the server achieves the QoE fairness by allocating bandwidth based on the advantages of the HTTP/3 protocol. To Challenge 1, VSiM adopts Mobility-profiled QoE-driven bandwidth allocation. It leverages clients' mobility profiles and QoE-related information to design an end-to-end scheme. At the client end, each client first collects its state information, including mobile profiles (*e.g.*, speed, location, and direction) from mobile devices by GPS and Inertial Measurement Unit (IMU) [27] as well as QoE-related information (*e.g.*., rebuffering and bitrate) from DASH video player. The collected state information is then grouped, encrypted, and sent along with the `HTTP Request` for downloading the chunk at a specific bitrate to the server. Utilizing these values and clients' QoE-related information, the proposed bandwidth allocation technique (see § 9.2.2) chooses the optimal allocated bandwidth for each client to maximize clients' QoE fairness dynamically in a mobile environment for real-time video streaming applications. To Challenge 2, VSiM adopts a high-compatibility server push strategy. It proposes a novel server push approach named Slow Degrade Fast Recovery (SDFR) (see § 9.2.3). Different from the traditional server push methods [28, 29], SDFR adds the buffer for needed clients in time dynamically without affecting the existing bandwidth allocation strategy and other clients' view quality. It is designed with a transparent mechanism compatible with all existing ABR algorithms. Specifically, based on clients' current stay time, handover time, and remaining buffer size, the server identifies clients who suffer high-frequent rebuffering and activates the server push function for them. To Challenge 3, VSiM adopts an online adaptive parameter update. It maps clients' mobile profile to hyper-parameters, such as staytime and handover time, which adapts to heterogeneous mobile wireless networks. Specifically, at the server end, the server calculates the trajectory of each client and further estimates the handover latency, staytime, and possible connection-less zones using its mobility profile and BSes' information. Furthermore, we propose the parameter update model (see § 9.2.4), based on Neural Networks (NNs), to decide the optimal parameters of the proposed model for each specific topology and the update period of bandwidth allocation.

VSiM is lightweight and easy to deploy in the real world without touching the underlying network infrastructure. It is pluggable to the server directly without caring and modifying any existing protocols or components. The adaptive end-to-end QoE fairness mechanism (Chapter 9) for the mobile video traffic with multiple mobile clients over a shared bottleneck link, named VSiM, consists of three key techniques:

1) dynamic and fair bandwidth allocation by incorporating clients' mobile profile and QoE-related information (§ 9.2.2); 2) quick buffer filling for clients with lower playback time according to the requirement of the buffer-sensitive clients (§ 9.2.3); 3) adaptiveness to heterogeneous wireless network environments, like varied mobility patterns and topologies of BSes (§ 9.2.4). The experiment results of VSiM in both simulation and prototype show that it improves more than 40% on QoE fairness (equal to resolution improvement of clients' viewing quality from 720p to 1080p) and ~20% on average of the averaged QoE compared to state-of-the-art solutions.

1.4.3. Analysis phase

Video-enhancement-participate approaches indeed bring low bandwidth cost and high accuracy. Nevertheless, trivially applying super resolution (SR) is time-consuming, and the extra latency introduces even more than analysis (also called DNN inference) time. To tackle this challenge, we present AccDecoder, *a new decoder that derives important content and enhances them from free block-based bits stream.* AccDecoder analyzes bits stream in compressed-video-space and expects to select and SR a subset of blocks, then leverages block dependencies and motion vectors to transfer the benefit from SR blocks to other blocks. This preliminary study wants to open the decoder and reveal how handy video coding information help speed up super resolution and DNN inference.

What is new about AccDecoder? (Contributions) Prior work (*e.g.*, [30–35]) reduces latency by hybrid using light-weight method (often consists of a *detector* and a *tracker*) and the heavy-weight method (a full DNN, *e.g.*, [36]). The light-weight method is like a shortcut to fast process unimportant data and reserve more time windows for heavy-weight but accurate inference. *Detector*, a fast classifier (*e.g.*, using learned features [33, 33, 34] or hand-craft low-level image features [30–32]), classifies if the current frame is worth feeding into full DNN for inference; if not, reusing the inference results from the previous nearest frame, but leverage *tracker* (*e.g.*, object tracker [37] moves the reused bounding box close to the ground truth) to make up for the accuracy loss from this shortcut. However, they always assume that bits stream are decoded into frames and the mechanism works on frame-space (see the prior VAPs in Fig. 1.5). These VAPs are from the perspective of the final DNNs, *i.e.*, the export of VAPs, whose inference is frame by frame. In contrast, we seek to take into account the mechanism works in compressed-video-space (see our decoder in Fig. 1.5); namely, we design VAP from the perspective of bits stream (the import of VAPs). So our approach can be viewed as complementary to this body of prior work.

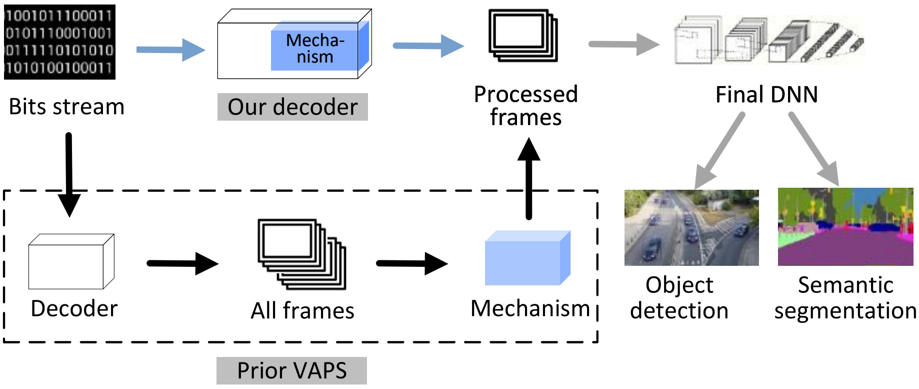


Figure 1.5.: **Comparison between our design and prior VAPs.**

The advantages of the design from the perspective of bits stream summarized as follows.

1) *Overcome the unstable network condition.* Frame-by-frame processing works well in AR/MR/VR applications, but this only suits ideal network condition whose delivery latency (3̃ms) and bandwidth (one-hop WiFi) is very well (see Table 1.1); VAPs crossing Wide Area Network (WAN), bits stream delivery tackles the challenge from dynamic bandwidth and network contention.

2) *Eliminate unnecessary decoding time costs.* VAPs deliver bits stream with TCP or UDP over WAN. Compressed-video-space mechanism avoids the time waste on decoding bits stream into unnecessary frames/regions (*e.g.*, the second frame of two continuous ones containing static content).

3) *Make the best of the information in hand.* Bits stream removes up to two orders of magnitude of superfluous information, making interesting signals prominent. In compressed-video-space, we account for correlation in video frames instead of prior mechanisms designed on *i.i.d.* frames. They repeatedly process near-duplicates to find out "important" content by eliminating redundant information among frames. Previous VAPs neglect the information from the codec. Obviously, there is a gap between codec and DNN model, however, our key intuition is that *the codec information in hand (*e.g.*, encoding types*[5]*), while not accurate, is sufficient to reveal "important" content, thereby speeding up video analytics.*

[5] In CV community, some approaches [49–52], leverage frame types (intra-frame (called I frame) and inter-frame (Non-I frame)) to label important frames. (Actually, H.26x formats adopt multiple types, we naively discussed here with 2 types.) I frames keep the most low-level (pixel-level) information in bits stream. Conceptually, I frames provide DNN model the most low-level information for abstract high-level (semantic-level) one.

Table 1.1.: **Comparison between AR/VR/MR and Video Analytics.**

Application	Latency Constraints	Network Condition	Streaming Unit
AR/VR/MR [**30**, **32**, **35**, **38**–**41**]	$\geq$ 60**fps** (**render one frame within** 16.66 **ms**). Because of the interaction between the user and their wearable device, latency constraints in AR/VR/MR is serious.	**Ideal network condition.** Desktop/Game Consoles placed in the same space with wearable devices. RTT commonly costs 3**ms** [35]	**Picture delivery.** Due to the latency constraint, AR/VR/MR can not wait to record and encode multiple frames then delivered. But the ideal network has ability to support picture delivery encoded by JPEG or PNG.
VAP [**31**, **42**–**47**]	25**fps**-30**fps**. Commonly, the video analytics system processes real-time video.	**Dynamic and unstable network condition.** Video delivery often cross Wide Area Network (WAN) which cost $20-50$**ms** RTT and the **bandwidth is not stable** [48].	**Video streaming.** Due to the delivery over WAN, streaming by frames too large to be feasible. Moreover, the latency constraint is not that serious.

1.5. Dissertation outline

This section outlines the three parts of this dissertation and the organization of chapters within these parts.

In Part I, we present the General Algorithm Design Schema to overcome the algorithm-level challenges of *Waypoint Planning Problem* like the non-convex optimization objective. Chap. 2 outlines the problem statement and related works, Chap. 3 presents the general algorithm design schema. Then, Chap. 4, Chap. 5, and Chap. 6 details and applies the schema to three scenarios.

In Part II, we present the bandwidth allocation framework for scaling mobile users to achieve QoE-fairness; it overcomes the system-level challenges associated with mobility impact and facilitates various ABR (adaptive bitrate) algorithms. Chap. 8 outlines the problem statement, state-of-the-art solutions and related work, Chap. 9 details the observation of mobility impact, and based on this, we design the QoE-optimized and fair framework.

In Part III, we present a decoder for analytics. Chap. 11 outlines the problem statement, and Chap. 12 compares and analyzes the limitation of the state-of-the-art solutions, then do a preliminary study on holistic designing video analytics pipeline, especially the decoder, with free information from the codec.

And finally, in Chap. 14, we revisit this dissertation's overall contributions and impact of this dissertation and outline the key future research prospects.

Part I.

Addressing Sensing Challenge in UAV Monitoring System: Waypoint Planning

Chapter 2

Problem Statement

This chapter initially introduces the waypoint planning problem. A comprehensive study of the state-of-the-art research is then provided with respect to the related research problems with waypoint planning. The study mainly identifies the issues in the state-of-the-art approaches and reveals the need for the solutions provided in the dissertation.

Contents

2.1. Introduction

Sensing (or called visual data collection) is the very beginning of UAV monitoring system. Capturing high-quality data and streaming to the ground server (in real-time or retrospective way) is dramatically significant for following analysis (See big picture in Chap. 1). To capture high-quality data, two natural questions occur: 1. Where to capture high-quality data? 2. How to efficiently capture these data? These two questions separately respond to two consecutive classic problems: waypoint selection problem and trajectory plan problem. From 2016, more and more companies, including DJI [8], Skydio [9], and Parrot [10], provide waypoint flying mode [11] to free UAV pilots from complicated operations. In this mode, pilots only need to set appropriate waypoints on map (usually 2D), then UAVs can automatically plan the trajectory, adjust flying height for obstacle avoidance, and cruise via each waypoint. Thus, in this dissertation, we consider the question: *How to set waypoints capturing high-quality data to improve the following analysis?*

Taking hazard response as an example in Fig. 2.1, spatiotemporal assessment of interests is a prerequisite for planning UAV waypoints. After utilize satellite establishing interest map [6] (*e.g.*, [53]), the next is UAV Waypoint planning with the knowledge of interest map[7].

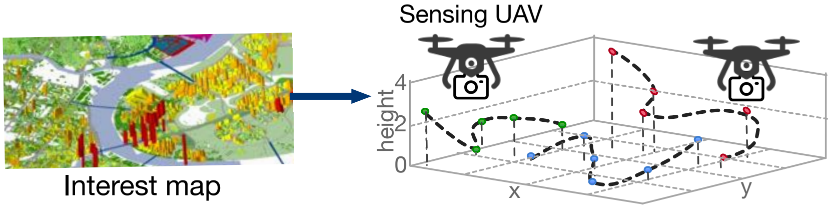


Figure 2.1.: **Workflow of Waypoint Planning Problem.**

There could be multiple variants of WPP (depending on the objectives and constraints), however, the general version can be formulated as below.

Waypoint Planning Problem (WPP): Given a set of PoIs $\mathcal{P}$, set a number of waypoints $\mathcal{W}$ to monitor the most PoIs, etc.

$$\begin{aligned} \max \quad & |\mathcal{P}_{monitored}| \\ \text{s.t.} \quad & |\mathcal{W}| = K. \end{aligned}$$

Some variant WPP, for instance, the physical sensing model variance (*e.g.*, full-view coverage, directional coverage) and mathematical sensing model variance (*e.g.*, continuously monitoring high emergency-level PoIs within a time interval), are also falls into above general version with some modification. The next section comprehensively surveyed and categorize these variances.

[6] Interest map is the spatiotemporal distribution of Point of Interests (PoIs, *e.g.*, possible survivor spots, fire locations, and on-ground sensors) with interest levels (e.g. emergency levels)

[7] In many other applications, such as joints and welding points in the inspection task of oil refinery management and electricity grids, these PoIs are fixed and their interest map can be established by experts.

2.2. Related Works

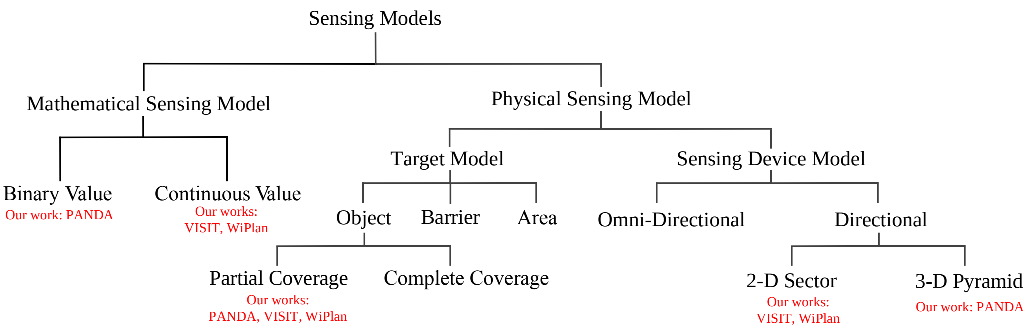


Figure 2.2.: **Classification of Related Works according to Sensing Models.**

This section comprehensively studies the state-of-the-art research. Because there are few works study waypoint planning problem and the somewhat similarity between the sensor coverage problem and waypoint planning, we expand our related works to sensor coverage problem. As shown in Fig. 2.2, we broadly classify the related literature by sensing model types.

2.2.1. Concept of Sensor Coverage Problem

The Sensor Coverage Problem [54] is a fundamental problem in wireless sensor networks, which can be briefly described as the coverage of objects, people, areas, and other targets by several sensors. In omni-directional sensor (always modeled as a disk) coverage, researchers have proposed many algorithms for this problem and introduce more objectives to the problem, such as using as few sensors as possible to ensure coverage, minimizing the energy consumption of the sensors, and so on. In directional sensor coverage, a sensor only work towards one specific direction (modeled as a sector) at a given moment t. In this case, the directional sensors need to be adjusted to the application requirements in order to obtain better performance. At the same time, how to arrange the sensors and adjust their operating directions to reduce the duplicated coverage area and obtain the maximum coverage is the main objective.

2.2.2. Mathematical Sensing Model

The mathematical sensing model is gradually changed from the binary coverage model [55] to probabilistic coverage model [56–58] (See the left top of Fig. 2.2). According to different application requirements, various sensors are used in sensor networks, such as temperature sensors, humidity sensors, infrared sensors, video sensors. Different sensors are built with different models according to their working characteristics.

Binary Value Model. The binary mathematical model is a simple model of the sensor coverage problem. The model defines that each target has only two states: 0 and 1. When the target lies in the coverage of sensors, the target is said to be covered (state is 1); otherwise, the target is uncovered (state is 0). Specifically, an area is covered if and only if all points in the area are covered by at least one sensor.

Continuous Value Model. The Continuous value mathematical model differs from binary model in that it only determines whether the target is within the sensor coverage area, but establishes a functional relationship between the sensor and the target. In [59], a probabilistic model is developed based on the signal propagation characteristics that decreases exponentially with the distance between the sensor and the target. The probabilistic model in [60] not only considers the propagation characteristics of the signal, but also more recently the rate of false alarm. In [61], the concept of correlation graph is introduced.

In this dissertation, PANDA (see Chap. 4) build on binary model since it is the first exploration on 3D model for UAV waypoint planning, we try to simplify the model. While, VISIT (see Chap. 5) and WiPlan (see Chap. 6) are built on Continuous value model.

2.2.3. Physical Sensing Model

2.2.3.1. Target Model

The coverage target are gradually divided into object coverage [62], barrier coverage [63, 64], and area coverage [65, 66]. In addition, considering different application scenarios, researchers have proposed and solved a series of problems on the original directed coverage problem, such as the full-view coverage problem [67] considering the coverage target orientation, the k-barrier coverage problem [68] considering the coverage quality, and so on.

In this dissertation, we focus on the object coverage work. PANDA, VISIT and WiPlan all concentrate on monitoring object target.

Object with Facing Direction. Some sensor deployment methods consider the facing direction of objects have been proposed in [1, 63, 69–76]. In [69], Wang *et al.* proposed the full-view coverage model by introducing objects' facing direction into the coverage model. Then, the full-view coverage model is extended to more scenarios in [1, 63, 70–76]. In [71] Hu *et al.* proposed an effective algorithm to solve full-view coverage problem in the mobile heterogeneous camera sensor networks. Wang *et al.* in [70], Ma *et al.* in [72], Yu *et al.* in [63], and Liu *et al.* in [74] focused on barrier coverage with minimum number of sensors. Yu *et al.* in [63] further considered intruders' faces for most intruders' trajectories crossing the barrier and Liu *et al.* in [74] considered the mobile camera sensors.

Object with Continuous Value Model (Called Quality of Monitoring, QoM). Some sensor deployment approaches consider the QoM in their sensing region [58, 62, 77–79]. Onur *et al.* in [77] first utilized signal propagation model to quantify the QoM of coverage, then proposed a sensor deployment algorithm achieving minimum ratio of false alarm and maximum ratio of alarm. Xing *et al.* in [58] first proposed a fusion model to fuse sensing value as QoM, then utilized this fusion model to develop a deployment algorithm which needs fewer sensors than algorithms without fusion. Yang *et al.* in [78] considered the energy consumption of sensors and combined fusion model in [58] to propose energy-efficient sensor deployment algorithm. Wang *et al.* in [62] defined QoM as coverage time of objects and proposed heuristics algorithm to deploy and schedule rotatable sensor to maximize the QoM. Fusco *et al.* in [79] defined QoM as the coverage time of one object covered by different sensors and proposed a sensor selecting and orientation assigning algorithm to achieve k-cover coverage.

Continuous Directional Model. A few sensor deployment approaches consider both QoM and facing direction of objects [68, 80–85]. Tao *et al.* in [80] proposed an algorithm to solve the problem that considers the priorities of sensing quality and sensing area. Saeed *et al.* in [81] considered the size of objects and proposed an algorithm to minimize the number of cameras and guarantee there is no occlusion among objects. Wang *et al.* in [68] quantified coverage QoM as k-barrier and studied the problem of deploying a minimum number of mobile and stationary directional sensors to achieve k-barrier coverage. Li *et al.* in [82] first combined previous k-coverage QoM and full-view coverage, then proposed a k-full-view coverage algorithm to address the problem of covering fixed number of objects with a minimum number of camera sensors for a special case. Cheng *et al.* in [83] first quantified the coverage QoM as breadth of barrier-coverage and proposed an algorithm to deploy sensors covering a belt-barrier achieving β breadth, then in [84] they introduced the facing direction of objects in coverage model.

2.2.3.2. Sensing Device Model

Omni-directional Sensing Model. The omni-directional physical model is suitable for temperature sensors, humidity sensors, watermark sensors, etc. It is built in the 2D plane as a "disc" with the sensor as the center and the sensing distance as the radius [86], and in the 3D space as a ball with the sensor as the center and the sensing distance as the radius. There are also some work in computer vision or cinematography applications [87–90]. [90] presents an end-to-end system to address the issue of optimizing the locations of two subjects captured in the photos by adjusting the trajectory of UAV and camera direction. [91] jointly optimizes 3D UAV motion plans and associated velocities. [87–89] consider omnidirectional camera model and fisheye camera model, to address computer vision issues.

2D Directional Sensing Model. Directional sensor coverage works can be classified into object coverage and area coverage, whose goals are maximizing the number of covered objects [62, 79] and area coverage ratio [68, 92], respectively. However, most of them do not take the objects' facing direction into account. Some camera sensor coverage works consider objects' facing direction [63, 69–76]. Wang *et al.* in [69] proposed a full-view coverage model by introducing objects' facing direction into the coverage model. Then, the full-view coverage model is extended to more scenarios in [1, 63, 70–76]. Hu *et al.* in [71] proposed an effective algorithm to solve full-view coverage problem in the mobile heterogeneous camera sensor networks. Some works [63, 70, 72, 74] focus on using minimum number of sensors to achieve barrier coverage, while [63] further considers intruders' faces for most intruders' trajectories crossing the barrier and [74] applies mobile camera sensors. Some works on autonomous cinematography consider the facing direction of multiple objects in one image. Joubert *et al.* in [90] presented a system to figure out the strategy of an UAV for capturing well-composed photos of two objects. Nageli *et al.* in [91] jointly considered monitoring quality and occlusion to optimize UAVs motion plans and associated velocities.

3D Directional Sensing Model. There exists a few works focusing on camera sensor coverage in 3D environment [93–101]. Ma *et al.* in [93] proposed the first 3D camera coverage model and developed an algorithm for area coverage on 2D plane with the projecting quadrilateral area of 3D camera coverage model. Based on this model, Yang *et al.* in [94] introduced coverage correlation model of neighbor cameras to decrease the number of cameras. Han *et al.* in [95] and Yang *et al.* in [96]

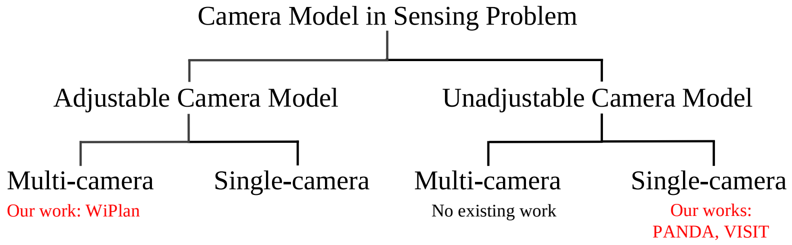


Figure 2.3.: **Problem Space.**

took energy and storage of camera sensors into account and proposed high-efficient resource utility coverage algorithm. Si *et al.* in [97] considered the intruders' facing direction and the size of face in barrier coverage. Hosseini *et al.* in [98] addressed the problem of camera selection and configuration problem for object coverage by binary integer programming solution. Li *et al.* in [99] and Peng *et al.* in [100, 101] established a more practical 3D camera coverage model, and studied three area coverage problems based on this model. Specifically, [99, 101] focus on maximize area coverage ratio, while [100] proposed a coverage hole detection and redeployment algorithm.

2.2.3.3. Camera Model - the specific directional model for UAV

Unadjustable Camera Model. Most of the existing work [1, 62, 63, 68, 70–72, 74, 102–104] study the sensor coverage problem with unadjustable camera model. [62, 63, 71, 72, 74, 102–104] focus on object coverage, while [1, 68, 70] focus on barrier or area coverage. In particular, [1, 63, 71, 72, 74, 103] further consider impact of objects' facing directions to quality of monitoring.

Adjustable Camera Model. Some work [15, 105–114] study deployment problem with adjustable camera model. Some of them consider the direction only adjustable camera model, *e.g.*, [105,106], which assume that sensors have been deployed in advance and design various scheduling mechanisms on cameras' directions. Some of them consider both focal length and direction adjustable camera [15, 107–114]. Specifically, [15] assumes the cameras are pre-deployed and focuses on adjusting the direction and focal length to maximize the number of covered objects. [108] utilizes the adjustable camera model to maximize the covered portion of a given area. [107] and [109] both build monitoring model based on the pixel requirement of computer vision application, *e.g.*, face detection, and design the photo selection and coverage algorithm based on their model, respectively. [114] and [110] both study the deployment in 3D space. [114] focuses on the large-scale scenario and design an efficient distributed algorithm, while [110] focuses on covering heterogeneous objects under budget constraint. Due to the space constraint, we briefly review the related work of camera models, but you can get more details in these two comprehensive surveys [115] and [116].

After a comprehensive survey, there is no existing work consider multi-camera model, thus we compare the related work on Multi-antenna model in wireless communication.

Multi-antenna Wireless Communication. We also review the multi-antenna model in the wireless communication domain [117–122]. Some work focus on improving performance in Multi-input Multi-output (MIMO) system and simultaneous wireless information and power transfer (SWIPT) system [117–120]. [118] and [120] survey the beamforming technique, aiming to improve the power transfer efficiency and communication robust, in the MIMO system and the SWIPT system. [117] bounds the performance gain of distributed antennas in one cell and proposes a deployment scheme to maximize the average users' communication rates. Some work study the connectivity between communication nodes by scheduling the antennas' directions and power [121, 122]. However, the antenna model is often modeled as a beam, which is quite different from camera models, and their deployment schemes focus on searching optimal one in discrete solution space, which also differ significantly from ours.

2.2.4. UAV related Monitoring Works

There are also some recent works considering the issues on autonomous cinematography with UAVs. Joubert *et al.* in [90] presented an end-to-end system for capturing well-composed photos of two subjects with UAV. They focused on addressing the issue of optimizing the locations of two subjects captured in the photos by adjusting the trajectory of UAV and camera direction. Nageli *et al.* in [91] jointly optimized 3D UAV motion plans and associated velocities. However, all these works on cinematography aim to solve the problems of taking good photos under physical limits in different scenes, which are quite different from our UAVs waypoint planning problem.

Some research efforts are dedicated to jointly deploying trucks and drones for delivery applications, and jointly deploying drones and drone base stations. [123–129] utilized trucks and drones for delivery problems and solve the problem with heuristic algorithms. Besides, Ghazzai *et al.* in [130] employed the particle swarm optimization algorithm to maximize the coverage area of drones. Kimura *et al.* in [131] deploy drones in a 3D space, where a drone acts as a base station to enhance the communication quality. Liu *et al.* in [132] studied the problem to maximize the collected data size and minimize the total energy consumption of drones. In general, few of the existing works take consideration of both the camera model and monitoring utility model for monitoring tasks. [133] considers the networking technologies when routing trucks and drones. Trotta *et al.* in [134] took the public transportation as base stations and maximized the system lifetime of a drone network with a heuristic algorithm. Ghazzai *et al.* in [130] employed the particle swarm optimization algorithm to determine the placement of drone docking stations. Liu *et al.* in [132] studied the problem to optimize the collected data size, geographic fairness, and the energy consumption simultaneously with given drone charging stations.

2.3. Summary

This dissertation focusing on the following three limitation proposes a general algorithm design schema (see Chap. 3) tackling the waypoint planning problem.

- **Continuous Directional Object Monitoring in 3D Space.** Current existing works lacks the study to 3D space monitoring task because of previous sensing without the support from UAV. To this end, we solve waypoints Planning of unmanned Aerial vehicles achieviNg 3D Directional coverAge (PANDA) problem in Chap. 4.
- **Anisotropic Directional Object Monitoring.** The popular application of computer vision in our life is based on high-quality visual data collection, such as face recognition [135] and license recognition [136]. Current existing works lacks exploration of the correlation between recognition accuracy and visual data quality. To address this, this dissertation tackles the waypoint planning of Unmanned Aerial VehIcles for aniSotropic monItoring Tasks (VISIT) in Chap. 5.
- **Monitoring with pioneering Multi-camera UAV model.** From 2016, more and more companies, including DJI, Skydio, and Parrot, release multi-camera UAVs, *e.g.*, DJI M-series [8], Skydio R1 [9], and Waldo XCAM [10], for improving the work efficiency. This dissertation explores the fundamental challenges of this novel multi-camera model; it address the Waypoints Planning for Adjustable Multi-camera UAVs (WiPlan) problem in Chap. **??**.

Chapter 3

Algorithm Design Schema

In this chapter, we propose our algorithm design schema for addressing waypoint planning problem. We utilize PANDA (See Chap. 4 for details) as an example to first introduce introduce the schema itself, second explain its feasibility.

Contents

3.1. Brief Background of PANDA

The Waypoints Planning of unmanned Aerial vehicles achieviNg 3D Directional coverAge (PANDA) is, given a set of objects with determined positions and orientations in a 3D space, plan the UAV's waypoints (*i.e.*, positions and orientations) to maximize the overall directional coverage utility. Utility model of a waypoint covering an object is built as: a straight rectangle pyramid (*i.e.*, the UAV coverage model) covers the vertex of a cone meanwhile the vertex of this pyramid lie in the cone (cone is the model of object). Take 1 waypoint and 3 objects as an example in Fig. 3.1, UAV c_i lies in the cone o_1, o_2, and o_3, meanwhile it covers o_1 and o_2 but not o_3 (o_1, o_2 lie in pyramid but o_3 does not); therefore, in such case, the utility is 2. PANDA is: given a set of deterministic cone (position, direction, and size), utilize a fixed number of fixed-size pyramid to cover the most of them.

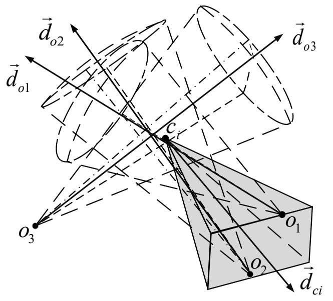


Figure 3.1.: **An example of 1 waypoint and 3 objects.**

3.2. Introduction

In this section, we use PANDA to illustrate our algorithm design schema. Our schema includes three steps.

Step 1: **Discretization.** Any object only can be efficiently covered by a waypoint in its efficient coverage space (*i.e.*, a cone). The whole coordinate solution space of waypoint is thus the union of all cones. We thus present a *space discretization approach* to partition the whole coordinate solution space into multiple cells, which can be considered separately. Especially, for each cell, the set of all possibly covered objects by adjusting the orientation of an UAV at any coordinate in the cell is exactly the same.

Step 2: **Dominating Set Extraction.** The same mathematic tools as Dominating Set we used in PANDA is Dominating Coverage Set (DCS), which covers the maximal set of objects and has no proper superset of covered objects by other arrangement (arrangement is the candidate waypoint). Then, the goal turns to find all candidate DCSs and their associated representative arrangements. Specifically, *the coordinates of associated arrangements of DCSs must lie on the boundaries of cells*, which serves as a constraint to help determine the representative arrangements. Therefore, adjust the coordinates and orientations of arrangements to create touching conditions and thus new constraints [8] Consequently, our problem is transformed into choosing fixed arrangements among the obtained candidate DCS arrangements to maximize the number of covered objects.

[8] In Sec 4.2.2, we intuitively give the formulation in seven-object case, as well as prove the Theo. 4.2.2 that it is sufficient to enumerate the 7 different cases, where 1 to 7 objects touch on the sides of pyramid, to extract all possible DCSs.

Step 3: **Approximate Algorithm.** Prove the submodularity and the properties of constraint of transformed problem (in PANDA it is uniform matroid constraint) and propose a greedy algorithm to solve it with performance guarantee.

3.3. Key Intuitions - Explanation of why our schema works

In this section, we still use PANDA (see Chap. 4) as an example to explain the feasibility of our schema. We only explain Step 2 because the Discretization step is easy to understand and the Approximate Algorithm is case by case.

3.3.1. Key intuitions in Dominating Set Extraction

1) **Focus on possible covered sets of objects of UAVs rather than candidate positions and orientations of UAVs.** As we can set waypoints/UAVs on any position and the orientation of their cameras arbitrarily, the number of candidate arrangements of UAVs is infinite, namely, the solution space of PANDA is infinite. However, many arrangements are essentially equivalent if they cover the same set of objects. Apparently, we only need to consider one representative arrangement among its associated class of all equivalent arrangements, and the number of all such representative arrangements is finite because the number of all possible covered sets of objects is finite.

2) **Focus on those arrangements that cover larger sets of objects.** If a representative arrangement covers the set of objects $\{o_1, o_2, o_3, o_4\}$, undoubtedly considering arrangements that cover its subsets, such as $\{o_1, o_2\}$ or $\{o_2, o_3, o_4\}$, is unnecessary. Our goal is to find the representative arrangements who possibly cover maximal covered sets.

3) **Find or "create" constraints to help determine representative arrangements.** However, even if we know the associated covered set of objects for a class of equivalent arrangements *how can we efficiently determine at least one representative arrangement?* Our solution is to imagine that given a feasible arrangement, we can adjust its position and orientation such that one or more objects touch some sides of the pyramid of the arrangement while keeping no objects out of coverage. If the arrangement adjust to satisfy one of following two situations, the representative arrangement has been found. 1. No matter how to adjust the arrangement, the covered objects won't change. 2. There will be at

least one object will get out of its coverage if the arrangement adjust any more, which means the arrangement has been fixed. Obviously, the obtained arrangement after the adjustment is also feasible and can be selected as a representative arrangement. Then, we in turn use the touching conditions as constraints and formulate them as equations to help determine the representative arrangements. Apart from such kind of constraints, we also find an additional constraint regarding positions of representative arrangements for their determination. During the enumeration, we find if there are 7 objects touches on the sides of pyramid and each side has on more than 3 objects, the pyramid can't adjust any more, otherwise some objects will be uncovered [9]. These representative arrangements can be found by adjusting from any equivalent arrangement in their associate class.

[9] For details, as the equations 4.14 listed in seven-objet case in Sec. 4.2.2, uniquely determine an arrangement, mathematically we need at least 9 equations (*e.g.*, Equ. (4.13 and Equ. 4.14)).

Chapter 4

Directional Coverage in 3D Space

This chapter considers the fundamental problem of Waypoints Planning of unmanned Aerial vehicles achieviNg 3D Directional coverAge (PANDA), that is, given a set of objects with determined positions and orientations in a 3D space, plan waypoints such that the overall directional coverage utility for all objects is maximized. This is the first work consider the 3D directional coverage model.

Contents

4.1. 3D Directional Coverage Statement

Suppose we have M objects $O = \{o_1, o_2, ..., o_M\}$ to be monitored in a 3D free space, each object o_j has a known orientation, which is denoted by a vector $\overrightarrow{d}_{oj}$. We also have N waypoints $C = \{c_1, c_2, ..., c_N\}$ can be set where to hover in the 3D free space. Because of hardware limitation, camera equipped on the UAV can only rotate in the vertical plane. However, this limitation has no influence to the coverage orientation, because UAV can hover in the air and rotate itself to face any horizontal orientation. By a little abuse of notation, c_i and o_j also denote the coordinate of waypoint/UAV and object. Table 5.1 lists the notations we use in PANDA.

Table 4.1.: **Notations used in Chap. 4.**

Symbol	Meaning
c_i	UAV i, or its 3D coordinate
o_j	Object j to be monitored, or its 3D coordinate
N	Number of waypoints to be planned
M	Number of objects to be monitored
$\overrightarrow{d}_{ci}$	Orientation of camera of UAV i
γ	Pitching angle of camera
γ_{min}	Minimum pitching angle
γ_{max}	Maximum pitching angle
α	Horizontal offset angles of the FoV around $\overrightarrow{d}_{ci}$
β	Vertical offset angles of the FoV around $\overrightarrow{d}_{ci}$
$\overrightarrow{d}_{oj}$	Orientation of object o_j
θ	Efficient angle around $\overrightarrow{d}_{oj}$ for directional coverage
Δ	Farthest sight distance of camera with guaranteed monitoring quality

4.1.1. Camera Model

More complicated than previous sector model of 2D directional sensor, camera model in 3D environment needs to be modeled as a straight rectangular pyramid as shown in Fig. 4.1. Due to cameras only can rotate in vertical plane, edge $\overline{AD}$ and $\overline{BC}$ are always parallel to the ground.

We use a 5-tuple (c_i, $\overrightarrow{d}_{ci}$, γ, α, β) to denote the camera model. c_i is coordinate (x_0, y_0, z_0) of an UAV in 3D space, $\overrightarrow{d}_{ci}$ is the orientation of the camera at the time, γ ($\gamma_{min} \leqslant \gamma \leqslant \gamma_{max}$) is the pitching angle of this orientation, α and β are the camera's horizontal and vertical offset angles of FoV (Field of View) around $\overrightarrow{d}_{ci}$.

As illustrated in Fig. 4.1, point c_i denotes the coordinate of UAV c_i, its value is (x_0, y_0, z_0). Vector $\overrightarrow{d}_{ci}$ denotes the orientation of c_i's camera which is perpendicular to undersurface $ABCD$ and its unit vector equals to (x_1, y_1, z_1). Point O is the centre of rectangle $ABCD$ and the distance $|\overline{c_iO}| = \Delta$, where Δ is the farthest distance from camera which can guarantee the quality of monitoring of every object on $ABCD$. Thus, the coordinate of point O is $(x_0 + \Delta x_1, y_0 + \Delta y_1, z_0 + \Delta z_1)$. Clearly, we can mathematically express plane $ABCD$ as

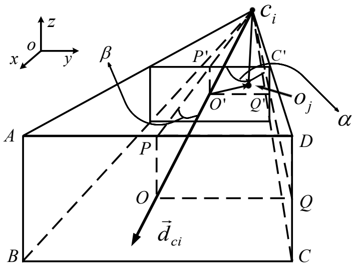


Figure 4.1.: **Camera model.**

$$\overrightarrow{d}_{ci} \cdot \begin{pmatrix} x - (x_0 + Dx_1) \\ y - (y_0 + Dy_1) \\ z - (z_0 + Dz_1) \end{pmatrix} = 0. \tag{4.1}$$

Connecting point c_i to midpoint P of $\overline{AD}$ and Q of $\overline{CD}$ respectively, we can get plane c_iOP and c_iOQ. As cameras only can rotate in vertical field, plane c_iOP is parallel to z-axis. Thus, plane c_iOP can be expressed as

$$-y_1x + x_1y + x_0y_1 - y_0x_1 = 0. \tag{4.2}$$

As shown in Fig. 4.1, $\overline{OQ} \perp \overline{OP}$,$\overline{OQ} \perp \overline{Oc_i}$, thus $\overline{OQ} \perp c_iOP$. By Equa. (4.2) and $\angle Oc_iQ$ equals to the horizontal offset angle α, so $|\overline{OQ}| = D \cdot \tan\alpha$. Thus, we can obtain

$$\overrightarrow{OQ} = \Delta \cdot \tan\alpha \cdot (-y_1, x_1, 0). \tag{4.3}$$

Similar, $\overline{OP} \perp c_iOQ$, $\angle Oc_iP$ equals to the vertical offset angle β, then we have $|\overline{OP}| = \Delta \cdot \tan\beta$. Combine the equation of plane c_iOP, vector $\overrightarrow{OP}$ can be obtained as

$$\overrightarrow{OP} = (x_1z_1, y_1z_1, -y_1^2 - x_1^2) \cdot \Delta \cdot \tan\beta. \tag{4.4}$$

To a given object o_j, if o_j is covered by c_i, it must be in some rectangle which is parallel to $ABCD$, *i.e.*, rectangle Ω between c_i and $ABCD$ in Fig. 4.1. According to this idea, we can illustrate the camera model as follows. Point O' is the centre of the rectangle and its coordinate is easy to figure out by o_j and normal vector $\overrightarrow{d}_{ci}$.

Utilize normal vectors $\overrightarrow{OQ}$ in Equa. (4.3) and $\overrightarrow{OP}$ in Equa. (4.4), if o_j satisfies the following constraint, point o_j is covered by camera c_i.

$$
\begin{aligned}
F_c(c_i,o_j,\overrightarrow{d}_{ci},\overrightarrow{d}_{oj}) &= \begin{cases} 1, & Prj_{\overrightarrow{OP}}\overrightarrow{O'o_j} \le |\overline{O'P'}|, \\ & Prj_{\overrightarrow{OQ}}\overrightarrow{O'o_j} \le |\overline{O'Q'}|, \\ & Prj_{\overrightarrow{d}_{ci}}\overrightarrow{c_io_j} \le \Delta. \\ 0, & otherwise. \end{cases} \\
s.t. \qquad & |\overline{c_iO'}| = Prj_{\overrightarrow{d}_{ci}}\overrightarrow{c_io_j}, |\overline{O'P'}| = |\overline{c_iO'}| \cdot \tan\beta, \\
& |\overline{O'Q'}| = |\overline{c_iO'}| \cdot \tan\alpha.
\end{aligned}
\tag{4.5}
$$

4.1.2. 3D Directional Coverage Model

First, we define 3D directional coverage as follows.

Definition 4.1.1. ***(3D directional coverage)*** *For an given object o_j and its facing direction $\overrightarrow{d}(x,y,z)$, there is an UAV c_i with camera orientation $\overrightarrow{d}_{ci}$, such that o_j is covered by c_i and $\alpha(\overrightarrow{d},\overrightarrow{o_jc_i}) \le \theta$ (θ is called the efficient angle), then object o_j is 3D directional covered by c_i.*

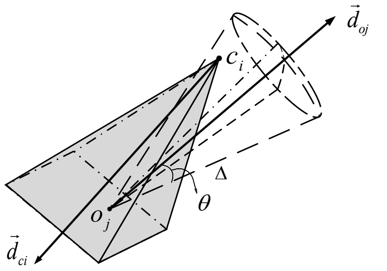


Figure 4.2.: **Directional coverage model.**

According to Def. 4.1.1, object model can be established as a spherical base cone as shown in Fig. 4.2. Let Object o_j be the vertex, rotate a sector of θ central angle and Δ radius around vector $\overrightarrow{d}$ for one revolution, then we obtain the object model as follows.

$$
\begin{cases} |\overrightarrow{c_io_j}| \le \Delta, \\ \alpha(\overrightarrow{o_jc_i},\overrightarrow{d}_{oj}) \le \theta. \end{cases}
\tag{4.6}
$$

Based on Equa. (4.6), UAV c_i can efficiently cover object o_j only when it locates in the cone of object o_j where can guarantee $|\overrightarrow{c_io_j}| \le \Delta$ and $\alpha(\overrightarrow{o_jc_i},\overrightarrow{d}_{oj}) \le \theta$. Thereby, combine Equa. (4.5) and (4.6), we can obtain the *directional coverage function* as

$$F_v(c_i,o_j,\overrightarrow{d}_{ci},\overrightarrow{d}_{oj}) = \begin{cases} 1, & Prj_{\overrightarrow{OP}}\overrightarrow{O'o_j} \leq |\overline{O'P'}|, \\ & Prj_{\overrightarrow{OQ}}\overrightarrow{O'o_j} \leq |\overline{O'Q'}|, \\ & Prj_{\overrightarrow{d}_{ci}}\overrightarrow{c_io_j} \leq |\overrightarrow{c_io_j}| \leq \Delta. \\ & \alpha(\overrightarrow{o_jc_i},\overrightarrow{d}_{oj}) \leq \theta. \\ 0, & otherwise \end{cases} \tag{4.7}$$
$$s.t. \quad |\overline{c_iO'}| = Prj_{\overrightarrow{d}_{ci}}\overrightarrow{c_io_j}, |\overline{O'P'}| = |\overline{c_iO'}| \times \tan\beta,$$
$$|\overline{O'Q'}| = |\overline{c_iO'}| \cdot \tan\alpha.$$

Then, the *directional coverage utility* can be defined as

$$\mathcal{U}_v(c_i,\overrightarrow{d}_{ci},o_j,\overrightarrow{d}_{oj}) = \begin{cases} 1, & \sum_{i=1}^{N} F_v(c_i,o_j,\overrightarrow{d}_{ci},\overrightarrow{d}_{oj}) \geq 1, \\ 0, & otherwise. \end{cases} \tag{4.8}$$

Similar to Camera Model, We also use Fig. 3.1 as an example to illustrate our 3D Directional Coverage Model and Directional Coverage Utility. In Fig. 3.1, UAV c_i lies in the spherical base cone of o_1, o_2, and o_3, thus the coordinate of c_i satisfies the Equa. (4.6) of all three objects. Combine the Camera Model, c_i, o_1, and c_i, o_2 both establish a 3D Directional Coverage subject to Equa. (4.7). Consequently, the directional coverage utility of c_i, o_1, $\mathcal{U}_v(c_i,\overrightarrow{d}_{ci},o_1,\overrightarrow{d}_{o1}) = 1$, and the directional coverage utility of c_i, o_2, $\mathcal{U}_v(c_i,\overrightarrow{d}_{ci},o_2,\overrightarrow{d}_{o2}) = 1$, but to c_i, o_3, the utility $\mathcal{U}_v(c_i,\overrightarrow{d}_{ci},o_3,\overrightarrow{d}_{o3}) = 0$.

4.1.3. Problem Formulation

In our problem, assume that we have obtained the orientation and coordinates of objects from the location and tracking technology of Internet of Things [137–140] or satellite images. We obtain o_j and $\overrightarrow{d}_{oj}$ in equations (4.5) and (4.7). Note that the parameters of cameras on UAVs are obtained in advance, thus the constants Δ, α, and β are also obtained. Our goal is to determine the coordinates c_i and orientations $\overrightarrow{d}_{ci}$ of UAVs.

Let the tuple $\langle c_i,\overrightarrow{d}_{ci}\rangle$, called *arrangement*, denotes the coordinate of UAV c_i and orientation of its camera $\overrightarrow{d_{ci}}$. Our task is to determine the arrangements for all N waypoints to optimize overall directional coverage utility for all M objects. Formally, the 3D Waypoints P̲lanning of Unmanned A̲erial Vehicle achieviN̲g D̲irectional coverA̲ge (PANDA) problem is defined as follows.

PANDA Problem (P1):

$$\textbf{(P1)} \quad max \quad \sum_{j=1}^{M} \mathcal{U}_v(\sum_{i=1}^{N} F_v(c_i, o_j, \overrightarrow{d}_{ci}, \overrightarrow{d}_{oj})),$$

$$s.t. \quad \gamma_{min} \leq \gamma \leq \gamma_{max}.$$

More generally, the goal of PANDA is setting N waypoints to directionally cover the maximum number of objects. If an object is covered by an UAV, the utility will increase by one, while overlapping coverage of the same object won't bring any contribution.

In the following theorem, we prove the PANDA problem is NP-hard.

Theorem 4.1.1. *The PANDA problem is NP-hard.*

Proof. To show the difficulty of the PANDA problem, we consider a simple case in which $\alpha = \beta = \theta = \pi$ and $\Delta = 1$, *i.e.*, the camera model is omnidirectional and each object can be covered from any orientation. Namely, as long as an object is located in the coverage of any camera, which is a unit ball, it can be covered by this camera. Our PANDA problem is transformed into using a fixed number of balls with radius of 1 to cover as many as objects in a 3D space. Note that the coverage problem on a 2D plane is a special case of constrained 3D space. This special case is exactly the well-known Unit Disk Coverage problem, which is NP-hard [14].

If we can design polynomial algorithm to address the original problem PANDA, obviously, we can address the NP-hard Unit Disk Coverage problem with this same algorithm. However, one NP-hard problem can't be addressed in polynomial time unless $P = NP$. Therefore, the PANDA is NP-hard problem, which doesn't have polynomial algorithm. □

4.2. Solution

In this section, we present an algorithm with approximation ratio $1-1/e$ to address PANDA.

4.2.1. 3D Space Discretization

As mentioned in 3D Directional Coverage Model, efficient coverage space of each object is modeled as a spherical base cone. These spherical base cones intersect among each other and form many 3D partitions called *cells*.

Due to geometric symmetry, only the waypoints locating in cells have chance to cover objects, and their potentially covered objects vary from one cell to another. For example, in Fig. 3.1, UAV c_i locates in the common cell of o_1 and o_2 and it can cover o_1 and o_2 simultaneously.

Theorem 4.2.1. *The number of partitioned cells is subject to $Z = O(M^2)$.*

Then we focus on the upper bound of the number of cells.

Proof. We first decrease the dimensions to 2D plane and analyze the upper bound of the number of partitioned cells on 2D plane by M uniform sectors intersecting with each other. Then, we prove that this upper bound is also the upper bound of the original 3D scenario by reduction.

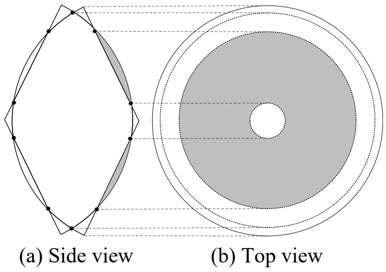


Figure 4.3.: **Intersection of two cones.**

Claim 4.2.1. *The number of partitioned cells on 2D plane by n uniform sectors intersecting with each other is at most $5n^2-5n+2$.*

Proof. First, we analyze the relationship between the number of cells and that of intersection points. Obviously, if there are three or more edges or arcs intersecting at same point, the number of cells must not be maximized. Thus, consider the condition that there are only two edges or arcs intersecting at one point, then one intersection point divides each edge into two parts, *i.e.*, the total number of added edges is 2 times that of intersection point. Let e denote the initial total number of edges, v denote the initial total number of vertices, f denote the initial total number of cells, (*i.e.*, faces in Graph Theory), and x denote the added intersection point. Due to the Euler characteristic [141], we have $f = e-v+2 = (e+2x)-(v+x)+2 = x+2$.

Furthermore, we observe that when the radian of sector is in $(\pi/2, \pi)$, there are the most intersection points for two sectors intersects with each other, *i.e.*, 10 intersection points. Thus, any pair among n sectors intersect at 10 different points, and there are at most $10 \cdot \binom{n}{2} = 5n^2 - 5n$ intersection points. By $f = x + 2$, the total number of cell is at most $5n^2 - 5n + 2$. □

In original 3D scenario, the side view and the top view of two cones intersecting with each other are depicted in Fig. 4.3. As shown in Fig. 4.3(a), from the side view, two cones intersect with each other by 10 intersection points. However, from the top view as shown in Fig. 4.3(b), the cells with grey color are also connected with each other in another dimension. This connection condition also happen in other symmetric cells. Therefore, the number of cells is less than the number of cells $5n^2 - 5n + 2$, *i.e.*, the number of cells $Z = O(M^2)$. □

4.2.2. Dominating Coverage Set (DCS) Extraction

After the space partition, we only need to consider the relationship between objects and UAVs in each cell, which depends on the coordinates and orientations of UAVs. In this subsection, we show that instead of enumerating all possible covered sets of objects, we only need to consider a limited number of representative covered sets of objects, which are defined as Dominating Coverage Sets (DCSs), and figure out their corresponding arrangements. Our ultimate goal is to reduce the problem to a combinatorial optimization problem which is selecting N arrangements from a limited number of arrangements obtained by DCS extraction.

Preliminaries. To begin with, we give the following definitions to assist analysis.

Definition 4.2.1. ***(Dominance)*** *Given two arrangements $\langle c_1, \overrightarrow{d}_{c1} \rangle$, $\langle c_2, \overrightarrow{d}_{c2} \rangle$ and their covered sets of objects O_1 and O_2. If $O_1 = O_2$, $\langle c_1, \overrightarrow{d}_{c1} \rangle$ is equivalent to $\langle c_2, \overrightarrow{d}_{c2} \rangle$, or $\langle c_1, \overrightarrow{d}_{c1} \rangle \equiv \langle c_2, \overrightarrow{d}_{c2} \rangle$; If $O_1 \supset O_2$, $\langle c_1, \overrightarrow{d}_{c1} \rangle$ dominates $\langle c_2, \overrightarrow{d}_{c2} \rangle$, or $\langle c_1, \overrightarrow{d}_{c1} \rangle \succ \langle c_2, \overrightarrow{d}_{c2} \rangle$; And if $O_1 \supseteq O_2$, $\langle c_1, \overrightarrow{d}_{c1} \rangle \succeq \langle c_2, \overrightarrow{d}_{c2} \rangle$.*

Definition 4.2.2. ***(Dominating Coverage Set)*** *Given a set of objects O_i covered by an arrangement $\langle c_i, \overrightarrow{d}_{ci} \rangle$, if there does not exist an arrangement $\langle c_j, \overrightarrow{d}_{cj} \rangle$ such that $\langle c_j, \overrightarrow{d}_{cj} \rangle \succ \langle c_i, \overrightarrow{d}_{ci} \rangle$, then O_i is a Dominating Coverage Set (DCS).*

For a given cell, it is possible only a few objects in the ground set of objects can be covered by an UAV locating in this cell. We formally give the following definition.

Definition 4.2.3. ***(Candidate Covered Set of Objects)*** *The candidate covered set of objects $\hat{O}_i$ for cell S_k are those objects possible to be covered by UAV c_i with some orientation $\overrightarrow{d_{ci}}$ in S_k.*

Obviously, any DCS of a cell is a subset of its candidate covered set of objects $\hat{O}_i$.

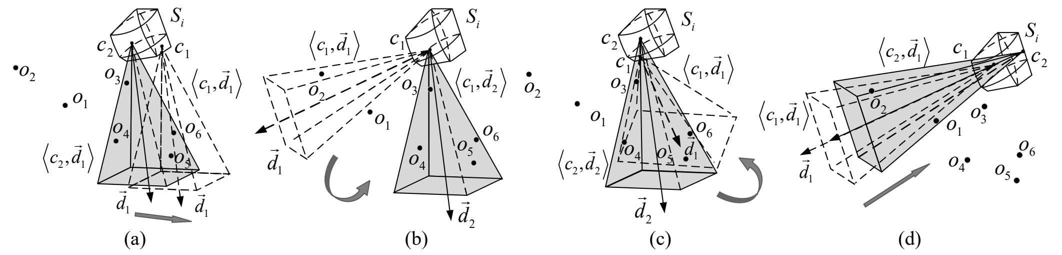

S_i
c_2
c_1
o_1
o_2
o_3
o_4
o_5
o_6
$\langle c_1,\vec{d}_1\rangle$
$\langle c_2,\vec{d}_1\rangle$
$\langle c_1,\vec{d}_2\rangle$
$\langle c_2,\vec{d}_2\rangle$
$\vec{d}_1$
$\vec{d}_2$
(a) (b) (c) (d)

Figure 4.4.: **Four kinds of transformations: (a) Translation, (b) Rotation Around Camera, (c) Rotation Around object(s), (d) Projection.**

As selecting DCSs is always better than selecting its subsets, we focus on figuring out all DCSs as well as their arrangements. In what follows, we first study two special cases where a coverage cell is reduced to a vertice (vertice case) and a line (line case) to pave the way for analyzing the general case.

DCS Extraction for the Vertice Case. First, we define four kinds of transformations as follows. Fig. 4.4 depicts four instances of these four transformations.

Definition 4.2.4. ***(Translation)*** *Given an arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$, keep the orientation unchanged and move the UAV from coordinate c_1 to coordinate c_2.*

Definition 4.2.5. ***(Rotation Around Camera (RAC))*** *Given an arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$, keep the coordinate unchanged and rotate the orientation from $\overrightarrow{d}_{c1}$ to $\overrightarrow{d}_{c2}$.*

Definition 4.2.6. ***(Rotation Around Objects (RAO))*** *Given an arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$, keep the object(s) on the touching side of pyramid,* i.e., *left side in Fig. 4.4(c), and move the UAV from $\langle c_1, \overrightarrow{d}_{c1}\rangle$ to $\langle c_2, \overrightarrow{d}_{c2}\rangle$.*

Definition 4.2.7. ***(Projection)*** *Given an arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$, keep the orientation unchanged and move the UAV along the reverse direction of orientation $\overrightarrow{d}_{c1}$ until reaching some point c_2 on the boundary of cell,* i.e., *$\langle c_2, \overrightarrow{d}_{c1}\rangle = f_{\perp}(\langle c_1, \overrightarrow{d}_{c1}\rangle)$.*

Obviously, *Projection* is a special case of *Translation.* Fig. 4.4(c) illustrates the *RAO* subject to objects o_3 and o_4.

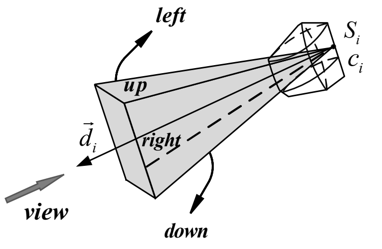


Figure 4.5.: **View orientation.**

Then, we present the DCS extraction algorithm for point case as shown in Alg. 1. Basically, the algorithm is a greedy algorithm which lets the UAV locate at the vertice and rotate around $o_j \in \hat{O}_i$ for a circle. Object o_j will slide on left, up, right, down sides orderly as illustrated in Fig. 4.5. During this process, Alg. 1 tracks the cur-

rent set of covered objects, and records all
DCSs. The input of DCS extraction for point case is the vertice S_i and its candidate covered set of objects $\hat{O}_i$. The output is the set of all DCSs.

Algorithm 1: DCSs Extraction for the Vertive Case

```
  Input: The vertice $S_i$, the candidate covered set of objects $\hat{O}_i$
  Output: All DCSs
1 Computer pitching angle of each object with $\overrightarrow{d}_{S_i o_j}$.
2 for every $o_j$ in $\hat{O}_i$ do
3     Initialize the rotated horizonal angle as the horizonal component of $\overrightarrow{d}_{S_i o_j} - \alpha$ and the
        vertical angle as $\gamma_j$.
4     Keep object $o_j$ sliding on the sides of the straight rectangle pyramid orderly, execute
        RAC transformation, where pitching angle changes between $\gamma_j + \beta$ and $\gamma_j - \beta$ and
        horizonal angle changes between $\overrightarrow{d}_{S_i o_j} - \alpha$ and $\overrightarrow{d}_{S_i o_j} + \alpha$, until there is at least one
        object will be uncovered. During the sliding process, if $o_j$ returns to the initialization
        value, add the current covered set of objects to the collection of DCSs and break.
5     Add the current covered set of objects to the collection of DCSs.
6     Keep $o_j$ continue sliding on the four sides of the straight rectangle pyramid orderly
        until a new object is covered.  During the sliding process, if $o_j$ returns to the
        initialization value, break. If not, goto step 4.
7 end
```

We use a toy example to illustrate the key idea of the DCS extraction for the vertice case. Fig. 4.6 illustrates the loop for object o_3, and the initial arrangement is $\langle S_i, \overrightarrow{d}_3^{left} \rangle$ as shown in Fig. 4.6(a). First, keep o_3 sliding on the left side of the straight rectangle pyramid and execute RAC transformation until o_3 touching up side, whose orientation is $\overrightarrow{d}_3^{up}$, as illustrated in Fig. 4.6(a). Second, keep o_3 sliding on the up side and execute RAC as shown in Fig. 4.6(b). During this sliding process, o_5 will get out of the pyramid. When o_5 touching the right side, add current DCS $\{o_3, o_4, o_5\}$ to the set of DCSs, then continue to slide until o_3 touching right side, whose orientation is $\overrightarrow{d}_3^{right}$. Third, keep o_3 sliding on the right side and execute

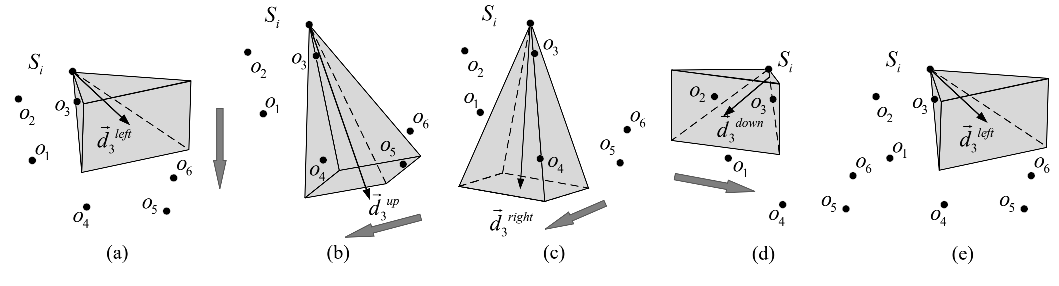
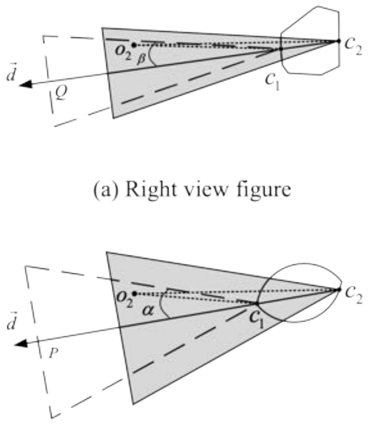


Figure 4.6.: **DCSs extraction for vertice case (object o_3 loop).**

RAC as shown in Fig. 4.6(c). Then, add DCSs $\{o_1, o_4, o_5\}$, $\{o_1, o_2, o_5\}$ to the set of DCSs orderly when o_4 and o_1 touch the down side in order. Forth, keep o_3 sliding on the down side and execute RAC as shown in Fig. 4.6(d). When o_2 touches the left side, add DCS $\{o_2, o_3\}$. At last, slide o_3 such that it assumes its initial arrangement as shown in Fig. 4.6(e), and add $\{o_3\}$. Note that for better readability, we just present the final situation of sliding on each side in Fig. 4.6.

DCS Extraction for Line Case. Line case is a specific instance of cell case. All the approaches to extract DCSs for any subcase are the same as following cell case. Thus, we omit this part.

DCS Extraction for the Cell Case. Now, we consider extracting DCSs for cell case.



(a) Right view figure

(b) Up view figure

Figure 4.7.: **Illustration for Lem. 4.2.1.**

According to the definition of *projection* , we have the following lemma.

Lemma 4.2.1. *If* $\langle c_2, \overrightarrow{d}_{c1}\rangle = f_\perp(\langle c_1, \overrightarrow{d}_{c1}\rangle)$, *then* $\langle c_2, \overrightarrow{d}_{c1}\rangle \succeq \langle c_1, \overrightarrow{d}_{c1}\rangle$.

Proof. First, we prove that no object will fall out of the straight rectangle pyramid through undersurface by the distance from objects to UAV. As mentioned in Sec. 4.2.1, every cell is formed by several spherical base cones with height of Δ. Thus, any point in given cell won't be farther than Δ from any object in $\hat{O}_i$ of this cell. As a result, no object will fall out of the pyramid through its undersurface during the process of projection. Second, we prove that no object will fall out of the pyramid through its four sides by the angle between objects and UAV. Fig. 4.7(a) illustrates the view from right of Fig. 4.4(d). In Fig. 4.7(b), it is obvious that $\angle o_2c_2Q < \angle o_2c_1Q$ since $\angle o_2c_2Q = \angle o_2c_1Q - \angle c_1o_2c_2$. So, the condition $\angle o_2c_1Q < \beta$ ensures $\angle o_2c_2Q < \beta$, *i.e.*, *projection* won't make objects fall out of pyramid through up or down side. Fig. 4.7(b) illustrates the view from up of Fig. 4.4(d). Similarly, as shown in Fig. 4.7(b), no objects will fall out of the pyramid through right or left side because $\angle o_2c_2P < \angle o_2c_1P < \alpha$. In summary, *projection* won't drop any initially covered object out of coverage, but, in contrast, it leads new objects to be covered. □

As Fig. 4.4(d) shows, after *projection* transformation, arrangement $\langle c_2, \overrightarrow{d}_{c1}\rangle$ covers o_1 and o_2 which are not covered by $\langle c_1, \overrightarrow{d}_{c1}\rangle$ before. By Lem. 4.2.1 we can get the following corollary.

Corollary 4.2.1. *Considering the case wherein UAVs lying on the boundaries of a cell is equivalent to considering the whole cell in terms of DCS extraction.*

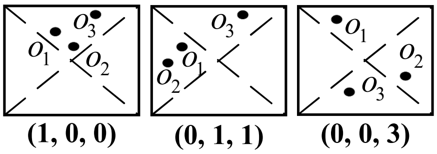
By Coro. 4.2.1, we only need to consider the arrangements wherein cameras lying on the boundaries of cell. We can perform the following transformation that begins with an arbitrary arrangement $\langle c, \overrightarrow{d}_c \rangle$ where c lies on the boundary. First, we execute RAC until there is at least one object touches some side of the straight rectangle pyramid (note that an object will never fall out of the pyramid through its undersurface as we discussed before). Next, keeping c lying on the boundary and former touched objects lying on their former touching sides, execute RAO and *translation* such that there is at least another object touches some side of pyramid. Execute above transformation of RAO and *translation* under given constraints repeatedly, such that as many as possible objects touch sides of pyramid until there is no objects will touch any side, we call it final condition. Finally, the position and orientation of straight rectangle pyramid, namely arrangement, can be either uniquely determined or not. For the former case, we can directly extract DCS of the unique arrangement. For the latter, we can select an arbitrary arrangement of final condition and extract DCS.

Because that during above transformation there is no object falling out of the pyramid, the set of covered objects of final condition dominates all sets under conditions of the process of transformation. Thus we only need to analyze the final condition which generate representative arrangement. In particular, we can enumerate all possible cases of final conditions for which there are 1 to 7 objects touching sides of the pyramid. Besides, one may be concerned about the possible performance loss as we select an arbitrary arrangement if a unique arrangement cannot be uniquely determined. We argue that there is NO performance loss and will prove it in Theo. 4.3.1.

In the following analysis, we use **(a, b, c)** to denote the case of final condition where a sides have three objects touching each of them, b sides have two objects touching each of them, and c sides have one object touching each of them. For example, in three-object cases, the solution of three coplanar objects lies on any one side of four sides are the same, so we just analyze they lying on the up side as show in Fig. 4.8 **(1, 0, 0)**. Fig. 4.8 to 4.12 depict the typical cases of final conditions, whose view is from the inverse direction of $\overrightarrow{d}_{ci}$ as shown in Fig. 4.5. The crossing dotted lines denote four edges of straight rectangle pyramid and their intersection point denotes the vertex of it.

- To one-object and two-object case, we only need to choose one point c_i on the boundary of cell S_i arbitrarily and execute the algorithm for point case.

- In three-object case, there exists three typical subcases.



(1, 0, 0) **(0, 1, 1)** **(0, 0, 3)**

Figure 4.8.: **Typical three-object cases.**

(1) **(1, 0, 0)**. As **(1, 0, 0)** in Fig. 4.8, o_1, o_2, and o_3 lie on the up side. Clearly, with the coordinates of three objects and expression of camera model, we have

$$\begin{cases} \overrightarrow{n}_{up} \cdot \overrightarrow{o_1o_2} = 0, \overrightarrow{n}_{up} \cdot \overrightarrow{o_2o_3} = 0, \\ \overrightarrow{d}_{ci} \cdot \overrightarrow{n}_{up} = \sin\beta, \overrightarrow{d}_{ci} \cdot \overrightarrow{n}_l = 0, \\ |\overrightarrow{n}_{up}| = 1, |\overrightarrow{d}_{ci}| = 1, |\overrightarrow{n}_l| = 1, \overrightarrow{n}_l // xOy. \end{cases} \tag{4.9}$$

where $\overrightarrow{n}_{up}$ is the normal vector of up side, $\overrightarrow{n}_l$ is the direction vector of the intersecting line of up side and the horizonal plane, and $\overrightarrow{d}_{ci} \cdot \overrightarrow{n}_l = 0$ describes camera can only rotate in the vertical plane we have discuss in Sec. 4.1. Hence, we can obtain the orientation $\overrightarrow{d}_{ci}$ with Equa. (4.9) and the candidate coordinates c_i can be expressed as follows:

$$\begin{cases} |\overrightarrow{c_io_1}| \leq \Delta, |\overrightarrow{c_io_2}| \leq \Delta, |\overrightarrow{c_io_3}| \leq \Delta, \\ \alpha(\overrightarrow{o_1c_i}, \overrightarrow{d}_{oj}) \leq \theta, \alpha(\overrightarrow{o_2c_i}, \overrightarrow{d}_{oj}) \leq \theta, \alpha(\overrightarrow{o_3c_i}, \overrightarrow{d}_{oj}) \leq \theta. \end{cases} \tag{4.10}$$

Then we only need to pick an arbitrary critical value of c_i that satisfies Inequality (4.10) to determine the arrangement.

(2) **(0, 1, 1)**. First, as **(0, 1, 1)** shown in Fig. 4.8, we can give the following equation:

$$\begin{cases} \overrightarrow{n}_{lf} \cdot \overrightarrow{o_1o_2} = 0, \overrightarrow{n}_{lf} \cdot \overrightarrow{n}_{up} = \frac{\tan\alpha\tan\beta}{\sqrt{\sec^2\alpha}\sqrt{\sec^2\beta}}, \\ \overrightarrow{n}_{lf} \cdot \overrightarrow{n}_l = \cos\alpha, \overrightarrow{n}_{up} \cdot \overrightarrow{n}_l = 0, \\ |\overrightarrow{n}_{up}| = 1, |\overrightarrow{n}_{lf}| = 1, |\overrightarrow{n}_l| = 1, \overrightarrow{n}_l // xOy. \end{cases} \tag{4.11}$$

where $\overrightarrow{n}_{lf}$ is the normal vector of left side. With Equa. (4.11), we can obtain an single-variable expression of the intersection line of left and up side. Then, combining Inequality (4.10) and the constraint of γ, we can determine the range of this parameter.

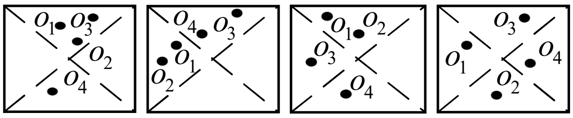
Finally, selecting a legal parameter to determine intersection line, we can determine c_i easily with Inequality (4.10). Therefore, the arrangement $\langle c_i, \overrightarrow{d}_{ci} \rangle$ can be determined. Subcases **(0, 1, 2)** and **(0, 1, 3)** can be solved by the same way. Here, we omit their analysis to save space.

(3) **(0, 0, 3)**. As **(0, 0, 3)** in Fig. 4.8, select a point c_i on the boundary of cell arbitrarily and connect c_io_1, c_io_2, and c_io_3, respectively. Then, this subcase is transformed into **(0, 3, 0)** with two objects on each side. We can obtain the equation

$$\begin{cases} \overrightarrow{n}_{up} \cdot \overrightarrow{c_io_1} = 0, \overrightarrow{n}_{rg} \cdot \overrightarrow{c_io_2} = 0, \overrightarrow{n}_{bt} \cdot \overrightarrow{c_io_3} = 0, \\ \overrightarrow{n}_{up} \cdot \overrightarrow{n}_{rg} = \frac{\tan\alpha\tan\beta}{\sqrt{\sec^2\alpha}\sqrt{\sec^2\beta}}, \overrightarrow{n}_{up} \cdot \overrightarrow{n}_{bt} = \cos 2\beta, \\ |\overrightarrow{n}_{up}| = 1, |\overrightarrow{n}_{rg}| = 1, |\overrightarrow{n}_{bt}| = 1. \end{cases} \tag{4.12}$$

where $\overrightarrow{n}_{rg}$ is the normal vector of right side. Thus, selecting a feasible solution arbitrarily, we can get an arrangement. Finally, execute *projection* until c_i reaching on the boundary of cell to determine the final $\langle c_i, \overrightarrow{d}_{ci} \rangle$. Moreover, subcase **(0, 0, 4)** can be solved by the same way.

- In four-object case, we have two typical subcases.



(1, 0, 1) **(0, 2, 0)** **(0, 1, 2)** **(0, 0, 4)**

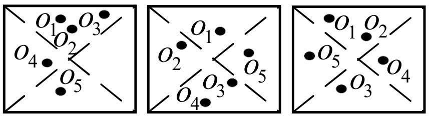
Figure 4.9.: **Typical four-object cases.**

(1) **(1, 0, 1)**. Similar to **(1, 0, 0)**, $\overrightarrow{d}_{ci}$ can be obtained, then normal vectors of four sides are easily to get. As **(1, 0, 1)** in Fig. 4.9, with the normal vector of down side and o_4, we can obtain the intersection line expression of up side and down side. Then, selecting a point on this intersection line and execute *projection*, we can determine the arrangement. Subcases **(1, 1, 0)** in five-object and **(2, 0, 0)** in six-object cases can be solved by the same way.

(2) **(0, 2, 0)**. As **(0, 2, 0)** in Fig. 4.9, combining $\overrightarrow{n}_{up} \cdot \overrightarrow{o_1o_2} = 0$ and Equa. (4.11), we can obtain the intersection line of up side and left side. Then, selecting c_i and $\overrightarrow{d}_{ci}$ by the same way as **(1, 0, 1)**, the arrangement can be determined.

- In five-object case, we have two typical cases.

(1) **(1, 0, 2)**. Similar to **(1, 0, 1)**, we can obtain the orientation $\overrightarrow{d}_{ci}$ and every normal vector of four sides. Thus, with the coordinates of o_4, o_5 and normal vectors of left and down sides as **(1, 0, 2)** in Fig. 4.10, we can determine c_i as well as an

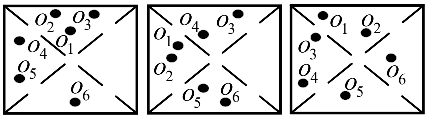


(1, 0, 2) (0, 2, 1) (0, 1, 3)

Figure 4.10.: **Typical five-object cases.**

arrangement. Then, execute *projection* until c_i reaching on the boundary of cell to determine the final $\langle c_i, \overrightarrow{d}_{ci} \rangle$.

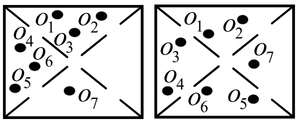
(2) **(0, 2, 1)**. Similar to **(0, 2, 0)**, we can obtain $\overrightarrow{d}_{ci}$ and normal vectors of four sides. With coordinates of o_1, o_3, o_5 and normal vectors of left, down, and right sides, we can obtain the intersection point of these three sides, saying c_i. Then, execute *projection* until c_i reaching on the boundary of cell, we can obtain the final arrangement.



(1, 1, 1) (0, 3, 0) (0, 2, 2)

Figure 4.11.: **Typical six-object cases.**

- Six-object case can be classified into two kinds of subcases. The first is **(2, 0, 0)**, it can be solved by the same way as **(1, 0, 1)**. The second kind includes **(1, 1, 1)**, **(0, 3, 0)**, and **(0, 2, 2)**, which has the only arrangement. Thus, we only need to solve their equations and execute *projection* to obtain the final arrangements.



(2, 0, 1) (0, 3, 1)

Figure 4.12.: **Typical seven-object cases.**

- In seven-object case, every subcase has the only arrangement, no matter how distributed on four sides. Thus, we only need to solve their equations and execute *projection* to obtain the final arrangements. There are two typical subcases in seven-object case. To **(2, 0, 1)**, we can obtain the following equations:

$$\begin{cases} \overrightarrow{n}_{up}\cdot\overrightarrow{o_1o_2}=0, \overrightarrow{n}_{up}\cdot\overrightarrow{o_1o_3}=0, |\overrightarrow{n}_{up}|=1,\\ \overrightarrow{n}_{lf}\cdot\overrightarrow{o_4o_5}=0, \overrightarrow{n}_{lf}\cdot\overrightarrow{o_4o_6}=0, |\overrightarrow{n}_{lf}|=1,\\ \overrightarrow{d}_{ci}\cdot\overrightarrow{n}_{up}=\sin\beta, \overrightarrow{d}_{ci}\cdot\overrightarrow{n}_{lf}=\sin\alpha, |\overrightarrow{d}_{ci}|=1. \end{cases} \tag{4.13}$$

With Equa. (4.13), we can obtain $\overrightarrow{d}_{ci}$. Due to o_7 lying on the bottom side, utilizing the positional relation of $\overrightarrow{c_io_7}$ and $\overrightarrow{n}_{up}$, $\overrightarrow{c_io_7}$ and $\overrightarrow{n}_{lf}$, and c_i must lie on the crossing line of up and left side, we can derive c_i. Then, combine $\overrightarrow{d}_{ci}$, the unique arrangement is obtained, *i.e.*, $\langle c_i, \overrightarrow{d}_{ci}\rangle$. Similarly, to **(0, 3, 1)**, we can obtain the following equations:

$$\begin{cases} \overrightarrow{n}_{up}\cdot\overrightarrow{o_1o_2}=0, \overrightarrow{n}_{lf}\cdot\overrightarrow{o_3o_4}=0, \overrightarrow{n}_{bt}\cdot\overrightarrow{o_5o_6}=0,\\ \overrightarrow{n}_{up}\cdot\overrightarrow{n}_{rg}=\frac{\tan\alpha\tan\beta}{\sqrt{\sec^2\alpha}\sqrt{\sec^2\beta}}, \overrightarrow{n}_{up}\cdot\overrightarrow{n}_{bt}=\cos 2\beta,\\ \overrightarrow{d}_{ci}\cdot\overrightarrow{n}_{up}=\sin\beta, |\overrightarrow{n}_{up}|=1, |\overrightarrow{n}_{lf}|=1, |\overrightarrow{d}_{ci}|=1. \end{cases} \tag{4.14}$$

With Equa. (4.14), we can obtain $\overrightarrow{d}_{ci}$. Similar to **(2, 0, 1)** case, we also can derive c_i then $\langle c_i, \overrightarrow{d}_{ci}\rangle$ as well. Due to space limit, we omit the repeated part.

Based on the above analysis for all cases, we present Alg. 2. Let Γ be the output set of DCSs in Alg. 2, then we have the following theorem.

Theorem 4.2.2. *Given any arrangement* $\langle c_i, \overrightarrow{d}_{ci}\rangle$, *there exists* $\langle c_k, \overrightarrow{d}_{ck}\rangle \in \Gamma$ *such that* $\langle c_k, \overrightarrow{d}_{ck}\rangle \succeq \langle c_i, \overrightarrow{d}_{ci}\rangle$.

Proof. Without loss of generality, we start from searching arrangements for three-object cases. Assuming objects o_1, o_2 and o_3 touching one side of straight rectangle pyramid, we select an arbitrary feasible arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$. Then, keeping the three objects on this side and execute transformations, there exists numerous conditions which can be classified into three classes.

Class 1. *There is the only one arrangement for objects o_1, o_2 and o_3.* Obviously, the selected feasible arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$ is unique, and it must generate the only DCS such that $\langle c_1, \overrightarrow{d}_{c1}\rangle \in \Gamma$.

Class 2. *There won't be any new object touch any side of pyramid.* This condition implies $\langle c_1, \overrightarrow{d}_{c1}\rangle = \langle c_k, \overrightarrow{d}_{k}\rangle (k \neq 1, k \in U)$, where U is the universe of all arrangements. Thus arrangement $\langle c_1, \overrightarrow{d}_{c1}\rangle$ generates DCS such that $\langle c_1, \overrightarrow{d}_{c1}\rangle \in \Gamma$.

Class 3. *Some new object(s) touch some side(s) of pyramid.* Assuming object o_4 touches one side and we arbitrarily select an arrangement $\langle c_2, \overrightarrow{d}_{c2}\rangle$ covering these four objects. Then, there exists two subclasses.

Algorithm 2: DCS Extraction for the Area Case

```
Input: The cell $S_i$, the candidate covered set of objects $\hat{O}_i$
Output: All DCSs
for $i \leq 7$ (number of objects $\leq$ the maximum of minimum number of objects on 4 sides to
  determine the only one arrangement) do
    for every combination $o_{k_1},...,o_{k_i}$ of all objects in $\hat{O}_i$ do
        if $i \geq 3$ then
            for every subcases $a+b+c$ subject to $3a+2b+c=i$ and $a+b+c \leq 4$ do
                for every possible arrangement of 4 sides taken $a+b+c$ to arrange $3,2,1$
                  objects respectively do
                    Execute the process following the corresponding subcase $a+b+c$.
                    if exists corresponding arrangement $\langle c_i, \overrightarrow{d}_{ci} \rangle$ then
                        Add the results to the candidate DCS set.    break
                    end
                end
            end
        end
        Select one point $p$ on the boundary of $S_i$ arbitrarily.
        if $i = 1$ then
            Build arrangement $\langle p, \overrightarrow{d}_p \rangle$ with object $o_k$ on the surface of straight rectangle
              pyramid.
        end
        if $i = 2$ then
            Decide if these objects can be in one camera coverage. If it is, build straight
              rectangle pyramid with line $\overline{po}_{k_1}$ and $\overline{po}_{k_2}$ on the surface.
        end
    end
end
```

Subclass 3.1. $\langle c_2, \overrightarrow{d}_{c2} \rangle = \langle c_1, \overrightarrow{d}_{c1} \rangle$. This indicates that object o_4 has been covered by $\langle c_1, \overrightarrow{d}_{c1} \rangle$, then $\langle c_1, \overrightarrow{d}_{c1} \rangle$ generates the DCS such that $\langle c_1, \overrightarrow{d}_{c1} \rangle \in \Gamma$.

Subclass 3.2. $\langle c_2, \overrightarrow{d}_{c2} \rangle \succeq \langle c_1, \overrightarrow{d}_{c1} \rangle$. This indicates that object o_4 isn't covered by $\langle c_1, \overrightarrow{d}_{c1} \rangle$, thus $\langle c_1, \overrightarrow{d}_{c1} \rangle$ is not a DCS. However, as Alg. 2, when searching arrangement $\langle c_2, \overrightarrow{d}_{c2} \rangle$ for objects o_1, o_2, o_3 and o_4 in the next round of all combination of four objects on the sides, arrangement $\langle c_1, \overrightarrow{d}_{c1} \rangle$ for objects o_1, o_2 and o_3 will be replaced by arrangement $\langle c_2, \overrightarrow{d}_{c2} \rangle$. If no object will touch any side of pyramid during continuous transformation, arrangement $\langle c_2, \overrightarrow{d}_{c2} \rangle$ generates the DCS such that $\langle c_1, \overrightarrow{d}_{c1} \rangle \preceq \langle c_2, \overrightarrow{d}_{c2} \rangle \in \Gamma$. Otherwise, similar to the above, arrangement $\langle c_2, \overrightarrow{d}_{c2} \rangle$ will be replaced by $\langle c_3, \overrightarrow{d}_{c3} \rangle, \ldots, \langle c_k, \overrightarrow{d}_{ck} \rangle$ iteratively until no object will touch any side or there is the only determined arrangement. arrangement $\langle c_k, \overrightarrow{d}_{ck} \rangle$ generates the DCS for $o_1, \ldots, o_k$, as well as, for o_1, o_2 and o_3. Consequently, $\langle c_1, \overrightarrow{d}_{c1} \rangle \preceq \langle c_k, \overrightarrow{d}_{ck} \rangle \in \Gamma$. □

4.2.3. Problem Reformulation and Solution

In this subsection, we discuss about how to select a given number of arrangements (serve as waypoints) from the obtained ones to maximize the number of coverage objects. We first reformulate the problem, then prove its submodularity, and finally present an effective algorithm to address this problem.

Let x_i be a binary indicator denoting whether the i_{th} arrangement in the arrangement set of DCSs Γ is select or not. For all DCSs from all cells in Γ, we can compute the coverage function with each object. The problem **P1** can be reformulated as

$$\begin{aligned} \textbf{(P2)} \quad & max \quad \sum_{j=1}^{M} \mathcal{U}_v \Big(\sum_{\langle c_i, \overrightarrow{d}_{ci} \rangle \in \Gamma} x_i F_v(c_i, o_j, \overrightarrow{d}_{ci}, \overrightarrow{d}_{oj}) \Big), \\ & s.t. \quad \sum_{i=1}^{|\Gamma|} x_i = N (x_i \in \{0,1\}). \end{aligned} \tag{4.15}$$

The problem is then transformed to a combinatorial optimization problem. Now, we give the following definitions to assist further analysis before addressing **P2**.

Definition 4.2.8. *[142] Let S be a finite ground set. A real-valued set function $f : 2^S \to \mathbb{R}$ is normalized, monotonic, and submodular if and only if it satisfies the following conditions, respectively: (1) $f(\emptyset) = 0$; (2) $f(A \cup \{e\}) - f(A) \geq 0$ for any $A \subseteq S$ and $e \in S \backslash A$; (3) $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ for any $A \subseteq B \subseteq S$ and $e \in S \backslash B$.*

Definition 4.2.9. *[142] A Matroid $\mathcal{M}$ is an arrangement $\mathcal{M} = (S, L)$ where S is a finite ground set, $L \subseteq 2^S$ is a collection of independent sets, such that (1) $\emptyset \in L$; (2) if $X \subseteq Y \in L$, then $X \in L$; (3) if $X, Y \in L$, and $|X| < |Y|$, then $\exists y \in Y \backslash X$, $X \cup \{y\} \in L$.*

Definition 4.2.10. *[142] Given a finite set S and an integer k. A uniform matroid $\mathcal{M} = (S, L)$ is a matroid where $L = \{X \subseteq S : |X| \leq k\}$.*

Then, our problem can be reformulated as

PANDA Problem (P3): Given Γ, select X (a M number set) from L such that $f(X)$ maximized

$$\begin{aligned} \textbf{(P3)} \quad & max \; f(X) = \sum_{j=1}^{M} \mathcal{U}_v \Big(\sum_{\langle c_i, \overrightarrow{d}_{ci} \rangle \in X} F_v(c_i, o_j, \overrightarrow{d}_{ci}, \overrightarrow{d}_{oj}) \Big), \\ & s.t. \quad X \in L, \\ & \qquad\; L = \{X \subseteq \Gamma : |X| \leq \mathcal{M}\}. \end{aligned} \tag{4.16}$$

Algorithm 3: Arrangements Selection

```
Input: The number of UAVs $N$, DCSs set $\Gamma$, objective function $f(X)$
Output: arrangement set $X$
$X = \emptyset$.
while $|X| \leq N$ do
    $e^* = \arg\max_{e \in \Gamma \setminus X} f(X \cup \{e\}) - f(X)$.
    $X = X \cup \{e^*\}$.
end
```

Lemma 4.2.2. *The objective function $f(X)$ in **P3** is a monotone submodular function, whose constraint is a uniform matroid.*

Proof. According to Def. 4.2.8, we need to verify the three listed requirements of $f(X)$ in order to prove that it is monotone submodular.

First, when the number of waypoints is 0, obviously $\mathcal{U}_v(\cdot) = 0$, thus we have $f(\emptyset) = 0$.

Second, let A be a set of arrangements in Γ, $e \in \Gamma \setminus A$ and $\varphi(X,i) = \mathcal{U}_v(\sum_{\langle c_i, \overrightarrow{d}_{ci}\rangle \in X} F_v(c_i, o_j, \overrightarrow{d}_{ci}, \overrightarrow{d}_{oj}))$. We first observe that the directional coverage utility $\mathcal{U}(\cdot)$ is non-decreasing. Moreover, it is clear that $\sum_{\langle c_i, \overrightarrow{d}_{ci}\rangle \in A \cup \{e\}} F_v(c_i, o_j, \overrightarrow{d}_{ci}, \overrightarrow{d}_{oj}) \geq \sum_{\langle c_i, \overrightarrow{d}_{ci}\rangle \in A} F_v(c_i, o_j, \overrightarrow{d}_{ci}, \overrightarrow{d}_{oj})$. Then, we have $\varphi(A \cup \{e\}, i) - \varphi(A, i) \geq 0$. Therefore,

$$f(A \cup \{e\}) - f(A) = \sum_{j=1}^{M} (\varphi(A \cup \{e\}, i) - \varphi(A, i)) \geq 0.$$

Third, let A and B be two sets where $A \subseteq B \subseteq \Gamma$ and element $e \in \Gamma \setminus B$. Since $\mathcal{U}_v(\cdot)$ is a binary function, we can analyze using exhaustive approach. If $\mathcal{U}_v(A, \cdot) = 0$ and $\mathcal{U}_v(A \cup \{e\}, \cdot) = 1$, then there must exist $\mathcal{U}_v(B \cup \{e\}, \cdot) = 1$. Moreover, regardless of the value of $\mathcal{U}_v(B, \cdot)$, we always have $(\mathcal{U}_v(A \cup \{e\}, \cdot) - \mathcal{U}_v(A, \cdot)) - (\mathcal{U}_v(B \cup \{e\}, \cdot) - \mathcal{U}_v(B, \cdot)) \geq 0$. For other cases, *i.e.*, $\mathcal{U}_v(A, \cdot) = \mathcal{U}_v(A \cup \{e\}, \cdot)$, there must exist $(\mathcal{U}_v(A \cup \{e\}, \cdot) - \mathcal{U}_v(A, \cdot)) = (\mathcal{U}_v(B \cup \{e\}, \cdot) - \mathcal{U}_v(B, \cdot))$. Thus, $(\varphi(A \cup \{e\}, i) - \varphi(A, i)) - (\varphi(B \cup \{e\}, i) - \varphi(B, i)) \geq 0$. Therefore,

$$\begin{aligned}
&(f(A \cup \{e\}) - f(A)) - (f(B \cup \{e\}) - f(B)) \\
=&\sum_{j=1}^{M} [(\varphi(A \cup \{e\}, i) - \varphi(A, i)) - (\varphi(B \cup \{e\}, i) - \varphi(B, i))] \geq 0.
\end{aligned}$$

To sum up, $f(X)$ is a monotone submodular function. □

Therefore, the reformulated problem falls into the scope of maximizing a monotone submodular function subject to matroid constraints, and we can use a greedy algorithm to achieve a good approximation [142]. The pseudo code of this arrangement selecting algorithm is shown in Alg. 3. In every round, Alg. 3 greedily adds an arrangement e^* to X to maximize the increment of function $f(X)$. We omit the proof to save space.

4.3. Theoretical Analysis

Theorem 4.3.1. *Algorithm PANDA achieves an approximation ratio of* $1-1/e$, *and its time complexity is* $O(NM^9)$.

Proof. First, we bound the approximation ratio of PANDA algorithm. Denote the overall directional coverage utility for all N UAVs under optimal solution to problem **P1** and the reformulated problem **P2** as $\mathbf{OPT}_{p2}$ and $\mathbf{OPT}_{p1}$, respectively. According to Coro. 4.2.1 and Theo. 4.2.2, Alg. 2 extracts all DCSs without loss. Thus, $\mathbf{OPT}_{p1} = \mathbf{OPT}_{p2}$. Denote the overall directional coverage utility by Alg. 3 to the problem **P2** (or **P3**) as **SOL**. According to the fact that a greedy algorithm of maximizing a monotone submodular function subject to a uniform matroid achieves $1-1/e$ approximation ratio [142], thus the approximation ratio of PANDA is $1-1/e$, formally,

$$\frac{SOL}{OPT_{p1}} = \frac{SOL}{OPT_{p2}} = 1 - \frac{1}{e}, \tag{4.17}$$

where e is the Napier's constant.

Next, we analyze the time complexity of PANDA. First, PANDA computes the total number of cells intersected by the cone of each object in 3D space, whose complexity is $O(M^2)$ according to Theo. 4.2.1. Second, in each cell, Alg. 2 will extract DCSs for each subcase. The total number of case is $\sum_{i=1}^{7}\binom{M}{i} = O(M^7)$ and each case has $O(1)$ subcases which generate the corresponding arrangements. Thus, the total time complexity of Alg. 2 is $O(M^7)$. Last, Alg. 3 will perform N times loop and in each time it will select the best one from the current remaining arrangements. Thereby, the time complexity of PANDA algorithm is $O(NM^9)$. □

4.4. Numerical Simulation

4.4.1. Evaluation Setup

In our simulation, objects are uniformly distributed in a $100m \times 100m \times 50m$ cuboid space. If no otherwise stated, we set $\alpha = \pi/3$, $\beta = \pi/12$, $\Delta = 25m$, $N = 10$, $\gamma_{min} = \pi/6$, $\gamma_{max} = \pi/3$, $\theta = \pi/6$, and $M = 20$, respectively. Note that both of the orientations of cameras and objects are considered with respect to the North. The orientations of objects are randomly selected from $[0, 2\pi]$ in horizonal plane and $[0°, 90°]$ in vertical plane. Each data point in evaluation figures is computed by averaging the results of 200 random topologies. As there are no existing approaches for PANDA, we compare four algorithms including three presented algorithms and an existing algorithm VPFCEA proposed by [99]. VPFCEA algorithm solves the optimal coverage problem on 2D plane. Randomized Coordinate with Orientation Discretization (RCOD) randomly generates coordinates of UAVs, and randomly selects orientation of UAVs from $\{0, \alpha, ..., k\alpha, ..., 2\pi\}$ in horizonal plane and $\{\gamma_{min}, \gamma_{min} + \beta, ..., \gamma_{min} + \lfloor(\gamma_{max} - \gamma_{min})/\beta\rfloor\beta, ..., \gamma_{max}\}$ in vertical plane. Grid Coordinate with Orientation Discretization (GCOD) improves RCOD by placing the UAVs at grid points. Grid Coordinate with Dominating Coverage Set (GDCS) further improves GCOD. It utilize DCS extraction algorithm for point case to generate candidate orientations and greedily selects the orientation with best coverage utility.

4.4.2. Performance Comparison

Impact of Number of Waypoints N. *Our simulation results show that on average, PANDA outperforms RCOD, GCOD, GDCS, and VPFCEA by* 12.35 *times,* 10.27 *times,* 3.51 *times, and* 87.56% *respectively, in terms of N.* Fig. 4.13 shows that the coverage utility for all algorithms increase monotonically with N. In particular, the coverage utility of PANDA first fast increases and approaches 1 when $N = 15$, and then becomes stable. GDCS increases relatively linearly because it can only choose among given grid coordinates for placing UAVs. VPFCEA performs better than GDCS because it can plan waypoints at any location, but much worse than PANDA because it only considers the object located on 2D plane. In contrast, the coverage utility of RCOD and GCOD always remain low, because their candidate coordinates of UAVs are limited and orientations are predetermined or randomly generated.

Impact of Number of Objects M. *Our simulation results show that on average, PANDA outperforms RCOD, GCOD, GDCS, and VPFCEA by* 12.37 *times,* 11.74

times, 41.5%*, and* 42.3%*, respectively, in terms of* M. From Fig. 4.14, the coverage utility decreases monotonically with increasing M. PANDA first performs well for no more than 13 objects but then decreases when M is larger than 13. The decreasing rate tends to be gentle and around 0.8. In contrast, GDCS and VPFCEA invariably degrades while RCOD and GCOD always keep low performance.

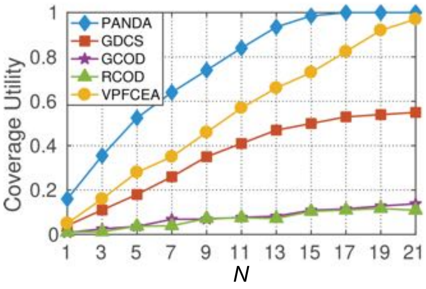


Figure 4.13.: **Number of Waypoints** N ***vs.* utility.**

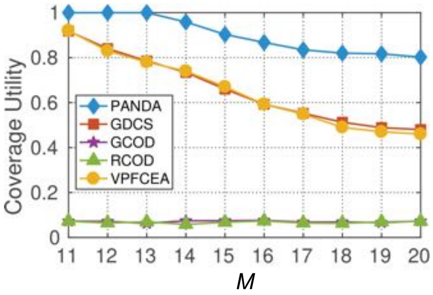


Figure 4.14.: **Number of Objects** M ***vs.* utility.**

Impact of Efficient Angle θ**.** *Our simulation results show that on average, PANDA outperforms RCOD, GCOD, GDCS, and VPFCEA by* 10.01 *times,* 9.94 *times,* 110.36%*, and* 86.36%*, respectively, in terms of* θ. As shown in Fig. 4.15, the coverage utility of four algorithms increases monotonically with θ. The coverage utility of PANDA first increases at a fast speed and approaches 1 when θ increases from 10° to 60°, and then keeps stable. However, the other four comparison algorithms increase slowly.

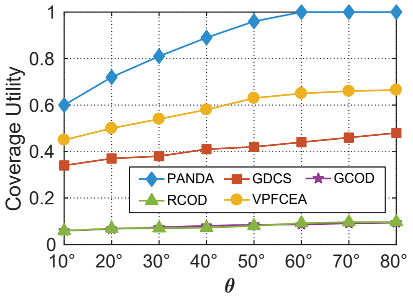


Figure 4.15.: **Efficient Angle** θ ***vs.* utility.**

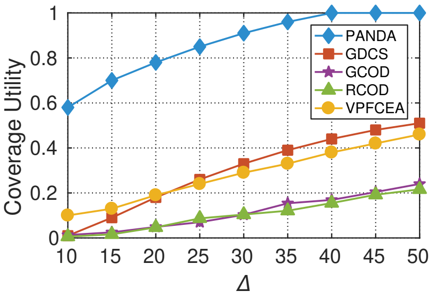


Figure 4.16.: **Farthest sight Distance** Δ ***vs.* utility.**

Impact of Farthest Sight Distance Δ**.** *Our simulation results show that on average, PANDA outperforms RCOD, GCOD, GDCS, and VPFCEA by* 11.20 *times,* 13.53 *times,* 84.35%*, and* 82.35%*, respectively, in terms of* Δ. Fig. 4.16 shows that the coverage utility of PANDA invariably increases with Δ until it approaches 1, while that of RCOD, GCOD, GDCS, and VPFCEA increase to about 0.2, 0.2, 0.55, and 0.55 respectively, and then keeps relatively stable.

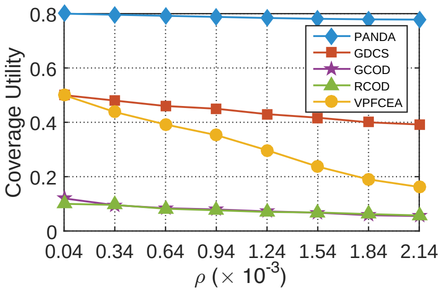


Figure 4.17.: **Density ρ *vs.* utility.**

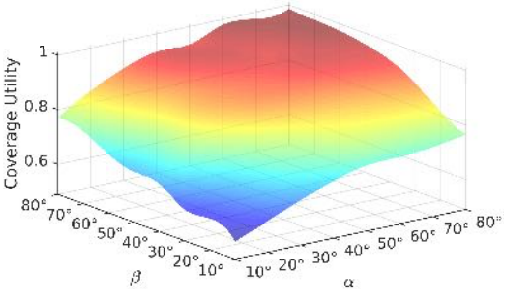


Figure 4.18.: **Horizontal Angle α and Vertical Angle β *vs.* utility.**

Impact of Density ρ. *Our simulation results show that on average, PANDA outperforms RCOD, GCOD, GDCS, and VPFCEA by* 6.18 *times,* 6.64 *times,* 81.35%*, and* 97.65%*, respectively, in terms of ρ (= number of objects ÷ volume of whole space).* Fig. 4.17 shows that the coverage utility of PANDA fluctuates slightly when ρ grows, but it is almost always near 0.8. The coverage utility of GDCS also fluctuates slightly, because GDCS uses our DCS Extraction algorithm at each grid. VPFCEA decreases a lot, because it doesn't consider the objects distributed in 3D space. When ρ increases, it covers objects much more difficultly. RCOD and GCOD always maintain bad coverage utility, because along with rho increasing they can cover objects more easier but the number of objects also increase. As coverage utility is the covered number of objects divides the sum number of objects, randomized planning like RCOD and GCOD can't get high coverage utility.

Impact of Horizontal Angle α and Vertical Angle β. Here we study the impact of α and β on coverage utility. Suppose the horizontal offset angle α and vertical offset angle β vary from 10° to 80°, respectively. Fig. 4.18 depicts the results and each point on the surface denote an average value of 100 experiment results. We observe that the coverage utility increases monotonically while either α or β increases. Indeed, with a larger α or β, the 3D coverage space gets larger and more potential objects can be covered.

Table 4.2.: **Coordinate and orientation of objects.**

Object	Coordinate	Orientation	Object	Coordinate	Orientation
o_1	(19.4,0.7.9,5.8)	$(7\pi/4,\pi/2)$	o_9	(83.0,2.7.9,5.0)	$(0,\pi/2)$
o_2	(21.0,5.9,3.3)	$(4\pi/5,2\pi/6)$	o_{10}	(84.3,3.2,4.6)	$(\pi/4,\pi/3)$
o_3	(2.1,0.9,5.8)	$(0,\pi/6)$	o_{11}	(81.5,19.3,1.7)	$(3\pi/4,\pi/2)$
o_4	(9.6,1.2,5.4)	$(3\pi/4,0)$	o_{12}	(16.2,66.9,1.7)	$(\pi/2,0)$
o_5	(11.4,3.9,4.2)	$(3\pi/4,0)$	o_{13}	(9.94,53.4,0.5)	$(\pi/2,\pi/2)$
o_6	(18.7,2.9,4.6)	$(\pi/4,\pi/6)$	o_{14}	(83.2,28.9,0.5)	$(\pi,\pi/3)$
o_7	(3.0,6.1,3.3)	$(3\pi/2,\pi/2)$	o_{15}	(36.6,63.8,0.5)	$(3\pi/4,0)$
o_8	(84.4,3.0.9,4.6)	$(\pi,\pi/6)$			

4.5. Field Experiment

(a) UAV (b) Object (c) Experiment site

Figure 4.19.: **Testbed.**

As shown in Fig. 4.19, our testbed consists of 7 DJI Phantom 4 advanced UAVs and 15 randomly distributed face figures as objects, and our experimental site is the playground of our school including its stands, whose size is $110m \times 80m$. Specifically, we set $\alpha = 35°$, $\beta = 20°$, $\Delta = 10m$, $\gamma_{min} = 10°$, $\gamma_{max} = 70°$, and $\theta = \pi/6$ based on real hardware parameters. The orientations of objects (θ, φ) are randomly generated where $\theta \in \{k\pi/4, k \in \{1,2,3,4,5,6,7,8\}\}$ is the angle between orientation and xOz and $\varphi \in \{k\pi/6, k \in \{0,1,2,3\}\}$ is the angle between orientation and xOy. Table 4.2 lists the obtained coordinates and orientations of all objects. Moreover, we draw a circle around the face on each face figure as shown in Fig. 4.19(b) to help

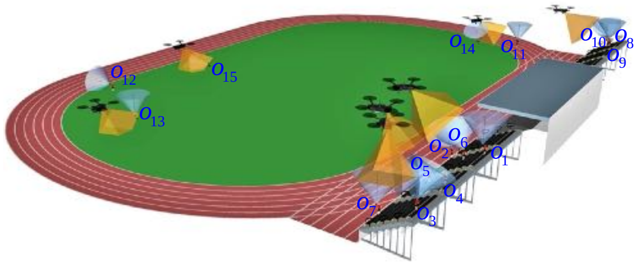


Figure 4.20.: **Objects distribution and waypoint planning by PANDA.**

demonstrate the coverage result by observing the circle's distortion degree.

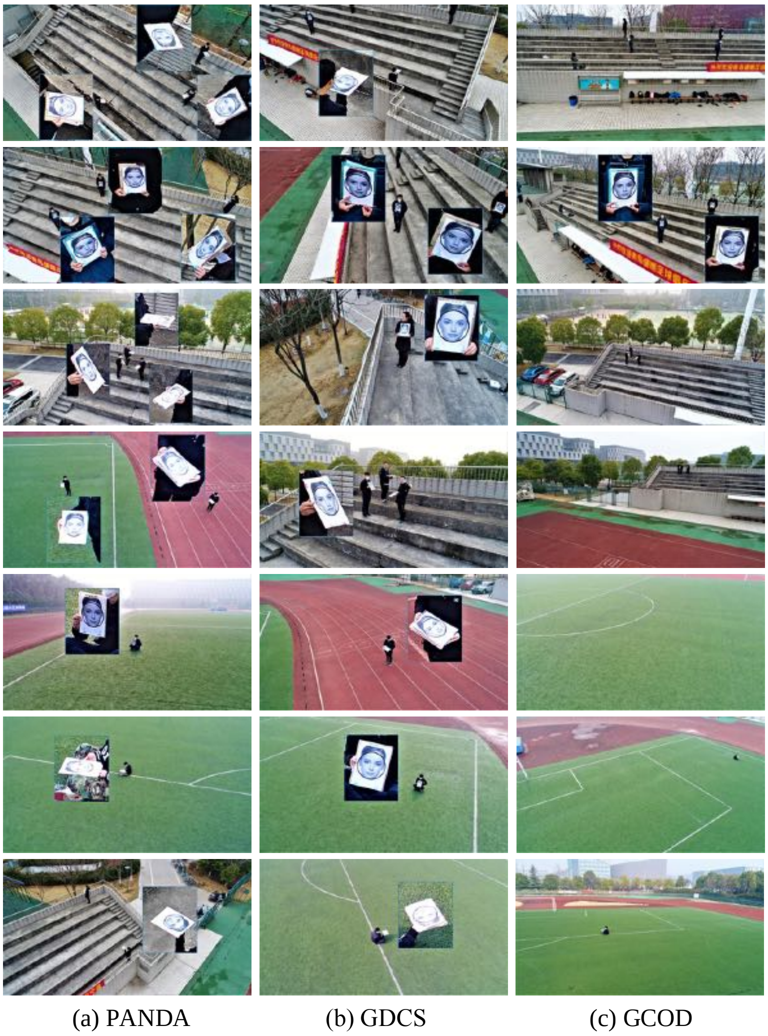

(a) PANDA (b) GDCS (c) GCOD

Figure 4.21.: **Experimental results of different algorithms.**

We respectively execute PANDA, GDCS, and GCOD offline and obtain their corresponding strategies. Fig. 4.20 illustrates the waypoint planning results for PANDA. Note that the spherical base cones of objects are depicted in grey while the straight rectangle pyramids of UAVs are in yellow. Fig. 4.21 shows the 7 pictures took by 7 UAVs for each of the three algorithms PANDA, GDCS, and GCOD. The rectangular enlarged views in each figure demonstrate the details of successfully efficient covered objects for the corresponding UAV. From Fig. 4.21, we can see that PANDA covers the most objects among all the three algorithms. The coverage utilities of PANDA, GDCS, and GCOD are 0.93, 0.53, and 0.20, respectively, which means PANDA outperforms GDCS and GCOD by 75.4% and 3.65 times.

4.6. Chapter summary

We solve the problem of 3D waypoint planning of UAVs to achieve directional coverage. The key novelty of PANDA is on proposing the first algorithm for waypoint planning with optimized directional coverage utility in 3D environment. The key contribution of PANDA is building the practical 3D directional coverage model, developing an approximation algorithm, and conducting simulation and field experiments for evaluation. The key technical depth of PANDA is in reducing the infinite solution space of this optimization problem to a limited one by utilizing the techniques of space partition and Dominating Coverage Set extraction, and modeling the reformulated problem as maximizing a monotone submodular function subject to a matroid constraint. Our evaluation results show that our algorithm outperforms the other comparison algorithms by at least 75.4%.

Chapter 5

Waypoints Planning for Anisotropic Visual Tasks

This chapter explores the correlation between current computer vision task and the quality of visual data. It solves the fundamental problem VISIT, that is, given a set of objects with determined coordinates and directions in 2D area, plan a fixed number of waypoints maximize the overall monitoring utility for all objects.

Contents

5.1. Problem Formulaion

5.1.1. Efficient Monitoring Model

Suppose N objects $\mathcal{O} = \{o_1, o_2, ..., o_N\}$ are deterministically distributed on a 2D plane Ω with known coordinates o_j and orientations $\overrightarrow{d}_{oj}$. We set M waypoints $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ of the UAV which can hover anywhere on Ω with any orientation $\overrightarrow{d}_{uj}$. The tuple $\langle u_i, \overrightarrow{d}_{ui} \rangle$, saying *strategy*, where u_i denotes the coordinate of waypoint i and $\overrightarrow{d}_{ui}$ denotes its orientation. By a little abuse of notation, u_i and o_j also denotes waypoint i and object j. Table 5.1 lists the notations we used.

Table 5.1.: **Notations used in Chap. 5.**

Symbol	Meaning
u_i	UAV i, or its coordinate
o_j	Object j to be monitored, or its coordinate
$\mathcal{U}$	Set of all UAVs
$\mathcal{O}$	Set of all objects
$\overrightarrow{d}_{ui}$	Orientation of UAV i
s_k	Strategy k $\langle u_i, \overrightarrow{d}_{ui} \rangle$ of UAV i with orientation $\overrightarrow{d}_{ui}$
A_l	Subarea l formed by a set of discretized sectors of O_l
$\mathcal{S}$	Set of selected strategies
γ	Monitoring angle around $\overrightarrow{d}_{ui}$
$\overrightarrow{d}_{oj}$	Orientation of object o_j
θ	Efficient angle around $\overrightarrow{d}_{oj}$ for monitoring
d_{min}	Minimum distance between UAV and object for safe
D	Monitoring distance of camera of UAV
ω	Key information in ω angle around object can be captured by a UAV
β	Distribution angle of key information of object

By incorporating the widely accepted empirical camera coverage sector model in [1, 63, 70, 72], we give the efficient monitoring definition as follows. In fact, to guarantee the safety for both objects and UAV, we should build the covered model of object as sector ring. Namely, the UAV should not fly in the area with a distance less than d_{min}. However, this safe distance (1-2m) is much shorter than Monitoring distance of UAV (30-40m) [143], thus we still use the sector model.

Definition 5.1.1. ***(Efficient monitoring)*** *An object o_j is efficiently monitored if for a given vector $\overrightarrow{d}(x,y)$ (its facing direction), there is a UAV u_i* [10]*, such that o_j is monitored by u_i and $\alpha(\overrightarrow{d},\overrightarrow{o_ju_i}) \leq \theta$ (θ is called the effective angle).*

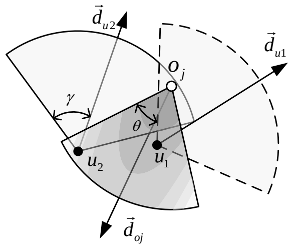


Figure 5.1.: **Efficient monitoring.**

According to the camera coverage sector model and Def. 5.1.1, a UAV u_i with orientation $\overrightarrow{d}_{ui}$ monitors objects with non-zero QoM in a shape of a sector with *monitoring angle* γ and *radius D*, saying *monitoring area.* An object o_j with orientation $\overrightarrow{d}_{oj}$ can be efficiently monitored with non-zero QoM in a shape of a sector with *efficient angle* θ and the same *radius D*, saying *monitored area* . Fig. 5.1 illustrates two pairs of efficient monitoring, u_1 with o_j and u_2 with o_j.

Image resolution can be defined as the ratio of the number of pixels and the size of image, whose unit is Pixels Per Inch (PPI) [12, 13]. To obtain high QoM UAVs need to capture high-resolution images of objects. Based on the definition of resolution, within the appropriate distance between objects and UAVs, the closer the distance, the more pixels of objects can be captured in the image. Particularly, the number of pixels of object is inversely proportional to the square of the decrease in distance between objects and UAVs, and the number of pixels of object's frontal view is inversely proportional to the decrease in monitoring angle, *i.e.*, $\alpha(\overrightarrow{d}_{oj},\overrightarrow{o_jc_i})$. Therefore, the QoM of efficient monitoring pair u_i and o_j can be modeled as follows.

$$\begin{aligned}&\mathcal{Q}(u_i,o_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{oj})\\&=\begin{cases}\frac{a}{(||u_io_j||+b)^2}\cos(\alpha(\overrightarrow{d}_{oj},\overrightarrow{o_ju_i})), 0 \leq ||u_io_j|| \leq D,\\ \qquad\qquad\qquad \overrightarrow{u_io_j}\cdot\overrightarrow{d}_{ui} - ||u_io_j||\cos\gamma \geq 0,\\ \qquad\qquad and \quad \overrightarrow{o_ju_i}\cdot\overrightarrow{d}_{oj} - ||o_js_i||\cos\theta \geq 0.\\ 0, \qquad\qquad otherwise.\end{cases}\end{aligned} \tag{5.1}$$

where a and b are two constants determined by the environment and the hardware, such as the electromagnetic interference from nearby base stations, the color contrast of cameras, the stability of UAV's air posture (these kinds of bias constant are also imported into the models of many other works [144, 145]). $||u_io_j||$ denotes the distance between u_i and o_j, and $\alpha(\overrightarrow{d}_{oj},\overrightarrow{o_ju_i})$ is the included angle between $\overrightarrow{d}_{oj}$

[10] We use UAV and waypoint interchangeably.

and $\overrightarrow{o_j u_i}$. Here using $\cos(\cdot)$ function is only for simplicity, because it represents the decreasing QoM by increasing monitoring angle. Other functions conforming to this characteristic can also work well.

5.1.2. Fusion Function

Multiple images from different views can provide different information for one object, thus fusing multi-viewed images can help us monitor this object better. However, fusing information is not simply linear superposition because images of an object captured from nearby strategies are often highly correlated.

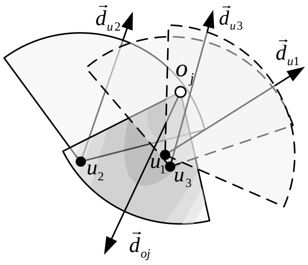


Figure 5.2.: **Simple Case.**

As shown in Fig. 5.2, images captured by $\langle u_1, \overrightarrow{d}_{u1}\rangle$ and $\langle u_3, \overrightarrow{d}_{u3}\rangle$ both monitor the left side of o_j from almost the same view angle, while $\langle u_2, \overrightarrow{d}_{u2}\rangle$ monitors the right side of o_j. In other words, images captured by $\langle u_1, \overrightarrow{d}_{u1}\rangle$ is highly correlated with $\langle u_3, \overrightarrow{d}_{u3}\rangle$ but lowly correlated with $\langle u_2, \overrightarrow{d}_{u2}\rangle$. Although $\langle u_3, \overrightarrow{d}_{u3}\rangle$ alone monitors o_j better than $\langle u_2, \overrightarrow{d}_{u2}\rangle$ alone, UAV flies at $\langle u_1, \overrightarrow{d}_{u1}\rangle$ and $\langle u_2, \overrightarrow{d}_{u2}\rangle$ can provide more information than $\langle u_1, \overrightarrow{d}_{u1}\rangle$ and $\langle u_3, \overrightarrow{d}_{u3}\rangle$.

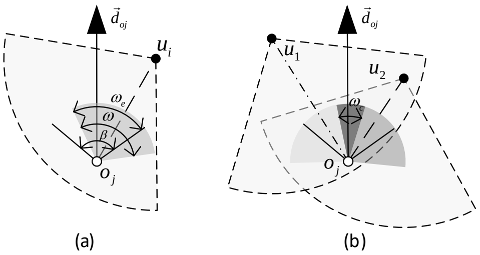


Figure 5.3.: **Fusion model.**

We use the amount of common information obtained by multiple strategies to quantify their correlation. Fig. 5.3(a) illustrates the key information distribution angle range β, the UAV extraction angle range ω, and the effective extraction angle range ω_e. The key information is distributed in the β angle range of o_j's facing direction. Each u_i can capture ω angle range of information of o_j but only the part of ω range in β captures key information, saying ω_e. β and ω are decided by types of objects of lens of cameras, thus they are different in different applications. For example, β in face recognition application is much smaller than in action recognition application because the key information is distributed on face while action can be also captured from the back view of human being. Fig. 5.3(b) and 5.3(c) give an instance. u_1 and u_2 in (b) and (c) respectively capture ω angle range information of o_j, but their common monitoring angle ω_c are different. Because ω_c in (b) is smaller than in (c), the strategies of u_1 and u_2 in (c) are more correlated. We quantify the correlation as $\frac{\omega_c}{\beta}$. If u_i monitors a set of objects $\mathcal{O}_i$ and u_j monitors a set of objects $\mathcal{O}_j$, where $\mathcal{O}_i \cap \mathcal{O}_j = \mathcal{O}_k$, then the correlation model can be expressed as:

$$\begin{aligned}
&\mathcal{K}(u_i, u_j, \overrightarrow{d}_{ui}, \overrightarrow{d}_{uj}, \mathcal{O}_k) = \frac{1}{(|\mathcal{O}_i| + |\mathcal{O}_j|)\beta} \sum_{k=1}^{|\mathcal{O}_k|} \int_{\overrightarrow{d}_{ok} - \frac{\beta}{2}}^{\overrightarrow{d}_{ok} + \frac{\beta}{2}} \omega_c d(\overrightarrow{v}) \\
&= \frac{1}{(|\mathcal{O}_i| + |\mathcal{O}_j|)\beta} \sum_{k=1}^{|\mathcal{O}_k|} (\omega - \alpha(\overrightarrow{o_k u_i}, \overrightarrow{o_k u_j})), \\
&s.t. \quad \mathcal{Q}(u_i, o_k, \overrightarrow{d}_{ui}, \overrightarrow{d}_{ok}) \neq 0, \mathcal{Q}(u_j, o_k, \overrightarrow{d}_{uj}, \overrightarrow{d}_{ok}) \neq 0,
\end{aligned} \tag{5.2}$$

where, $|\mathcal{O}_k|$ is the size of $\mathcal{O}_k$ and $\overrightarrow{v}$ is the angle variable changing from $\overrightarrow{d}_{ok} - \frac{\beta}{2}$ to $\overrightarrow{d}_{ok} + \frac{\beta}{2}$. If there exists no same object monitored by u_i and u_j, their correlation $\mathcal{K}(\cdot) = 0$.

It needs to be mentioned that other correlation models, such as using sampling data training correlation model [146] and establishing correlation model with classic mathematical models [147, 148], are also suitable for our solution framework.

5.1.3. Monitoring Utility - Variance Reduction Model

Different subsets of strategies provide different information of objects. The more different high-quality information is captured, the better anisotropic monitoring performance can be obtained. Therefore, we need to monitor the objects not only with higher QoM but also with a lower correlation of capturing key information.

Fundamental of Variance Reduction. Gaussian Process (GP) is a powerful tool to illustrate our real world. An important property of GP is that given a set of random variables $\mathcal{S}$ follows GP, the joint distribution over its subset $\mathcal{A} \in \mathcal{S}$ is also Gaussian. Assuming we measure a set of data d_A corresponding to the subset $\mathcal{A}$, based on this property we can estimate the value at every point $y \in \mathcal{S}$ conditioned on these data, $P(y|\mathcal{A})$. Meanwhile, the entropy of a Gaussian random variable y conditioned on a set of Gaussian random variables $\mathcal{A}$ can be expressed as [149]:

$$H(y|\mathcal{A}) = \frac{1}{2} log((2\pi e)\sigma^2_{y|\mathcal{A}}). \tag{5.3}$$

It only depends on the covariance $\sigma^2_{y|\mathcal{A}}$. And according to the Probability Theory, the conditional covariance is given by:

$$\sigma^2_{y|\mathcal{A}} = \sigma^2_s - \Sigma_{y\mathcal{A}} \Sigma^{-1}_{\mathcal{A}\mathcal{A}} \Sigma_{\mathcal{A}y}, \tag{5.4}$$

where $\Sigma_{\mathcal{A}\mathcal{A}}$ is the covariance matrix of $\mathcal{A}$ with itself and $\Sigma_{y\mathcal{A}} = \Sigma^T_{\mathcal{A}y}$ is a row vector of the covariances of y with all variables in $\mathcal{A}$.

Variance Reduction. *Variance reduction* is a typical method for the optimal entropy problem which is widely adopted in previous work [150, 151]. Different strategies provide different reductions in the variance of the accuracy of recognition. The higher reduction of variance, the higher monitoring performance can be obtained. Moreover, according to the Entropy Theory, given a fixed covariance matrix, the conditional covariance does not depend on the actual observed values $\mathcal{A}$. This provides us with a good opportunity to plan waypoints in advance.

Because the practical factors such as the limited accuracy of GPS, the bias of orientation, the influence of wind in the air and the noise over the channel for image transmission [152], the captured images are biased. We assume the aggregate effect of these factors follows Gaussian distribution which is widely accepted in many literatures [153, 154].

We can establish the kernel matrix as follows:

$$\Sigma = \begin{pmatrix} \Sigma_{\mathcal{O}\mathcal{O}} & \Sigma_{\mathcal{O}\mathcal{U}} \\ \Sigma_{\mathcal{U}\mathcal{O}} & \Sigma_{\mathcal{U}\mathcal{U}} \end{pmatrix},$$

where,

$$\Sigma_{\mathcal{U}\mathcal{U}} = \begin{pmatrix} \mathcal{K}(u_1,u_1) & \mathcal{K}(u_1,u_2) & \dots & \mathcal{K}(u_1,u_m) \\ \mathcal{K}(u_2,u_1) & \mathcal{K}(u_2,u_2) & \dots & \mathcal{K}(u_2,u_m) \\ \vdots & \vdots & & \vdots \\ \mathcal{K}(u_m,u_1) & \mathcal{K}(u_m,u_2) & \dots & \mathcal{K}(u_m,u_m) \end{pmatrix} \tag{5.5}$$

in which $\mathcal{K}(u_i,u_j)$ is the abbreviation of $\mathcal{K}(u_i,u_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{uj},\mathcal{O}_k)$, and

$$\Sigma^T_{\mathcal{U}\mathcal{O}} = \Sigma_{\mathcal{O}\mathcal{U}} = \begin{pmatrix} \mathcal{Q}(o_1,u_1) & \mathcal{Q}(o_1,u_2) & \dots & \mathcal{Q}(o_1,u_m) \\ \mathcal{Q}(o_2,u_1) & \mathcal{Q}(o_2,u_2) & \dots & \mathcal{Q}(o_2,u_m) \\ \vdots & \vdots & & \vdots \\ \mathcal{Q}(o_n,u_1) & \mathcal{Q}(o_n,u_2) & \dots & \mathcal{Q}(o_n,u_m) \end{pmatrix} \tag{5.6}$$

in which $\mathcal{Q}(o_j,u_i)$ is the abbreviation of $\mathcal{Q}(u_i,o_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{oj})$. Thus, to objects set $\mathcal{O}$ and UAV set $\mathcal{U}$, the conditional covariance is:

$$\sigma^2_{\mathcal{O}|\mathcal{U}} = tr(\Sigma_{\mathcal{O}\mathcal{O}}) - tr(\Sigma_{\mathcal{O}\mathcal{U}} \Sigma^{-1}_{\mathcal{U}\mathcal{U}} \Sigma_{\mathcal{U}\mathcal{O}}). \tag{5.7}$$

where $tr(\cdot)$ is the *trace function* of a matrix.

We state that the assumption of Equa. (6.2) and Equa. (6.3) is only one kind of metric to quantify the QoM and correlation. Generally speaking, the solution and algorithms proposed in VISIT can be applied to any QoM and fusion function.

5.1.4. Problem Formulation

Recall our ultimate objective is to maximize the accuracy of recognition of objects and the given condition is the distribution of objects. According to the Variance Reduction, if we want to maximize the accuracy of recognition, we need to minimize the variance of $\mathcal{O}$ given $\mathcal{U}$. Namely, we need to maximize the negation of the variance. Combine the $tr(\Sigma) = tr(\Sigma_{\mathcal{O}\mathcal{O}}) + tr(\Sigma_{\mathcal{U}\mathcal{U}})$, which is,

$$maximize \quad tr(\Sigma_{\mathcal{U}\mathcal{U}}) + tr(\Sigma_{\mathcal{O}\mathcal{U}}\Sigma_{\mathcal{U}\mathcal{U}}^{-1}\Sigma_{\mathcal{U}\mathcal{O}}). \tag{5.8}$$

Then, we define the overall monitoring utility as Equa. (5.8), thus our task is to find the optimal strategies for all M UAVs to maximize the overall monitoring utility. With all above, the waypoint planning of Unmanned Aerial V̲ehI̲cles for aniS̲otropic monI̲toring T̲asks (VISIT) problem is defined as follows.

VISIT Problem (P1):

$$\textbf{(P1)} \quad \begin{aligned} max \quad & tr(\Sigma_{\mathcal{U}\mathcal{U}}) + tr(\Sigma_{\mathcal{O}\mathcal{U}}\Sigma_{\mathcal{U}\mathcal{U}}^{-1}\Sigma_{\mathcal{U}\mathcal{O}}), \\ s.t. \quad & |U| = M. \end{aligned}$$

where, $tr(\cdot)$ is the trace function of matrix, $\Sigma_{\mathcal{U}\mathcal{U}}$ and $\Sigma_{\mathcal{O}\mathcal{U}}$ are establish as Equa. (5.5) and (5.6), and $\Sigma_{\mathcal{U}\mathcal{U}}^{-1}$ is the inverse matrix of $\Sigma_{\mathcal{U}\mathcal{U}}$. In the following theorem, we prove the VISIT problem is NP-hard.

Theorem 5.1.1. *The VISIT problem is NP-hard.*

Proof. To show the difficulty of the VISIT problem, we consider a simple case in which $\omega = \beta = \gamma = \theta = 2\pi$, $D = 1$ and the QoM of efficient monitoring for each pair of UAV and object is the same. Namely, as long as an object is efficiently monitored by a UAV and its monitoring utility has reached the maximum value, adding any other UAVs will not improve its monitoring utility. Our VISIT problem changes to using a fixed number of disks with radius of 1 to cover as many as objects in a 2D plane, which is exactly the NP-hard Unit Disk Coverage problem [14].

If we can propose a polynomial algorithm to address the original problem VISIT, obviously, we can address the NP-hard Unit Disk Coverage problem with this same algorithm. However, one NP-hard problem cannot be addressed in polynomial time unless $P = NP$. Therefore, the VISIT problem is NP-hard. □

5.2. Solution

In this section, we present an algorithm with approximation ratio $1 - 1/e - \epsilon$ to address VISIT which consists of three steps. First, we approximate the QoM as a piecewise constant function of distance and angle. By doing so, the monitoring region is divided into many subareas in which the distance between real QoM and approximate QoM is bounded by ϵ, in which the approximated QoM at any point in each subarea becomes constant. Second, we reuse the idea from Sec. 4.2.2 in Chap. 4, like the "divide and conquer" method, in which extracts Monitoring Dominating Set and transform the problem into combinatorial optimization problem. Third, we prove that the transformed problem falls into the realm of maximizing a monotone submodular optimization problem subject to a uniform matroid constraint, and propose a greedy algorithm to solve VISIT problem.

5.2.1. Area Discretization

In this section, we approximate the QoM as a piecewise constant function and bound the approximation error in each interval (see Theo. 5.2.1 and Theo. 5.2.2). Then the fusion function and the number subarea can be bounded by a polynomial of the same approximation error (see Theo. 5.2.3 and Theo. 5.2.4).

Piecewise Constant Approximation of QoM. Let $\mathcal{Q}(d,\alpha)$ denote the QoM of an object is monitored by a UAV with distance d and angle α, *i.e.*, $\mathcal{Q}(d,\alpha) = \frac{a}{(d+b)^2}\cos\alpha$ where $0 \le d \le D, 0 \le \alpha \le \theta$, and $\mathcal{Q} = 0$ otherwise. We use multiple piecewise constant segments $\tilde{\mathcal{Q}}(d,\alpha)$ to approximate $\mathcal{Q}(d,\alpha)$ and bound the approximation error and the computational overhead.

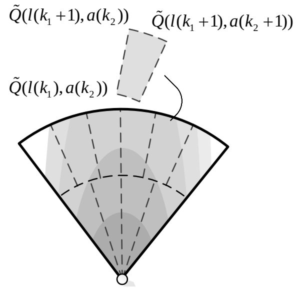

$\tilde{Q}(l(k_1+1), a(k_2))$
$\tilde{Q}(l(k_1+1), a(k_2+1))$
$\tilde{Q}(l(k_1), a(k_2))$

Figure 5.4.: **Approximation.**

Fig. 5.4 depicts the key idea of the approximation method of $\mathcal{Q}(d,\alpha)$. Let $l(0), l(1), \cdots, l(K_1)$ be the end points in distance domain of K_1 constant segments, which divides the sector of objects into K_1 sector rings. In each sector ring, *i.e.* between $l(k)$ and $l(k+1)$, let $a(0), a(1), \cdots, a(K_2)$ be the end points in angle domain of constant segments, which divides each sector ring into K_2 segments both sides of $\overrightarrow{d}_{oj}$. Thus, the sector of object is divided into $K = 2\sum_{j=0}^{K_1-1} K_2$ number of segments. For example, in Fig. 5.4, K_1 is set to 2 and K_2 for $l(0) \leq d \leq l(1)$ and $l(1) \leq d \leq l(2)$ are set to 2 and 3 respectively. Obviously, when K is larger, the less approximation error but more computational overhead is introduced.

Definition 5.2.1. *Setting $l(0) = 0$, $l(K_1) = D$, $a(0) = 0$ and $a(K_2) = \theta$, the piecewise constant QoM function $\tilde{\mathcal{K}}(d, \alpha)$ can be defined as:*

$$\tilde{\mathcal{Q}}(d,\alpha) = \begin{cases} \mathcal{Q}(l(1), a(1)), & d = l(0), \alpha = a(0) \\ \mathcal{Q}(l(k), a(k)), & l(k-1) < d \leq l(k), \\ & a(k-1) < \alpha \leq a(k) \\ 0, & d > l(K_1), \alpha > \theta. \end{cases} \tag{5.9}$$

We bound the approximation error and the computational overhead with two steps. First, we regard α as a constant and bound the approximation error of $\mathcal{Q}(d,\alpha)$ in distance domain. The following theorem ensure that the approximation error in distance domain is less than ϵ_d.

Theorem 5.2.1. *To any given $\alpha = c$, setting $l(0) = 0$, $l(K_1) = D$, and $l(k_1) = b((1+\epsilon_d)^{k_1/2} - 1)$, ($k_1 = 1, \cdots, K_1 - 1$, and $K_1 = \lceil \frac{ln(\mathcal{Q}(0,\alpha)/\mathcal{Q}(D,\alpha))}{ln(1+\epsilon_d)} \rceil$), we have the approximation error:*

$$1 \leq \frac{\mathcal{Q}(d,c)}{\tilde{\mathcal{Q}}(d,c)} \leq 1 + \epsilon_d, (d \leq D). \tag{5.10}$$

Proof. Fix $\alpha = c$. Without loss of generality, suppose that we have $l(k_1) < d \leq l(k_1+1)$ for a given distance d. As $Q(d,c)$ monotonically decrease with distance d, on one hand, $\frac{Q(d,c)}{\tilde{Q}(d,c)} = \frac{Q(d,c)}{Q(l(k_1)+1,c)} \geq \frac{Q(l(k_1+1),c)}{Q(l(k_1+1),c)} = 1$; on the other hand,

$$\begin{aligned} \frac{Q(d,c)}{\tilde{Q}(d,c)} &= \frac{Q(d,c)}{Q(l(k_1)+1),c)} \\ &\leq \frac{Q(l(k_1),c)}{Q(l(k_1+1),c)} \\ &\leq \frac{(b((1+\epsilon_d)^{k_1+1/2}-1)+b)^2}{(b((1+\epsilon_d)^{k_1/2}-1)+b)^2} \\ &= 1+\epsilon_d, \end{aligned}$$

the second inequality hold because substitute QoM (Equ. (5.1) and the expression of $l(k_1)$. Thus, the result follows. □

Then, we bound the approximation error in each segment. Similar, we obtain the following theorem which offers the sufficient condition to guarantee that the approximation error in each segment is less than ϵ_1.

Theorem 5.2.2. *In each piecewise* $l(k_1) \leq d \leq l(k_1+1)$*, setting* $a(0)=0$*,* $a(K_2)=\theta$*, and* $a(k_2)=\arccos(\frac{1+\epsilon_d}{1+\epsilon_1})^{k_2}$*,* $k_2=1,\cdots,K_2-1$*, and* $K_2=\lceil\frac{ln(\mathcal{Q}(l(k_1),0)/(\mathcal{Q}(l(k_1+1),\theta)\cdot(1+\epsilon_d)))}{ln((1+\epsilon_1)/(1+\epsilon_d))}\rceil$*, we have the approximation error:*

$$1 \leq \frac{\mathcal{Q}(d,\alpha)}{\tilde{\mathcal{Q}}(d,\alpha)} \leq 1+\epsilon_1, (d \leq D, \alpha \leq \theta). \tag{5.11}$$

Proof. Without loss of generality, suppose $l(k_1) < d \leq l(k_1+1)$ for a given distance d and $a(k_2) < \alpha \leq a(k_2+1)$ for a given angle a . As $Q(d,\alpha)$ monotonically decrease with angle α and distance d, on one hand, $\frac{Q((d,\alpha)}{\tilde{Q}(d,\alpha)} = \frac{Q(d,\alpha)}{Q(l(k_1+1),a(k_2+1)} \geq \frac{Q(d,a(k_2+1))}{Q(l(k_1+1),a(k_2+1))} \geq \frac{Q(l(k_1+1),a(k_2+1))}{Q((l(k_1+1),a(k_2+1))} = 1$; on the other hand,

$$\begin{aligned}
\frac{Q(d,\alpha)}{\tilde{Q}(d,\alpha)} &= \frac{Q(d,\alpha)}{Q(l(k_1+1),a(k_2+1))} \\
&\leq \frac{Q(l(k_1),a(k_2))}{Q(l(k_1+1),a(k_2+1))} \\
&= \frac{Q(l(k_1),a(k_2))}{Q(l(k_1+1),a(k_2))} \cdot \frac{Q(l(k_1+1),a(k_2))}{Q(l(k_1+1),a(k_2+1))} \\
&\leq (1+\epsilon_1) \cdot \frac{a \cdot (((1+\epsilon_2)/(1+\epsilon_1))^{(k_2+1)/2})^2}{a \cdot (((1+\epsilon_2)/(1+\epsilon_1))^{(k_2)/2})^2} \\
&= \left(\frac{1+\epsilon_1}{1+\epsilon_d}\right) \cdot (1+\epsilon_d) = 1+\epsilon_1.
\end{aligned}$$

Thus, the result follows. □

Approximation for Fusion Function. By Theo. 5.2.1 and 5.2.2, the approximation error in angle domain is bounded by $\frac{1+\epsilon_1}{1+\epsilon_d}$. Thus, the fusion function can be bounded as following theorem.

Theorem 5.2.3. *In each segment* $a(i) \leq \alpha(\overrightarrow{d}_{ok}, \overrightarrow{o_k u_i}) \leq a(i+1)$ *and* $a(j) \leq \alpha(\overrightarrow{d}_{ok}, \overrightarrow{o_k u_j}) \leq a(j+1)$ *in Equa. (6.3), setting* $\alpha(\overrightarrow{d}_{ok}, \overrightarrow{o_k u_i}) = a(i)$ *and* $\alpha(\overrightarrow{d}_{ok}, \overrightarrow{o_k u_i}) = a(j)$*, then we have the approximation error:*

$$1 \leq \frac{\mathcal{K}(u_i, u_j, \overrightarrow{d}_{ui}, \overrightarrow{d}_{uj}, \mathcal{O}_k)}{\tilde{\mathcal{K}}(u_i, u_j, \overrightarrow{d}_{ui}, \overrightarrow{d}_{uj}, \mathcal{O}_k)} \leq \frac{1+\epsilon_1}{1+\epsilon_d}, (\alpha(\cdot) \leq \theta). \tag{5.12}$$

Proof. Note that it always holds $\epsilon_1 \geq \epsilon_d$. According to Theo. 5.2.1 and 5.2.2, $1+\epsilon_1 = \frac{\tilde{Q}(l(k_1),a(k_1))}{\tilde{Q}(l(k_1+1),a(k_1+1))}$ and $1+\epsilon_d = \frac{\tilde{Q}(l(k_1),a(k_1))}{\tilde{Q}(l(k_1+1),a(k_1))}$, then $\frac{1+\epsilon_1}{1+\epsilon_d} = \frac{\tilde{Q}(l(k_1),a(k_1))}{\tilde{Q}(l(k_1+1),a(k_1+1))} \div \frac{\tilde{Q}(l(k_1),a(k_1))}{\tilde{Q}(l(k_1+1),a(k_1))} \geq 1.$

Following from Equa. (5.2) and Theo. 5.2.3, the approximation error satisfies

$$\frac{\mathcal{K}(u_i,u_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{uj},\mathcal{O}_k)}{\tilde{\mathcal{K}}(u_i,u_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{uj},\mathcal{O}_k)} = \frac{\frac{1}{(|\mathcal{O}_i|+|\mathcal{O}_j|)\beta}\sum_{k=1}^{|\mathcal{O}_k|}(\omega-\alpha(\overrightarrow{o_k u_i},\overrightarrow{o_k u_j}))}{\frac{1}{(|\mathcal{O}_i|+|\mathcal{O}_j|)\beta}\sum_{k=1}^{|\mathcal{O}_k|}(\omega-\alpha(a(i),a(j)))} \le \frac{1+\epsilon_1}{1+\epsilon_d}, \tag{5.13}$$

where $\alpha(a(i),a(j))$ is the included angle between $a(i)$ and $a(j)$, and $\alpha(\cdot) \le \theta$. As well as $\frac{\mathcal{K}(u_i,u_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{uj},\mathcal{O}_k)}{\tilde{\mathcal{K}}(u_i,u_j,\overrightarrow{d}_{ui},\overrightarrow{d}_{uj},\mathcal{O}_k)} \ge 1$. Then the result follows. □

Discretizing Area. In this subsection, we show how to discretize area based on piecewise constant approximation of $\mathcal{Q}(d,\alpha)$. Then, by this discretization method, we bound the solution space.

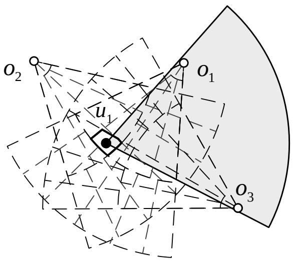


Figure 5.5.: **Area discretization.**

Fig. 5.5 illustrates the key idea of area discretization. First, we draw concentric sectors with radius $l(0), l(1), \cdots, l(K_1)$ and central angle θ centered at each object, respectively. Then, between each pair of consecutive concentric arcs $l(k_1)$ and $l(k_1+1)$, we draw line segments around both sides of orientation of objects, respectively. The extension lines of these line segments cross o_j and their included angles with $\overrightarrow{d}_{oj}$ are $a(0), a(1), \cdots, a(K_2)$, respectively. Due to geometric symmetry, if a UAV lies between two concentric sectors with radius $l(k_1)$ and $l(k_1+1)$ of an object, then this object must also lies between two sectors with the same radiuses centered at this UAV. Moreover, if the UAV monitoring this object between line segments $a(k_2)$ and $a(k_2+1)$ of this object, it will lead to a constant approximated QoM. As Fig. 5.5, UAV u_1 locates between sectors with radius $l(0)$ and $l(1)$ centered at objects o_1 and o_2 as well as $l(1)$ and $l(2)$ centred at object o_3, and u_1 monitors o_1 and o_3. The approximated QoM at o_1 and o_3 by u_1 is equal to $\mathcal{Q}(l(1),a(1))$ and $\mathcal{Q}(l(2),a(1))$.

Consequently, we have the following theorem. Here I omit the proof since the same as Theo. 4.2.1 in Chap. 4.

Theorem 5.2.4. *The number of discretizing subarea for N objects is $O(N^2\epsilon_d^{-2}\epsilon_1^{-2})$.*

5.2.2. Monitoring Dominating Set (MDS) Extraction

After the area discretization, QoM in each subarea is approximated to be constant. Therefore, we can reuse the idea from Sec. 4.2.2 in Chap. 4, like the “divide and conquer” method, in each sub area considering the representative monitored sets of objects rather than enumerate all possible covered sets of objects (here called Monitoring Dominating Sets (MDSs) but DCS in Chap. 4), and figure out these corresponding strategies. Our ultimate goal is also reducing the problem to a combinatorial optimization problem which is finding M strategies from a limited number of strategies extracted by MDS.

Preliminaries. First, we repeat the definitions of Candidate Monitored Set.

Two strategies $\langle u_1, \overrightarrow{d}_{u1}\rangle$ and $\langle u_2, \overrightarrow{d}_{u2}\rangle$ and their monitored object sets O_1 and O_2. If $O_1 = O_2$, then $\langle u_1, \overrightarrow{d}_{u1}\rangle$ is *equivalent* to $\langle u_2, \overrightarrow{d}_{u2}\rangle$; If $O_1 \supseteq O_2$, then $\langle u_1, \overrightarrow{d}_{u1}\rangle$ *dominates* $\langle u_2, \overrightarrow{d}_{u2}\rangle$. To a strategy $\langle u_i, \overrightarrow{d}_{ui}\rangle$ and its set O_i, if there does not exist any strategy $\langle u_j, \overrightarrow{d}_{uj}\rangle$ such that $\langle u_j, \overrightarrow{d}_{uj}\rangle$ dominates $\langle u_i, \overrightarrow{d}_{ui}\rangle$, then O_i is a maximum dominating set and we call it *Monitoring Dominating Set (MDS).*

Definition 5.2.2. ***(Candidate Monitored Set)*** *The candidate monitored set $\tilde{O}_i$ for subarea A_k are those objects that possible to be monitored by a UAV u_i with some orientation $\overrightarrow{d_{ui}}$ in A_k.*

In what follows, we first study a special case where a subarea is reduced to a point (point case) and then the general case (area case).

MDS Extraction for Point Case. Alg. 4 clarifies the MDS extraction for point case. Basically, it is essentially a greedy algorithm which anticlockwise rotates orientation of a UAV located at the point subarea from 0 to 2π. During this process, it tracks the current set of monitoring objects, and records all MDSs. The input of this algorithm is the subarea point A_i and its candidate monitoring set $\tilde{O}_i$ and the output is all MDSs.

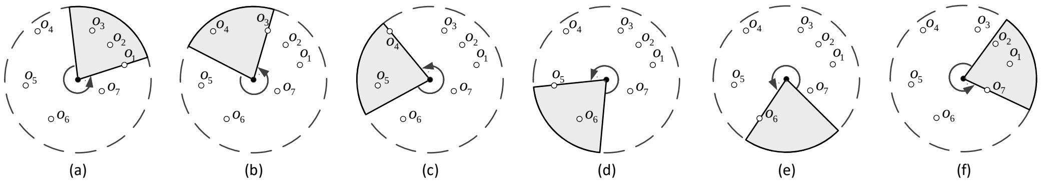


Figure 5.6.: **An example of MDS extraction for point case.**

Fig. 5.6 illustrates a toy instance to show the process of MDS extraction for point case. As shown in Fig. 5.6(a), the algorithm starts with monitoring $\{o_1\}$, then rotate

Algorithm 4: MDS Extraction for Point Case

```
Input: The subarea point $A_i$, the candidate monitored set $\tilde{O}_i$
Output: All MDSs
Set a reference ray originating from UAV as $0°$ and compute the angle between this
  reference ray and the line from the UAV to each object.
According to their angles sort all candidate monitored objects.
Initialize the orientation of the UAV to $0°$.
while rotated angle is less than $360°$ do
    Rotate the UAV anticlockwise to monitor objects one by one until there is some
      monitored object will fall out of monitored.
    if rotated angle is larger than $360°$ then
        terminate.
    end
    Add the current covered set of objects to the collection of MDSs.
    Rotate the UAV anticlockwise until a new object is included in the covered set.
    if rotated angle is larger than $360°$ then
        terminate.
    end
end
```

UAV anticlockwise to monitor o_2 and o_3 one by one. But o_4 cannot be added into the current monitoring set otherwise $\{o_1, o_2\}$ will fall out of the monitoring region. Thus, $\{o_1, o_2, o_3\}$ is an MDS and it will be added to the collection of MDSs. Then, Alg. 4 continues rotating the UAV and removes $\{o_1, o_2\}$ from current monitoring set and adds o_4 into it, as shown in Fig. 5.6(b). Since o_5 is beyond the current monitoring region of UAV, $\{o_3, o_4\}$ is added to the collection of MDSs. Next, the algorithm extracts MDSs of $\{o_4, o_5\}$ and $\{o_5, o_6\}$ orderly via rotating the UAV as shown in Fig. 5.6(c) and (d). After that, Alg. 4 removes $\{o_5\}$ from current monitoring set and try to monitor a new object, saying o_7. However, due to the limitation of monitoring angle, o_7 cannot be added in the current monitoring set as illustrated in Fig. 5.6(e). Thus, $\{o_6\}$ is added to the collection of MDSs. The algorithm proceeds until the UAV rotates larger than 360° as depicted in Fig. 5.6(f). Finally, the obtained collection of MDSs are $\{o_1, o_2, o_3\}, \{o_3, o_4\}, \{o_4, o_5\}$, $\{o_5, o_6\}$, $\{o_6\}$, and $\{o_7, o_1, o_2\}$.

MDS Extraction for Area Case. Then, we discuss the general area case and present the algorithm of MDS extraction for area case in Alg. 5. We first give a toy instance of the process of Alg. 5 in Fig. 5.7, then prove its correctness. As shown in Fig. 5.7(a), suppose there are six objects in the candidate monitored set for subarea A_i. First, we draw lines crossing each pair of objects, *e.g.*, o_1 and o_2 in Fig. 5.7(b), and set a UAV at intersection points u_1 and u_2 with two objects lying on UAV's clockwise boundary Thus, we obtain two MDSs $\{o_1, o_2, o_4\}$ and $\{o_1, o_2, o_4, o_5, o_6\}$ as well as their strategies $\langle u_1, \overrightarrow{d}_{u1}\rangle$ and $\langle u_2, \overrightarrow{d}_{u1}\rangle$. Second, we draw arcs crossing each pair of objects with circumferential angle equals to $2 \times \gamma$, *e.g.*, o_1 and o_5 in Fig. 5.7(c),

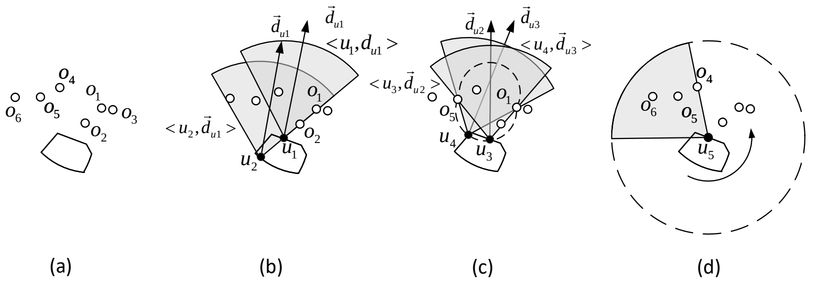


Figure 5.7.: **An example of MDS extraction for area case.**

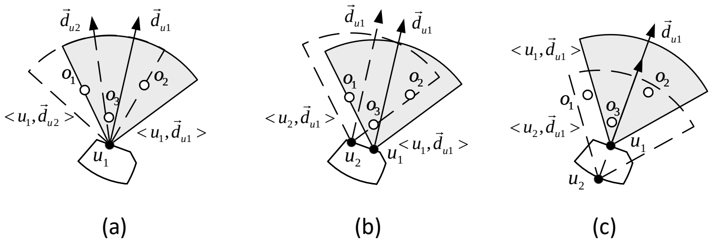


Figure 5.8.: **Three kinds of transformation: (a) Rotation, (b) Translation, (c) Projection.**

and set a UAV at intersection points u_3 and u_4 with two objects respectively lying on UAV's two radiuses. Thus, we obtain two MDSs $\{o_1, o_2, o_4, o_5\}$ and $\{o_1, o_4, o_5\}$ as well as their strategies $\langle u_3, \overrightarrow{d}_{u2}\rangle$ and $\langle u_4, \overrightarrow{d}_{u3}\rangle$. Third, we randomly choose a point on the boundary of subarea and execute MDS extraction algorithm for point case, as Fig. 5.7(d). Finally, we check all the obtained MDSs and remove the MDSs which are subsets of some other MDS. In this toy instance, we reserve $\{o_1, o_2, o_4, o_5, o_6\}$ and $\{o_1, o_2, o_3, o_4, o_5\}$.

What follows, we prove the correctness of Alg. 5.

To begin with, we give three transformations to assist our proof. As shown in Fig. 5.8, there are three transformations in MDS extraction for area case. All three transformations are transformed from strategy $\langle u_1, \overrightarrow{d}_{u1}\rangle$ in Fig. 5.8. Rotation transformation, as illustrated in Fig. 5.8(a), keeps the coordinate of UAV u_i unchanged and rotate the orientation of UAV from $\overrightarrow{d}_{u1}$ to $\overrightarrow{d}_{u2}$. Translation transformation, as illustrated in Fig. 5.8(b), keeps the orientation of UAV $\overrightarrow{d}_{ui}$ unchanged and move the coordinate of UAV from u_1 to u_2. Projection transformation is a special case of translation transformation, as illustrated in Fig. 5.8(c). It keeps the orientation unchanged and moves the coordinate of UAV along the reverse direction of orientation $\overrightarrow{d}_{u1}$ until reaching some point u_2 on the boundary of subarea.

Algorithm 5: MDS Extraction for Area Case

```
Input: The subarea $A_i$ and its candidate monitored set $\tilde{O}_i$
Output: MDSs and their corresponding strategies
for all pairs of objects in $\tilde{O}_i$, say $o_i$ and $o_j$ do
    Draw a straight line crossing $o_i$ and $o_j$, and intersect with the boundaries of subarea.
    Hover UAV at these intersection points and adjust their orientations, let right radius of
      monitoring area crossing $o_i$ and $o_j$.
    Add the MDSs and the corresponding strategies under current setting into the solution
      set.
    Draw two arcs crossing $o_i$ and $o_j$ with circumferential angle $2 \cdot \gamma$ and intersect the
      boundaries of subarea.
    Hover UAV at these intersection points and adjust their orientations, let the two
      radiuses cross $o_i$ and $o_j$ respectively.
    Add the MDSs and the corresponding strategies under current setting into the solution
      set.
end
Choose a point on the boundary of the subarea randomly and execute MDS extraction for
  point case algorithm and add the results into the solution set.
Filter the solution set and remove the subsets of some MDSs and their corresponding
  strategies.
```

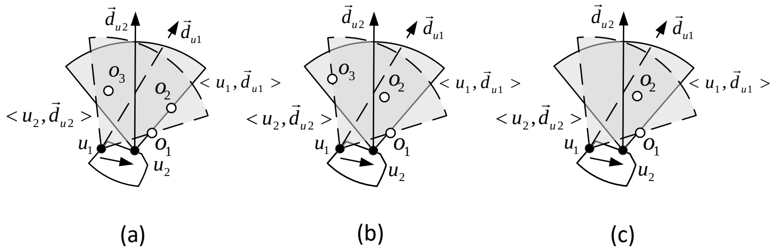


Figure 5.9.: **Three critical conditions of transformation: (a) Two objects both touch the right radius; (b) Two objects respectively touch the left and right radius; (c) Only one object touches the boundary.**

According to the transformation of projection, we have the following lemma. We omit the proof of it, as the idea is totally the same as Lem. 4.2.1.

Lemma 5.2.1. *If $\langle u_2, \overrightarrow{d}_{u1} \rangle$ is the projection of $\langle u_1, \overrightarrow{d}_{u1} \rangle$, then $\langle u_2, \overrightarrow{d}_{u1} \rangle$ dominates $\langle u_1, \overrightarrow{d}_{u1} \rangle$.*

By Lem. 5.2.1, we can get the following corollary.

Corollary 5.2.1. *The MDSs extracted under the case wherein UAV located on the boundaries of a subarea dominate the MDSs extracted under the case wherein UAV located in the whole subarea.*

Thus, we only need to consider $\langle u_i, \overrightarrow{d}_{ui} \rangle$ where u_i are on the boundaries of subarea. Let Γ denote the output set of Alg. 5. We have the following theorem.

Theorem 5.2.5. *Given any strategy* $\langle u, \overrightarrow{d}_u \rangle$*, there exists* $\langle u_2, \overrightarrow{d}_{u2} \rangle \in \Gamma$ *such that* $\langle u_2, \overrightarrow{d}_{u2} \rangle$ *dominates* $\langle u, \overrightarrow{d}_u \rangle$*.*

Proof. By Coro. 5.2.1, we only need to consider the strategies wherein their coordinates lie on the boundaries of the subarea. Then, for an arbitrary strategy $\langle u, \overrightarrow{d}_u \rangle$, we execute the following transformations.

1) Keep the coordinate u fixed and rotate the orientation $\overrightarrow{d}_u$ anticlockwise, which is a rotation transformation as Fig. 5.8(b) , until there is at least one object, saying o_1, is going to fall out of monitoring area through the right radius. Suppose the obtained strategy is $\langle u_1, \overrightarrow{d}_{u1} \rangle$, where $u_1 = u$. Obviously, $\langle u_1, \overrightarrow{d}_{u1} \rangle$ dominates $\langle u, \overrightarrow{d}_u \rangle$.

2) Keep the right radius of the UAV's monitoring area crossing o_1 and move the UAV along the boundaries of subarea, until at least another object is going to fall out of the monitoring area through either right or left radius. If no other object is going to fall out, then stop the transformation. In other words, during the process of translation and rotation transformations, any object monitored currently won't fall out of the monitoring area, formally, the newly obtained strategy $\langle u_2, \overrightarrow{d}_{u2} \rangle$ dominates $\langle u_1, \overrightarrow{d}_{u1} \rangle$.

After the above transformations, there are three possible conditions we may encounter.

1) Another object touches the right radius of the monitoring area (Fig. 5.9(a)).

2) Another object touches the left radius of the monitoring area (Fig. 5.9(b)).

3) None of other object touches the boundary of the monitoring area (Fig. 5.9(c)).

Cases 1 and 2 are critical conditions that an object which is monitored by $\langle u_1, \overrightarrow{d}_{u1} \rangle$ is going to fall out the monitoring area. While Case 3 is the situation that no objects will fall out, formally $\langle u_2, \overrightarrow{d}_{u2} \rangle$ is always equivalent to $\langle u_1, \overrightarrow{d}_{u1} \rangle$. Note that, objects won't fall out of monitoring area through the arc boundary as we have proved in Coro. 5.2.1.

In Alg. 5, we can see that Step 2-4 and Step 5-7 correspond to Case 1 and 2, respectively. For Case 3, arbitrary points on the boundaries of subarea are equivalent, thus Step 9 can extract all MDSs resulted from this case. Consequently, the corresponding monitored set of objects of strategy $\langle u_2, \overrightarrow{d}_{u2} \rangle$ must be included in Γ before Step 10. Since it is dominated by some MDS in the final obtained Γ, the result follows. □

Algorithm 6: Strategy Selection

```
Input: The number of UAVs $M$, MDS set $\Gamma$, object function $f(X)$
Output: Strategy set $X$
$X = \emptyset$.
while $|X| \leq M$ do
    $e^* = \arg\max_{e \in \Gamma \setminus X} f(X \cup \{e\}) - f(X)$.
    $X = X \cup \{e^*\}$.
end
```

5.2.3. Problem Reformulation and Solution

After the second step, our problem has been reduced to a combinatorial optimization problem which is finding M strategies from a limited number of strategies extracted by MDS. In the third step, we first reformulate the problem, then prove its monotonicity and submodularity, and finally present an effective algorithm with $1-1/e$ approximation ratio to address this reformulated problem.

Let $\mathcal{S}$ be the selected set of strategies from Γ. For all possible $\mathcal{S}$ in Γ, we can compute their overall monitoring utility. The problem **P1** can be reformulated as

$$\begin{aligned} \textbf{(P2)} \quad & max \quad tr(\Sigma_{\mathcal{S}\mathcal{S}}) + tr(\Sigma_{\mathcal{O}\mathcal{S}}\Sigma_{\mathcal{S}\mathcal{S}}^{-1}\Sigma_{\mathcal{S}\mathcal{O}}), \\ & s.t. \quad \mathcal{S} \subseteq \Gamma, |\mathcal{S}| = M, \end{aligned}$$

where $\Sigma_{\mathcal{O}\mathcal{S}}$ and $\Sigma_{\mathcal{S}\mathcal{S}}^{-1}$ can be easily obtained by Equa. (5.9) of corresponding o_i and $s_i = \langle u_i, \overrightarrow{d}_{ui} \rangle$ and Equa. (5.12) of s_i, s_j and their monitoring objects set $\mathcal{O}_i$ and $\mathcal{O}_j$.

The problem is then transformed to a combinatorial optimization problem. Follow the same problem transformation idea of Problem Reformulation (Sec. 4.2.3 in PANDA Chap. 4), we give further analysis and transform **P2** into **P3**. We briefly repeat the Submodularity (Def. 4.2.8) and Matroid (Def. 4.2.9 and Def. 4.2.10) definition as follows.

(1) A real-valued set function $f : 2^S \to \mathbb{R}$ (S is a set) is nonnegative, monotonic, and submodular [142] iff.: 1. $f(A) \geq 0$; 2. $f(A \cup \{e\}) \geq f(A)$ ($A \subseteq S$,$e \in S \setminus A$); 3. $f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B)$ ($A \subseteq B \subseteq S$,$e \in S \setminus B$). (2) A Uniform Matroid [142] $\mathcal{M} = (S, L)$ where S is a finite ground set, $L \subseteq 2^S$ is a collection of independent sets, k is an integer, such that 1. $\emptyset \in L$; 2. if $X \subseteq Y \in L$, then $X \in L$; 3. if $X, Y \in L$, and $|X| < |Y|$, then $\exists y \in Y \setminus X$, $X \cup \{y\} \in L$. 4. $L = \{X \subseteq S : |X| \leq k\}$.

Then, our problem can be reformulated as

VISIT Problem (P3): Given Γ, select X (a M number set) from L such that $f(X)$ maximized

$$\begin{aligned} \textbf{(P3)} \quad & max\ f(X) = tr(\Sigma_{XX}) + tr(\Sigma_{\mathcal{O}X}\Sigma_{XX}^{-1}\Sigma_{X\mathcal{O}}), \\ & s.t. \quad X \in L, \\ & \qquad\quad L = \{X \subseteq \Gamma : |X| \leq \mathcal{M}\}. \end{aligned}$$

Lemma 5.2.2. *The objective function $f(X)$ is a monotone submodular function, and the constraint is a uniform matroid constraint.*

Proof. We verify the three listed requirements in submodularity. First, when the number of UAVs is 0, obviously $\Sigma = 0$, thus $f(\emptyset) = 0$. Next, we present the following claim.

Claim 5.2.1. *Given a* semi-positive definite (s.p.d.) *matrix Σ in a block separated form as*

$$\Sigma = \begin{pmatrix} A & E & F & G \\ E^T & B & H & I \\ F^T & H^T & C & J \\ G^T & I^T & J^T & D \end{pmatrix},$$

we have:

$$\begin{aligned} 1.\quad & tr(A) + tr((EFG)(EFG)^T A^{-1}) \\ & \leq tr(A+B) + tr(\begin{pmatrix} F & G \\ H & I \end{pmatrix}\begin{pmatrix} F & G \\ H & I \end{pmatrix}^T \begin{pmatrix} A & E \\ E^T & B \end{pmatrix}^{-1}), \end{aligned} \tag{5.14}$$

$$\begin{aligned} 2.\quad & tr(A+B) + tr(\begin{pmatrix} F & G \\ H & I \end{pmatrix}\begin{pmatrix} F & G \\ H & I \end{pmatrix}^T \begin{pmatrix} A & E \\ E^T & B \end{pmatrix}^{-1}) \\ & - (tr(A) + tr(EFG)(EFG)^T A^{-1}) \leq \\ & tr(A+B+C) + tr(\begin{pmatrix} G \\ I \\ J \end{pmatrix}\begin{pmatrix} G \\ I \\ J \end{pmatrix}^T \begin{pmatrix} A & E & F \\ E^T & B & H \\ F^T & H^T & C \end{pmatrix}^{-1}) \\ & - (tr(A+C) + tr(\begin{pmatrix} G \\ J \end{pmatrix}\begin{pmatrix} G \\ J \end{pmatrix}^T \begin{pmatrix} A & F \\ F^T & C \end{pmatrix}^{-1})). \end{aligned} \tag{5.15}$$

Proof. For InEqua. (5.14), we set $M = (FG)$,

$$\begin{pmatrix} M \\ N \end{pmatrix} = \begin{pmatrix} F & G \\ H & I \end{pmatrix}, \begin{pmatrix} O & P \\ P^T & Q \end{pmatrix} = \begin{pmatrix} A & E \\ E^T & B \end{pmatrix}^{-1}.$$

Then, expanding (5.14) and combining like terms, we get

$$\begin{aligned}&tr(A)+tr(E^TA^{-1}E+M^TA^{-1}M)\\&\quad\leq tr(A+B)+tr(M^TOM+N^TP^TM+M^TPN+N^TQN)\\=&tr(A+E^TA^{-1}E)+tr(M^TA^{-1}M)\\&\quad\leq tr(A+B)+tr(M^TOM+N^TP^TM+M^TPN+N^TQN).\end{aligned}$$

Now, we prove it by two parts.

(1) $tr(A+E^TA^{-1}E)\leq tr(A+B)$: As Σ is an *s.p.d.* matrix, for any real vectors x and y, we have $(x,y)^T\begin{pmatrix}A & E\\E^T & B\end{pmatrix}(x,y)>0$. As an *s.p.d.* matrix can always be factorized as a matrix times its transpose, we can rewrite the left of (5.14) as

$$\begin{aligned}&x^TR^TRx+x^TR^TR^{-T}Ey+y^TE^TR^{-1}Rx\\&+y^TBy+y^TE^TR^{-1}R^{-T}Ey-y^TE^TA^{-1}Ey,\end{aligned}$$

where R is invertible and $R^{-T}=(R^T)^{-1}$. Thus, (5.14) can be rewritten as $(Rx+R^TEy)^T(Rx+R^{-T}Ey)+y^T(-E^TA^{-1}E+B)y>0$. As A is *s.p.d.*, $Ax+Ey=0$ with respect to x always has a solution for any y, then vector $(Rx+R^TEy)$ equals to 0 for any given y. That means, for any given y, there always exists $y^T(-E^TA^{-1}E+B)y>0$. So, $(-E^TA^{-1}E+B)$ is *s.p.d.*, which implies $tr(-E^TA^{-1}E+B)>0$ and then $tr(E^TA^{-1}E)<tr(B)$.

(2) $tr(M^TA^{-1}M)\leq tr(M^TOM+N^TP^TM+M^TPN+N^TQN)$: Let the right of the inequality minus the left, we need to prove $tr(\Phi)\geq 0$, $\Phi=M^T(O-A^{-1})M+N^TP^TM+M^TPN+N^TQN$. Because $AO+EP^T=I$ and $AP+EQ=0$, which implies $O+A^{-1}EP^T=A^{-1}$ and $PQ^{-1}P^T+A^{-1}EP^T=0$, then we have

$$M^T(O-A^{-1})M=M^T(PQ^{-1}P^T)M.$$

Then $\Phi=M^T(PQ^{-1}P^T)M+N^TP^TM+M^TPN+N^TQN$, which leads to

$$(M^TN^T)\begin{pmatrix}P & 0\\Q & I\end{pmatrix}\begin{pmatrix}Q^{-1} & 0\\0 & 0\end{pmatrix}\begin{pmatrix}P & Q^T\\0 & I\end{pmatrix}(MN)^T.$$

As Q^{-1} is *s.p.d.*, the result follows. Combine the above two parts, the whole result follows. □

By InEqua. (5.14), we obtain $f(A\cup\{e\})-f(A)\geq 0$, where, $A\subseteq\Gamma$ and element $e\in\Gamma\backslash A$. And according to (5.15), we have $(f(A\cup\{e\})-f(A))-(f(B\cup\{e\})-f(B))\geq 0$, where A and B are two sets such that $A\subseteq B\subseteq\Gamma$ and element $e\in\Gamma\backslash B$. We omit the proof of InEqua. (5.15) as it is similar to that of InEqua. (5.14).

To sum up, $f(X)$ is a monotone submodular function. □

Consequently, the reformulated problem falls into the scope of maximizing a monotone submodular function subject to a matroid constraint, which can be addressed by a greedy algorithm which achieves a good approximation [142]. Alg. 6 shows the pseudo code of strategy selecting algorithm. In each round, Alg. 6 greedily adds a strategy e^* to X to maximize the increment of function $f(X)$.

5.3. Theoretical Analysis

Theorem 5.3.1. *The VISIT algorithm achieves an approximation ratio of* $1-\frac{1}{e}-\epsilon$*, where* $\epsilon = 3\epsilon_1$*, and its time complexity is* $O(MN^{12}\epsilon^{-12})$*.*

Proof. First, we bound the approximation ratio of VISIT algorithm. Denote the two sets of strategies of all M UAVs under **OPT** (optimal solution) to problem **P1** and the reformulated problem **P2** as $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$, respectively. Denote the obtained strategies of VISIT to the problem **P2** (or **P3**) as $\mathcal{S}_2$. According to the fact that a greedy algorithm of maximizing a monotone submodular function subject to a uniform matroid achieves $1-1/e$ approximation ratio [142], thus the approximation ratio of Alg. 6 is $1-1/e$, namely,

$$\begin{aligned}&tr(\Sigma_{\mathcal{S}_2\mathcal{S}_2})+tr(\Sigma_{\mathcal{O}\mathcal{S}_2}\Sigma_{\mathcal{S}_2\mathcal{S}_2}^{-1}\Sigma_{\mathcal{S}_2\mathcal{O}})\\ \geq&(1-\frac{1}{e})(tr(\Sigma_{\mathcal{S}_2^*\mathcal{S}_2^*})+tr(\Sigma_{\mathcal{O}\mathcal{S}_2^*}\Sigma_{\mathcal{S}_2^*\mathcal{S}_2^*}^{-1}\Sigma_{\mathcal{S}_2^*\mathcal{O}})).\end{aligned} \tag{5.16}$$

Further, by Theo. 5.2.2, we have $\tilde{\mathcal{Q}}(\cdot) \geq \frac{1}{1+\epsilon_1}\mathcal{Q}(\cdot)$. By Theo. 5.2.3, we have $\tilde{\mathcal{K}}(\cdot) \geq \frac{1+\epsilon_d}{1+\epsilon_1}\mathcal{K}(\cdot) \geq \frac{1}{1+\epsilon_1}\mathcal{K}(\cdot)$. Thus, each entry in $\Sigma_{\mathcal{S}_2^*\mathcal{S}_2^*}$,$\Sigma_{\mathcal{O}\mathcal{S}_2^*}$, $\Sigma_{\mathcal{S}_2^*\mathcal{S}_2^*}^{-1}$, and $\Sigma_{\mathcal{S}_2^*\mathcal{O}}$ is approximated to the entry in problem **P1** with at most $\frac{1}{1+\epsilon_1}$ error. Then, we have

$$\begin{aligned}&tr(\Sigma_{\mathcal{S}_2^*\mathcal{S}_2^*})+tr(\Sigma_{\mathcal{O}\mathcal{S}_2^*}\Sigma_{\mathcal{S}_2^*\mathcal{S}_2^*}^{-1}\Sigma_{\mathcal{S}_2^*\mathcal{O}})\\ \geq&\frac{1}{1+\epsilon_1}\cdot tr(\Sigma_{\mathcal{S}_1^*\mathcal{S}_1^*})\\ &+\frac{1}{1+\epsilon_1}\cdot\frac{1}{1+\epsilon_1}\cdot\frac{1+\epsilon_d}{1+\epsilon_1}\cdot tr(\Sigma_{\mathcal{O}\mathcal{S}_1^*}\Sigma_{\mathcal{S}_1^*\mathcal{S}_1^*}^{-1}\Sigma_{\mathcal{S}_1^*\mathcal{O}})\\ \geq&\frac{1}{(1+\epsilon_1)^3}\cdot(tr(\Sigma_{\mathcal{S}_1^*\mathcal{S}_1^*})+tr(\Sigma_{\mathcal{O}\mathcal{S}_1^*}\Sigma_{\mathcal{S}_1^*\mathcal{S}_1^*}^{-1}\Sigma_{\mathcal{S}_1^*\mathcal{O}})).\end{aligned} \tag{5.17}$$

Combining InEqua. (5.16) and (5.17), it can be bounded as follows

$$\begin{aligned}
&tr(\Sigma_{\mathcal{S}_2\mathcal{S}_2})+tr(\Sigma_{\mathcal{O}\mathcal{S}_2}\Sigma^{-1}_{\mathcal{S}_2\mathcal{S}_2}\Sigma_{\mathcal{S}_2\mathcal{O}})\\
\geq&(1-\frac{1}{e})\cdot\frac{1}{(1+\epsilon_1)^3}\cdot(tr(\Sigma_{\mathcal{S}^*_1\mathcal{S}^*_1})+tr(\Sigma_{\mathcal{O}\mathcal{S}^*_1}\Sigma^{-1}_{\mathcal{S}^*_1\mathcal{S}^*_1}\Sigma_{\mathcal{S}^*_1\mathcal{O}}))\\
\geq&(1-\frac{1}{e})\cdot(1-\epsilon_1)^3\cdot(tr(\Sigma_{\mathcal{S}^*_1\mathcal{S}^*_1})+tr(\Sigma_{\mathcal{O}\mathcal{S}^*_1}\Sigma^{-1}_{\mathcal{S}^*_1\mathcal{S}^*_1}\Sigma_{\mathcal{S}^*_1\mathcal{O}}))\\
\geq&(1-\frac{1}{e})\cdot(1-3\epsilon_1+3\epsilon_1^2-\epsilon_1^3)\cdot\\
&(tr(\Sigma_{\mathcal{S}^*_1\mathcal{S}^*_1})+tr(\Sigma_{\mathcal{O}\mathcal{S}^*_1}\Sigma^{-1}_{\mathcal{S}^*_1\mathcal{S}^*_1}\Sigma_{\mathcal{S}^*_1\mathcal{O}})).
\end{aligned}\tag{5.18}$$

For any $\epsilon_1 \leq 3$, $3\epsilon_1^2-\epsilon_1^3 \geq 0$, we have

$$\begin{aligned}
&(1-\frac{1}{e})\cdot(1-3\epsilon_1+3\epsilon_1^2-\epsilon_1^3)\cdot\\
&(tr(\Sigma_{\mathcal{S}^*_1\mathcal{S}^*_1})+tr(\Sigma_{\mathcal{O}\mathcal{S}^*_1}\Sigma^{-1}_{\mathcal{S}^*_1\mathcal{S}^*_1}\Sigma_{\mathcal{S}^*_1\mathcal{O}}))\\
\geq&(1-\frac{1}{e})\cdot(1-3\epsilon_1)\cdot(tr(\Sigma_{\mathcal{S}^*_1\mathcal{S}^*_1})+tr(\Sigma_{\mathcal{O}\mathcal{S}^*_1}\Sigma^{-1}_{\mathcal{S}^*_1\mathcal{S}^*_1}\Sigma_{\mathcal{S}^*_1\mathcal{O}}))\\
\geq&(1-\frac{1}{e}-3\epsilon_1)\cdot(tr(\Sigma_{\mathcal{S}^*_1\mathcal{S}^*_1})+tr(\Sigma_{\mathcal{O}\mathcal{S}^*_1}\Sigma^{-1}_{\mathcal{S}^*_1\mathcal{S}^*_1}\Sigma_{\mathcal{S}^*_1\mathcal{O}})).
\end{aligned}\tag{5.19}$$

Therefore, by setting $\epsilon = 3\epsilon_1$, the approximation ratio of the VISIT algorithm is $1-\frac{1}{e}-\epsilon$.

Next, we analyze the time complexity of the VISIT algorithm. First, VISIT computes the subareas by the discretization method. The complexity of this step is $O(N^2\epsilon^{-4})$ according to Theo. 5.2.4. Second, in each subarea MDS extracted by Alg. 5 the combination of objects pair for Case 1 and Case 2 is $\binom{N}{2} = \frac{N(N-1)}{2}$ and each case generates $O(1)$ number of MDSs. For Case 3, its associated time complexity is also the number of MDSs in each subarea which is $O(N)$. Thus, the time complexity of Alg. 5 (as well as the number of MDSs) is the number of subarea times the computation complexity of each subarea, *i.e.*, $O(N^4\epsilon^{-4})$ $(= O(N^2\epsilon^{-4}) \times \max(\binom{N}{2}, O(N)))$. Third, VISIT computes correlation of each pair of strategies. Each MDS is corresponding to a strategy, thus the combination of strategies pair is $\binom{|\mathcal{S}|}{2} = \frac{|\mathcal{S}|(|\mathcal{S}|-1)}{2}$ and each selection costs $O(1)$ time. Thus, the time complexity of this step is $O(N^8\epsilon^{-8})$ $(= \binom{O(N^4\epsilon^{-4})}{2})$. At last, Alg. 6 will perform M times loop to select M strategies. In each loop, it will multiply three matrixes generating $O(N^{12}\epsilon^{-12})$ times computation and select the best one. Thereby, the time complexity of VISIT algorithm is $O(MN^{12}\epsilon^{-12})$. □

5.4. Numerical Simulation

5.4.1. Evaluation Setup

In our simulation, objects are uniformly distributed in a $10m \times 10m$ 2D square area and their orientations are randomly selected among $[0, 2\pi]$. If no otherwise stated, we set $N = 12$, $M = 9$, $\gamma = \pi/3$, $\theta = \pi/6$, $D = 3m$, $\omega = \pi/18$, $\beta = \pi/3$, $\epsilon = 0.2$, and $\epsilon_d = \sqrt{\epsilon/3}$, respectively. We also simulate the coordinates of UAVs which follow a 2D Gaussian distribution with both x- and y- coordinate randomly selected from a Gaussian distribution with $\mu = u_i$ and $\sigma_x = \sigma_y = 3$. Moreover, each data point in evaluation figures is computed by averaging results of 200 random topologies and normalized by dividing the best total QoM, which is $\alpha \times N$.

Since there are no existing approaches for VISIT, we present three algorithms for Comparison: (1) Grid Coordinate Monitoring Algorithm (GCMA) first divides whole 2D square area into small grids, whose side length is $D/\sqrt{2}m$, then extracts MDSs at each vertex of grids with MDS extraction algorithm of point case, and last greedily selects the strategies which monitors the most number of objects. In other words, GCMA doesn't consider the QoM and defines the monitoring utility of every object is just classified of monitored or not. (2) Number-Objective Monitoring Algorithm (NOMA) improves GCMA by dividing area with the intersection of monitoring sectors and extracting MDSs in each subarea with MDS extraction algorithm of area case, namely, the coordinates of UAVs are not just grid, and the objective of NOMA is maximizing the number of efficient monitored objects. (3) Isotropic-QoM Monitoring Algorithm (IQMA) improves NOMA by introducing QoM which is influenced by distance between object and UAV, but the QoM is isotropic which means it doesn't vary with different monitoring angle.

5.4.2. Performance Comparison

We compare the VISIT algorithm with IQMA, NOMA, and GCMA in terms of almost every parameter in Table 5.1. In terms of approximation ratio ϵ, the simulation results show that the monitoring utility is stable along with ϵ increases. This phenomenon let us set a relatively high ϵ to reduce the computational overhead. In addition, in terms of efficient angle θ, the monitoring utility doesn't get better along with increasing θ as we wish. The reason is that the VISIT always provides high monitoring utility with performance bound in common θ value, but if we set the extreme value of θ (*i.e.* 0 or 2π) the utility may change a lot. The detailed simulation results are discussed as follows.

Impact of Number of Waypoints M. *Our simulation results show that on average, VISIT outperforms IQMA, NOMA, and GCMA by* 68.03%, 2.28 *times, and* 5.05 *times, respectively, in terms of* N. Fig. 5.10 shows that the monitoring utility of VISIT invariably increases with M until it approaches 1, while that of IQMA and NOMA increase to about 0.58 and 0.22, respectively, and then keep relatively stable. However, the monitoring utility of GCMA always remains low because the candidate coordinate of waypoints are limited.

Impact of Number of Objects N. *Our simulation results show that on average, VISIT outperforms IQMA, NOMA, and GCMA by* 1.24 *times,* 3.74 *times, and* 7.11 *times, respectively, in terms of* N. Fig. 5.11 shows that the monitoring utility decreases monotonically as the number of objects increases for all four algorithms. Particularly, both the monitoring utilities of GCMA decrease more slowly than that of VISIT, IQMA, and NOMA. This is because VISIT, IQMA, and NOMA select positions that can generate more monitoring utility, but GCMA cannot. Moreover, the monitoring utility of VISIT decreases more slowly than that of IQMA and NOMA when N gets larger.

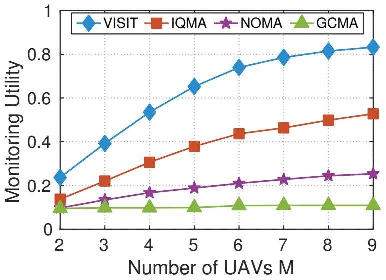


Figure 5.10.: **Number of Waypoints M vs. utility.**

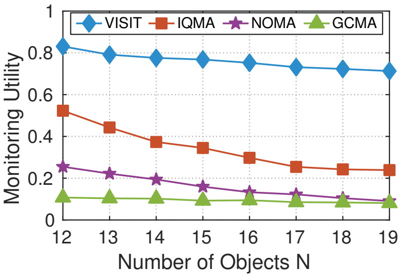


Figure 5.11.: **Number of Objects N vs. utility.**

Impact of Approximation Loss ϵ. *Our simulation results show that on average, VISIT outperforms IQMA, NOMA, and GCMA by* 55.18% , 2.15 *times, and* 5.83 *times, respectively, in terms of* ϵ. As shown in Fig. 5.12, the monitoring utility of VISIT fluctuates slightly when ϵ grows, but it is almost always larger than 0.8. This provides us with a good opportunity to select a larger ϵ to reduce the computation overhead of VISIT without noticeable degradation of performance.

Impact of Farthest Sight Distance D. *Our simulation results show that on average, VISIT outperforms IQMA, NOMA, and GCMA by* 57.25%, 2.33 *times, and* 5.83 *times, respectively, in terms of* D. Fig. 5.13 shows that the monitoring utility of VISIT monotonically increases with D until it approaches 1, while that of IQMA increase a little because it neglects the influence of monitoring angle. Then

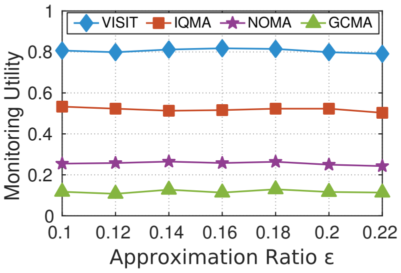


Figure 5.12.: **Approximation Loss ϵ *vs.* utility.**

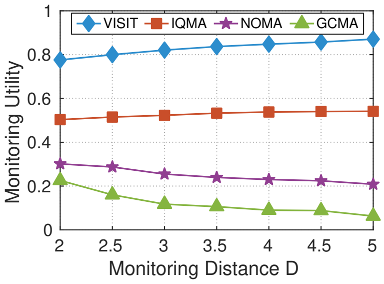


Figure 5.13.: **Farthest Sight Distance D *vs.* utility.**

the monitoring utility of NOMA decreases because it only maximizes the number of monitoring objects but the QoM of additional monitoring objects in each strategy decreases when D increases. The monitoring utility of GCMA decreases because the side length of grids decreases when D increases, namely, the distance between objects and UAVs increases.

Impact of Efficient Angle θ. *Our simulation results show that on average, VISIT outperforms IQMA, NOMA, and GCMA by* 89.23%*,* 3.13 *times, and* 3.15 *times, respectively, in terms of θ.* As shown in Fig. 5.14, when θ grows, VISIT always maintains high monitoring utility which is almost always larger than 0.8. However, when θ grows, objects have more opportunities to be monitored. The monitoring utility of IQMA and NOMA decrease because they do not consider QoM or only consider isotropic QoM and UAV may hover at the strategies where more objects can be monitor but with very large monitoring angles. The monitoring utility of GCMA is always low, because UAV can hover at grids.

Impact of Monitoring Angle γ. *Our simulation results show that on average, VISIT outperforms IQMA, NOMA, and GCMA by* 32.23%*,* 2.03 *times, and* 4.15 *times, respectively, in terms of γ.* Fig. 5.15 shows that the monitoring utility of VISIT monotonically increases with γ increasing, while the monitoring utilities of

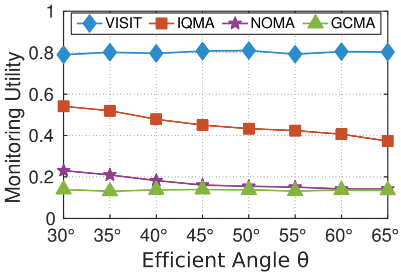


Figure 5.14.: **Efficient Angle θ *vs.* utility.**

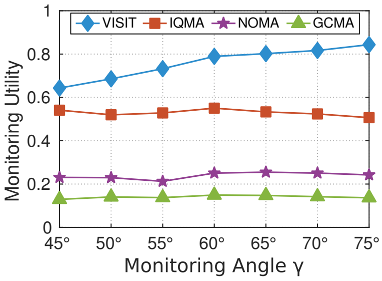


Figure 5.15.: **Monitoring Angle γ *vs.* utility.**

IQMA, NOMA, and GCMA almost do not change. Indeed, when γ grows, UAVs have more chance to monitoring objects, thus the monitoring utility of VISIT increases. However, IQMA, NOMA, and GCMA do not consider anisotropic QoM, thus they may plan waypoints at strategies where can monitor more objects but low QoMs.

Impact of Object's Key Information Distribution Angle β and Captured Key Information Angle by Each UAV ω Here we study the impact of ω and β on monitoring utility. Fig. 5.16 depicts the results and each point on the surface denotes an average value of 200 experiment results. We observe that the monitoring utility increases invariably when ω increases, while monitoring utility decreases invariably when β decreases. Indeed, with a larger ω, each UAV can capture more information of objects, and with a smaller β, each object needs fewer number of UAVs to capture information.

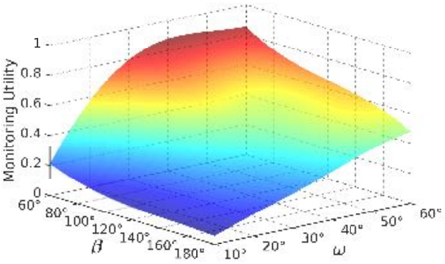
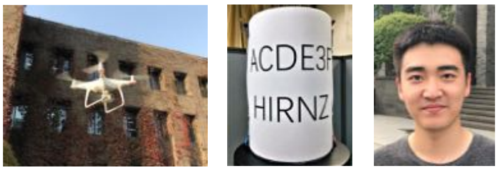


Figure 5.16.: **Key Information Distribution Angle β and Captured Key Information Angle by Each UAV ω *vs.* utility.**

5.5. Field Experiment

To verify our framework, we execute two field experiments with kinds of different objects, and thus different information of two kinds of objects and fusion functions. As shown in Fig. 5.17, experiment equipments consists of DJI Phantom 4 advanced UAV and two different kinds of objects, *i.e.*,



(a) UAV (b) Text Object (c) Face Object

Figure 5.17.: **UAV & Objects.**

Table 5.2.: **Coordinates and orientations of objects.**

Object	Coordinate	Orientation	Object	Coordinate	Orientation
o_0	(4.32, 11.58)	234.21°	o_8	(3.55, 4.71)	197.91°
o_1	(11.93, 3.22)	114.43°	o_9	(7.09, 3.16)	282.47°
o_2	(9.94, 14.34)	256.84°	o_{10}	(3.97, 0.44)	69.88°
o_3	(2.44, 3.75)	133.28°	o_{11}	(8.97, 5.60)	194.79°
o_4	(4.40, 13.43)	294.20°	o_{12}	(10.32, 6.08)	341.20°
o_5	(12.50, 0.73)	103.95°	o_{13}	(3.28, 4.12)	114.87°
o_6	(5.56, 12.53)	168.34°	o_{14}	(8.87, 14.05)	278.88°
o_7	(7.62, 0.45)	240.33°			

text objects and face objects. In our experiment, 15 objects are randomly distributed in our experiment field, whose size is $15\,m \times 15\,m$, and the coordinate and orientation of objects are shown in Table 5.2.

Fig. 5.18 illustrates the distribution of 15 objects and the strategies of 8 waypoints for VISIT, IQMA, and NOMA are drawn in the red, blue, and yellow sectors respectively. Because the accuracy of GPS, the bias of orientation, the influence of wind in the air and the noise over the channel for image transmission lead to bias of pictures captured by the same strategy, we capture 5 pictures of each waypoint and add them together into one picture as shown in Fig. 5.21-5.23 and Fig. 5.28-5.30. To keep both object and UAV safe we set isolation distance between objects and UAV to 2 meters. Moreover, we set $\epsilon = 0.2$ and $\epsilon_d = \sqrt{\epsilon/3}$ in both experiments.

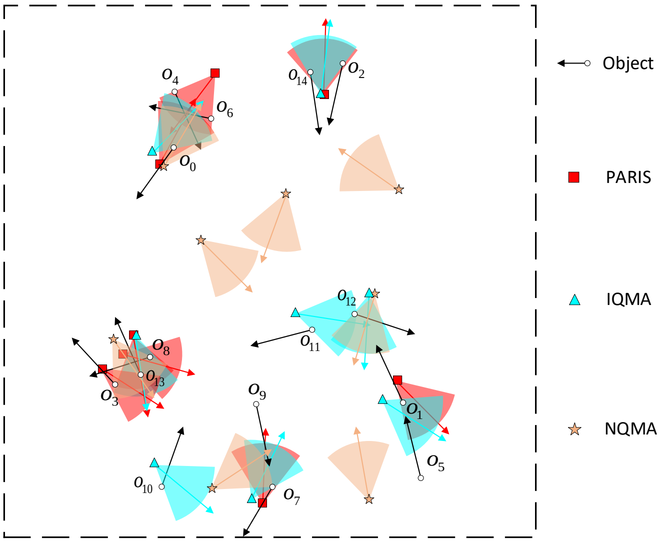


Figure 5.18.: **Objects distribution and waypoints of VISIT, IQMA, and NOMA.**

5.5.1. Text Experiment

In text experiment, we use 15 objects as shown in Fig. 5.17(b). Each of them is made of hard distinguishable text as shown in Fig. 5.19 printed on an $A4$ paper and then pasted on a semicircle cylinder. Fig. 5.20 shows the real experiment field of text experiment which involves 15 text objects. According to the real hardware of UAV and the size of semicircle cylinder, parameters in text experiment are set to $\gamma = \pi/6$, $D = 7m$, $\omega = \pi/18$, $\beta = 4\pi/5$ and $\theta = 2\pi/5$.

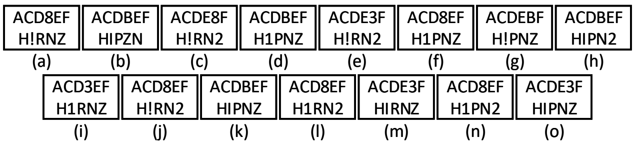


Figure 5.19.: **Text objects.**

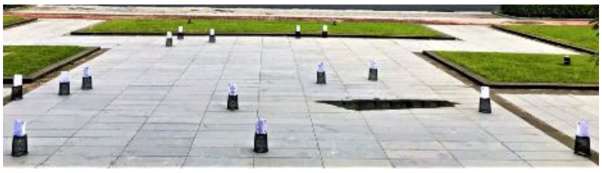


Figure 5.20.: **Text experiment field.**

Fig. 5.21-5.23 show the pictures taken by the selected strategies of 8 UAVs with our algorithm VISIT and two compared algorithms IQMA and NOMA.

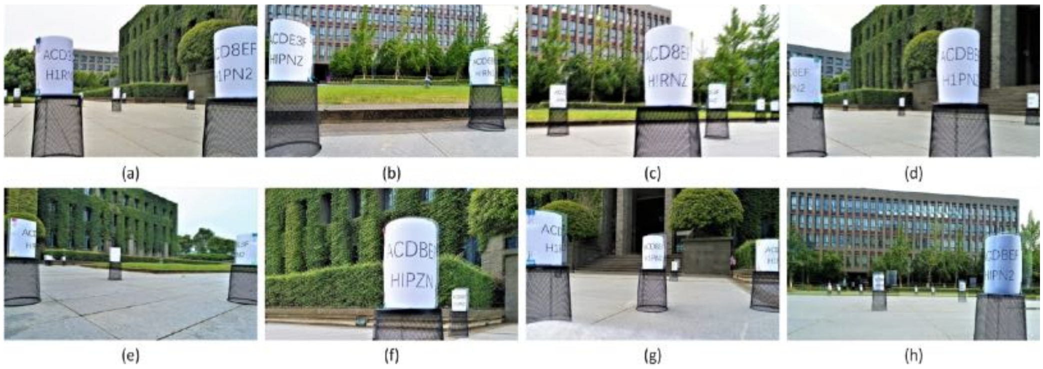


Figure 5.21.: **Captured text pictures by VISIT.**

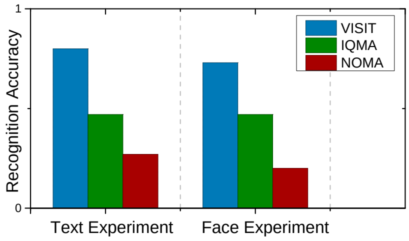


Figure 5.24.: **Recognition results.**

Then, we utilize these sampling pictures of each algorithm to execute text recognition experiment. The fusion function and the text recognition algorithm we used are the existing method respectively proposed in [155] and [136]. Before executing text recognition, we also add 35 other similar texts into candidate set of recognition results. Fig. 5.24 illustrates the recognition accuracy of text experiment by VISIT, IQMA, and NOMA.

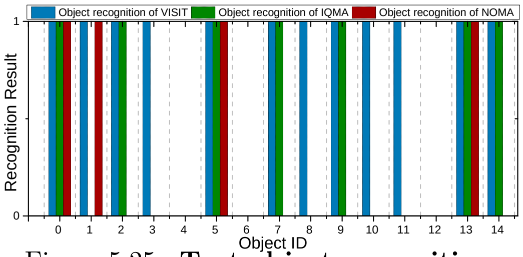


Figure 5.25.: **Text object recognition.**

In text experiment, the recognition accuracies of VISIT, IQMA, and NOMA are 0.80, 0.47, and 0.27, respectively. It means VISIT outperforms IQMA and NOMA by 41.3% and 66.3%, respectively. Fig. 5.25 depicts the recognition results of each object. 1 indicates correct text objet recognition while 0 indicates not.

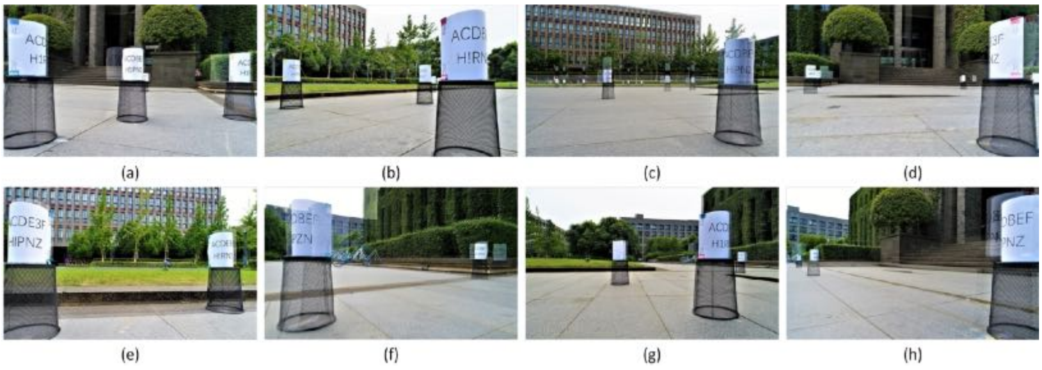


Figure 5.22.: **Captured text pictures by IQMA.**

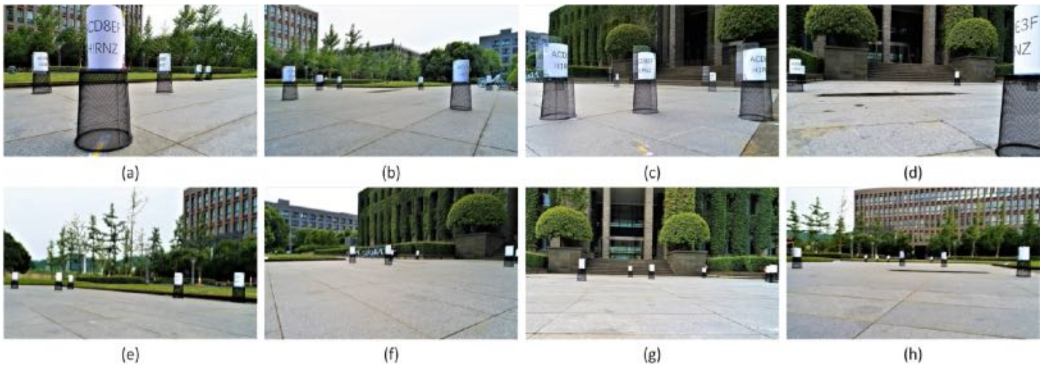


Figure 5.23.: **Captured text pictures by NOMA.**

5.5.2. Face Experiment

In face experiment, as illustrated in Fig. 5.17(c), we invite 15 students to attend our experiment and stand on a square, which include 3 females and 12 males whose stature are respective around $160cm - 165cm$ and $170cm - 188cm$. 15 frontal views of these students are shown in Fig. 5.26. Fig. 5.27 shows the real experiment field of face experiment which involves 15 students with known facing directions. According to the real hardware of UAV and the distribution of features in face recognition [156], parameters in face experiment are set to $\gamma = \pi/6$, $D = 7m$, $\omega = \pi/18$, $\beta = 2\pi/3$ and $\theta = \pi/3$.

Fig. 5.28-5.30 show the pictures taken by the 8 selected strategies(waypoints) for each of VISIT, IQMA, and NOMA. Then we use the value fusion model proposed in [155] as our fusion function and face recognition approach proposed in [135] as recognition algorithm. Similar to text recognition, we also add 85 other front view faces into candidate set of recognition results before executing face recognition experiment. Fig. 5.24 illustrates the recognition accuracy of by VISIT, IQMA, and NOMA.

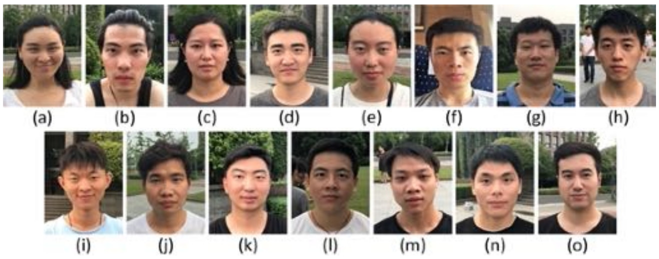


Figure 5.26.: **Face objects.**

Figure 5.27.: **Face experiment field.**

In face experiment, the recognition accuracies for VISIT, IQMA, and NOMA are 0.73, 0.47, and 0.20, respectively, which means VISIT outperforms IQMA and NOMA by 35.6% and 72.6%. Fig. 5.31 depicts the distance of features for each object. Some bars of objects are not depicted in figure, *i.e.* o_{12} and o_2 of NOMA, *etc.*, because the distances of features in the sampling pictures of these objects are too large. From Fig. 5.31, we can see the distances of features of some object in VISIT are not better than IQMA or NOMA, *i.e.*, o_6, o_7, o_9, o_{11}, and o_{14}, this is because the objective of our algorithm is maximizing overall monitoring utility.

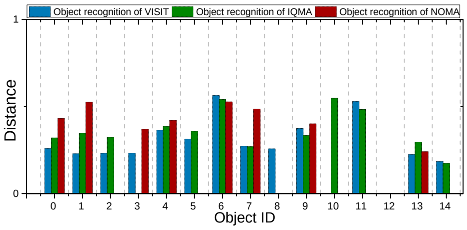


Figure 5.31.: **Face object recognition.**

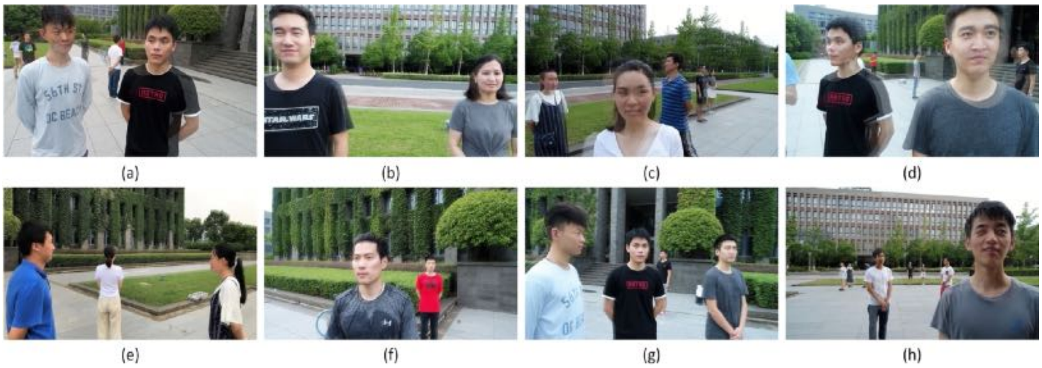


Figure 5.28.: **Captured face pictures by VISIT.**

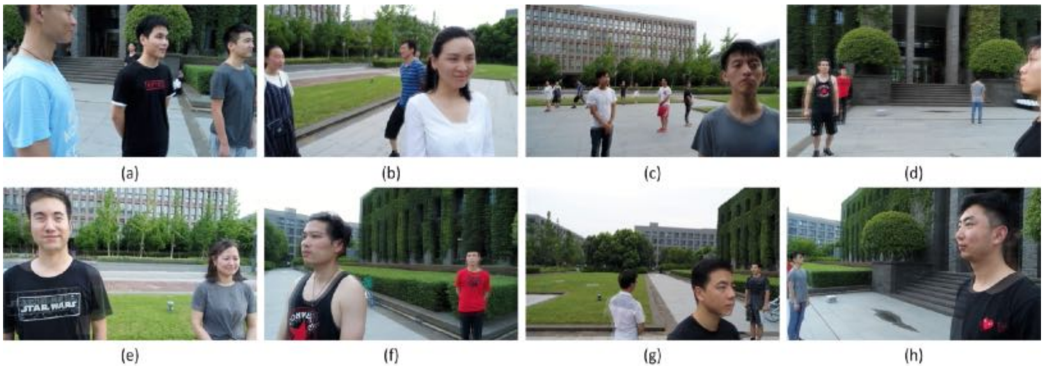


Figure 5.29.: **Captured face pictures by IQMA.**

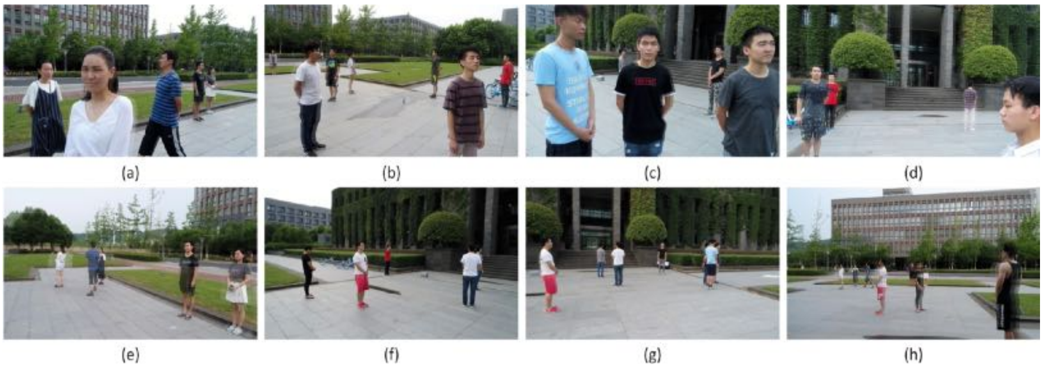


Figure 5.30.: **Captured face pictures by NOMA.**

5.6. Chapter summary

In this paper, we solve the problem of waypoint planning for anisotropic monitoring tasks. The key novelty of this paper is on proposing the first algorithm for considering anisotropic quality of monitoring. The key contribution of this paper is: First, we build the anisotropic monitoring framework which can be used in various QoM and fusion model; Second, we develop an approximation algorithm with performance guarantee; Third, we conduct simulation and two kinds of field experiments for evaluation. The key technical depth of this paper is in reducing the infinite solution space of this optimization problem to a limited one by utilizing the techniques of area partition and Monitoring Dominating Set extraction, and modeling the reformulated problem as maximizing a monotone submodular function subject to a matroid constraint. The experiment results show that our algorithm outperforms other comparison algorithms by at least 41.3%.

Chapter 6

Waypoints Planning with Adjustable Multi-Camera UAVs

This chapter explores the challenges and improvements of adjustable multi-camera UAV model in monitoring tasks. It studies a fundamental problem of Waypoints Planning for Adjustable Multi-camera UAVs (WiPlan), that is, given a set of objects with known locations, plan a set of waypoints for an adjustable multi-camera UAV (*i.e.*, determine the locations and adjust each camera's direction and focal length) such that the overall monitoring utility for all objects is maximized.

Contents

Table 6.1.: **Notations used in Chap. 6.**

Symbol	**Meaning**
u_i	UAV i, or its location
o_j	Object j to be monitored, or its location
ϕ_{ui}^k	Direction of k-th camera on UAV i
f_{ui}^k	Focal length of k-th camera on UAV i
Φ_{ui}	Set of cameras' direction on UAV i
$\mathcal{F}_{ui}$	Set of cameras' focal length on UAV i
$\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle$	Subarrangement of camera k on UAV i
$\langle u_i, \Phi_{ui}, F_{ui} \rangle$	Arrangement of UAV i
K	Number of cameras on each UAV
$\overrightarrow{d}_{ui}$	Orientation of UAV i
γ	Angle of view (AOV) of camera's field of view
R	Range of distance of camera's field of view
φ	Aspect of the object
ΔA	Angle of aspect combining
A_l	Subarea l formed by a set of discretized sectors of O_l
Λ	Set of extracted representative locations
Γ_l	Set of location l's representative subarragements

6.1. Problem Formulation

Network Model. Suppose $\mathcal{O} = \{o_1, o_2, ..., o_N\}$ are distributed in target area. one UAV equipped with K cameras plans M waypoints $\mathcal{U} = \{u_1, u_2, ..., u_M\}$ to monitor these objects. Each state of a waypoint is described with a tuple $\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle$ (called *arrangement*), in which u_i denotes the location, Φ_{ui} and $\mathcal{F}_{ui}$ denote the directions set and focal lengths set of the cameras, respectively. The $k-$th camera at waypoint i is described with $\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle$ (called *subarrangement*), in which ϕ_{ui}^k and f_{ui}^k denote its direction and focal length. Table 6.1 lists the notations we used in WiPlan.

Camera Model. We adopt the camera model used in many literature [107, 157–159]. As shown in Fig. 6.1(a), the monitoring region of the camera model is a sector, which is determined by the angle of view (AOV) γ and the range of distance R. Both γ and R are determined by the camera's focal length f. Specifically, γ depends on f and the dimension of image i (Fig. 6.1(b)), while R depends on γ and the requirement of applications (Fig. 6.1(c)). Formally,

$$\gamma \triangleq 2\arctan(\frac{i}{2f}), R \triangleq \cot(\frac{\gamma}{2}) \cdot \frac{P \cdot r}{2p} \cdot z, \tag{6.1}$$

where the digital zoom ratio z, the total pixels of the entire image L are the hardware determined, l and r are the required pixels and required ratio predefined by different applications.

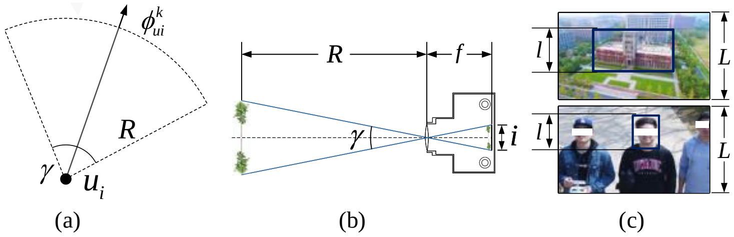


Figure 6.1.: **Camera Model. (a) illustrates the monitoring region of camera. (b) illustrates the relationship between f, R, and γ. (c) illustrates the parameters l and L in applications on large-scale building monitoring and human face recognition.**

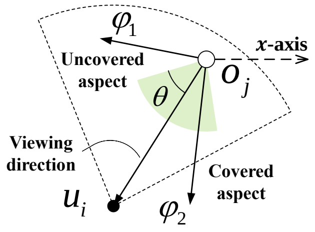


Figure 6.2.: **Aspect Coverage. φ_2 of o_j is a covered aspect by u_i while φ_1 is not.**

Quality of Monitoring (QoM). In most cases, the QoM in the sector-shape monitoring region is not uniform. On one hand, QoM varies with the distance between the camera and the object d and the focal length f [15, 160], and their relation satisfies $QoM \propto f^2/d^2$. On the other hand, QoM varies with the angle between the object's facing direction and the camera's viewing direction. Specifically, along with this angle increase, the QoM, *e.g.*, face recognition accuracy, drops dramatically [156]. To sum up, the best way is to monitor objects with high QoM from as many angles of view as possible. To quantify the monitored angle of view, we utilize the *aspect coverage* concept proposed by [158, 159]. An *aspect* φ of an object is a direction that can be represented by an angle in $[0, 2\pi)$ with 0 degree indicating the one pointing to the positive direction of x-axis, as shown in Fig. 6.2. An aspect φ is covered if there is a camera at waypoint u_i guaranteeing two conditions: (1) o_j is covered by u_i's camera; (2) $\alpha(\varphi, \overrightarrow{o_j u_i}) < \theta$ ($\overrightarrow{o_j u_i}$ is the *viewing direction*, $\alpha(,)$ is the angle between two vectors, and θ is the *efficient angle*).

Formally, the QoM of o_j's aspect φ is

$$Q(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) \triangleq \begin{cases} \frac{a \cdot (f_{ui}^k)^2}{(||u_i o_j|| + b)^2}, 0 \leq ||u_i o_j|| \leq R, \\ \qquad \overrightarrow{u_i o_j} \cdot \phi_{ui}^k - \|u_i o_j\| \cos(\frac{\gamma}{2}) \geq 0, \\ \qquad and \quad \overrightarrow{o_j u_i} \cdot vec(\varphi) - \|o_j u_i\| \cos\theta > 0. \\ 0, \qquad otherwise. \end{cases}$$

where a and b are two constants determined by environment and hardware of devices, $||u_i o_j||$ is the distance between u_i and o_j, $vec(\varphi)$ is the vector with the angle of φ and length of $\|o_j u_i\|$, and γ and R are from Eq. (6.1) at $f = f_{ui}^k$.

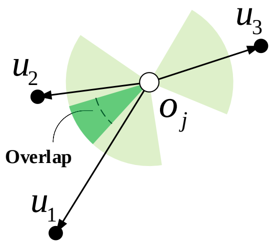


Figure 6.3.: **Monitoring Utility. Both u_1 and u_2 monitor the overlapped aspects of o_j.**

Monitoring Utility. By the QoM funtion, an object's monitoring utility is an integration of aspects' QoM from 0 to 2π. However, according to the aspect coverage's conditions, we observe that if an aspect φ is covered, then its surrounding aspects in the θ interval, *i.e.*, $[\varphi-\theta,\varphi+\theta)$, are all covered. Meanwhile, multiple cameras may cover the overlapped aspects with different QoMs, as shown in Fig. 6.3. Thus, quantifying the obtained information in the overlapped aspects, *i.e.*, information fusion, is an issue to be addressed. We prove that any functions following submodularity applies to our algorithm (See §6.3), *e.g.*, trivial linear addition, cross-entropy, mutual information, even training a function with empirical experiment results. Here, we use max() function as an example, because when multiple UAVs monitor the same aspect, the image with maximum QoM contains complete information of all other images. Thus, the monitoring utility of an object o_j is

$$U(\mathcal{U},o_j) \triangleq \int_0^{2\pi} \max_{\mathcal{A}} Q(u_i,o_j,f_{ui}^k,\phi_{ui}^k,\varphi)d\varphi \tag{6.2}$$

where $\mathcal{A}$ is the arrangements set of $\mathcal{U}$.

Problem Formulation. Then, the problem of Waypoints Plan for Adjustable Multi-camera UAVs (WiPlan) is, *given the distribution of objects, determine the arrangements for all M waypoints of a UAV equipped with K cameras to optimize overall monitoring utility for all N objects.* Here, we define the overall monitoring utility as the normalized sum of N objects' monitoring utility.

WiPlan Problem (P1):

$$\max \quad \frac{1}{2\pi N}\sum_{j=1}^{N}\int_0^{2\pi} \max_{\mathcal{A}} Q(u_i,o_j,f_{ui}^k,\phi_{ui}^k,\varphi)d\varphi$$

$$s.t. \quad |\mathcal{A}| = M, 0 \le \phi_{ui}^k < 2\pi, 0 \le \varphi < 2\pi.$$

The following theorem indicates the hardness of WiPlan.

Theorem 6.1.1. *The WiPlan problem is NP-hard.*

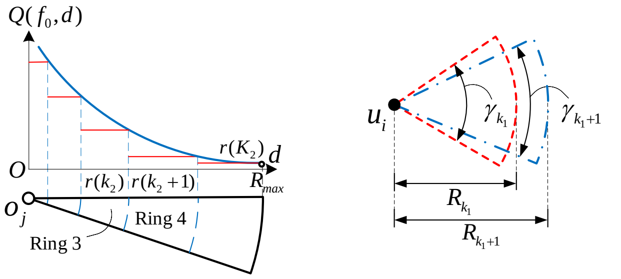
Proof Sketch. Consider a simple case in which focal length f is fixed, $\gamma=\theta=2\pi$, $R=1$, and QoM is set to be a $0-1$ function. In this case, if an object monitored by an UAV, all its aspects have been captured and other UAVs will not contribute even if they also monitor this object. WiPlan changes to the well-known NP-hard Unit Disk Coverage problem [14]. Thus, WiPlan problem is also NP-hard. □

6.2. Solution

Overall roadmap. First, we present piecewise constant function and aspect combining technique to approximate the nonlinear QoM function and monitoring utility. By doing so, the whole solution space is discretized into many cells (§ 6.2.1). Then, we present a Representative Arrangements Extraction (RAE) method, including Representative Locations Extraction and Representative Subarrangements Extraction, to construct the decision space (§ 6.2.2). Thus, the WiPlan problem is transformed a combinatorial optimization problem WiPlan-T (§ 6.2.3). Finally, we design a two-level greedy algorithm to address it, prove WiPlan-T's properties, and bound the algorithm's approximation ratio and time complexity (§ 6.3).

6.2.1. Step 1: Discretization.

In this step, we use the same tech as Sec. 5.2.1 in Chap. 5 and novel aspect combining to approximate the nonlinear solution space to linear one; then discretize whole solution space into many cells.



(a) Distance Discretization (b) Focal Length Discretization

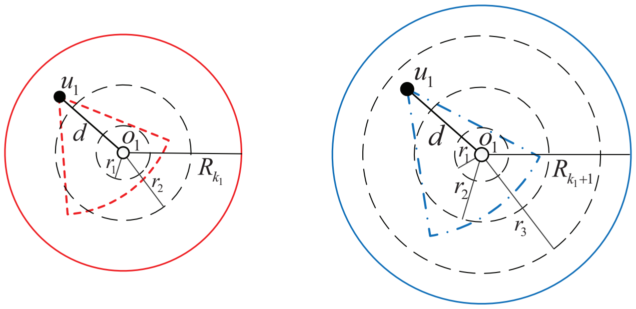
Figure 6.4.: **QoM Approximation. (a) the monitored region of o_j is divided into 5 rings. (b) two types of sectors responding to $f = l_{k_1}$ and $f = l_{k_1+1}$.**

Preliminary 1: Piecewise Constant Approximation for QoM. Let $Q(f.d) = \frac{af^2}{(d+b)^2}$ denote the QoM of an aspect monitored by a camera respect to focal length f and distance d. We use multiple piecewise constant segments $\tilde{Q}(f,d)$ to approximate the QoM $Q(f,d)$, respectively. The distance dimension is divided into $K2$ constant segments (Fig. 6.4(a)); accordingly, R_{max} is divided into little segments which divides the monitored region of an object into K_2 many rings. The focal length dimension is divided into $K1$ constant segments (Fig. 6.4(b)); each focal length l_{k_1} $(0 \leq k_1 \leq K_1)$ generates a γ-AOV R-radius camera (sector) (γ and R calculated via Eq. (6.1)); accordingly, there are total K_1 types of sectors. Formally, the piecewise constant function is defined as follows.

Definition 6.2.1. *Setting $l(0) = F_{min}$, $l(K_1) = F_{max}$, $r(0) = 0$, and $r(K_2) = R_{max}$*[11]*, the piecewise constant QoM function $Q(f,d)$ is defined as follows:*

$$\tilde{Q}(f,d) \triangleq \begin{cases} Q(l(1), r(1)), & f = l(0), d = r(0) \\ Q(l(k_1-1), r(k_2)), & l(k_1-1) < f \leq l(k_1), \\ & r(k_2-1) < d \leq r(k_2) \\ 0, & f > l(K_1), d > r(K_2). \end{cases} \tag{6.3}$$

After two dimensional approximation, any point in each ring monitored by one same type of sector, the QoM is a same constant. Fig. 6.5(a) and Fig. 6.5(b) show an example of approximate QoMs of o_1 under two focal lengths.



(a) Approximate QoM under f_{k_1} (b) Approximate QoM under f_{k_1+1}

Figure 6.5.: **Example of Step 1-1. The approximate QoMs of o_1 are respectively $Q(l_{k_1}, r_3)$ and $Q(l_{k_1+1}, r_3)$ since $R_{k_1} \leq r_3$.**

Now, we provide a sufficient condition to bound the approximation error in following two theorems. We omit their proofs as they are the same as the proofs of Thoe. 5.2.2 and Theo. 5.2.1.

Theorem 6.2.1. *Holding fixed $f = f_0$, setting $r(0) = 0$, $r(K_2) = R_{max}$, and $r(k_2) = b((1+\epsilon_1)^{k_2/2} - 1)$, $(k_2 = 1, \cdots, K_2 - 1$, and $K_2 = \left\lceil \frac{ln(Q(f,0)/Q(f,R_{max}))}{ln(1+\epsilon_1)} \right\rceil)$, the approximation error is*

$$1 \leq \frac{Q(f_0,d)}{\tilde{Q}(f_0,d)} \leq 1 + \epsilon_1.$$

[11] K_1 and K_2 control the approximate error, and obviously.

Theorem 6.2.2. *Setting* $l(0) = F_{min}$*,* $l(K_1) = F_{max}$*, and* $l(k_1) = F_{min} \cdot \left(\frac{1+\epsilon_2}{1+\epsilon_1}\right)^{k_1/2}$*,* $(k_1 = 1, \cdots, K_1 - 1$*, and* $K_1 = \left\lceil \frac{ln(F_{max}/F_{min})}{ln((1+\epsilon_2)/(1+\epsilon_1))} \right\rceil)$*, we have the approximation error:*

$$1 \leq \frac{Q(f,d)}{\tilde{Q}(f,d)} \leq 1+\epsilon_2.$$

Preliminary 2: Monitoring Utility Approximation for an Object. Fig. 6.6 illustrates the basic idea of the *aspect combining* to approximate the monitoring utility of an object. We use $\frac{2\pi}{\Delta A}$ directions centered at o_j with even space ΔA to divide o_j's 2π aspects. Efficient angle θ in aspect coverage model reduce to the completely covered ΔA intervals. For example, in Fig. 6.6, θ is approximated as interval $[\Delta A, 4\Delta A]$ since $[0, \Delta A)$ and $[4\Delta, 5\Delta A)$ are not completely covered. Theo. 6.2.3 provides a sufficient condition to bound the error from aspect combining.

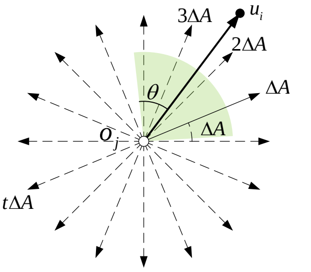


Figure 6.6.: **Aspect combining.**

Theorem 6.2.3. *Setting* $\Delta A = cd(360, \theta)$ *(*$cd(\cdot,\cdot)$ *is the Common Divisor of two integers) and* $\epsilon_\Delta = \frac{2\Delta A}{\theta - 2\Delta A}$*, then,* u_i *monitors* o_j *with identical efficient interval provided* $\overrightarrow{o_j u_i} \in (t\Delta A, (t+1)\Delta A)$*. The approximation error is*

$$1 \leq \frac{U(\tilde{Q})}{\tilde{U}(\tilde{Q})} \leq 1+\epsilon_\Delta.$$

where $U(\tilde{Q}) = \int_0^{2\pi} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi$.

Proof. We prove this theorem by constructing the extreme cases. Fig. 6.7 and Fig. 6.8 illustrates two examples of extreme cases. Without loss of generality, given $\tilde{Q}$ we have $U(\tilde{Q}) = \int_0^{2\pi} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi$ and $\tilde{U}(\tilde{Q}) = \Delta A \cdot \sum_{t=1}^{\frac{2\pi}{\Delta A}} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A)$. In the best case, Fig. 6.7, $\tilde{U}(\tilde{Q})$ is at most the value of $U(\tilde{Q})$, *i.e.*, every covered θ exactly covers the integral multiple of ΔA. In such case, $\frac{U(\tilde{Q})}{\tilde{U}(\tilde{Q})} = 1$. In the worst case, Fig. 6.8, in which θ is very close to the integral multiple of ΔA and there is no overlapped covered aspects. Formally, $\forall i \leq K, i \in \mathbb{Z}^+, K \leq 2 \times \frac{2\pi}{\theta}, \beta_i \to 0$, *e.g.*, $K = 6$. Let $\Phi(x) \triangleq \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, x)$, then in the worst case, $U(\tilde{Q}) = \int_0^{2\pi} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, \varphi) d\varphi = \sum_{t=1}^{\frac{2\pi}{\theta}} \frac{\theta}{\Delta A} \cdot \Delta A \cdot \Phi(t\Delta A)$ because the limit subjects to $\beta_i \to 0$ and there are at most K (constant) β_i s, namely, the $\frac{K}{2}$ intervals from $(t-1)\Delta A + \beta_i$to $t\Delta A$ and $\frac{K}{2}$intervals from $(t-1)\Delta A$to $t\Delta A - \beta_i$all cover com-

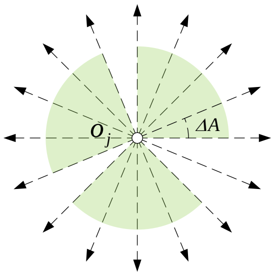


Figure 6.7.: **Best Case.**

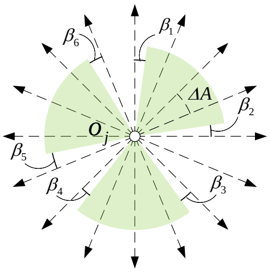


Figure 6.8.: **Worst Case.**

plete aspects in ΔA. Additionally, $\tilde{U}(\tilde{Q}) = \Delta A \cdot \sum_{t=1}^{\frac{2\pi}{\Delta A}} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) = \sum_{t=1}^{\frac{2\pi}{\theta}} (\frac{\theta}{\Delta A} - 2) \cdot \Delta A \cdot \Phi(t\Delta A)$because the $\frac{K}{2}$intervals from $(t-1)\Delta A + \beta_i$to $t\Delta A$and $\frac{K}{2}$intervals from $(t-1)\Delta A$to $t\Delta A - \beta_i$all cover zero aspects in ΔA. Thus, we have $\frac{U(\tilde{Q})}{\tilde{U}(\tilde{Q})} = \frac{\sum_{t=1}^{\frac{2\pi}{\theta}} \frac{\theta}{\Delta A} \cdot \Delta A \cdot \Phi(t\Delta A)}{\sum_{t=1}^{\frac{2\pi}{\theta}} (\frac{\theta}{\Delta A} - 2) \cdot \Delta A \cdot \Phi(t\Delta A)} = 1 + \frac{2\Delta A}{\theta - 2\Delta A} = 1 + \epsilon_{\Delta}$. □

After the aspect combining, monitoring utility of o_j with a UAV set U reduces from an integral to a weighted sum:

$$\tilde{U}(\tilde{Q}) = \tilde{U}(\mathcal{U}, o_j) = \Delta A \cdot \sum_{t=1}^{\frac{2\pi}{\Delta A}} \max_{\mathcal{A}} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A). \tag{6.4}$$

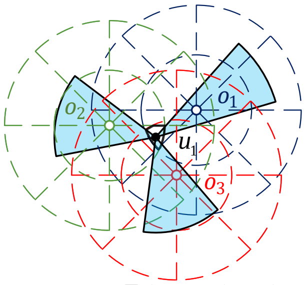


Figure 6.9.: **Discretization.**

Discretizing to Cells. Following above two preliminaries, we discretize the entire area into *cells*. Fig. 6.9 illustrates the basic idea of it, which contains two steps. First, following the distance-dimension discretization, we draw concentric circles with radius $r(0)$, $r(1)$, $\cdots$, $r(K_2)$ at each object, respectively. Second, following the aspect combining, we draw aspects with $\Delta A, 2\Delta A, \cdots, 2\pi$ angles centered at each object. For example. in Fig. 6.9, $\Delta A = \pi/4$, thus the entire area is divided into 148 cells. UAV u_1 lies in the ring 1 of o_1, o_2, and o_3, and its three cameras respectively monitor o_1 with discretized focal length $l(k_1+1)$, and o_2 and o_3 with $l(k_1)$. Due to geometric symmetry, if a UAV lies in a ring with radius $r(k_2)$ and $r(k_2+1)$ of an object, then this object must also lie in a ring with the same radius centering on this UAV, leading to a constant approximate QoM $Q(f, r(k_2+1))$ to this object. The approximated QoM of o_1 from u_1 is $Q(l(k_1+1), r(1))$, and the one of o_2 and o_3 is $Q(l(k_1), r(2))$.

Now, we state the cells number and properties of each cell enforced by aforementioned scheme in the following theorem.

Theorem 6.2.4. *Following the cell discretization scheme, N objects partition the whole space into $O(N^2\epsilon_1^{-2}\epsilon_\Delta^{-2})$ cells; UAV with the certain sector at any point in each cell contribute the same utility to surrounding objects.*

Proof. We use the Claim 4.2.1 in Chap. 4 of the upper bound of the number of discretized area on 2D plane by M uniform sectors intersecting with each other as follows. We repeat this claim as follow.

Claim 1.2. 1. *The number of partitioned cells on 2D plane by n uniform sectors intersecting with each other is at most $5n^2-5n+2$.*

Given N objects, there are at most $\frac{2\pi}{\Delta A}\cdot NK_2$ number of sectors. By Claim 4.2.1, Theo. 6.2.1, and Theo. 6.2.3, we have $\frac{2\pi}{\Delta A}=O(\epsilon_\Delta^{-1})$ and $K_2=O(\epsilon_1^{-1})$. Substitute $O(\epsilon_\Delta^{-1})$ and $O(\epsilon_1^{-1})$ into upper bound in Claim 4.2.1, the number of cells is at most $5\times\left(\frac{2\pi}{\Delta A}\cdot NK_2\right)^2-5\times\left(\frac{2\pi}{\Delta A}\cdot NK_2\right)+2=O(N^2\epsilon_1^{-2}\epsilon_\Delta^{-2})$. Therefore, the number of cells is $O(N^2\epsilon_1^{-2}\epsilon_\Delta^{-2})$. □

6.2.2. Step 2: Representative Arrangements Extraction (RAE).

The core idea of RAE is finding out the representative arrangements (RA) who contribute maximal monitoring utility. By Theo. 6.2.4, the RA only relies on the geometry relationship between objects and waypoints. We prove the necessary condition of RA; design two algorithms to respectively extract all representative locations and representative subarragements; and prove the output of RAE contains all RAs. Our goal is converting WiPlan to a combinatorial optimization problem.

Preliminaries. We first define maximal set to assist analysis. Given two arrangements $\langle u_1,\Phi_{u1},\mathcal{F}_{u1}\rangle$ $\langle u_2,\Phi_{u2},\mathcal{F}_{u2}\rangle$ and their monitored object sets O_1, O_2. If $O_1=O_2$, we say two arrangements are *equivalent*. If $O_1\supseteq O_2$, we say $\langle u_1,\Phi_{u1},\mathcal{F}_{u1}\rangle$,*dominates* $\langle u_2,\Phi_{u2},\mathcal{F}_{u2}\rangle$. Specially, if there is no arrangement $\langle u_i,\Phi_{ui},\mathcal{F}_{ui}\rangle$ such that $\langle u_i,\Phi_{ui},\mathcal{F}_{ui}\rangle$ dominates $\langle u_1,\Phi_{u1},\mathcal{F}_{u1}\rangle$, we say O_1 is a *Maximal Set (MS)* .

By definition of maximal set, monitoring MSs is always better than monitoring its subsets; the arrangements monitoring MSs are called *representative arrangements (RAs).* Accordingly, our task is extracting all RAs.

Step 2-1: Representative Locations Extraction. We first extract the location of RAs.

Algorithm 7: Representative Locations Extraction

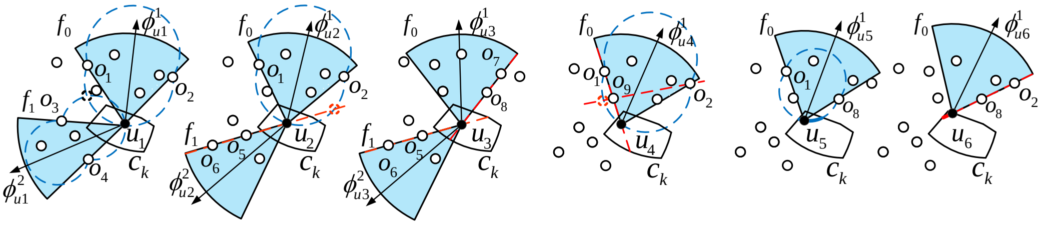
```
Input: Cell $c_k$ and corresponding $\mathcal{O}_{max}$ ($\mathcal{O}_{max}$ for $c_k$ is the set of objects can be
       monitored by a UAV $u_i$ with $2\pi$ AOV and $R_{max}$ range at arbitrary points in $c_k$).
Output: $\Lambda$ and corresponding fixed subarrangements.
for all types of discretized focal length, say $f_0$ do
    for all two pairs of objects $o_p$, $o_q$ and $o_s$, $o_t$ in $\mathcal{O}_{max}$ do
        for all types of discretized focal length $f_1$ do
            Draw two arcs crossing $o_p$, $o_q$ with inscribed angle $\gamma_{f_0}$, and two arcs crossing
              $o_s$, $o_t$ with $\gamma_{f_1}$, respectively.
        end
        for each pair of objects in the two pairs do
            Draw two arcs crossing this pair of objects with inscribed angle $\gamma_{f_0}$, and a
              straight line crossing the rest pair, respectively.
        end
        Draw straight line crossing $o_p$, $o_q$ and $o_s$, $o_t$, respectively.
    end
    for all three objects $o_p$, $o_q$, and $o_s$ in $\mathcal{O}_{max}$ do
        for all pairs $o_1$, $o_2$ from three objects do
            Draw two arcs crossing $o_1$, $o_2$ with inscribed angle $\gamma_{f_0}$, and two straight lines
              crossing $o_1$, $o_3$ as well as $o_2$, $o_3$, respectively.
        end
    end
    for all pairs of $o_p$, $o_q$ in $\mathcal{O}_{max}$ do
        Draw two arcs crossing $o_p$ and $o_q$ with $\gamma_{f_0}$ to get the intersecting curve with cell
          $c_k$; Randomly select a point on intersecting curve, add it to $\Lambda$ and record
          corresponding fixed subarrangements;
        Draw a straight line crossing $o_p$ and $o_q$ to get the intersecting curve with cell $c_k$;
          Randomly select a point on intersecting line, add it to $\Lambda$ and record
          corresponding fixed subarrangements.
    end
    Randomly select a point inside cell $c_k$ and add it to $\Lambda$.
end
```

Definition 6.2.2. ***Representative Location (RL):*** *Given an arrangement $\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle$, u_i is a representative location iff at least one camera with $\langle u_i, \phi^k_{ui}, f^k_{ui} \rangle$ monitor a maximal set.*

By Def. 6.2.2, we design the Representative Locations Extraction (RLE) algorithm in Alg. 7. Fig. 6.10 shows an example of how algorithm operates in one loop of focal length f_0. Now, we state the guarantees enforced by RLE. Let Λ be the extracted representative locations set.

Theorem 6.2.5. Λ *contains all the representative locations that can possibly generate Maximal Sets.*

Proof. Def. 6.2.2 defines that the candidate locations satisfy at least one of its cameras monitoring a maximal set. Thus, we borrow the insight of Theo. 5.2.5 from Chap. 5 and claim the following lemma. We omit its proof as it is the same as Theo.



(1) Two Arcs (2) Arc & Line (3) Two Lines
(a) Two Pairs of Objects (b) Three Objects
(1) Arc (2) Line
(c) Two Objects

Figure 6.10.: **Representative Locations Extraction. (a), (b), and (c) correspond to Step 2-7, Step 8-10, and Step 11-13 of Alg. 7, respectively. Step 14 is omitted because the space constraint and it is easy to understand.**

5.2.5

Lemma 6.2.1. *The two necessary conditions for one camera monitoring a maximal set are: 1. two objects located on the same one radius; 2. two objects respectively located on two radiuses.*

With above Lemma, we prove our theorem as follows. Apparently, we shall stay at one of following three statuses after above transformation. We use *Case a-b* to denote the case *b* under *a* sector(s) status. **Case 1-1:** Three objects hit on the radiuses. In this case, there is the only candidate location, such as Fig. 6.10(b), corresponding to Step 8-10 in Alg. 7. **Case 1-2:** Two objects hit on the radiuses. In this case, the center of the sector can slide on the arc or straight line, such as Fig. 6.10(c), corresponding to Step 11-13 in Alg. 7. If during the sliding there are other objects hitting on radius, it falls into the scope of Case 1-1. **Case 1-3:** One object hits on the radius. In this case, the center of the sector can be located at any point in the cell, corresponding to Step 14 in Alg. 7.

Now, we consider that *there are object(s) hitting on the radiuses of other sectors during above transformation.* By one-sector analysis, the total cases of two sectors are the combinations of above one-sector condition, *i.e.*, 2^3 cases. Yet, they can combine to 4 cases. **Case 2-1:** As long as three objects hit on the radiuses of one sector. This sector can determine the only location, then it falls into the scope of Case 1-1. **Case 2-2:** Two pairs of objects hit on the radiuses of two sectors. There are three subcases, but each of them can determine the only location, such as Fig. 6.10(a), which corresponds to Step 2-7 in Alg. 7. **Case 2-3:** One pair objects hit on the radius of one sector and an other object hits on the radius of the other. In this case, the center of the sector can slide on the arc or straight line while keeping the individual object sliding on the radius, so this case falls into the scope of Case 1-2. If during the sliding, there are other objects hitting on radius, it falls into the scope of Case 2-1 or Case 2-2. **Case 2-4:** Two objects hit on the radius of each sector, respectively. This case falls into the scope of Case 1-3. All the cases of more than two sectors fall into aforementioned cases. Consequently, Λ contains all the candidate locations. □

Step 2-2: Representative Subarrangements Extraction (RSE). After Alg. 7, we obtain all the locations of RAs. The representative subarrangements extraction extracts all directions and focal lengths that may monitor maximal sets for all locations in Λ. The details of the candidate subarrangements extraction procedure are presented in Alg. 8. Fig. 6.11 shows an example of extracting representative subarrangements (RSs) with focal length f_0 at representative location u_i. It rotates the sector at location u_i such that its direction varies from 0° to 360°. During this progress, it tracks the current monitored objects while identifies if it covers MSs and records corresponding subarrangements. All extracted subarrangements construct set Γ, and Γ_l denotes the RSs set of location l.

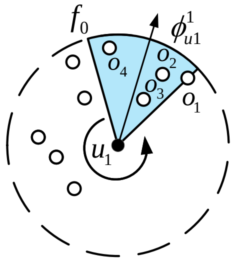


Figure 6.11.: **Representative Subarrangements Extractio.**

Algorithm 8: Representative Subarrange Extraction

```
Input: Cell $c_k$, $\mathcal{O}_{max}$, $\Lambda$, and representative location $u_i$.
Output: All candidate arrangements.
for all representative locations in $\Lambda$ do
    for all types of discretized focal length $l_{k1}$ do
        Initialize the direction of sector to 0°.
        while rotated angle is less than 360° do
            while no object will fall out of sector do
                Rotate the UAV anticlockwise.
            end
            Add current direction as well as focal length to the candidate arrangements
              set, and record the corresponding monitored objects.
            while no object will enter the sector do
                Rotate the UAV anticlockwise.
            end
        end
    end
end
```

6.2.3. Step 3: Problem Reformulation and Solution.

After discretization and RSE, WiPlan is transformed to a combinatorial optimization problem **WiPlan-T**, that is, *select a set L with M locations from Λ, and for each location l in L select K subarrangements from Γ_l to construct arrangement A_l, such that the monitoring utility for N objects is maximized.* Yet, the challenge is that both selecting M locations from Λ and selecting K subarrangements from Γ_l are variants of NP-hard *Set Cover* problem [161]. Further, locations selection and subarrangements selection are tightly coupled because selected locations and their arrangements impact the following locations and subarrangements selection, resulting in a much more significant challenge.

Algorithm 9: WiPlan-T Algorithm

```
   Input: Representative locations set $\Lambda$, representative subarrangements set $\Gamma$, objects set $\mathcal{O}$.
   Output: Selected arrangements $A_L$, selected locations $L$.
1  Initialization: $\Delta_H^a := H_{L\cup\{l\}}(A_L \cup \{a\}) - H_{L\cup\{l\}}(A_L)$; $\Delta_f^{A_L} := f(A_L \cup \{A_l\}) - f(A_L)$;
     $A_L := \emptyset$; $L := \emptyset$.
2  while $|A_L| \leq M$ do
3      foreach $l \in \Lambda$ do
4          $i :=$ number of fixed subarrangements in Alg. 7; $A_l := \emptyset$.
5          while $|A_l| \leq K - i$ do
6              $a^* = \arg\max\{\Delta_H^a | a \in \Gamma_l\}$;
7              $A_l := A_l \cup \{a^*\}$.
8          end
9      end
10     $A_{l^*} = \arg\max\{\Delta_f^{A_l} | l \in \Lambda\}$;
11     $A_L := A_L \cup \{A_{l^*}\}$; $L := L \cup \{l^*\}$; $\Lambda := \Lambda \backslash L$.
12 end
```

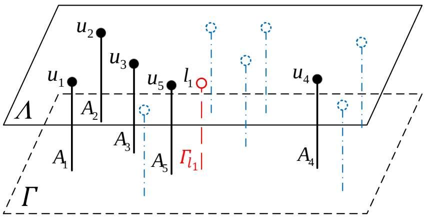


Figure 6.12.: **WiPlan-T Problem. The bottom dotted plane indicate the lower level subproblem in problem P2, while the top plane indicate the upper one in problem P2.**

To address this challenge, we present a two-level greedy algorithm in Alg. 9. In each iteration, Alg. 9 first evaluates the monitoring utility $H(\cdot)$ (defined in Section 4.5) of each location $l \in \Lambda \backslash L$ by greedily selecting $K - i$ subarrangements from Γ_l under current selected arrangements A_L (Step 3-7). For example, Fig. 6.12 illustrates the 6-th iteration which is evaluating l_1 under current selected arrangements set $\{A_1, A_2, A_3, A_4, A_5\}$. After evaluating all the locations in $\Lambda \backslash L$, Alg. 9 selects arrangement A_l^* of location l^* who contributes the maximum local marginal benefit of $f(\cdot)$ (defined in (**P3**) in Section 4.5) and adds A_l^* and l^* to A_L and L respectively, while deletes l^* from Λ (Step 8-9).

6.3. Theoretical Analysis

As shown in Fig. 6.12, WiPlan-T can be regarded as a two-level optimization problem. Let Π be the set of all arrangements constructed by lower level subproblem (we omit the formulation of lower level subproblem as the limited space), x_i be the indicator variable denoting whether the i-th location is selected or not. Then, upper level subproblem can be reformulated as:

$$\begin{aligned}\max \quad & \frac{\Delta A}{2\pi N}\sum_{j=1}^{N}\sum_{t=1}^{\frac{2\pi}{\Delta A}}\Big(\max_{\langle u_i,\Phi_{ui},\mathcal{F}_{ui}\rangle\in\Pi} x_i\tilde{Q}(u_i,o_j,f_{ui}^k,\phi_{ui}^k,t\Delta A)\Big)\\ s.t. \quad & \sum_{i=1}^{|\Lambda|} x_i = M \ (x_i \in 0,1).\end{aligned}$$

Properties of WiPlan-T Problem. We first give following definition to assist the properties analysis. (1) A real-valued set function $f: 2^S \to \mathbb{R}$ (S is a set) is nonnegative, monotonic, and submodular [142] iff.: 1. $f(A) \geq 0$; 2. $f(A\cup\{e\}) \geq f(A)$ $(A \subseteq S, e \in S\backslash A)$; 3. $f(A\cup\{e\}) - f(A) \geq f(B\cup\{e\}) - f(B)$ $(A \subseteq B \subseteq S, e \in S\backslash B)$. (2) A Uniform Matroid [142] $\mathcal{M} = (S,L)$ where S is a finite ground set, $L \subseteq 2^S$ is a collection of independent sets, k is an integer, such that 1. $\emptyset \in L$; 2. if $X \subseteq Y \in L$, then $X \in L$; 3. if $X,Y \in L$, and $|X| < |Y|$, then $\exists y \in Y\backslash X$, $X\cup\{y\} \in L$. 4. $L = \{X \subseteq S : |X| \leq k\}$.

Following the above definitions, we rewrite the two level subproblems as follows.

(P2). Lower Level Subproblem: Given L selected locations with arrangements A_L, and an additional location $e \in \Lambda\backslash L$, select K subarrangments from Γ_e forming arrangements A_e to maximize the overall monitoring utility for N objects.

$$\begin{aligned}&\max\ H_{L\cup\{e\}}(Y)\\ &=\frac{\Delta A}{2\pi N}\sum_{j=1}^{N}\sum_{t=1}^{\frac{2\pi}{\Delta A}}\Big(\max_{\langle u_i,\Phi_{ui},\mathcal{F}_{ui}\rangle\in A_L\cup Y}\tilde{Q}(u_i,o_j,f_{ui}^k,\phi_{ui}^k,t\Delta A)\Big)\\ &s.t. \quad Y \in \Gamma_e,\ A_e = \{Y \subseteq \Gamma_e : |Y| \leq K\}.\end{aligned}$$

Upper Level Subproblem: Based on the constructed arrangements on the lower level, select M arrangements to maximize the overall monitoring utility of N objects.

$$\begin{aligned}&\max\ f(X) = \frac{\Delta A}{2\pi N}\sum_{j=1}^{N}\sum_{t=1}^{\frac{2\pi}{\Delta A}}\Big(\max_{\langle u_i,\Phi_{ui},\mathcal{F}_{ui}\rangle\in X}\tilde{Q}(u_i,o_j,f_{ui}^k,\phi_{ui}^k,t\Delta A)\Big)\\ &s.t. \quad X \in L,\ L = \{X \subseteq \Lambda : |X| \leq M\}.\end{aligned}$$

We observe both lower and upper level subproblems follow the following property.

Lemma 6.3.1. *The objective function $f(X)$ and $H_{L\cup\{e\}}(X)$ are nonnegtive, monotone and submodular, and their constraint are both uniform matroids.*

Proof. According to the definition of submodular, we need to check the three listed requirements of $f(X)$ to prove it is monotone and submodular. For ease of analysis, define

$$\tau(X,j) = \sum_{t=1}^{\frac{2\pi}{\Delta A}} \Big(\max_{\langle u_i, \phi_{ui}^k, f_{ui}^k \rangle \in X} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) \Big). \tag{6.5}$$

First, when the number of UAVs is 0, obviously $\tau(X,j) = 0$, thus, we have $f(\emptyset) = 0$.

Second, let A be a set of arrangements in Γ, $e \in \Gamma \backslash A$. Then, $\tau(X \cup e, j) - \tau(X,j) \geq 0$ since: **1.** the $\sum \cdot$ function is non-decreasing for $\cdot \geq 0$; **2.** max function is also non-decreasing, *i.e.*, $\max_{\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle \in X \cup \{e\}}$ $\tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) - \max_{\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle \in X} \tilde{Q}(u_i, o_j, f_{ui}^k, \phi_{ui}^k, t\Delta A) \geq 0$ to any specific t. Therefore,

$$f(A \cup \{e\}) - f(A) = \frac{1}{N} \sum_{j=1}^{N} (\tau(A \cup \{e\}, j) - \tau(A, j)) \geq 0.$$

Third, let A and B be two sets of arrangements in Γ where $A \subseteq B \subseteq \Gamma$, and $e \in \Gamma \backslash B$. Recall the definition of $\tau(X,j)$ in Eq. (6.5), we have

$$\tau(X \cup \{e\}, j) - \tau(X,j) = \begin{cases} 0, & \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e\Delta A) \leq \tilde{Q}_{max}(X), \\ \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e\Delta A) - \tilde{Q}_{max}(X), & otherwise. \end{cases} \tag{6.6}$$

where $\tilde{Q}_{max}(X) = \max_{\langle u_i, \Phi_{ui}, \mathcal{F}_{ui} \rangle \in X} \tilde{Q}(u_i, o_e, f_{ui}^k, \phi_{ui}^k, t_e\Delta A)$, *i.e.*, the maximum QoM provided by some arrangements in X who monitoring the same aspect $t_e\Delta A$ of the same object o_e as arrangement e. We observe that: (1) for given e, its QoM $\tilde{Q}(u_e, o_e, \phi_{ue}^{ke}, t_e\Delta A)$ is a constant; (2) $\tilde{Q}_{max}(A) \leq \tilde{Q}_{max}(B)$ since $A \subseteq B$. Then, we use the proof by cases.

Case 1: $\tilde{Q}(u_e, o_e, f_{ui}^k, \phi_{ue}^{ke}, t_e\Delta A) \leq \tilde{Q}_{max}(A)$. By Eq. (6.6), we have $\tau(A \cup \{e\}, j) - \tau(A, j) = 0$. The observation (2) implies that $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e\Delta A) \leq \tilde{Q}_{max}(A) \leq \tilde{Q}_{max}(B)$. Thus, by Eq. (6.6), $\tau(B \cup \{e\}, j) - \tau(B, j) = 0$. Therefore, $\tau(A \cup \{e\}, j) - \tau(A, j) = \tau(B \cup \{e\}, j) - \tau(B, j)$.

Case 2: $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e\Delta A) > \tilde{Q}_{max}(A)$. According to Eq. (6.6), we have $\tau(A \cup \{e\}, j) - \tau(A, j) = \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e\Delta A) - \tilde{Q}_{max}(A)$. By observation (2), we have $\tilde{Q}_{max}(A) \leq \tilde{Q}_{max}(B)$, which implies $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) \leq \tilde{Q}_{max}(B)$ or $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) > \tilde{Q}_{max}(B)$.

Case 2.1: $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) \leq \tilde{Q}_{max}(B)$. Based on Eq. (6.6), $\tau(B \cup \{e\}, j) - \tau(B, j) = 0$. Thus, $\tau(A \cup \{e\}, j) - \tau(A, j) \geq \tau(B \cup \{e\}, j) - \tau(B, j)$ since $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) > \tilde{Q}_{max}(A) > 0$.

Case 2.2: $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) > \tilde{Q}_{max}(B)$. Based on Eq. (6.6), $\tau(B \cup \{e\}, j) - \tau(B, j) = \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) - \tilde{Q}_{max}(B)$. Furthermore, according to the observation (1) and (2), we have $\tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) - \tilde{Q}_{max}(A) \geq \tilde{Q}(u_e, o_e, f_{ui}^{ke}, \phi_{ue}^{ke}, t_e \Delta A) - \tilde{Q}_{max}(B)$, *i.e.*, $\tau(A \cup \{e\}, j) - \tau(A, j) \geq \tau(B \cup \{e\}, j) - \tau(B, j)$.

Because the $\tau(A \cup \{e\}, j) - \tau(A, j) \geq \tau(B \cup \{e\}, j) - \tau(B, j)$ holds in all cases, we have

$$\begin{aligned}
&(f(A \cup \{e\}) - f(A)) - (f(B \cup \{e\}) - f(B)) \\
=&\frac{1}{N}\sum_{j=1}^{N}((\tau(A \cup \{e\}, j) - \tau(A, j)) \\
&- (\tau(B \cup \{e\}, j) - \tau(B, j))) \geq 0.
\end{aligned}$$

To sum up, we can conclude that $f(X)$ in **P2** is monotone and submodular. In addition, the constraint of **P2** is clear a uniform matroid constraint. Thus, proof completes. □

Approximation Ratio By Lem. 6.3.1, both upper and lower level optimizations fall into the scope of maximizing a monotone submodular function subject to uniform matroid constraint, which can be solved by an approximate algorithm with greedy policy [142]. Then, we have the following theorem. We only prove the most tricky part, *i.e.*, the approximation ratio between Algothim 9 and problem **P2**.

Theorem 6.3.1. *The WiPlan solution achieves an approximation ratio of* $\frac{1}{2} + \frac{1}{2e^2} - \frac{1}{e} - \epsilon$, *where* $\epsilon = \frac{e^2\tau + \tau - 2e\tau}{2e^2 + 2e^2\tau}$ *and* $\tau = \epsilon_\Delta + \epsilon_2 + \epsilon_\Delta \epsilon_2$, *and its time complexity is* $O(MKN^6\epsilon^{-4})$.

Proof. Let **OPT** be the optimal solution of **P2**.

Let $\{l_1, l_2, \cdots, l_M\}$ denote the locations set of selected arrangements, where l_i is the location of selected i-th arrangement in a fixed order generated by scheme L. Here, $\{l_1^{++}, l_2^{++}, \cdots, l_M^{++}\}$ denotes the locations of selected arrangements set generated by scheme L^{++} that greedily selects subarrangements and greedily selects location, *i.e.*, Alg. 9. $\{l_1^{*+}, l_2^{*+}, \cdots, l_M^{*+}\}$ denotes the locations generated by scheme L^{*+} that selects *optimal* subarrangements for each location and greedily selects locations. Note that the orders of l_i in the two sets are not the same since the different schemes.

Next, we introduce three auxiliaries to assist our proof. **1.** $\Delta L_{i-1}(\widetilde{l_i})$ denotes the marginal utility increment of adding arrangement $\widetilde{l_i}$ to arrangements set L_{i-1}. For example, $\Delta L^{++}_{i-1}(\widetilde{l_i^{++}})$ denotes the marginal utility increment of adding $\widetilde{l_i^{++}}$ in the i-th iteration of Alg. 9. **2.** $\Delta L^{++}_{i-1}(l_i^{*+})$ denotes the marginal utility increment of adding an arrangement associated with scheme L^{++} at location l_i^{*+} rather than arrangement $\widetilde{l_i^{*+}}$. In other words, it is a created arrangement that greedily selects subarrangements at location l_i^{*+}. **3.** $L^{++} \bowtie L^{*+} \triangleq \{l_1^{++}, l_2^{++}, \cdots, l_M^{++}, l_1^{*+}, l_2^{*+}, \cdots, l_M^{*+}\}$ which concatenates two locations set L^{++} and L^{*+}. Its physical meaning is the locations generated by scheme that *deploy* $2M$ *of UAVs, the first* M *UAVs deploy by scheme* L^{++} *and the second* M *UAVs deploy by scheme* L^{*+}. Then, $\Delta L^{++} \bowtie L^{*+}_{i-1}(l_i^{*+})$ denotes the increment of marginal utility by greedily selecting subarrangements at location l_i^{*+} on $M+i-1$ locations including M UAVs by scheme L^{++} and $i-1$ UAVs by L^{*+}.

Now, let us bound the approximation ratio. First, $\Delta L^{++}_{i-1}(\widetilde{l_i^{++}}) \geq \Delta L^{++}_{i-1}(l_i^{*+})$ since the greedy property, namely, each selected location with scheme L^{++} contribute the most marginal utility increment, of course, its marginal utility is larger than any other locations following scheme L^{++}. Second, $\Delta L^{++}_{i-1}(l_i^{*+}) \geq \Delta L^{++} \bowtie L^{*+}_{i-1}(l_i^{*+})$ since the submodularity of $f(X)$ and the fact $L^{++}_{i-1} \subseteq L^{++} \bowtie L^{*+}_{i-1}$. Third, because of the definition $\Delta L^{++} \bowtie L^{*+}_{i-1}(l_i^{*+}) = H_{L^{++} \bowtie L^{*+}_{i-1} \cup \{l_i^{*+}\}}(X)$, Lem. 6.3.1, and greedily subarrangements selection, we have$\Delta L^{++} \bowtie L^{*+}_{i-1}(l_i^{*+}) \geq (1-1/e) \cdot \Delta L^{++} \bowtie L^{*+}_{i-1}(\widetilde{l_i^{*+}})$ [142], where $\widetilde{l_i^{*+}}$ is the optimal subarrangements selection at location l_i^{*+}. Combine above three inequalities, we have

$$\Delta L^{++}_{i-1}(\widetilde{l_i^{++}}) \geq (1-\frac{1}{e}) \cdot \Delta L^{++} \bowtie L^{*+}_{i-1}(\widetilde{l_i^{*+}}). \tag{6.7}$$

Then, summate all marginal utility and by Inequality (6.7) we have

$$\begin{aligned}
\sum_{i=1}^{M} \Delta L^{++}_{i-1}(\widetilde{l_i^{++}}) &\geq \sum_{i=1}^{M} (1-\frac{1}{e}) \cdot \Delta L^{++} \bowtie L^{*+}_{i-1}(\widetilde{l_i^{*+}}) \\
2 \cdot \sum_{i=1}^{M} \Delta L^{++}_{i-1}(\widetilde{l_i^{++}}) &\geq \sum_{i=1}^{M} \Delta L^{++}_{i-1}(\widetilde{l_i^{++}}) + (1-\frac{1}{e}) \cdot \sum_{i=1}^{M} \Delta L^{++} \bowtie L^{*+}_{i-1}(\widetilde{l_i^{*+}}) \\
2 \cdot f(L^{++}) &\geq (1-\frac{1}{e}) \cdot f(L^{++} \bowtie L^{*+}) \\
f(L^{++}) &\geq (\frac{1}{2}-\frac{1}{2e}) \cdot f(L^{++} \bowtie L^{*+}).
\end{aligned}$$

The third inequality holds since $f(L^{++}) = \sum_{i=1}^{M} \Delta L^{++}_{i-1}(\widetilde{l_i^{++}})$ and $f(L^{++} \bowtie L^{*+}) = \sum_{i=1}^{M} \Delta L^{++}_{i-1}(\widetilde{l_i^{++}}) + \sum_{i=1}^{M} \Delta L^{++} \bowtie L^{*+}_{i-1}(\widetilde{l_i^{*+}})$.

By Lem. 6.3.1 and [142], we have $f(L^{*+}) \geq (1 - \frac{1}{e}) \cdot \mathbf{OPT}$. Then,

$$\begin{aligned} f(L^{++}) &\geq (\frac{1}{2} - \frac{1}{2e}) \cdot f(L^{++} \bowtie L^{*+}) \\ &\geq (\frac{1}{2} - \frac{1}{2e}) \cdot f(L^{*+}) \geq (\frac{1}{2} - \frac{1}{2e}) \cdot (1 - \frac{1}{e}) \cdot \mathbf{OPT}. \end{aligned}$$

Hence, we bound the approximation ratio between Alg. 9 and **P2**. Then, combining Theo. 6.2.2, and Theo. 6.2.3, the ultimate approximation ratio can be bounded as follows. □

Time Complexity Analysis. According to Theo. 6.2.4, WiPlan divides the entire space into $O(N^2\epsilon_1^{-2}\epsilon_\Delta^{-2})$ cells. To generate cells, it takes $O(1)$ and thus, it totally takes $O(N^2\epsilon_1^{-1}\epsilon_\Delta^{-1})$ time. Then, Alg. 7 extracts all representative locations (RLs) and maximal set in each cell; it traverses all combinations of four objects, three objects, and two objects (Step 2-13), with all types of focal lengths, and randomly select a point (Step 14).Thus, Alg. 7 takes no more than $O(N^2\epsilon_1^{-2}\epsilon_\Delta^{-2}) \times ((\binom{N}{4} \cdot \binom{K_1}{2} + \binom{N}{3} \cdot K_1 + \binom{N}{2} \cdot K_1) = O(N^6\epsilon_1^{-2}\epsilon_\Delta^{-4})$ time (See Theo 6.2.2 for K_1 and ϵ_2). Next, RSE extracts subarrangements at each RL which takes $O(1)$ hence totally takes $O(N^6\epsilon_1^{-2}\epsilon_\Delta^{-4})$ time. At last, Alg. 9 greedily selects M locations from $O(N^6\epsilon_1^{-2}\epsilon_\Delta^{-4})$ locations, and at each of M locations greedily selects K subarrangements from constant ones, thus takes $O(KMN^6\epsilon_1^{-2}\epsilon_\Delta^{-4})$. Substitute $\epsilon = \Theta(\epsilon_2\epsilon_\Delta)$ in Theo. 6.3.1, the time complexity is $O(KMN^6\epsilon^{-4})$.

6.4. Numerical Simulation

6.4.1. Evaluation Setup

Parameters Setup. In our simulation, objects are uniformly distributed in a $400m \times 400m$ square area. If no otherwise stated, $M = 7, N = 15, K = 3, \theta = 30°, \epsilon_1 = \frac{1}{6}, \epsilon_2 = \frac{1}{4}, \epsilon_\Delta = \frac{1}{3}, i = \sqrt{3}, f \in [30, \frac{\sqrt{3}\times 10}{4-2\sqrt{3}}], \frac{P \cdot r \cdot z}{p} = \frac{10}{\sqrt{3}}$in Equ. (6.1), respectively. The above parameters are set up based on our experimental results with real hardware [8], which let the monitoring region of one camera vary in $\gamma \in [30°, 60°]$and $R \in [5m, 10.2m]$. Each data point in the figures is computed by averaging the results of 200 random topologies.

Baseline Setup. We compare WiPlan with four comparison algorithms. (1) *Random Location and Random Subarrangement (RLRS)* randomly generates locations of UAVs, and randomly selects K directions from $\{0, \alpha, ..., k\alpha, ..., 2\pi\}$and focal lengths from $[30, \frac{\sqrt{3}\times 10}{4-2\sqrt{3}}]$for UAVs, respectively. (2) *Constraint Location and Ran-*

dom Subarrangement (CLRS) discretizes entire monitoring area into square grids whose length is $\sqrt{2}/2 \cdot R_{max}$. Then, randomly select Mlocations from grid points and guarantee all selected locations within at least one circle centering at an object and radius with R_{max}. The subarrangements are generated with the same method of RLRS. (3) *Constraint Location and Greedy Subarrangement (CLGS)* improves CLRS by placing UAVs with the same way of CLRS but greedily selecting subarrangement for each location. (4) *VISIT [104]* the state-of-the-art algorithm with single-camera UAV; we modify it by adding aspect combining as it is designed for directional coverage.

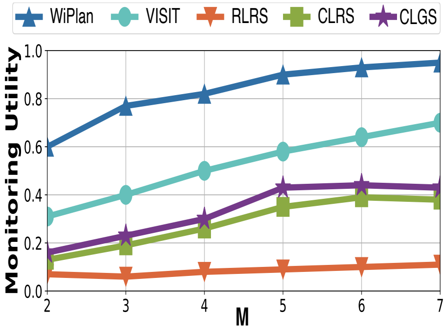


Figure 6.13.: **Number of Waypoints M vs. Utility.**

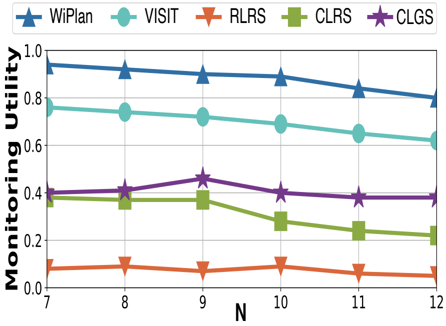


Figure 6.14.: **Number of Objects N vs. Utility.**

6.4.2. Performance Comparison

Impact of Number of Waypoints (M). *Our simulation results show that on average, WiPlan outperforms VISIT, CLGS, CLRS, and RLRS by* 1.76×, 2.13×, 2.83×, *and* 10.05×, *respectively, in terms of* M. Fig. 6.13 shows that the monitoring utility of all algorithms increases monotonically. In particular, the monitoring utility of WiPlan starts at a higher value than comparison algorithms because it has a performance guarantee, *i.e.*, approximation ratio, then it increases until very close to 1. The monitoring utility of comparison algorithms also increases but relatively limited and fluctuant because of their randomness.

Impact of Number of Objects (N). *Our simulation results show that on average, WiPlan outperforms VISIT, CLGS, CLRS, and RLRS by* 1.32×, 2.03×, 3.28×, *and* 9.31×, *respectively, in terms of* N. Fig. 6.14 shows that WiPlan achieves high monitoring utility and performs consistently better than comparison algorithms. However, the monitoring utility of WiPlan has a slight trend of decreasing, and it seems the decrease is more considerable than comparison algorithms. But actually,

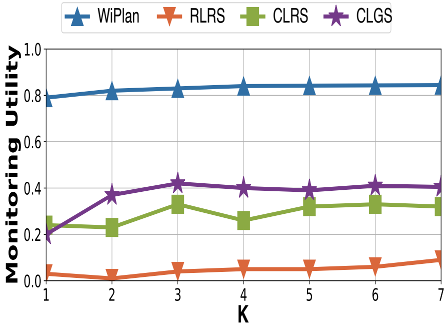


Figure 6.15.: **Number of Cameras K *vs.* Utility.**

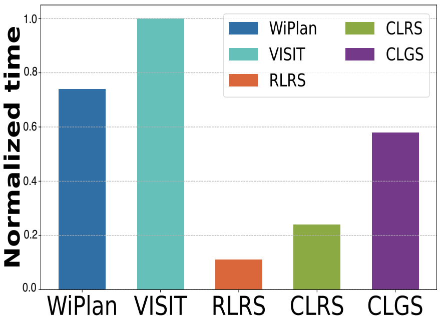


Figure 6.16.: **Computation Time cost.**

the WiPlan only drops 12.3% while the three comparison algorithms drop 19.2% on average.

Impact of Number of Cameras (K). *Our simulation results show that on average, WiPlan outperforms CLGS, CLRS, and RLRS by* 2.93×, 3.16× *and* 9.13×, *respectively, in terms of K.* Fig. 6.15 shows that WiPlan's monitoring utility increases first and then stays at a high value. However, comparison algorithms have a slight but unstable increment because of their random selection mechanism.

Comparison of time cost. *Our simulation results show that on average, WiPlan outperforms CLGS, CLRS, and RLRS by* 2.93×, 3.16× *and* 9.13×, *respectively, in terms of K.* Fig. 6.16 shows that WiPlan cost only 73% computation time of VISIT but get 1.73× utility (see Fig. **??**). The reason is VISIT consists of matrix multiplication which cost the most part of time. Fig. 6.16 also illustrates that WiPlan only use 17% more computation time than random mechanism CLGS but get 2.13× performance. Although RLRS and CLRS cost very short computation time, they trade from exreme low monitoring utility.

Impact of approximate error. We study the impact of approximation errors ϵ_Δ and ϵ_1; both of them vary from 0.1 to 0.5. Fig. 6.17 depicts the results, where each point on the surface plots an average value of 200 experiment results of different topologies. We observe that the monitoring utility always stays at a high value (≥ 0.85) when ϵ_1 and ϵ_Δ are both ≤ 0.4.

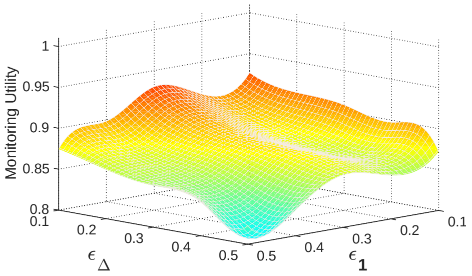
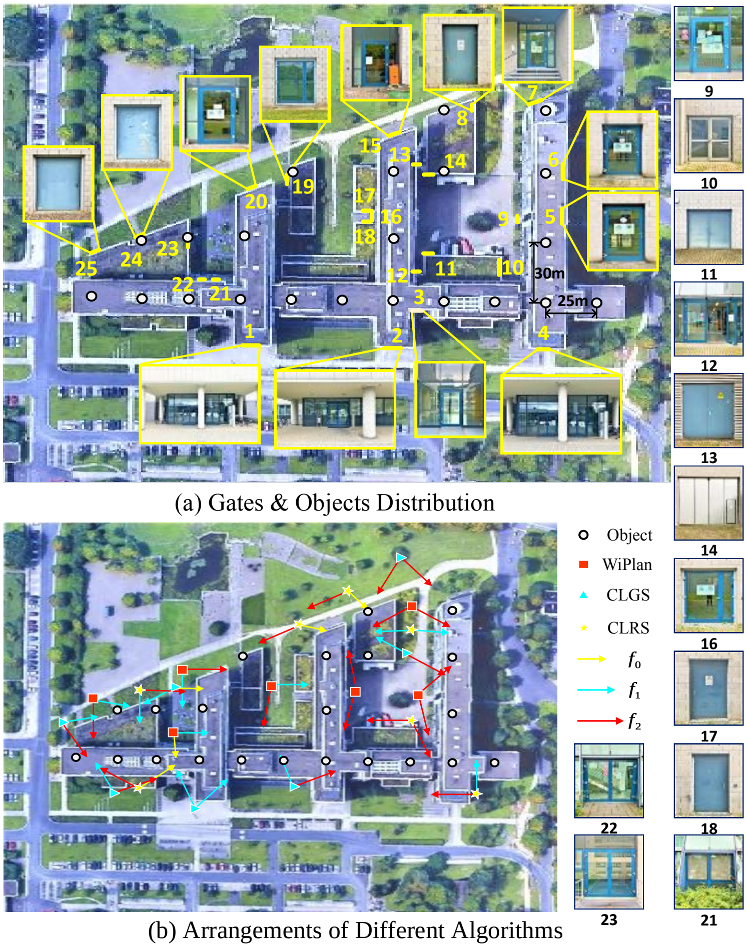


Figure 6.17.: **Approximate Error impacts.**



(a) Gates & Objects Distribution

(b) Arrangements of Different Algorithms

Figure 6.18.: **Ground Truth and Arrangements.**

6.5. Field Experiment

Experiment Setup. As shown in Fig. 6.18, our testbed consists of a teaching building with 25 gates and one two-camera UAVs with 7 waypoints. The experiment field is $150m \times 200m$, and the parameters of the camera, $\gamma \in [30°, 60°]$and $R \in [20m, 45m]$, are set up based on our experimental results with real hardware [8]. The approximate errors are set to $\epsilon_1 = 1/3$, $\epsilon_2 = 4/11, \epsilon_\Delta = 2/3$, which leads the range of focal length value discretized into 3 intervals, *i.e.*, 3 different camera models. In Fig. 6.18, 3 arrows named f_0, f_1, and f_2 with 3 different colors and lengths denote 3 types of camera models. Their AoVs γ and ranges R are 60° and $20m$, 45° and $30m$, and, 30° and $45m$. Since the teaching building is too large, we discretize it into 22 parts indicated with 22 objects, as shown in Fig. 6.18(a). To avoid collision with the building, UAVs fly in the outside air and keep $15m$ away from the objects.

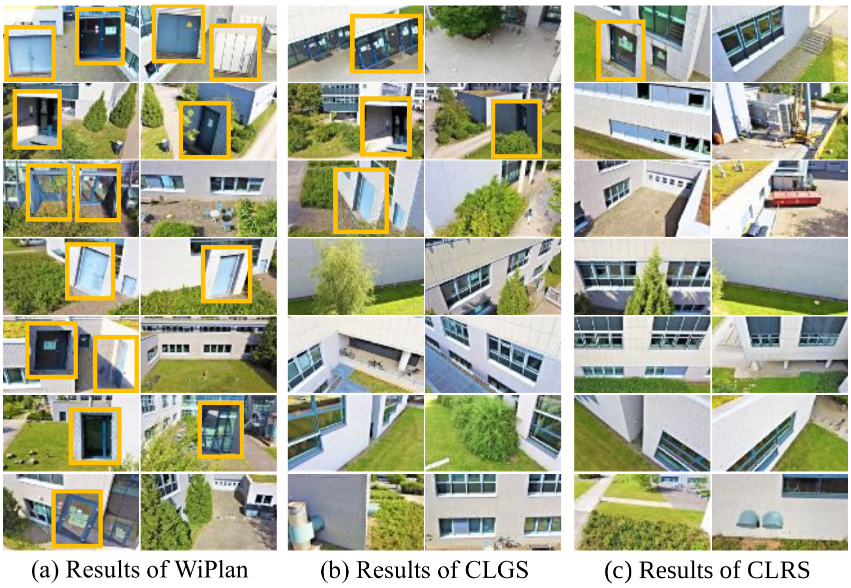


Figure 6.19.: **Captured Images of Three Algorithms.**

UAVs fly in the $7.5m$ sky, and the angle of pitch is set to $20°$.

Experimental Results. The locations of 22 objects, the UAV number 7 and each UAV's camera number 2, and 3 types of cameras' γ, R input three algorithms, WiPlan, CLGS, and CLRS. The output arrangements of three algorithms are depicted in Fig. 6.18(b) with different labels. We use the number of monitored gates as a metric to compare the performance of three algorithms, and thus, the ground truth is 25. The captured images of each UAV following the outputted arrangements by three algorithms are shown in Fig. 6.19. In each image, we enlarge the captured gates bounded in rectangle boxes for clearness. If there is no gate in the captured image, we show the original photos. The experimental results shows that WiPlan monitors 15 gates while CLGS and CLRS only monitor 4 and 1. WiPlan achieves 65% of ground truth and outperforms CLGS and CLRS 4.25× and 15×.

6.6. Chapter summary

WiPlan solves the problem of the waypoint plan of adjustable multi-camera UAVs for monitoring tasks. More specifically, WiPlan focus on the practical scenarios in which critical inspection points are fixed, (*e.g.*, joints and welding points in the inspection task of infrastructure). In such cases, a time-consuming but careful waypoint selection schema can be reused repeatedly. Before UAV flies to collect data of these critical inspection points (inputs), WiPlan generates the waypoints (outputs) for the

UAV hovering and monitoring.This is the first work for waypoint planning problem with multi-camera UAV. The key insight is that the multi-camera model expands huge solution space in waypoint planning solution space compared to single-camera model. Tackle this challenge, WiPlan proposes a polynomial time algorithm with constant approximation ratio algorithm. The results show that multi-camera UAVs with careful algorithm design achieving much better performance than single-camera model; but, the efficiency (time complexity) still exists much room.

Chapter 7

Discussion & Future Prospects

In this chapter, we discuss two issues one may occurred in your mind and propose future prospects of waypoint planning problem for UAV monitoring system.

Contents

7.1. Discussion

Planning Minimum Waypoints to Achieve a Required Coverage Utility Consider a problem which is slightly different from PANDA, VISIT, and WiPlan, that is,planning minimum waypoints to achieve a required coverage utility. The solution is almost the same as them, except Step 3 Approximate Algorithm in Chap. 3. Instead, we greedily select arrangements one by one until the required coverage utility is achieved, then output selected arrangements. According to the classical results in [162], the adapted algorithm achieves $\frac{1}{\ln n}$ approximation ratio, where n is the number of candidate arrangements. With similar analysis, we can prove that the overall solution for this variant problem can also achieve $\frac{1}{\ln n}$ approximation ratio.

Applying PANDA to Real-world Scenarios In this section, we consider the case where the 3D directional coverage in actual scenarios. In actual scenarios, there will be many problems occurred, such as obstacles, electronic interference, and sky above the sidewalk, etc. However, PANDA is a basic problem and our algorithm of PANDA can also address these issues occurred in actual scenarios. In briefly, these actual problems introduce some constraints to PANDA that the obstacles, the regions of electronic interference, and the sky above the sidewalks can't place UAVs or obstruct the field of view. Mathematically, they decrease the solution space of PANDA, but union with these constraints the algorithm of PANDA can also work because we just need to search the remaining solution space.

7.2. Future Prospects

7.2.1. Heterogeneous UAVs and Objects

In this dissertation, all we considered is the homogeneous UAVs and homogeneous objects, however, in most scenarios, objects and UAVs are heterogeneous. Here, we take PANDA as an example.

Heterogeneous UAVs. Heterogeneous camera models are consisted of different parameters, *i.e.*, α, β, and Δ. The PANDA problem of heterogeneous camera version is, given a set of UAVs containing Φ types of cameras, the ϕ-th $(1 \leq \phi \leq \Phi)$ type of camera model with parameter α^{ϕ}, β^{ϕ}, and Δ^{ϕ}, and the number of UAVs of this type is N_{ϕ}. The objective function is the same as the one of PANDA, which is placing these heterogeneous UAVs in 3D space to maximize the overall directional coverage utility for all M objects.

For this version, we may combine the focal length discretization in WiPlan together with the general algorithm design schema. In Step 1, for each type of camera, we divide the whole 3D space into multiple cells with each Δ^{ϕ} and the homogenous efficient angle θ. In Step 2, we extract DCSs of each type of camera model, *i.e.*, α^{ϕ} and β^{ϕ}, and obtain the set of DCSs, *i.e.*, Γ_{ϕ}, of each type of camera; and obtain Φ different sets of DCSs because of the heterogeneous parameters of camera models. In Step 3, to PANDA problem, we reformulate it into a problem of maximizing a monotone submodular function subject to a uniform matroid constraint; but to this heterogeneous version, we reformulate it into a maximizing monotone submodular function subject to a Partition Matroid constraint (see the following definitiom). The, we can design a greedy algorithm to maximize the monotone submodular function with performance bound [142].

Definition 7.2.1. *[142] Given $\mathcal{S} = \bigcup_{i=1}^{k} \mathcal{S}'_i$ is the disjoint union of k sets, $l_1, l_2, \cdots, l_k$ are positive integers, a Partition Matroid $\mathcal{M} = (\mathcal{S}, \mathcal{I})$ is a matroid where*

$\mathcal{I} = \{X \subset \mathcal{S} : |X \cap \mathcal{S}'_i| \leq l_i \text{ for } i \in [k]\}$.

Heterogeneous Objects. Heterogeneous objects problem is much more easier than heterogeneous UAVs version PANDA, because the only heterogeneous parameter of objects is θ. To this problem, we can also use the same method of PANDA. The only difference is the space discretization step. Formally, given a set of objects containing Ψ types, the ψ-th ($1 \leq \psi \leq \Psi$) type of object model with parameter θ^{ψ} and Δ, and the number of objects of this type is M_{ψ}. Then, we divide the whole 3D space with all objects but corresponding heterogeneous θ not the homogeneous one in PANDA. The following step is the same as the method of PANDA. Thus, the time complexity and the approximation ratio are both the same as PANDA, which is $O(NM^9)$ and $1-1/e$.

7.2.2. Considerable Volumes of Objects

In three problems we addressed in this dissertation, we ignore both the volumes of UAVs and objects. However, in real system, we have to consider the real volumes of objects in the coverage region of UAVs. Fortunately, there are some works consider the obstruction issue in coverage problem. We take VISIT as an example and address it via original method plus some extra preprocessing. The preprocessing for our preprocessing which includes three steps. The key idea of the preprocessing is the same as calculus. First, it discretizes the surface of each objects into λ_j number of intervals. Second, to each interval, the preprocessing method computes its norm vector and combine this interval with its norm vector to be an object. After the above two steps, we obtain $\sum_{ß=1}^{M} \lambda_j$ number of intervals but some of them are obstructed with each other. Third, preprocessing utilizes the method in [81] to analyze the geometry relationship among these objects and drop the obstructed objects. After the preprocessing, the rest of the objects are not obstructed, thus we can treat them as point like VISIT. Now, the problem can be reformulated to VISIT and solved with the method of VISIT.

7.2.3. Time Complexity Reduction

As we introduce in Chap. 2, our algorithm design schema is suitable for those static (offline) monitoring tasks, such as infrastructure routine check and power generator unit routine check, whose requirements of monitoring are high accuracy rather than real-time. To these tasks, one time-consuming but carefully track point selection schema can be reused repeatedly.

However, the time complexity of our algorithm is truly high for those real-time (online) monitoring tasks. Meanwhile, if the visual recognition is executing on UAVs, the power consumption is a considerable problem. To address these two issues, we need to design an energy efficient algorithm with much lesser time complexity. Fortunately, there are some techniques to reduce the time complexity, such as the correlation graph [61, 163]. In future system works, we will highlight the tradeoff between monitoring quality and algorithm cost time and reduce the time complexity with some techniques.

Part II.

Addressing Communication Challenge in UAV Monitoring System: Study for Scaling Mobile Clients

Chapter 8

Problem Statement

High mobility is the most prominent characteristic in UAV system. It is important to explore how mobility impacts the communication and further Quality of Experience (QoE) in our target monitoring system. Before deploy the video streaming system on real UAV monitoring system, in Part II (Chapter 8 and Chapter 9), we study the video streaming on Dynamic Adaptive Streaming over HTTP (DASH) framework in mobile environment.

Contents

8.1. Introduction

As the prevalence of mobile devices (e.g., smartphones, tablets, and laptops) and the emerging high-rate multimedia applications including video streaming for mobile gaming [164] and social networks [165], such as Internet live broadcast and video dating, mobile video traffic increases significantly in recent years. In 2017, video traffic accounted for 75% of all Internet traffic. It is estimated to rise to 82% by 2022 [166]. Meanwhile, mobile video traffic accounted for 59% of all mobile data traffic in 2017, which is estimated to rise to four-fifths (79%) of the world's mobile data by 2022 [166]. The rapid growth of mobile video traffic and user demand leads

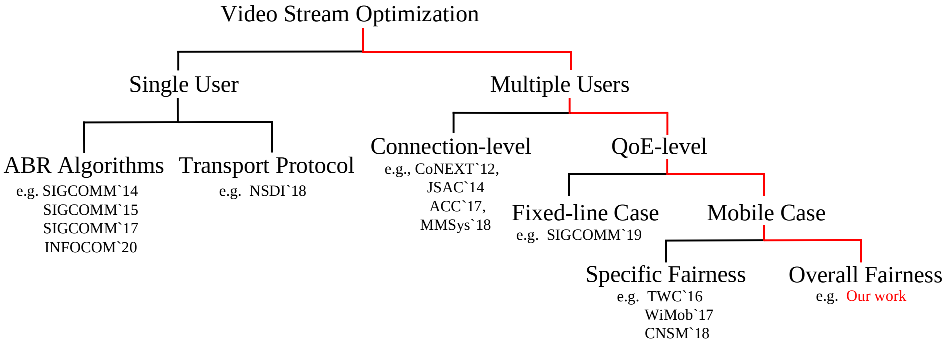


Figure 8.1.: **Classification of related video streaming optimization works.**

to a higher probability for multiple video streaming clients sharing a bottleneck link. The experience of multiple users can be affected greatly by the network conditions and users' high mobility, such as high fluctuation in the available bandwidth and high moving speed of clients when multiple clients simultaneously compete for the shared bottleneck link, in mobile video streaming applications [167].

This problem becomes more severe in 5G networks, where clients are subject to high mobility, the base station (BS) is typically of a smaller size and the directional antenna is often employed to prevent the severe propagation loss and ensure a good transmission rate [21]. Because of the directional antenna, video content can only be transmitted when the antennas of both the base station and the mobile user are directed towards each other. In this case, handoffs[12] occur more frequently due to the small cell region (e.g., picocells with range under 100 meters [168]) and clients' high mobility characteristics (e.g., in highway and rail environments). Frequent handoffs may cause rebuffering, which will diminish the QoE significantly. Besides, some client may obtain lower QoE, which means there is QoE unfairness among users. However, QoE and QoE fairness are two key metrics to evaluate the performance of video traffic for clients. QoE is an important aspect in keeping a single customer's satisfaction in isolation while QoE fairness is especially of importance for video content providers where operators want to keep all users sufficiently satisfied (i.e., high QoE) in a fair manner. Therefore, optimization models are needed to achieve the maximum (or at least ensure an acceptable) QoE, while ensuring fairness among users for mobile video streaming applications in terms of resource (e.g., bandwidth) allocation in shared bandwidth links.

8.2. Related Work

We broadly classify the related video streaming optimization literature into the following two categories. A pictorial representation of this classification is shown in the Fig. 8.1.

8.2.1. Single User

Since the videos are encoded into multiple different bitrates and stored on the servers as separate files, selecting the video chunks with the optimal bitrate as per the network throughput becomes the primary goal of the Adaptive Bitrate selection (ABR) algorithm [169] to improve the user's QoE during video streaming. In this regard. Huang et al. [170] proposed a method to improve the QoE but only consider the clients' video player buffer to choose the next video chunk and fail to account for other parameters (e.g., bottleneck bandwidth) that also affect the QoE. Yin et al [171] leveraged the model predictive control theory to predict key environment variables and solved an exact optimization problem to select the next chunks while to overcome the environment prediction accuracy, Mao et al. [172] presented a system Pensieve, which trains a neural network model to select future video chunks based on the current environment state. Dong et al. [173] proposed an online-learning congestion control algorithm called PCC Vivace to improve the video streaming performance. [16] attempted to optimize QoE by selecting the optimal initial video segment using deep reinforcement learning according to the network conditions (*e.g.*, signal strength). To improve QoE, [17] proposed to integrate the video super-resolution algorithm into the adaptive video streaming strategy by using the deep reinforcement learning approach.

8.2.2. Multiple Users

A simple fairness definition would be to provide connection-level fairness, which ensures an equal allocation of network resources among competing flows [174, 175]. In this regard, Jiang et al. [176] proposed an algorithm to improve the fairness, while methods in [177, 178] try to ensure the QoE fairness for competing for flows by exploiting TCP-based bandwidth sharing. [179] proposed a method based on game theory to avoid selfish behavior, which achieves stable viewer QoE during video streaming. Vikram et al. [180] built a system named Minerva, which optimizes max-

[12] A handoff occurs when the mobile device moves between two BSes or cells.

min QoE fairness by taking into consideration users' priorities in wired networks. QoE optimization is achieved by leveraging the load balance in base stations in [20], where Jain's fairness is used to achieve the bitrate-level fairness for the video streaming traffic. [21] predicted the next base station for mobile clients by using their mobility information and pre-store video in the next base station's cache to achieve the video quality consistency. [22] considered the video content, playing buffer, and channel status to optimize the QoE and achieve buffer-level fairness for HTTP adaptive streaming applications. Inspired by the congestion control of transmission control protocol, [23] considered the buffer filling rate, network capacity, congestion avoidance, and detection to optimize the QoE and QoE fairness.

8.2.3. Limitation of State-of-the-art

However, single user-based work [16, 17, 170, 172, 173] did not consider the QoE fairness for the optimization. Besides, they are not suitable for multi-user scenarios with constrained bandwidth. For the multiple-user-based work, [178] and [180] are designed to work optimally for only wired networks. [176–178,180] are not suitable for mobile environments. [21] did not consider QoE fairness of clients. Other works [20], [22], and [23] only considered the specific fairness for clients, such as buffer-level or bitrate-level. Besides, all of them did not incorporate clients' mobility profiles into the QoE fairness optimization for video streaming applications. However, clients have various mobile patterns in wireless networks, which can assist providers in improving clients' QoE and QoE fairness in a high-mobility environment.

Chapter 9

Video Streaming in Mobile Environment

Contents

9.1. Background and Motivation

9.1.1. Background

HTTP/3 (HTTP/2 over QUIC). HTTP adaptive streaming (HAS) has appeared in the form of notable standard to deliver video contents over the network in the past few years [181]. HTTP/3 (HTTP/2 over QUIC) resolves the major issue of Head of Line (HoL) blocking along with multiple other improvements compared to HTTP/2. The video streaming approaches over HTTP/3 are promising, since with Quick UDP Internet Connections (QUIC), HTTP/3 has some good features like HoL Elimination, Forward Error Correction, Connection Identifier, and Server Push, benefiting today's network communications significantly.

Quality of Experience (QoE). During video transmission, a video V is divided into a stream of smaller segments or chunks, $V = \{1, 2, ..., K\}$ where each chunk contains S seconds of the original video. Each chunk is further encoded at different bitrates for streaming by the publisher. During streaming, the video player selects the optimal bitrate for improving the perceived video quality of the client. Higher bitrates indicate higher video qualities. Hence, a common goal of video players is to request higher quality chunks whenever the network conditions are favorable. However, the QoE of video during streaming is also affected by additional factors, especially, rebuffering and smoothness. During video streaming, rebuffering is said to occur when the video player's buffer runs out before the next chunk is downloaded, i.e., when the download time of a chunk is greater than the video player buffer's playout time. Smoothness on the other hand refers to the perceived variations between video segments during playtime. Hence, when requesting a video segment at higher/lower bitrates, the video players requested quality should not vary significantly from the previous one.

For a video V of length L, let c_k represents the k-th chunk at a bitrate r where $r \in \{r_1, r_2, ...r_m\}$, and R_k denote the time spent for rebuffering. Then according to the video streaming literature [171, 180], the QoE observed by a client for the k-th chunk is calculated as follows:

$$QoE(c_k) = q(c_k) - \beta R_k - \gamma ||q(c_k) - q(c_{k-1})||, \tag{9.1}$$

where $q(c_k)$ refers to the improvement in quality with the requested bitrate for the chunk c_k, $R_k = \frac{c_k}{r} - b$ refers to the rebuffering time calculating by the difference between downloading time $\frac{c_k}{r}$ and remaining playing time b. $q(c_k) - q(c_{k-1})$ represents the smoothness between the chunk c_k and c_{k-1}. The parameter β penalises the gain in QoE with $q(c_k)$ for rebuffering while γ penalises the QoE gain with the loss of smoothness between c_k and c_{k-1}. Therefore, as per Eq. (9.1), in order to maintain a good QoE, a video player must ensure higher bitrates, low rebuffering and higher smoothness during video streaming.

QoE Fairness. Let B denote the bottleneck bandwidth and N be the client's number. At a time period T, each client i watches a video consisting of M chunks. The total QoE of i-th client for viewing this video is denoted as QoE $_i$. Then, the max-min QoE fairness, [13] a standard QoE fairness metric, is to *maximize* $min_{i \in [N]} \frac{QoE_i}{M}$, where $[N]$ is the set of positive integers $\leq N$. Max-min QoE fairness reflects the QoE improvement of the worst performing clients, which helps service providers to offer a fairer service for clients and encourages their engagements [180]. In order to achieve that in a high-mobility environment, the mobility profile of each client i should take into account the resource (e.g., bandwidth) allocation.

9.1.2. Motivation

For mobile video streaming applications, current models, like [17,18,20,21,23], with the connection-level fairness, i.e., occupying equal shared bandwidth of competing flows, may not ensure the QoE fairness for all clients, especially for those content providers with a larger number of users. In fact, in order to encourage more users to participate, video service providers are more inclined to improve the viewing quality of users with lower bitrates, rather than improving the viewing quality of users with higher bitrates [180]. Netflix, one of the largest video content providers, already considered this problem and adopts a series of techniques [182, 183] (e.g., three parallel TCP connections) to allocate a larger bandwidth for Netflix videos instead of considering connection-level fairness, reducing the rebuffering probability for low-buffer clients at video startup. Nevertheless, the fair clients' view quality among competitive network traffic is not incorporated, especially in a mobile network environment. Specifically, from the perspective of the service provider, there are two major drawbacks.

Clients' QoE can be affected by their mobility. Clearly, users may have different bandwidth allocation requirements in different scenarios [180,184]. In mobile wireless networks, the mobility of users significantly affects network performance including the QoE and QoE fairness. From the bottom to up layer, it affects physical SNR (Signal Noise Ratio) strength [185], the access time of user [186–188], the convergence speed of routing [189], and QoE [190], etc. For instance, compared to

[13] VSiM is flexible and different QoE fairness metrics can be used to evaluate VSiM. We use the max-min QoE fairness as an example to demonstrate the performance of VSiM.

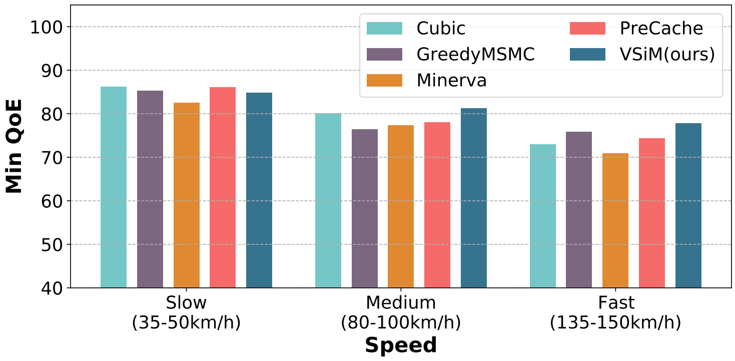


Figure 9.1.: **High-speed clients may experience lower QoE.**

low-speed clients, high-speed clients may require more allocated bandwidth within the same time period to accomplish the same viewing quality, due to the frequent handoffs or possible connection loss between BSes. Fig. 9.1 shows the view quality of clients over various speeds at time T with uniform linear motion. For simulation settings details, please refer to Sec. 9.4. It is obvious to see that a client with high speed $v \in [135km/h, 150km/h]$ is more likely subject to low QoE. Specifically, compared to clients with $v \in [35km/h, 50km/h]$, the minimum QoE achieved by the clients with speeds $v \in [135km/h, 150km/h]$ is 77.64% (about 10 points with QoE normalization, see § 9.3.1) lower on average. Furthermore, we observe that VSiM outperforms other state-of-the-art approaches for high-speed clients by sacrificing slightly the benefit of some low-speed clients to improve the clients' with low QoE.

Mobile clients have different buffer-sensitive levels. The existing approaches with the equal shared bandwidth between connections did not consider the state of the buffer-sensitive mobile video clients, like the playback buffer size, hence they are blind to the guidance information in the application-level, such as increasing the playback buffer size for buffer-sensitive clients. For example, a mobile video client with a short staytime in a BS and will experience a handoff time or a connection loss area to go next BSes or networks. This may result in a higher chance for this client to suffer the rebuffering, which reduces its QoE in a high mobile scenario. For such kind of clients, increasing the playback buffer size is more critical to improve their QoE compared to requesting the high-quality chunks. Besides, the whole QoE fairness can also be improved because of the improvement of minimum QoE. Therefore, *dynamic* and *adaptive* buffer update strategy is a good choice to help the buffer-sensitive clients quickly replenish their buffer size.

Server push in QUIC is a promising strategy to accomplish this requirement. However, traditional server push approaches did not consider the mobile characteristics into their design. Some server push strategies, like [28, 191] transmits the same

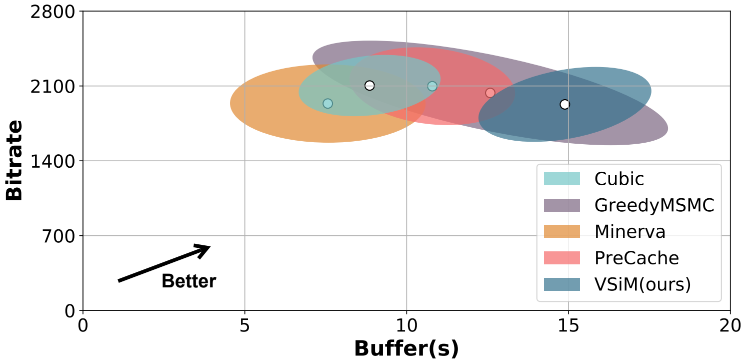


Figure 9.2.: **The trade-off space between bitrate and buffer with the bottleneck bandwidth in a highly mobile environment.**

quality chunk with that of the previous chunk, resulting in high downloading time and not suitable for the mobile video clients. Besides, [191] may drop all pushed chunks and wastes bandwidth resource. A novel server push strategy in VSiM is proposed to update the buffer size adaptively without affecting the existing bandwidth allocation strategy and other clients' viewing experience. Fig. 9.2 illustrates the trade-off space between the bitrate and buffer when facing bottleneck bandwidth in mobile environments. The larger the ellipse is, the greater the variance, resulting in the worse performance of the model. We can see that VSiM is able to increase the buffer size of clients significantly while slightly affecting the average bitrate.

9.2. VSiM System

This section introduces the design goal and overview of our system VSiM, then discuss its three key techniques namely bandwidth allocation, server push, and parameter update.

9.2.1. Overview of VSiM

VSiM is an end-to-end solution for improving the QoE and QoE fairness of video streaming in a highly mobile environment. The main design goals that we wish to achieve in VSiM are: 1) efficiently incorporate various factors in mobile environments that can potentially impact the QoE fairness of clients during video streaming, 2) easy to deploy and configure in the real world, 3) maximize the QoE fairness while ensuring the total QoE, 4) adapt to the uncertainties in heterogeneous mobile wireless networks.

System architecture. At the client end, we exploit the ABR controller ❶ to collect the bitrate of the requested chunk and the buffer state of Dash player. We further collect the information about each client's mobility profile from the sensors ❷ in their mobile devices. Since many smart devices collect such information using GPS and IMU [27], we assume that our system can also have access to such information on the clients' devices. The collected state information ❸ is then grouped, encrypted, and sent along with `HTTP Request` ❹ for downloading the chunk c_k at bitrate b_i to server.

At the server end, for each arriving `Request` from clients, the server decrypts the state information and trajectory prediction ❺ calculates clients' trajectories using mobility profile information and topology information of base stations ❻. Once the trajectory is known, the server identifies the BSes the client connects with and the associated parameters such as the handover latency, staytime, and possible connection-less zones that will impact the QoE of the mobile client. We assume that the server is aware of the information of its needed BSes. Utilizing these values and the information from clients' DASH players (e.g., buffer level and bitrate level), utility computation module ❼ applies utility function to calculate the optimal weight w_i for each client and transfer them to the bandwidth allocation module ❽ (see § 9.2.2).

Since the server has a global view of all clients, it efficiently calculates and allocates the available bandwidth among the participating clients by considering their mobility profiles and QoE-related information. We allocate the bandwidth by the weight w_i for each client using the Cubic congestion control approach [174, 180] such that the link capacity is utilized completely and there is an improvement in the QoE fairness of all participated clients. The weight w_i is updated over a period t, when the topology of BSes has a significant change. Besides, the optimal value of t and utility function parameters to quantify each factor's (e.g., bitrate, rebuffer, and smoothness) contribution for bandwidth allocation, like β and λ, are produced by the parameter update strategy ❾ (see § 9.2.4).

Meanwhile, based on these values and information, server also identifies the clients who are more likely to experience increased rebuffering due to a short staytime in a BS, handover latency, or connection-less zones. In order to improve the QoE of such clients, our system tries to fill their player buffer to increase playtime and thereby reduce the effect of rebuffering. Server push module ❿ identifies these potential clients, prioritizes such clients, and pushes extra chunks to them. The original chunks and the pushing chunks will be stored in two buffers (Please refer to § 9.2.3 for details).

Once the optimal bandwidth is allocated for each client and the server push gives

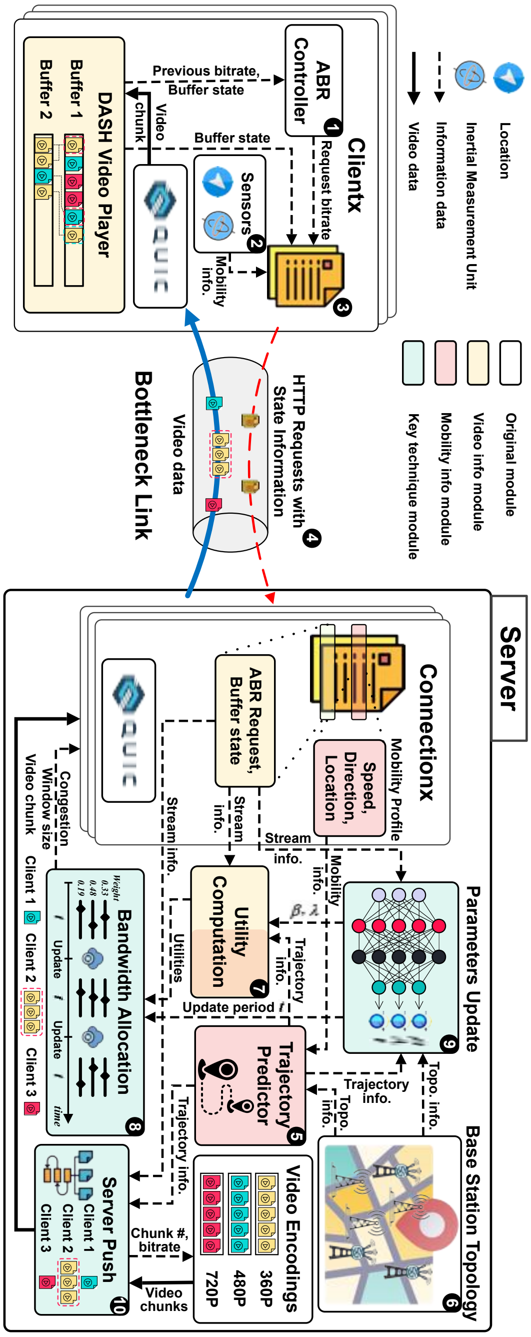


Figure 9.3.: **VSiM improves the QoE and QoE fairness in a mobile environment by three key techniques. Bandwidth Allocation considers clients' mobile profile and QoE-related information, Server Push algorithm avoids rebuffering, and Parameter Update mechanism adapts system parameter.**

an optimal push strategy, the prepared chunks are transmitted to the client over the allocated bandwidth using the QUIC transport protocol. However, the push bandwidth is allocated only when necessary, and if the QoE fairness of all clients that share the same bottleneck bandwidth can be guaranteed.

Discussion. In VSiM, we place the mobility predictor on the server side because the server with high storage and computation ability is easy to get BSes' information to predict clients' trajectories and ensures the global QoE fairness for all clients. However, when clients and BSes share their mobility and topology information respectively to the server, privacy leakage might happen. Privacy algorithms like differential privacy can protect users' information while ensuring models' performance, but it is beyond the scope of VSiM. We will consider this in our future work.

9.2.2. Bandwidth Allocation

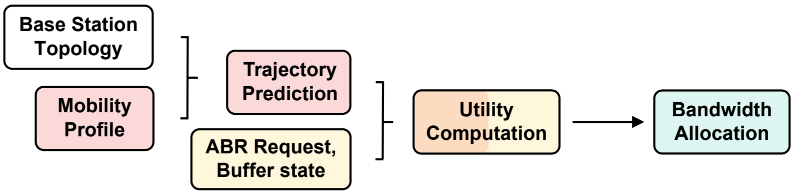


Figure 9.4.: **Bandwidth Allocation strategy considering clients' mobile profile and QoE-related information.**

VSiM ensures a fair QoE experience for all participating clients by considering users' mobility profile, buffer-level, bitrate, and smoothness during video streaming, which is described in Fig. 9.4. Let c_k be the current video chunk requested at a bitrate r_m and b_k be the remaining playtime in the video player's buffer, where k is the kth video chunk and m is the mth bitrate level. During video streaming, for every video segment c_k, the following state information S_{c_k} given in Eq. (9.2) is collected at the client module in the system consisting of QoE-related information $\{c_k, r_m, b_k\}$ and mobile profile $\{v, a, \overrightarrow{d}, l_{x,y}\}$, where c_k denotes the requested chunk, r_m is the bitrate of c_k, b_k is the buffer state, $I_{x,y}$ denotes the location. v, α, and $\overrightarrow{d}$ represents speed, acceleration, and direction.

$$S_{c_k} = \left[c_k, r_m, b_k, v, a, \overrightarrow{d}, l_{x,y}\right]. \tag{9.2}$$

Utility Computation. In this section, we build the mathematical model between the original QoE fairness optimization problem of VSiM and bandwidth allocation. Let $U(r)$ be the utility function for bandwidth allocation. Based on the QoE definition in Eq. (9.1), $U(r)$ is built as a function of each client's download rate r, which is optimized by leveraging the clients' mobility profile information and QoE-related information (e.g., bitrate level and buffer level). $U(r)$ is defined as Eq. (9.3).

$$U(r) = q(c_k) - \beta R'_k - \lambda |q(c_k) - q(c_{k-1})|, \tag{9.3}$$

where $q(c_k)$ represents the requested bitrate for chunk c_k, $\beta R'_t$ represents the rebuffering penalty, and $\lambda|q(c_k) - q(c_{k-1})|$ represents the penalty for variance in smoothness. The bandwidth allocation is dynamically changing in real-time.

Intuitively, rebuffering time aggrades along with handover time and chunk download time, degrades along with playtime remaining and staytime. Formally, rebuffering time is:

$$R^{'}_k = \frac{c_k}{r} - b - t_s + t_h, \tag{9.3a}$$

where $\frac{c_k}{r}$ denotes the time to download the remaining chunk of c_k at a download rate of r, b denotes the playtime remaining in the clients' video player buffer. The parameter $t_s = \frac{d_s}{v}$ denotes the remaining time within the connection zone of BSes, where v is the client's moving speed and d_s is the remaining distance within the connection zone of BSes. t_h denotes the handover latency between the current and the next BS. Therefore, We mapped users' mobility characteristics to the rebuffer parameter R'_k. Both the staytime t_s and handover time t_h are decided by users' mobility profile, like the speed, direction, and acceleration. The higher t_s is, the higher QoE while t_h is inversely proportional to QoE.

The handover time t_h is defined as:

$$t_h = \begin{cases} \frac{d_h}{v} + \tau, & \text{no overlap between BSes,} \\ \tau, & \text{overlap between BSes,} \end{cases} \tag{9.3b}$$

where d_h is the distance occurring connection loss between BSes. τ is the time when clients switch between BSes. For example, a client travels from its current BS to next BS, if there exists the connection loss between these two BSes, then $t_h = \frac{d_h}{v} + \tau$. Otherwise, $t_h = \tau$. Please note that since the trajectory of each client is varying over time by changing the speed, direction, and acceleration, the calculation results of both t_h and t_s are also varying over time.

Bandwidth allocation. The information required for Bandwidth allocation is described in Fig. 9.4. Besides, given the above definition for the utility computation, the bandwidth weight for client i is calculated as $w_i = \frac{r_i}{\widetilde{U}(r_i)}$ where $\widetilde{U}(r_i) = \frac{U(r_i)}{U(B)}$ and the allocated bandwidth will be $r_i = \frac{w_i}{\sum_{i=1}^{n} w_i} B$, where i, n, and B represents the i-th client, clients number, and server's total bandwidth. We put the convergence proof of bandwidth allocation in the Appendix (§ 9.5). Please note that the time complexity of VSiM is very small, *i.e.*, $O(cn)$, where n is the number of clients sharing the bottleneck link and $c$is the iteration times required to converge to the optimal bandwidth allocation that maximizes the QoE fairness. The space complexity is also very small since clients' state information is refreshed on the server for each bandwidth allocation.

9.2.3. Server Push

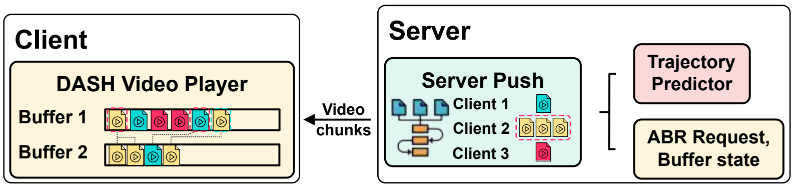


Figure 9.5.: **Server push module utilizes clients' mobility and video stream info to determine if and how to push extra chunks.**

Server push module is employed to decrease the frequency of rebuffering for buffer-sensitive mobile video clients. It significantly increases these clients' playback buffer by pushing multiple lower-bitrate chunks when they drop into an emergency situation. Meanwhile, server push should be compatible to arbitrary ABR algorithm and should not offset the benefit from bandwidth allocation. Therefore, we carefully design a novel server push algorithm called Slow Degrade Fast Recovery (SDFR). The core thinking includes: 1. *Multiple chunks encapsulations.* Server encapsulates multiple lower-bitrate chunks back to the client according to client's state. 2. *Slow degrade.* During server push, the bitrate of pushing chunk degrades level by level to control the smoothness.

Workflow. Fig. 9.5 illustrates the overview of server push. At the clients' end, Dash player maintains two buffers logically, in which Buffer 1 stores the original request video chunks and Buffer 2 stores the pushing video chunks, records the corresponding relation of each chunk in Buffer 2, i.e., the encapsulation relation, and delivers fake bitrate information to ABR algorithms. At the sever end, after receiving `HTTP Request` message from the client, the server first estimates the client's state with the buffer state information and the trajectory predicted by mobility profile from the message. Then, it requests and encapsulates multiple lower-bitrate chunks from video encoding according to the SDFR algorithm and the bitrate request from `HTTP Request` message. Fig. 9.5 depicts a toy example: Client 2 requests one 720P bitrate chunk while server push encapsulates and responds three 360P chunks; Dash player stores three 360P chunks into buffers and sends a piece of fake information "received one 720P bitrate chunk as request" to ABR algorithm.

Fig. 9.6 shows the server push algorithm with a state machine model. SDFR sorts the bitrate level with an descending order resulting in a list $[c_{max}, c_2, \cdots, c_{min}]$. The bandwidth demand $B(\cdot)$ of these bitrate levels follows a total order: $B(c_i) < B(c_j)$ if $c_i < c_j$. c_p and c_R denote the pushing chunk bitrate and the client's bitrate request, respectively. Variable `k` quantifies client's emergency level. `isEMER(k)` denotes the event of the k-level emergency, whose condition is $\texttt{t}_\texttt{h}$`-b==k*`$\texttt{t}_\texttt{s}$ (The physical meaning is that the client needs to download *extra k* chunks per request in the following staytime such that its playback buffer have enough video to play during the handover time). Similarly, `overEMER(k)` and `underEMER(k)` denote the event of emergency exceeds k-level or lacks k-level, whose conditions are $\texttt{t}_\texttt{h}$`-b>k*`$\texttt{t}_\texttt{s}$ and $\texttt{t}_\texttt{h}$`-b<k*`$\texttt{t}_\texttt{s}$, respectively. SDFR also receives *messages* from other modules: bitrate request of ABR algorithm c_R in `HTTP Request` from client; `FULL` when client's buffer is full; `RESET` when trajectory prediction module detects t_h changes. Sever maintains a server push state machine for each client and the state transition happens when the server receives the corresponding client's `HTTP Request`. Below, we describe each state and transition.

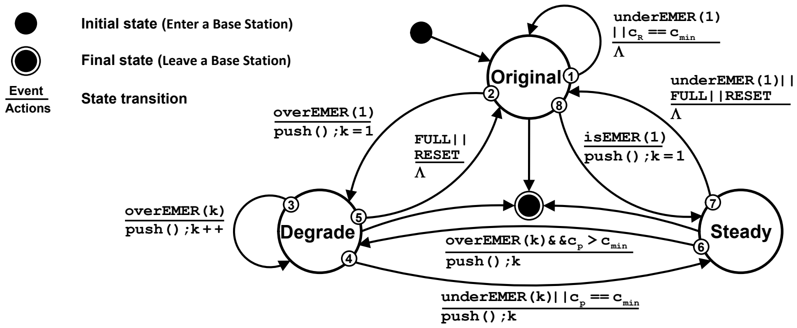


Figure 9.6.: **Illustration on the server push algorithm Slow Degrade Fast Recovery (SDFR) as a state machine.**

- **Initial.** Client enters a base station, then server push starts execution and changes to `Original`.
- **Original: responding chunk with `Request` bitrate.** Server push holds sending chunk with c_R bitrate. It stops at `isEMER(1)` (to `Steady`) or `overEMER(1)` (to `Degrade`). If $\texttt{t}_\texttt{h}$`==0`, it changes to `Final`.
- **Degrade: estimating the emergency level.** Degrading c_R directly to c_p may significantly decrease client's QoE if $c_R >> c_p$. To allow a smooth decrease, SDFR probe the emergency level by level, i.e., adding one additional chunk controlled by `k++`. A larger k allows more chunks delivery. During emergency,

server push runs the `push()` procedure to encapsulate k chunks with bitrate c_p into `HTTP Response` message, where c_p satisfies max c, $s.t.\ k \cdot B(c) \leq B(c_R)$. After emergency estimated (`underEMER(k)||`$\mathtt{c_p}$`==`$\mathtt{c_{min}}$), server push changes to `Steady`. If server push receives `FULL` message or `RESET` message, it changes to `Original`. If $\mathtt{t_h}$`==0`, it changes to `Final`.

- **Steady: multiple chunks pushing.** Client achieves high buffer fill rate by server push encapsulating and sending multiple c_p bitrate chunks with $B(c_R)$ bandwidth. It changes to `Degrade` when `overEMER(k)&&` $\mathtt{c_p}$`>`$\mathtt{c_{min}}$. If `underEMER(1)` or server push receives `FULL` message or `RESET` message, it changes to `Original`. If $\mathtt{t_h}$`==0`, it changes to `Final`.
- **Final.** Client leaves a base station, server push ends.

Fig. 9.7 illustrates an example of a client's state transitions.

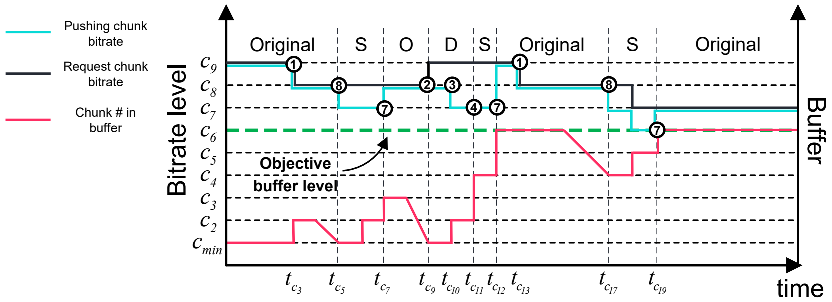


Figure 9.7.: **Illustration on an example of one client's server push process (the dotted lines split its states).**

9.2.4. Parameter Update

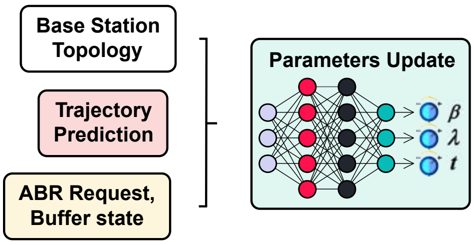


Figure 9.8.: **Parameter update module uses BS and clients info to update VSiM's parameters by neural network algorithms.**

The parameter update module is the key contribution for VSiM, which is employed to generate the optimal values of parameters required by the bandwidth allocation module (§ 9.2.2) based on the BS topology, the trajectory prediction information (*e.g.*, staytime and handover time), ABR request information (*e.g.*, bitrate), and buffer state (*e.g.*, remaining buffer size), given in Fig. 9.8. The updated parameters include weights β and λ of the utility function in Eq. (9.3) denoting the contribution of factors (*e.g.*, bitrate, rebuffer, and smoothness) to control a mobile video's rate shares. Besides, these parameters updated frequency t is also produced by the parameter update module.

Machine learning approaches (*e.g.*, Neural Networks for Multi-Output Regression [192]) can be utilized in this module. The machine learning model is trained offline with the dataset collected from various simulations over different BS topologies considering clients' QoE and QoE fairness. To generate the training dataset, we use the random search strategy [193] on the weights of utility function defined in Eq. (9.3), where $\beta \in \{0,1,2,\cdots,50\}$ and $\lambda \in \{0,0.1,0.2,\cdots,1\}$ represent the contribution for both rebuffer and smoothness, respectively. Besides, the weights update frequency $t \in \{0,1,2,\cdots,15\}$ is also considered into the random search strategy. When the utility function produces a good performance, we record the weight set $\{\beta,\lambda,t\}$. The QoE metric is mainly taken into account for parameter value selection. Then, for each set of good weights, we run 3000 simulation instances to collect the training data $\{c_r,r,b,q(c_k),q(c_{k-1}),t_s,t_h,\beta,\lambda,t\}$, where $\{\beta,\lambda,t\}$ can be treated as labels and the rest can be used as features. $\{c_r,r,b,q(c_k),q(c_{k-1})\}$ is the clients' QoE-related information and $\{t_s,t_h\}$ reflects mobility characteristics calculated based on clients' mobility profile and BSes Topology. Please notice that VSiM only needs to retrain the parameter update module when the BSes' topology significantly changes.

9.3. Implementation

We implement VSiM in both simulation and prototype. VSiM sits between the low-level functions (QUIC protocol) and the high-level applications (Dash video player on the client end and Video encoding on the server end). On the client end, VSiM modifies Dash player (Version 3.1.0) [194] by maintains two buffers logically [14]. On the sever end, all the modules of VSiM in Fig. 9.3 are implemented in Go language (Version 1.13.8) based on QUIC_GO (Version 0.17.1) [195]. Two penalty parameters for rebuffering and smoothness in Eq. (9.3) are $\beta = 20$ and $\gamma = 0.1$ for VSiM without triggering the parameter update strategy while these two values are set as $\beta \in [0,50]$ and $\lambda \in [0,1]$ for VSiM over varied topologies. The period time of param-

[14] Note that, this modification does not break the easy deployment characteristic of VSiM because this modification is implemented on the server and the buffer allocation happened at the start of clients building connection with the server.

eter update in the bandwidth allocation strategy (§ 9.2.2) is $t \in [1, 15]$. These values are set according to the simulation experience. Besides, the movement of clients incorporates three groups: slow movement ($v \in [35, 50] km/h$), medium movement ($v \in [80, 100] km/h$), fast movement ($v \in [135, 150] km/h$). The accelerations of car/-motorcycle and train are within $[-8, 2.5] m/s^2$ and $[-7, 0.5] m/s^2$, respectively.

9.3.1. System Settings, Metrics, Dataset, and Benchmarks

System settings. We use a server equipped with Intel Core i7-5930K CPU at 3.5GHz, 32GB (DDR4 3000MHz) of RAM, Killer E3000 2.5Gbps Ethernet network port. All clients use Google Chrome (Version 83) with QUIC (HTTP/3) support enabled. We use 10 devices including iPhone XR, Xiaomi Mi 8, Surface Go 2, 2 × IPad Air 4, IPad Mini 4, ThinkPad X1, and 2 × MacBook Pro, as the mobile clients for the prototype test (see § 9.4.3).

Evaluation Metrics. To better see the performance of VSiM, we regularize the scope of QoE within [0,100] [180] by Equation $a \times ln(x) - b$, where $a = 16.61$ and $b = 42.94$ for our employed datasets. For instance, about 5.8 points improvement with QoE normalization can accomplish a video quality jump from 720p to 1080p. It is defined by a large number of experimental statistics over the employed dataset. VSiM is evaluated by the following metrics: 1) QoE: It is employed to describe clients' viewing experience for the mobile streaming video, calculated in Eq. (9.1) (see § 9.1.1). 2) Max-min QoE fairness: It reflects the QoE improvement of clients with the minimum QoE (see § 9.1.1). 3) Minimum QoE: It represents the QoE of the client with the minimum value among all clients. 4) Average QoE: it is the average QoE over all participated clients. 5) Cumulative Distribution Function (CDF): It reflects the QoE fairness improvement.

Datasets. VSiM work well on videos with varied bitrates from different sources. Here, we evaluate VSiM by a standard test dataset [196], which reflects the real-world distribution. It includes 20 videos. The value of c_k in our dataset ranges from 45kbps to 3936kbps. It includes low, middle, and high levels of bitrates.

Benchmarks. We have four benchmarks for comparison to prove the performance of VSiM: (1) Cubic [197] with the average bandwidth; (2) Minerva [180], where QoE fairness is targeted for video streaming with a bottleneck link but without mobility consideration; (3) GreedyMSMC [20] achieving the QoE improvement by leveraging the load balance in base stations in a mobile environment; (4) PreCache [21] improves the QoE performance by pre-storing video in next base station's cache in mobile wireless networks. Cubic, GreedyMSMC, and PreCache are originally suitable for

Table 9.1.: **The 20 videos used in VSiM evaluation..**

Level	Bitrate(kbps)	Resolution	Level	Bitrate(kbps)	Resolution
V_1	45	320*240	V_{11}	782	1280*720
V_2	88	320*240	V_{12}	1008	1280*720
V_3	128	320*240	V_{13}	1207	1280*720
V_4	177	480*360	V_{14}	1473	1280*720
V_5	217	480*360	V_{15}	2087	1920*1080
V_6	255	480*360	V_{16}	2409	1920*1080
V_7	323	480*360	V_{17}	2944	1920*1080
V_8	378	480*360	V_{18}	3340	1920*1080
V_9	509	854*480	V_{19}	3613	1920*1080
V_{10}	577	854*480	V_{20}	3936	1920*1080

mobile scenes. For Minerva, we transplant its utility function and perceptual quality concept in our mobile experimental scenes. All algorithms are implemented in the same mobile environment for comparison.

9.3.2. Mobility Pattern

VSiM adapt to various mobility patterns, which are mapped to staytime and handover time, to fulfill the QoE fairness for high mobile clients. In this paper, we use three mobility models as examples to evaluate our mechanism. 1) freeway mobility model [189] and railway mobility model [198]: mobile users are restricted to their lanes on the freeway/railway and its velocity is temporally dependent on its previous velocity; 2) random waypoint model [199]: at every instant, a user randomly selects a destination and moves towards it with a velocity selected uniformly randomly from $[0, v_{max}]$, where v_{max} is the preset maximum velocity for each user. It is commonly used in simulations.

9.4. Evaluation

/ In this section, we first introduce the simulation and prototype scenario. Then, we verify the contribution of each key technique in VSiM and robustness of VSiM by simulations. Following that, the comparison of VSiM against state-of-the-art solutions with various metrics is illustrated over a prototype wireless network. For all results, we repeated the experiment for each bandwidth for 20 runs. Please note that if there is no specified topology, topology 1 is employed to evaluate models' performance. The parameter update strategy of VSiM is only triggered when the topology is changed.

Key takeaways. Key takeaways of our evaluations are:

- SDFR server push approach that fulfills the minimum QoE in VSiM is about 2.4 points (equal to clients' viewing experience jump from the bitrate level 2944kbps to 3340kbps in resolution 1080p) (Fig. 9.9).
- VSiM is robust for heterogeneous wireless networks including various topologies (Fig. 9.11), various video lengths and clients scale (Fig. 9.12), various ABR algorithms and mobility patterns (Fig. 9.13).
- VSiM improved more than 40% QoE fairness (equal to resolution improvement of clients' viewing quality from 720p to 1080p) compared to state-of-the-art while ensuring about 20% improvement on average for the averaged total QoE (Fig. 9.15 and 9.17).

9.4.1. Simulation and Prototype Scenarios

Simulations and prototype tests are implemented along the railway or highway direction. Two different scenarios to verify our system are as follows.

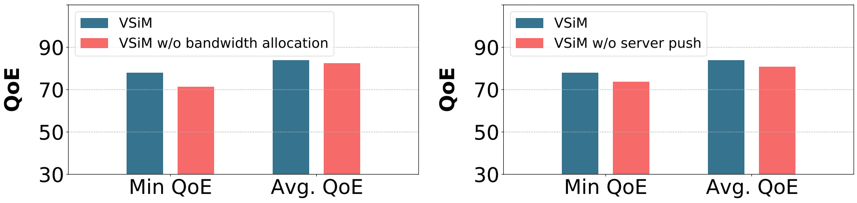
Topology 1: BSes with the connection loss area. We select the railway and highway from the train station in city A to the train station in city B (around 110km). Along the road, we give the assumption that this area is covered with 237 BSes consisting of 37 4G BSes and 200 5G BSes. The reason is that the 5G BSes are deployed every 500 meters and 4G BSes every 3km, depending on the communication range of BSes and area requirements (*e.g.*, high density of BSes in an urban area while low density in a rural area). The transmission ranges of 4G and 5G BS are 2km and 300m, respectively [200].

Topology 2: BSes without connection loss area. All BSes have a perfect overlap in a developed urban area. We select the railway and highway from city A to city B (around 24km). Along the road, we give the assumption that this area is covered with 48 5G BSes, which are deployed every 500 meters depending on the communication range of BSes. The transmission range of 5G BS is 300m

9.4.2. Baseline Comparison

We verify the contribution of each key technique in VSiM and robustness of VSiM by simulations, where 20 to 60 clients are employed in the topology 1 and 2.

Contribution of each key technique. We first measure the contribution of VSiM's two key techniques, i.e., Bandwidth Allocation and Server Push, with 60 clients and 150Mbps bandwidth in Topology 1. The results are described in Fig. 9.9. Then, the third technique, i.e., parameter update, is triggered in VSiM when the topology is changed, which is verified in Fig. 9.10 with 60 clients and 150Mbps bandwidth.



(a) (Bandwidth allocation

(b) Server push

Figure 9.9.: **Bandwidth Allocation and Server Push techniques contribute to VSiM QoE improvement on both QoE fairness and average QoE in Topology 1.**

Specifically, in Fig. 9.9(a), we observe that the minimum QoE in VSiM is about 33% (about 4.8 points with the QoE normalization) higher compared to that in VSiM without the bandwidth allocation technique while accomplishing a desirable average QoE. This is equivalent to the minimum viewing quality of clients in VSiM without the bandwidth allocation strategy is 240p while the minimum viewing quality of clients with that is 360p, thanks to the bandwidth allocation technique (§ 9.2.2), which leverages users' mobility profiles, requested bitrate, and playback buffer size to allocate the bandwidth fairly among clients. In Fig. 9.9(b), we notice that the minimum QoE and average QoE per client in VSiM is about 15% (about 2.4 points, equal to a viewing experience jump from the bitrate level 2944kbps to 3340kbps in resolution 1080p) and 13% higher than that in VSiM without the server push technique. Thanks to our proposed SDFR server push strategy (§ 9.2.3) given the current buffer level, staytime, and handover time. This mechanism greatly improves the QoE fairness of clients.

Fig. 9.10 gives the minimum QoE and average QoE comparison between VSiM and VSiM w/o paraUpdate over different topologies. We notice that both VSiM and VSiM w/o paraUpdate perform well in topology 1 while VSiM achieves a higher value in terms of both minimum QoE and average QoE compared with those of VSiM w/o paraUpdate in Topology 2. This is because we assign a set of selected optimal parameters (e.g., $\beta = 20$ and $\lambda = 0.1$) for both VSiM and VSiM w/o paraUpdate regarding Topology 1. However, when the topology changes, the parameter update strategy is triggered in VSiM to adapt to different topologies dynamically. As for

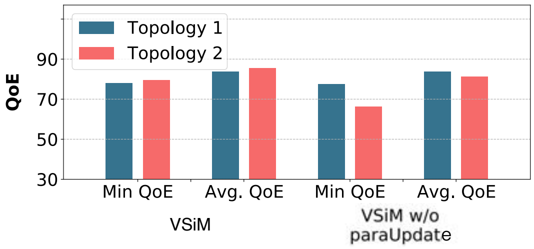
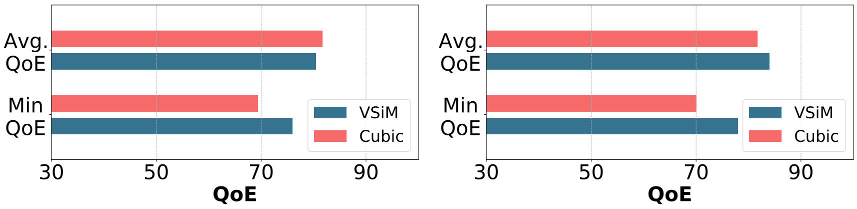


Figure 9.10.: **Parameter update technique brings contribution to VSiM's QoE improvement on both QoE fairness and average QoE over varied topologies.**

VSiM w/o paraUpdate, the previous selected parameters for Topology 1 may not be suitable for Topology 2.

Robustness of our system. VSiM is robust over various uncertainties, like different video lengths and BSes' topologies, different ABR algorithms, and different clients' mobility patterns and the number of clients.



(a) Topology 1.

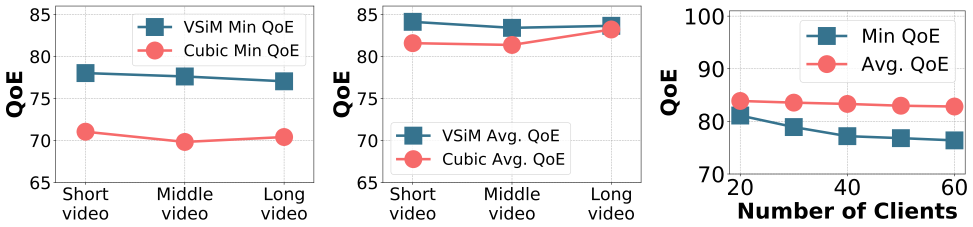
(b) Topology 2

Figure 9.11.: **VSiM can handle various topologies and maintain a sizeable gain on QoE fairness.**

- *Impact of various topologies.* In Fig. 9.11, the minimum QoE and average QoE of VSiM and Cubic over two different topologies (§ 9.4.1) over 60 clients and 150M bandwidth. It is obvious that compared to Cubic, the minimum QoE in both topologies A and B of VSiM achieves a significant improvement of the minimum QoE (about 6 points on average, equal to the clients' viewing experience with resolution jump from 720p to 1080p). Besides, The average QoE of VSiM is close to that of Cubic in these two topologies.

- *Impact of various video lengths.* Fig. 9.12(a) and (b) illustrates QoE of clients with different video length. we observe that VSiM accomplishes a stable minimum QoE and average QoE over various lengths of videos, which are much better compared with those of Cubic, especially for the minimum QoE.

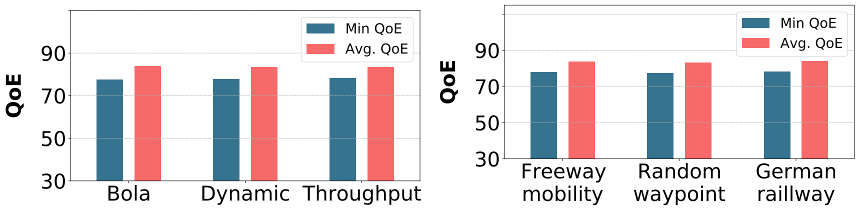
- *Impact of large-scale clients numbers.* In Fig. 9.12(c), we find that the perfor-



(a) Various video lengths (Min. QoE).
(b) Various video lengths (Avg. QoE).
(c) Various clients scale.

Figure 9.12.: **VSiM achieves high QoE under various video lengths; It also ensure stable, high QoE fairness under large-scale number of clients.**

mance (*e.g.*, minimal QoE or average QoE per client) of VSiM is almost constant under a various number of clients, showing the stability of VSiM against the variant number of clients. The slightly reducing trend of the minimum QoE and average QoE with the increasing number of clients in Fig. 9.12(c) is caused by the probability that the greater the number of users, the greater the probability that some clients will obtain lower QoE.



(a) Various ABR algorithms.
(b) Various mobility patterns.

Figure 9.13.: **VSiM maintains stable and high QoE under various mobility patterns of clients and various ABR algrithms.**

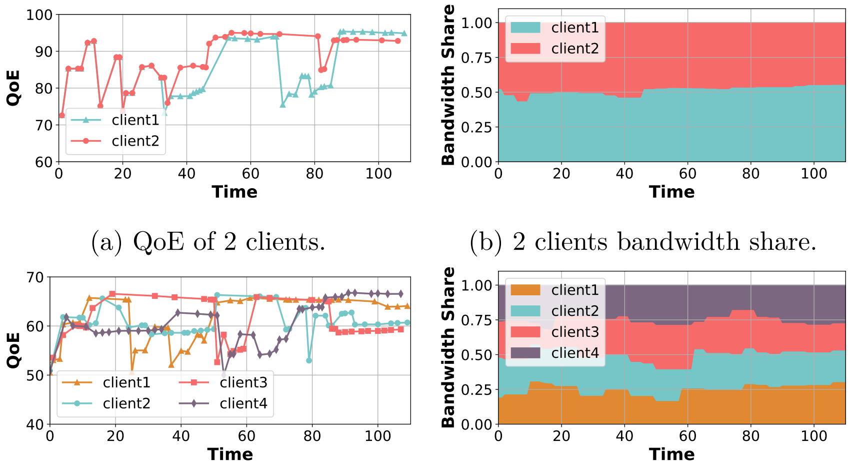
- *Impact of various ABR algorithms.* VSiM is transparent to ABR algorithms. In Fig. 9.13(a), we can clearly see that VSiM achieves good performance over different ABR algorithms regarding both the minimum QoE and average QoE. VSiM is transparent to ABR algorithms. The ABR algorithm should be abstracted away so that VSiM can work with any ABR algorithm. VSiM may have access to the ABR algorithm, but can only use it as a black box.

- *Impact of various mobility patterns.* In VSiM, we convert the users' mobility profiles to the staytime and handover time, which adapts VSiM to various mobility patterns, which is given in Fig. 9.13(b). Besides, a new parameter adjust model is proposed to ensure VSiM adapt to various BSes topology and number of nodes.

9.4.3. Prototype Test

We build a prototype test in a lab testbed to check VSiM's performance in real-world scenarios in the Topology 1 (§ 9.4.1) over 10 clients with bandwidths [10Mbps, 15Mbps, 25Mbps, 35Mbps] bandwidth. We run the experiment under a multi-user scenario who travel between two German railway stations with 110km distance and run VSiM over an actual wireless network link in mobile networks.

Sensitivity to network settings. The impact of network uncertainties, like bandwidth variance and latency variance, are tested in this section.



(a) QoE of 2 clients.

(b) 2 clients bandwidth share.

(c) QoE of 4 clients.

(d) 4 clients bandwidth share.

Figure 9.14.: **QoE and bandwidth allocation of VSiM videos by a real wireless link over 10Mbps bandwidth in mobile networks.**

- *Impact of bandwidth variance.* We report the bandwidth and QoE variations over time by a real wireless link in mobile networks at bandwidth 10Mbps with two or four mobile clients in Fig. 9.14 to show how the system assigns the bandwidth and the impact of bandwidth allocation on the QoE changes.

In Fig. 9.14(a) and (b), we observe that both the average QoE and allocated bandwidth of two mobile clients C_1 and C_2 are close at an initial period of time t(*e.g.*, $t \in [0, 60s]$). This is because both C_1 and C_2 at this period of time are moving inside the BS with a long staytime (*e.g.*, greater than 60s). In this case, we give the same staytime value (*e.g.*, 60s) for these two clients, which is given to avoid one client occupying the whole bandwidth and further improve the QoE fairness. Then, after a period of time, C_2 is still inside the BS, but the staytime of C_1 is short and may go to next BSes or experience some connection loss area because of the fast movement. VSiM captures this and utilizes the optimization strategy to improve the allocated bandwidth and QoE of C_1. In Fig. 9.14(b), it is clear to see that C_1's bandwidth increases. Because of the fixed total bandwidth, C_2's bandwidth is reduced. Similarly, in Fig. 9.14(c) and (d) with 4 clients, we can see that after some time, C_1 and C_4 are allocated with higher bandwidth, which improves their QoE. Because of their mobility, they may experience low viewing quality (*e.g.*, experience connection loss zone or frequent handoffs) after a period of time. VSiM improves the clients with lower viewing quality to maximize the QoE fairness for all clients.

Algorithm	100ms	200ms	300ms
VSiM (Min QoE)	**76**	**73**	**68**
Cubic (Min QoE)	69	67	63
VSiM (Avg. QoE)	**82**	**80**	**78**
Cubic (Avg. QoE)	80	78	73

Table 9.2.: **VSiM maintains high Minimum (Min) and Avg. (Average) QoE than Cubic under various latency conditions.**

• *Impact of latency variance.* Table 9.2 illustrates the impact of latency variance on VSiM and Cubic. For Table 9.2, we observe that VSiM achieves better performance in terms of both the minimum QoE and average QoE than those of Cubic. Specifically, the minimum QoE achieved by VSiM has increased by about ∼7 points, ∼6 points, and ∼6 points compared to cubic with 100ms, 200ms, and 300ms, which means that the clients' viewing quality of VSiM can obtain at least 1080P while the clients' viewing quality of Cubic is 720p with these different latency. Meanwhile, VSiM fulfills a better average QoE.

Compare with state-of-the-art. In this section, we compare VSiM with state-of-the-art regarding the average QoE, QoE fairness, and CDF over various bandwidths.

• *Average QoE Comparison.* In Fig. 9.15, we observe that: 1) increasing the bandwidth will improve the mobile client's total QoE. This is because a higher bandwidth value leads to the DASH requesting a higher bitrate, which further improves each client's QoE; 2) VSiM outperforms state-of-the-art solutions with both ten mobile clients over different bandwidths regarding the average QoE.

Specifically, the average QoE improvement at bandwidth 25Mbps of VSiM is about 13% (about 2.0 points) and 30% (about 4.3 points) higher than that of Cubic [197] and Minerva [180] while around 11% (about 1.8 points) and 16% (about 2.4 points) on average improvement is achieved by VSiM compared with PreCache [21] and

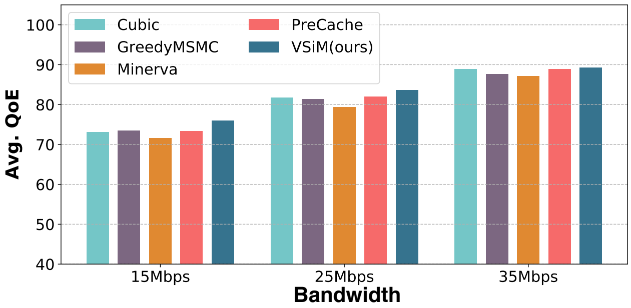


Figure 9.15.: **VSiM fulfills a higher average QoE compared with various algorithms over different bandwidths.**

GreedyMSMC [20]. VSiM with more than 2.0 points improvement can at least jump one level of bitrate compared with state-of-the-art in terms of 1080p video. This means that in the same bottleneck bandwidth and mobile networks, the average bitrate value of all mobile clients that can be used is 3340kbps in Cubic while all mobile clients in VSiM can at least watch videos with 3613kbps for the average bitrate value regarding 1080p. This is because VSiM considers the mobility pattern and HTTP/3 characteristics (such as server push) to optimize the bandwidth allocation for different mobile users.

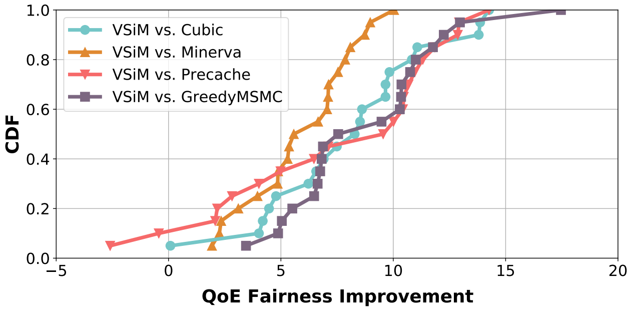


Figure 9.16.: **QoE fairness improvement achieved by clients under various algorithms with 25Mbps bandwidth.**

- *CDF Comparison.* Fig. 9.16 illustrates the CDF of QoE fairness improvement over Cubic, Minerva, GreedyMSMC, and PreCache with ten mobile clients collected by 20 runs in a real wireless network at 25Mbps bandwidth. As we discussed in Section 9.3.1, VSiM can accomplish a video quality jump from 720p to 1080p if the improvement with QoE normalization is greater than 5.8 points. We notice that there are $\sim 55\%$ (about 6.7 points), $\sim 40\%$ (about 6.5 points), $\sim 30\%$ (about 6.3 points), and $\sim 25\%$ (about 6.4 points) probability for VSiM to achieve the value of QoE fairness improvement being larger than 5.8 points compared to Minerva, PreCache, Cubic, and GreedyMSMC. VSiM fulfills a video quality jump from 720p to 1080p with these probabilities. For example, suppose the minimum video quality of Minerva over all clients is 720p. In that case, the minimum video quality of VSiM over all clients has a probability of $\sim 55\%$ to fulfill 1080p in the same mobile bottleneck environment. This significant improvement depends on the key designed techniques in VSiM for a mobile environment.

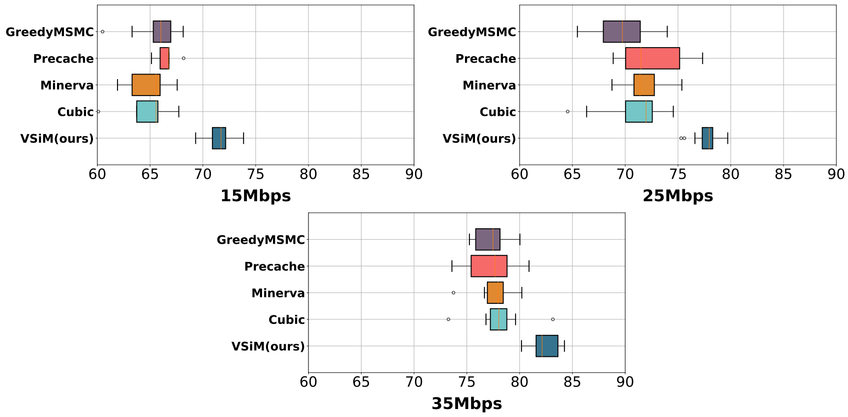


Figure 9.17.: **VSiM fullfils a significant improvement of QoE fairness comparison with various algorithms.**

- *QoE fairness comparison.* Fig. 9.17 records the clients with minimum QoE for each run. The longer the box, the greater the variance of the experimental results of different runs is, which means that the results are worse. As expected, the lowest bandwidth has the lowest QoE and vice versa. Besides, we observe that VSiM achieves good QoE fairness over varied bandwidths. For example, in Fig. 9.17, the QoE fairness for the median value of VSiM improves an average of about 40% (about 5.9 points with QoE normalization) for all the bandwidth than that of Minerva, which means that VSiM can accomplish a jump from 720p to 1080p. Especially for 15M bandwidth, the median QoE fairness of VSiM improves about 51% (about 6.9 points with QoE normalization) than Cubic. Compared to Minerva, GreedyMSMC, and PreCache, VSiM fulfills about 49% (about 6.6 points), 43% (about 5.9 points), and 36% (about 5.0 points) QoE fairness improvement on the average with aspect to the median value overall bandwidth. Additionally, we observe that the variance (*e.g.*, box size) of VSiM is small over different bandwidths.

9.5. Appendix

9.5.1. Convergence Proof

In the following part, we prove that the bandwidth allocation method in Sec. 9.2.2 will converge to utility fairness.

Definition 9.5.1. *There are n clients in a mobile network and all the clients share the same bottleneck bandwidth to download videos from a server. The egress bandwidth of the server is fixed and it is denoted by B. The i^{th} client has a utility function $U_i\left(r_i\right)$ in which r_i denotes its available bandwidth. We wish to find a fair bandwidth allocation for the client with the intention to maximize the QoE of the clients with minimum QoE.*

It is reasonable to say that the available bandwidth for client c_i can be B at most, thus, we could get

$$\arg\max_{r_i} U_i(r_i) = B. \tag{9.4}$$

Therefore, we could normalize the utility function as follows:

$$\widetilde{U_i}\left(r_i\right) = \begin{cases} 0, & r_i = 0, \\ \dfrac{U_i\left(r_i\right)}{U_i\left(B\right)}, & 0 < r_i < B, \\ 1, & r_i = B. \end{cases} \tag{9.5}$$

With the above utility function definition, our optimization problem can be modeled as

$$\max \quad \min_i \widetilde{U_i}(r_i) \tag{9.6a}$$

$$\text{s.t.} \quad \sum_i r_i = B. \tag{9.6b}$$

It is reasonable to infer that the utility function is concave since clients' experience diminishes the marginal utility as the bandwidth increases. Then we have the following theorem

Theorem 9.5.1. *$\widetilde{U}(r_i)$ is non-decrease concave function, for $0 \leq x \leq y \leq B$, $0 < \alpha \leq 1$ we have*

$$\left(\frac{y}{x}\right)^{\alpha} \leq \frac{U\left(y\right)}{U\left(x\right)} \leq \frac{y}{x}. \tag{9.7}$$

Theorem 9.5.2. *There exists an optimal allocation $\{r_i^*\}$ that reaches the goal in which $\{\widetilde{U_i}(r_i)\}$ are equal for all participating clients. At each time window, we could get a series of weights using*

$$w_i = \frac{r_i}{\widetilde{U_i}(r_i)},$$

then we could allocate the egress bandwidth as

$$r_i = \frac{w_i}{\sum_{i=1}^{N} w_i} B.$$

The above allocation ensures that r_i will converge to r_i^* after t iterations.

9.5.2. Proof of Theorem 9.5.1

Proof. Let $f(t) = \frac{U(t)}{t}$, then we can get

$$f'(t) = \frac{U'(t)t - U(t)}{t^2},$$

$$\frac{\partial (U'(t)t - U(t))}{\partial t} = U''(t) + U'(t) - U'(t) = U''(t) \leq 0.$$

Hence, for the term $U'(t)t - U(t)$, we know it takes the maximal value at $t = 1$, i.e.,

$$\max_{t \in [0,1]} \left(U'(t)t - U(t) \right) = U'(0) * 0 - U(0) \leq 0.$$

So for $t \in [0,1]$, $U'(t)t - U(t) \leq 0$, then $f'(t) \leq 0$. Thus, $f(t)$ is decrease function, we then can get $\frac{U(y)}{y} \leq \frac{U(x)}{x}$. Thus, we prove

$$\frac{U(y)}{U(x)} \leq \frac{y}{x}.$$

Similarly, we can prove the left side and thus, the theorem is proved. □

9.5.3. Proof of Theorem 9.5.2

Proof. We first prove the convergence for special case, i.e., for two clients. We denote the two clients' utility functions as $\widetilde{U_1}$ and $\widetilde{U_2}$, respectively. The bandwidth of client i in t iteration is denoted by r_i^t. There exists an optimal bandwidth allocation (r_1^*, r_2^*) satisfying the condition that $\widetilde{U_1}(r_1^*) = \widetilde{U_1}(r_2^*)$.

Without loss of generality, we hope to prove the convergence that $r_1^t \to r_1^*$ and $r_2^t \to r_2^*$. It is equivalent to prove that $\frac{r_2^t}{r_1^t} \to \frac{r_2^*}{r_1^*}$.

In each iteration of the weight updates, if the clients compute their weights $w_i = \frac{r_i}{\widetilde{U_i}(r_i)}$, then we could get

$$\frac{r_2^{t+1}}{r_1^{t+1}} = \frac{w_2}{w_1} = \frac{\widetilde{U_1}(r_1^t)}{\widetilde{U_2}(r_2^t)} \frac{r_2^t}{r_1^t}. \tag{9.8}$$

.

We denote $X_1^t = \frac{r_1^*}{r_1^t}$ and $X_2^t = \frac{r_2^t}{r_2^*}$, from Equation (9.8), we could get

$$X_1^{t+1} X_2^{t+1} = \left(\frac{\widetilde{U_1}\left(r_1^t\right)}{\widetilde{U_1}\left(r_1^*\right)} X_1^t \right) \left(\frac{\widetilde{U_2}\left(r_2^*\right)}{\widetilde{U_2}\left(r_2^t\right)} X_2^t \right). \tag{9.9}$$

On the other hand, from Theorem (9.5.1), we could get

$$\frac{r_1^t}{r_1^*} \le \frac{\widetilde{U_1}\left(r_1^t\right)}{\widetilde{U_1}\left(r_1^*\right)} \le \left(\frac{r_1^t}{r_1^*}\right)^{\alpha}, \tag{9.10}$$

$$\left(\frac{r_2^*}{r_2^t}\right)^{\alpha} \le \frac{\widetilde{U_2}\left(r_2^*\right)}{\widetilde{U_2}\left(r_2^t\right)} \le \left(\frac{r_2^*}{r_2^t}\right). \tag{9.11}$$

Then we could get

$$1 \le X_1^{t+1} X_2^{t+1} \le \left(X_1^t X_2^t\right)^{1-\alpha} \le \left(X_1^0 X_2^0\right)^{(1-\alpha)t}. \tag{9.12}$$

As $t \to 1$, $X_1^{t+1} X_2^{t+1} = 1$, then we can conclude that $r_1^t \to r_1^*$ and $r_2^t \to r_2^*$.

The above procedures ensure that the proposed bandwidth allocation method could realize fairness in the end. Using the above procedures recursively we can conclude that VSiM will allocate bandwidth fairly i.e., optimally in terms of utility. Thus, Theorem (9.5.2) is proved. □

9.6. Chapter summary

In this chapter, we propose VSiM, the first end-to-end QoE fairness scheme for mobile video traffic with multiple mobile clients. VSiM leverages clients' mobility profiles, QoE-related information, and SDFR server push strategy to allocate bandwidth that maximizes the QoE fairness in real-time. VSiM is easy to deploy in the real world without touching the underlying network infrastructure. We implement VSiM in both simulation and prototype tests on top of HTTP/3. In the simulation, we verify the contribution of each key technique and robustness of VSiM, like different topologies, different video lengths, various mobility patterns, as well as various clients number and ABR algorithms. In the prototype, we find that VSiM outperforms state-of-the-art approaches, with about 40% QoE fairness improvement (equal to clients' viewing experience in resolution from 720p to 1080p). Meanwhile, VSiM ensures about 20% improvements on average of the averaged QoE (equal to the bitrate level improvement of clients' viewing experience from 2087kbps to 2409kbps in 1080p resolution over the public dataset). In future work, we plan to test and deploy VSiM in real-world service provider networks.

Chapter 10

Future Prospects

In this chapter, we present possibility of how to transfer VSiM to MoniSys. We consider and present the future prospects of the proposed communication-phase system and its possible variances.

Contents

10.1. Video on Demand *vs.* Live Video Streaming

DASH framework is proposed for video on demand (*e.g.*, Netflix, Youtube), but application in MoniSys is closer to live streaming. Visual data sensed by UAVs then encode into video in real time and feeds into network. Rather than pre-encode video on demand with various bitrates, in MoniSys, system should holistic decide the bitrate and encode current video chunk according to the network condition, analysis requirement and overhead. WebRTC is one of the most popular protocol for live streaming, and Adaptive QP mechanism is also widely used in video analytics system (see 12.2.1 for details). In next step, we are going to change the objective function in VSiM and explore another fairness system (*e.g.*, accuracy-fairness, or weighted summation of impacts).

10.2. Under Unstable Environment

In this section, we consider the possible area of extension under unstable environments.

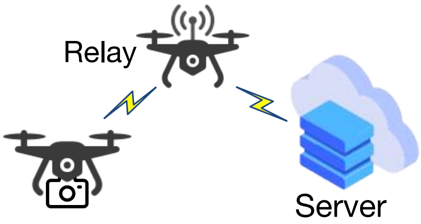


Figure 10.1.: **Data forward by relay UAVs.**

Non-direct Connection among UAVs and Server. MoniSys is commonly tackling monitoring under emergency and dangerous scenarios. Majority of possible environments do not allow UAVs directly connect to the server because of the possible electromagnetic interference or hostile environment condition. In this context, UAVs often categorize into sensing UAVs and relay UAVs. Sensing UAVs are still in charge of capturing environment data, while relay UAVs are in charge of forwarding the sending data to server. The advantage of this architecture is improving the system robustness. However, it leads to more challenges in network management, *e.g.*, the bandwidth allocation is VSiM. Specially, the bottleneck among all UAVs may be not the same. In VSiM, there is an assumption that bottleneck occurs at the last mile on the Internet, *i.e.*, the access link from the users to base station, while in such ad hoc architecture, it is not appropriate any more because of the routing. In the future, we may deploy the bandwidth allocation on the relay UAVs, but instead of centralized mechanism in VSiM, a distributed decision negotiation of bandwidth allocation if friendly to limited power and computation resource limited UAVs.

Part III.

Addressing Analysis Challenge in UAV Monitoring System: Preliminary study for Video Analytics Pipeline

Chapter 11

A Preliminary Study of Video Analytics Pipelines

In this chapter, we focus on the video analytics pipeline. As we are still working on this project, we can only give some preliminary study here. First, we give the introduction.

Contents

11.1. Introduction

Advances in computer vision presents a great opportunity to process and analyze huge amount of video data generated by pervasive video cameras [201–203]. Deep Neural Networks (DNNs) [36, 204–207] has improved the accuracy of many vision tasks dramatically but at high computational cost of forward inference. This resulting accuracy-latency-computation tradeoff necessitates distributed video analytics pipelines (VAPs) [31, 41–43, 46, 208], in which compute-intensive inference tasks and necessary videos are offloaded and streamed to edge or cloud.

Prior VAPs focus on reducing the bandwidth cost and latecy when stream video from camera to edge/cloud, and increasing accuracy when execute inference on cloud. From coarse-tuning camera configuration (*e.g.*, resolution and frame rate) [42, 45, 209] to fine-filtering invalid frames (*e.g.*, frames contain empty street in traffic video) [31, 33, 34, 208] and invalid content (pixels) in frames (*e.g.*, street part is invalid compared to vehicles) [35, 43, 44, 209, 210] of video data, the networking community has brought great advancements in bandwidth saving. Compared to bandwidth saving, however, the community is still at the early stage in accuracy increasing. Inspired by the observation from computer vision (CV) community — running object recognition related tasks on high-resolution images can largely increase the detection accuracy [211] — VAPs [38, 39, 46, 47] tries to utilize image (frame) enhancing model (*e.g.*, Super Resolution (SR) [206,207,212] and Generative Adversarial Network (GAN) [213, 214]) to enhance image details before fed them into inference model. Nevertheless, image enhancement causes additional latency resulting in 100ms-500ms end-to-end latency [46], which is far away from real-time requirement (24fps-30fps).

We believe that enhancement is a promising way to increase inference accuracy along with rapid advances in DNN [215] and falling GPU cost [216], however, it still leaves large room for improvement. Prior image enhancement mechanisms no matter to improve watchers' quality of experience [217–219] or to increase the accuracy of computer vision tasks [38, 39, 46, 47] treats every received pixel in frame equally. Although [38] enhances only small objects (*e.g.*, small faces), the method it uses to find out small objects still on frame-level. In other words, their basic processing unit is frame.

While the frame-level enhancement mechanism has served us well, we argue that it is suboptimal for enhancement-participated video analytics. The frame-based approach hinges on one premise: the analytics model process videos frame by frame. It needs to be revisited in enhancement-participated video analytics.

Different from user-centric video streaming pursuing smoothness of frame quality, the contribution of different part (pixels) of a frame to inference accuracy varies widely. In object detection, for example, only the foreground part containing vehicles are valuable to traffic flow analysis; on the contrary, enhancing background pixels (*e.g.*, streets and trees) is worthless except to increase system latency. As a result, the VAPs equipped with frame-level enhancement mechanisms can never get rid of being suboptimal. To tackle this problem, VAPs need a *subframe-based mechanism* to find out and enhance the "important" content.

In this context, we bring holistic thinking of video codec and inference tasks into VAPs. The fundament of video encoding is using signal processing techniques to eliminate *block-level* (a small region contains 16×8 or 8×8 pixels in frame, see 12.1.1 for details) spatial-temporal redundancies on that occur in videos; coincidentally, distributed VAPs transmit video from the camera to the edge/cloud for compute-intensive inference, which naturally necessitates video encoding and decoding. Bits stream (the data format of compressed video stream in distributed VAPs) contains plenty of free information about key blocks can help us find out the "important" content. On the other hand, how video encoding eliminates spatial and temporal redundancies hidden in the bits stream can accelerate the "important" content enhancement and inference.

The challenge of subframe-enhancement mechanism, however, is *how to derive important content and enhance them from free block-based bits stream.* To demonstrate the feasibility of the approach, we prototype the system and quantify benefits and cost of it. We open the decoder and take a first attempt to answer the following question: *How overlooked but handy information in decoder helps accelerate and improve video analytics system?* In answering this question, we dig out and make use of frame also macro block dependency from decoder. We believe that there is still many information unexplored to achieve larger improvement (see §13).

Chapter 12

AccDecoder: Accelerated Decoding for Neural-enhanced Video Analytics

In this chapter, we first present the background knowledge for video analytics pipeline. Then, we analyze the design space and comprehensively discuss and compare the advantages and limitations of state-of-the-art video analytics pipelines. After, we propose our AccDecoder and its key design choices. Unfortunately, we have not finished this project when I submit this dissertation, therefore, I cannot offer full version of this project.

Contents

12.1. Background

We start with setting up the basic knowledge (§12.1.1) then its status quo (§12.1.2).

12.1.1. Preliminary knownledge

Video Analytics Pipeline (VAP). Many computer vision tasks are considered in VAPs, such as traffic control [220], surveillance and security [38, 221], as well as digital assistant [222]. We consider two tasks as running examples −*object detection* and *semantic segmentation*. Object detection aims to identify objects of interests (*i.e.*, their locations and classes) on each frame in the video, whereas semantic segmentation labels each pixel with one class. Selecting these two tasks has two major reasons: first, both of them play the core role in computer vision community because a wide range of high-level tasks (*e.g.*, autonomous driving) are built on them; second, we seek to keep consistent with prior video analytics work [43, 45] to let the performance comparison be more straightforward and convincing.

Distributed Architecture. The proliferation of VA is facilitated by the advances of deep learning and the low prices of high-resolution network-connected cameras. However, the accuracy improvement from deep learning is at the high computational cost. Although the state-of-the-art smart cameras can support deep learning method, the deployed surveillance and traffic camera paint a much bleaker resource picture. For example, DNNCam [223] that ships with a high-end embedded NVIDIA TX2 GPU [224] costs more than $2000 while the price of deployed traffic cameras today ranges $40-$200 [225, 226]; these cameras typically loaded with a single-core CPU only provide very scarce compute resource[15]. Because of this huge gap, the typical *video analytics pipeline (VAP) follows the distributed architecture. E.g.*, a vehicle detection pipeline consists of: a front-end traffic camera compresses and streams live video to a compute-powerful edge/cloud GPU server upon wire/wireless network; and, a back-end server decodes received video into frames and feeds them into models (*e.g.*, Faster RCNN [36]) to detect vehicles.

Video codec basics. Video codecs, composed of encoder and decoder, are software/hardware program used to compress size of video files for easier storage or delivery over network. Encoders compress video data and wrap them into common video formats (*e.g.*, H.26x [227], VPx [228], AVx [229]), while decoders decompress the compressed video data into frames before post-process (*e.g.*, playback or analy-

[15] Professional UAVs may equipped with high-end GPUs but most of civilian UAVs are not

sis). By selecting the encoder, its coding standard and algorithm will determine the computational cost and compression effect. For example, MPEG4 encoders usually have low hardware requirements and are easy to implement, while H.264 ones have better compression. This compression process is usually lossy, which strikes the balance between video quality and compression ratio according to encoding settings of users (*e.g.*, bitrate, frame rate, and group of pictures).

Let us taking H.264, one of the most popular codecs, as an example explain the compression process. During encoding, each video frame is first divided into non-overlapped macroblocks (16×16 pixels), then to each macroblock, the encoder searches for the optimal compression method (including the block division types and encoding types of each block) according to the pixel-level similarity and encoding settings. One macroblock may be further divided into non-overlapped blocks (8×8 or 8×16 pixels) and each of them is encoded with intra- or inter-frame types. The intra-coded block is encoded with the most pixel-value similar reference block in the same frame, and the offsets between these two blocks are encoded into *motion vector (MV)* and *residual.* With the same procedure, inter-coded block locate the most pixel-value similar block searched by reference index and motion vector from other frames. Motion vector indicates the spatial offset between the target block and its reference while the difference in pixel values of two blocks is encoded as residual for decoding[16].

Performance metrics. Under this distributed architecture, the focal point of VAP is the tradeoff among three performance metrics: *accuracy*, *resource (bandwidth & computation) cost*, and *latency.*

- *Accuracy.* We comparing output on each compressed frame of this VAP (under limited bandwidth) with the output of the state-of-the-art analytics model on the same raw frame[17]. Using the output of this *golden configuration*, the state-of-the-art model and raw video input, instead of human-annotated labels as *"ground truth"* is useful in real applications and consistent with recent works [42, 45, 230]. This way, any inaccuracy will be due to VAP designs (*e.g.*, video compression, DNN itself). We identify *F1 score* as metric in objects detection task (the harmonic mean of precision and recall for detected objects' location and classes) and semantic segmentation (the intersection over union of pixels associated to the same class).

[16] Conceptually, {target block}= ({reference block}+{residual})·{MV} (here we represent the MV as a matrix).

[17] Raw frames often encodes into video with quality loss by setting quantization parameter (QP). The larger QP value, the smaller video file size but lower quality. Here, we define highest quality video when setting QP as 0.

- *Resource (Bandwidth & Computation) cost.* We define the total size of the video file delivered from the camera to the server of each VAP as its bandwidth cost. As GPU is the major hardware running DNN today, we use GPU time (100% usage ratio) to measure the computation cost of VAPs.
- *Latency.* In general, the end-to-end latency is the time interval from the time of camera capturing one image to the model on sever outputs its inference result, which consists of the camera processing (capturing images and encoding them into video) time, video delivering time, and server processing (decoding video into frames and run inference on each of them) time. In this paper, we focus on designing a tool (*AccDecoder*) which can plug into the server of any VAPs to accelerate the inference, thus we measure the latency includes the video delivering time and server processing time.

12.1.2. Video analytics pipeline: status quo

Here, we categorize the status quo of video analytics pipelines in four classes and analyze their limitation on purchasing *high accuracy, low resource cost*, and *low latency*.

Camera-side analytics-aware video compression. Unlike traditional video encoding designed for human visual quality, analytics-aware one opens new design space. Filtering similar frames and reusing the inference result from one representative is one popular method. For example, [34, 208] run light-weight DNN and [30, 31, 231] calculate the inter-frame pixel-level difference on camera to binary classify the frames into delivered and undelivered classes. However, these cheap methods may cause false positive (*e.g.*, pixel-level distance changes by background may trigger camera to send irrelevant frames) and false negative (*e.g.*, small appeared objects may be missed) results leading to unnecessary bandwidth cost or inference accuracy reduction. Other solutions (*e.g.*, [232–234]) design new framework encode on feature-maps-level, unfortunately, they work well on classification task not advanced location-sensitive tasks (object detection).

Server-driven video compression. To tackle the limited computational resource of camera, other approaches leverage server-side rich resource to generate feedback guiding camera adjusts video configuration (*e.g.*, the resolution and the frame rate adaption [42, 45]), or more fine-grained, adjust each pixels configuration (*e.g.*, high quality for pixels in the region of interest, such as target object, but low quality for the background [35,41,43,210]). This approach (*i.e.*, server control mode) introduce extra latency, which is especial large when the delivery between camera and cloud cross wide area network.

Server-training camera-executing video compression. Pioneer methods [44, 209] divide the VAPs into two stages: offline training and online executing. [209], on server, periodically probes inference DNN with accumulated images in time slot and search the best assignment for the weights of YUV channels; the broadcast the results to all cameras for encoding images in the next slot. [44], trains a shallow DNN which can assign high-or-low quality for all macroblocks (a small region containing 16×16-pixel image) in a frame on server; then deploy this DNN on camera to assign qualities for encoder.

Server-side data enhancement. Inspired by their succeed of image enhancement (via Super Resolution (SR) [206,207,212] or Generative Adversarial Networks (GAN) [213,214]) in video streaming for human-centric quality of experience (*e.g.*, [218] and [217] considers video on demand, [219] works for live video streaming), some studies [38,46,47] expand this approach to machine-centric video analytics system. It indeed increases accuracy, although, as we will elaborate in §12.2.1, the latency it causes much offset the accuracy increasing. The root cause is that image enhancement models are much heavier than other computer vision models. For instance, the complexity of SR model is as high as 1000× heavier than image classification and object detection model in terms of MultAdds [39].

Server-side model retraining. Continues learning tackles the data drift[18] issue by periodically retrains inference models thus improve model performance. Some work (*e.g.*, [235]) periodically retrain the final inference model with the labels generated from an expensive model. This teacher-student paradigm is only effective under sufficient network bandwidth to deliver raw video for generating high-quality labels. Ekya [235] periodically retrain the cheap final inference model ResNet-18 with the labels generated from expensive model ResNet-152. The teacher-student paradigm is feasible under sufficient network bandwidth for raw video utilized to generate high-quality labels.

In our system, we take a pragmatic stance to focus on the backend (edge/sever) in the design space – *decoder*. With no camera-side processing, no server-driven, and no model retraining, only data enhancement, we systematically exploits the abundant information in bits stream to improve enhancement mechanism. The reason is that: first, as we discussed in §12.1.1, majority of installed surveillance or traffic cameras only equipped with cheap CPU, which cannot support SOTA

[18] Data drift indicates that there exists bias between the distribution of training set and testing set. For instance, under different weather, day/night etc, pixels value may be affected and further impact inference results.

camera-side analytics-aware video compression method. Second, decoder plays the pioneer role for analyzing the bits stream.

12.2. Motivation

In this section, we elucidate the pros and cons from enhancement (§12.2.1) and use empirical measurements to illustrate the potential improvement (§12.2.2).

12.2.1. Pros and Cons from Enhancement

Pro: Accuracy Increment. Existing studies has shown enhancement improves the task accuracy a lot; *e.g.*, in object detection, image enhancement in [46] improves the 4% accuracy while enhance only small objects mechanism in [236] improves the accuracy of small objects up to 9%. We preliminary study the accuracy improvement by image enhancement. We compare the results from original resolution video with 320×180 resolution (lr) to SR video with three upscaling factors 2× (to 640×360 res), 3× (to 960×540 res), and 4× (to 1280×720 res). As the upscaling factor increases, the accuracy also increases but the accuracy gain emerge diminishing marginal utility; in particular, compared to lr, 2× increase 7% average accuracy, while the benefit of 3× and 4× is diminishing.

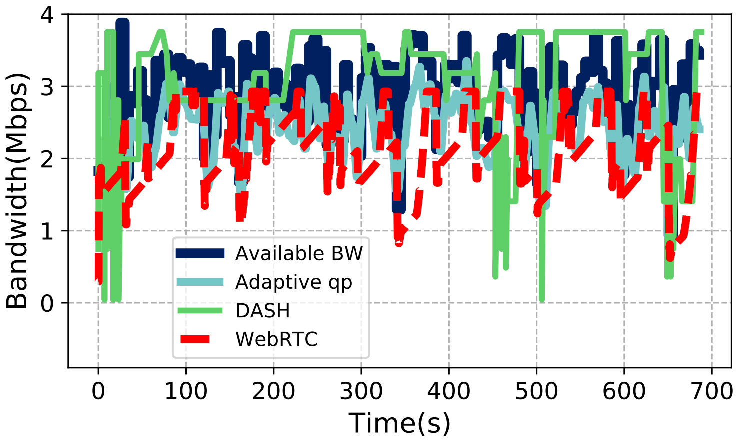


Figure 12.1.: **Streaming Protocol Comparison.**

Pro: Bandwidth Save. Video streaming in analytics pipeline is sensitive to the variation in available network bandwidth. Fig. 12.1 shows the the available bandwidth and video bitrate for video streaming for analytics (here use DDS [43] as an example), compared with live video using WebRTC for user viewing and on-demand video using adaptive streaming (DASH) [194, 237] given a FCC broadband network trace [238]. Video streaming for analytics, comparing to WebRTC, uses bandwidth more aggressively by sharply adjusting the encoding quality (*e.g.*, quantization parameter, qp, in [42, 43]) because it does not need to smooth the quality of successive chunks (2-10 second video) like WebRTC does for better user quality of experience. However, compared with on-demand video streaming, two live video streaming use bandwidth much more conservatively. The root cause is that pre-encode chunks with various bitrates in on-demand video streaming provide the chance to probe and re-transmit the same-content but different-bitrate chunk. As a result, the bandwidth utilization of video streaming in analytics is pretty well, but the backend of VAPs still has plenty room to explore.

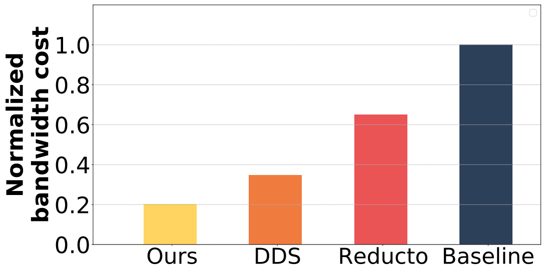


Figure 12.2.: **Bandwidth Saving.**

In backend, enhancement can significantly improve accuracy without requiring more bandwidth. Fig. 12.2 plots the bandwidth usage of enhancement mechanism (frontend transmits low-resolution video then backend enhances into high-resolution one) and two state-of-the-art VAPs (*i.e.*, DDS [43] and Reducto [31]) under the same target accuracy. The frontend in baseline directly transmits all high-resolution frames as a benchmark. DDS iteratively transmits low-resolution video and partial high-resolution video while Reducto only transmits necessary frames. In figure, enhancement mechanism only cost near half bandwidth of DDS and one third of Reducto, which demonstrates enhancement-participated mechanism saves considerable bandwidth compared to server-driven video compression (*e.g.*, DDS) and camera-side analytics-aware video compression (*e.g.*, Reducto).

Con: High Latency. Along with the accuracy improvement, latency significantly increases. Even from 2× upscaling, the total 102ms is much higher than 33ms of real-time (30fps) requirement; the latency of 4× increases up to 126ms. We further studies the latency and accuracy tradeoff of high resolution (1960×1080) images.

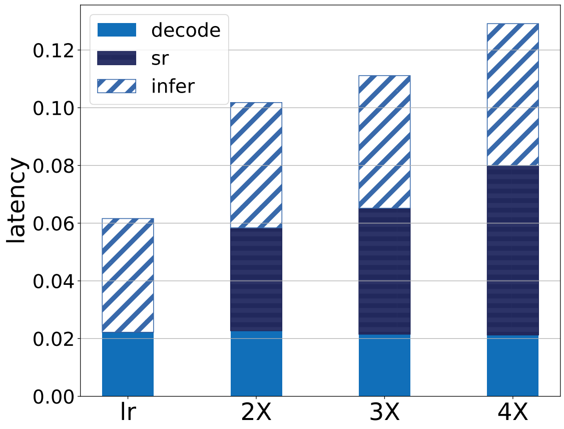


Figure 12.3.: **Latency Break Down.**

Compared to upscaling factors, the original resolution dominates the latency of image enhancement; namely, along with the resolution increasing, though upscaling factor decreases, latency even grows more. For example, upscaling 980×540 image to 1960×1080 costs 214ms. Current enhancement-based solution works well in video on demand (*e.g.*, [217–219]), but to more latency-serious analytics task (not only enhancement but also includes inference latency), prior approaches (*e.g.*, [38,46,47]) only achieve $2-3$ fps.

12.2.2. Potential Improvement

Traditional enhancement-participated video analytics based on frame-level, they enhance and infer frame by frame in order of video ingestion. In this section, we use three real datasets [239–241] upscaling them from 640×360 to 1280×720 to demonstrate the potential improvement of subframe-level enhancement mechanism from spatial. From temporal, it brings more potential which we leave to the final experiment.

12.2.2.1. Spatial Latency Reduction opportunities.

In one single frame, compared to video on demand enhances human quality of experience (QoE) by scaling each frame's resolution [217–219], in video analytics, however, it is crucial that the server-received video has sufficient video quality in the regions that heavily affect the DNN's ability to identify/classify objects; however, the received video does not have to be smooth or have high quality everywhere. This contrast has a profound implication – video analytics could achieve high "quality" (*i.e.*, accuracy) with much less latency in enhancement. Each frame can be spatially enhanced with non-uniform quality levels. In object detection, for instance, one may give low quality (*e.g.*, bilinear or bicubic interpolation [242]) to the areas other than the objects of interest. Fig. 12.4 shows that across three datasets (three scenarios), in $60\% - 85\%$ frames, the time cost of enhancing only the objects of interest less than 20% of the time cost of enhancing entire frame.

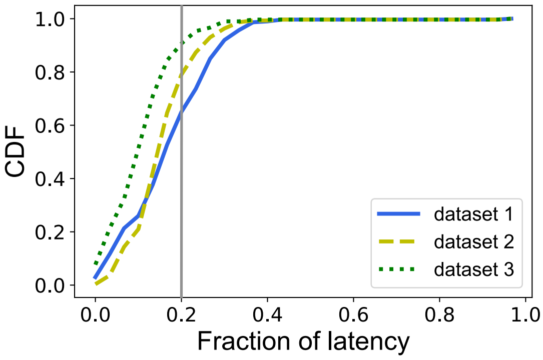


Figure 12.4.: **Latency saving opportunities from spatial.**

12.3. Key Design Choices

To overcome the limitations of frame-level enhancement in 12.2.2. AccDecoder applies enhancement (SR) only to partial "important" content in a subset of frames and transfer them to the remaining frames; it also applies inference DNN only on a subset of frames and transfer the inference results to other frames. The goal is to amortize the computational overhead of a SR DNN across the video. Moreover, we want to provide a guarantee that the resulting inference accuracy within a small margin compared to the per-frame enhance and infer; at the same time, guarantee the whole mechanism subject to the real-time constraint. We ask ourselves a series of pivotal questions that lead to the key design choices we make in achieving the goal.

12.3.1. On which granularity in bits stream to find out "important" content for enhancement?

Key Observations. "Important" content is the image after enhancement may improve the inference accuracy. Recent studies from computer vision community (*e.g.*, [236]) and networking community (*e.g.*, [43, 46]) find that the enhancement perfectly improve the inference results of small objects (*e.g.*, the size of traffic signs in Tsinghua-Tencent 100K dataset [243] are only 32×32 pixels but occupy 42% total number of objects). Coincidentally, *macroblock (MB)* is the quantization parameter (QP) assignment unit in video codec to determine the image quality of this 16×16-pixel content. On macroblock-grained, it is not too fine-grained (compared to pixels) to build huge solution space for searching important content, and also not too coarse-grained (compared to frame) to waste computational resource for enhancing unimportant content.

Approach. AccDecoder applies enhancement only to a subset of MBs, referred to as *anchor MBs*. The remaining MBs reuse the enhanced MBs according to the block dependencies (See Sec. 12.3.2) to up-scale their resolutions. Because enhancement models [206, 207, 212] often use overlapping blocks to extract frequent information, AccDecoder expands each anchor MB to each direction by β blocks (if they are not already selected). At the same time, to match the rectangle image shape feeding into enhancement model, AccDecoder fills the holes by switch the unselected blocks to selected one.

12.3.2. What to Reuse and How?

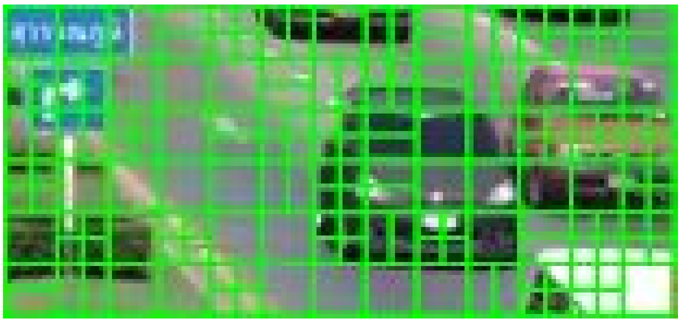


Figure 12.5.: **Each big 16×16-pixel green square is a *Macroblock*. Small 8×8-pixel squares or 8×16-pixel rectangles are *blocks*..**

What to reuse. To make the most out of caching, we *cache the final output* (*i.e.*, the enhanced high-resolution image) follows the same idea in [49, 217]. Their results show most of the latency occurs at the last couple of layers which means caching and reusing the output of enhancement model (*e.g.*, SR model) is the most effective in computation reduction.

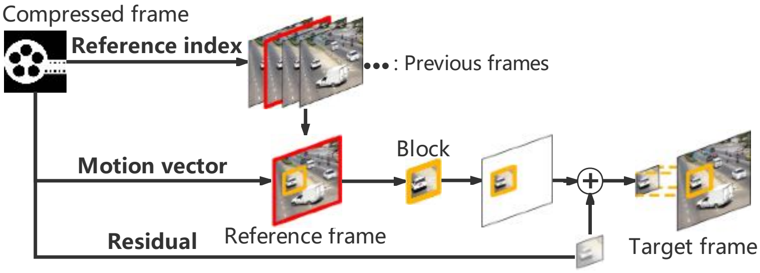
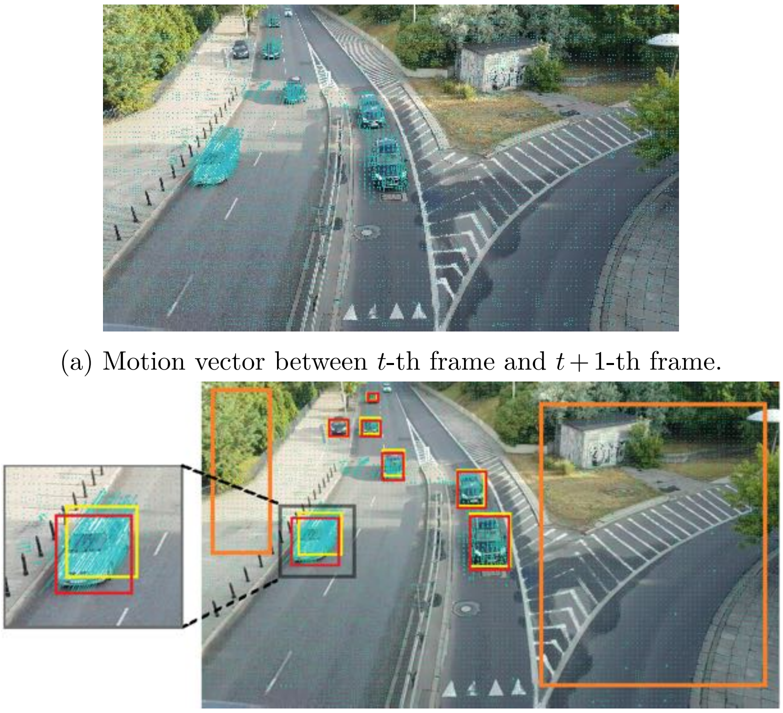


Figure 12.6.: **Illustration of inter-frame decoding. Block dependencies in decoder present an implication to reuse..**

How to reuse. Inter-frame encoding in common codecs (*e.g.*, H.26x [227, 244] and VPx [245]) leverages temporal redundancy among frames (in a group of picture) to "reuse" similar regions from reference frames. Figure 12.6 illustrates the process of decoder decoding an inter-coded block with dependency information. Decoder first uses reference index to extract a reference frame from previous decoded frames; then, it applies motion vector to the source block in the reference frame to transfer the block to the target frame; at last, it adds residual to the the transferred block to recover the target frame. Inter-frame dependencies allow us to enhance only some MBs, but reuse these enhanced MBs to reconstruct others.

12.3.2.1. Additional Inference result reuse

How to reuse. Motion vector indicates the offset of pixels among blocks in target frame and reference frame. Although pixel-level offset is very difficult mapping to semantic meaning (*e.g.*, object) in object [246], we argue that motion vector provides enough information to speed up inference without compromising accuracy especially in static traffic or surveillance cameras. Figure 12.7 shows the relation between motion vector and bounding box (Bbox) of detected objects. From this relation, we have two observations to speed up inference. Figure 12.7a illustrates the motion vectors between t-th frame and $t+1$-th frame; 12.7b shows the inference results of object detection on both t-th and $t+1$-th frame.



(a) Motion vector between t-th frame and $t+1$-th frame.

(b) Object detection results. Yellow boxes are from t-th frame, red one from $t+1$-th frame.

Figure 12.7.: **The motion vector implies the movement of bounding box and the static region in image.**

What to reuse in Object Detection. From the enlarged image of white moving car, we observe that the motion vectors of object always gather together and perfect match the movement of objects' Bboxes. Accordingly, VAP does not necessarily execute object detection on current frame but instead move the Bboxes of the reference frame along with the motion vector of each object; then, its inference latency gets amortized with the reference frame. We find that accuracy of motion vector decreases as the scene varies drastically, but taking a pragmatic stance to focus on static cameras, the performance has been maintained high lever.

What to reuse in Semantic Segmentation. To some other CV applications, like semantic segmentation whose output is the class of each pixel, reuse the output of reference frames is not practical. However, the motion vector also presents an opportunity to reuse intermediate results of inference model. We observe that macro blocks with small motion vectors often has similar pixel values. Follows this observation, we can cache the feature map (*i.e.*, output of convolutional layer) of these macro blocks in reference frame and load them when execute convolution on the current frame.

12.3.3. How to Guarantee Performance?

Key Observations. We would like to ensure that the accuracy performance AccDecoder delivers is within a small margin (*e.g.*, ≤ 0.1 f1-score) compare to that of per-frame enhancement while the constraint of the average latency (from receiving the bit streams to rendering the inference results on each frame) is no more than 33ms (*i.e.*, 30fps).

Given a video contains frame set $\mathcal{F}$, *i.e.* macroblocks set $\mathcal{MB}$ where $Decode(\mathcal{MB}) = \mathcal{F}$, select partial macroblocks $\mathcal{MB}_s$ to be enhanced such that average accuracy of this video is maximized:

$$\textbf{(P1)} \quad max \quad \frac{1}{|\mathcal{F}|}\sum_{i=1}^{|\mathcal{F}|} Acc_i(D(SR(\mathcal{MB}_i)+\overline{\mathcal{MB}}_i), D(SR(F_i))),$$
$$s.t. \quad \mathcal{MB}_i \in \mathcal{MB}_s, t_{AVG} \leq 33ms,$$

where, $\mathcal{MB}_i$ is the selected macroblock in frame $\mathcal{F}_i$, $\overline{\mathcal{MB}}_i = \mathcal{F}_i - \mathcal{MB}_i$.

Approach. We view it as two-level optimization problem that the upper-level frame selection problem categorizes all frames into three classes – SR frames, detect frames, reuse frames; the lower-level anchor MBs selection occurs on the select frames belong to SR class. Obviously, there is a gap between our approach and optimal solution of **P1** and we are focusing this gap when I am writing this dissertation. Hope we have good results.

Select anchor MBs in SR frames. We use AccModel, proposed in [44], to measure how sensitive of the quality change of macroblocks (16×16 pixels region) in one frame to the accuracy. AccModel is a light-weighted DNN (MobileNet-SDD [247] append three convolution layers). It maps a frame into a matrix that each element is the sensitivity value of the macroblock at the corresponding position. The sensitivity of the very top left 16×16 pixels, for instance, corresponds to the column-one-row-one element in the matrix. Fig. 12.8 visualizes the matrix values in the 4-th frame in dataset [240], in which the small object (*e.g.*, the cars in the top half frame) is sensitive to quality change, namely, these macroblocks should be enhanced.

Categorize all frames into three classes and their execution pipelines. For each frame in SR class, selected anchor MBs are enhanced by SR model and the remaining MBs are bicubic interpolated; after this, feeds the up-scaling frame into final DNN (*e.g.*, object detection) for inference. To the frames in detect class, they earn the benefits

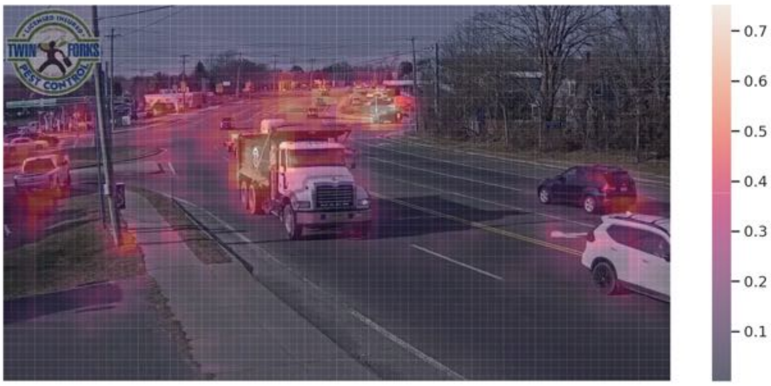


Figure 12.8.: **Sensitivity of quality change of macroblocks to accuracy.**

from SR MBs following the dependencies in Fig. 12.6 then feeds into final DNN for inference. Their accuracy still increases a lot due to the benefits transferring from SR MBs. Note that the transferring is quite fast (the time cost is the same as normal frame decoding) as it only includes additional bicubic interpolation on residual per frame compared to normal frame decoding. To those frames in reuse class, *e.g.* object detection, we get the bounding box of each object in the last (playback order) detect frame, calculate the mean of all motion vectors that reside in the bounding box, and use it to shift the old position to the current position. While, to the semantic segmentation for reuse frames, we get the class of each pixel in the last (playback order) detect frame, calculate the mean of all motion vectors for each macroblock, and use it to determine whether the pixels in this macroblock is static (compared to last detect frame) or not; if it is static, then set all pixels in this macroblock black.

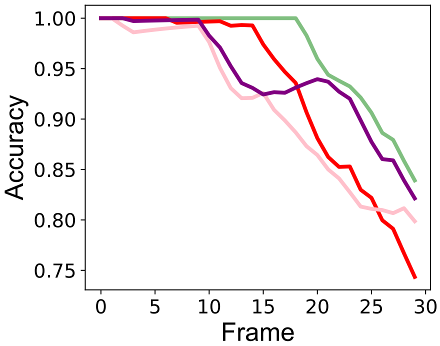


Figure 12.9.: **Reuse error accumulation of MVs.**

Categorize metrics. The accuracy of reuse frames drop dramatically along with error accumulation from cache of SR MBs or inference results. In Fig. 12.9, taking

Object Detection as an instance, reusing bounding box detected in the first frame to the following frames causes significant accuracy reduction. Consequently, *we should carefully categorize three classes.*

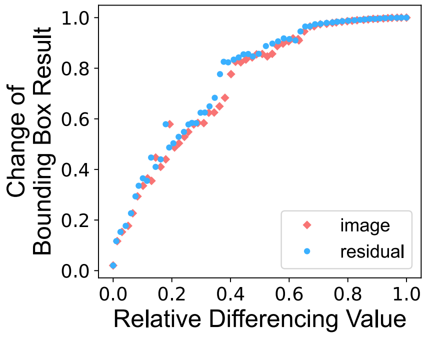


Figure 12.10.: **Correlation between differencing values of Laplacian on both image and residual and changes in Bbox results.**

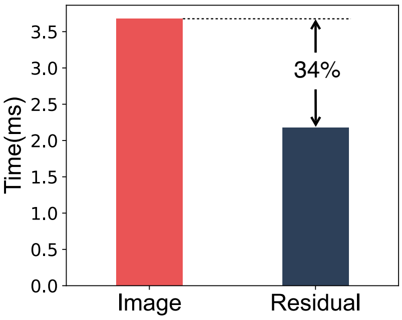


Figure 12.11.: **Time comparison of executing Laplacian on residual and image.**

The constraint of real-time (30fps) video analytics restricts us not to use heavy-weighted metrics to categorize frames. Here, we extract the light-weighted features on the residual of each frame. Our intuition is information of residual is sparse and de-redundent. It preserves difference among frames but not too dense to process, thus provides us a good opportunity to efficient filtering frames for free. We tried many light-weighted features (*e.g.*, Pixel, Edge, Area, HoG used in [31]) and find out the Laplacian (*i.e.*, Edge) on residual has a high correlation with the inference accuracy. Fig. 12.10 plots the correlation is very strong which implies that Laplacian on residual profiles the impacts of reuse error very well. At the same time, executing Laplacian operator on residual is 34% faster than that on image as Fig. 12.11.

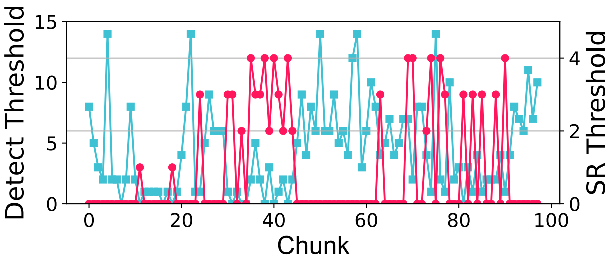
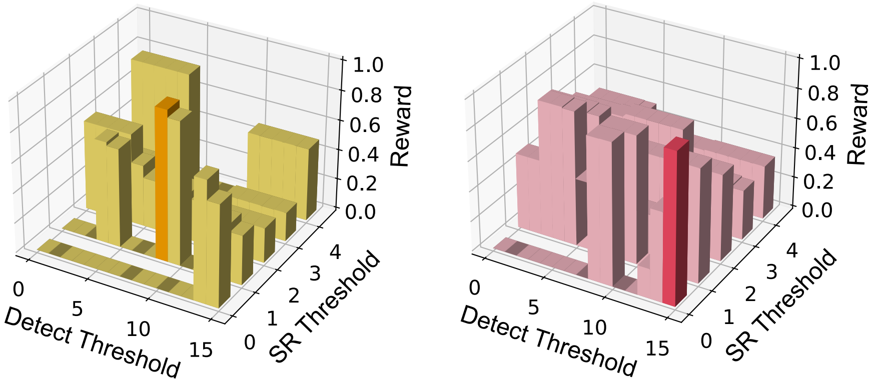


Figure 12.12.: **Best threshold vary across chunks (even adjacent).**

Categorize controller. Categorizing frames into three classes is not trivial. Optimal threshold of feature difference (*e.g.*, pixel, *i.e.*, Laplacian operator, feature) among chunks in one video varies a lot. Fig. 12.12 plots the best thresholds on video [239], which implies we should dynamic adjust the threshold.

On the other hand, across various videos, their best threshold combination is more dynamic (As shown in Fig. 12.13).



(a) Video 1

(b) Video 2

Figure 12.13.: **Best threshold vary across videos..**

In this context, we use Deep Reinforcement Learning (DRL) as the classifier to adaptively adjust the threshold of categorize metric according to the video content. AccDecoder exploits and leverages the free information from bits stream as illustrated in Fig. 12.14. Diff extractor compares the value differs of Laplacian on continuous residuals, while content feature extract on reconstruct/decode frames by VGG16 [248], and motion vector is used to reuse the inference results.

We formulate the adaptive pipeline selection problem in a way to maximize the accuracy given latency limitation, which can be expressed as:

$$\max_{\mathbf{x}} \sum_{f \in F} \alpha_1 Acc(x_f) - \alpha_2 D(x_f), \tag{12.1}$$

where $\mathbf{x} = \{x_1, ..., x_F\}$, F is the set of frames, ACC is the accuracy of a frame, D is the latency of the pipline selection, and α_1 and α_2 are weights.

We model trial-and-error learning as a Markov decision process (MDP). At each time t, the agent observes the current state s_t of the interactive environment and gives an action a_t according to its policy. Then, the environment returns reward r_t

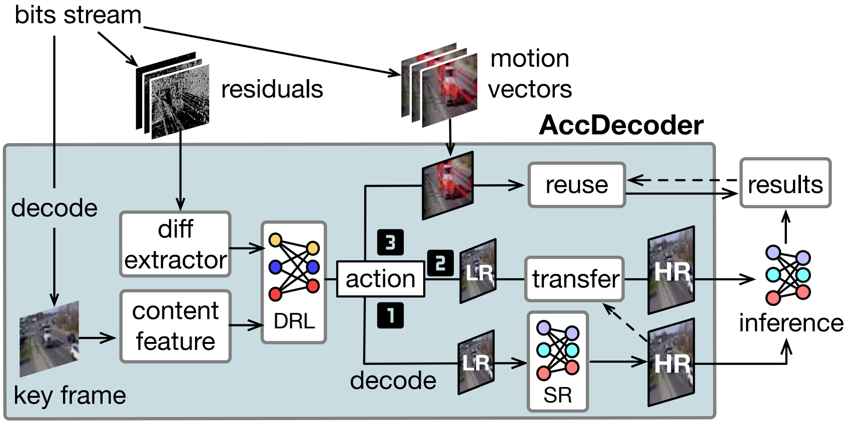


Figure 12.14.: **Workflow of AccDecoder.**

as feedback, and moves to the next state s_{t+1} according to the transition probability $P(s_{t+1}|s_t, a)$.

The goal to find an optimal policy can thus be formulated as the mathematical problem of maximizing the expectation of cumulative discounted return $R_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$, where $\gamma \in [0,1]$ is a discount factor for future rewards to dampen the effect of future rewards on the action; r_k is the reward of each step, and T is the time horizon before game over.

In particular, for the RL algorithm used by the agents, we will consider the well-known REINFORCE algorithm of. REINFORCE, which is often considered a special case of actor-critic algorithms, was originally proposed for single-agent reinforcement learning problems. Here, we present a variant with extension to multi-agent environments which incorporates communication.

- **State**: The feature of the first frame of the current chunk is extracted through the 1x1x1000 fully connected layer of vgg16. Since the dimension of this feature is too large, we use PCA to reduce it to 128 dimensions. The accumulated “edge” difference (1 dimension) between the frame selected for detection in the previous chunk and the first frame of the current chunk plus the edge difference between every two frames in a chunk (29 dimensions), a total of 30 dimensions. (Used to count which frames to do detect). Similar to the above, find the edge (with the Laplacian operator) on the residual as the state. (Used to calculate which frames do SR).
- **Action**: Select two thresholds P_1 and P_2, we combine these two elements to form an action space a=(t1, t2), first select which frames need to be detected through t1, and then apply t2 to these frames, select which frames Frames need to be SR,

and the remaining frames are multiplexed.

- **Reward**: Consider two aspects, the average f1score of the current chunk, and then the time it takes to get the detection results of these frames. Our goal is to achieve real-time detection, so the chunk needs to complete all tasks before the arrival of the next chunk (complete within 1s), Negative rewards will be given when the specified time is exceeded.

$$R_t = \alpha_1 \times f1score - \alpha_2 \times (1 \ if \ time > 1 \ else \ 0), \tag{12.2}$$

  where α and $\alpha 2$ are the weight factors to balance the preference to delay and accuracy. The definition of this reward is very liberal as it allows us to model different user preferences according to different situations.

12.4. Chapter Summary

In this chapter, we argue that the lack of holistic think in prior studies neglects handy information in decoder to speed up the video analytics pipeline. Our goal is to highlight the encoding information, while not accurate, is sufficient to reveal what information can be reused, thereby speed up DNN models. Through case studies, we explore preliminary design of task-driven super-resolution model, the SR frame reuse based on frame dependencies, and the potential inference acceleration based on motion vectors. The preliminary results demonstrate the benefit of our design.

Chapter 13

Future Prospects

In this chapter, we present several issues under study and future directions.

Contents

13.1. Issues to be addressed

The gap between two-level solution and macroblock selection over entire video. In Sec. 12.3.3, **P1** formulate the MBs selection problem. However, our two-level solution, upper-level frame categorization and lower-level MBs selection in one frame, is two heuristic which cannot guarantee the performance differs from **OPT**. For example, if the optimal MBs selection are distributed over per frame, our solution locates so far from OPT in the solution space. Under this consideration, we plan to from two aspects: 1) mathematically prove the properties of objective function in P1 and bound the gap; 2) statically study the impacts of MBs selection on unbiased large video set.

13.2. Future Direction: Online training

Real-time video analytics can be treated as a live streaming system. In this scenario, pre-trained task-driven super-resolution model may not work well because of dynamic the video content may cause non-iid data problem [249]. Non-iid indicates that there exists bias between the training set and testing set. For instance, under different weather, day/night etc, pixels value may be affected and further impact DNN inference. To tackle this challenge, content-aware SR model is an optional method. Leveraging online training is a promising way to train such model but resource allocation [235] is challenging.

End-to-end enhancement plus inference model. The super resolution model often consists of three elements: deep convolutional networks to extract features, non-linear mapping from extracted features to feature maps, and reconstruction. The inference DNN model also needs to use deep convolutional networks extracting features. Many prior studies [233, 250, 251] find that convolutional operation is the most time costly in most DNNs. Consequently, open DNN in layer-grained and merge the same part to design an end-to-end super-resolution plus inference model may speed up the video analytics system.

13.3. Future Direction: Offload intermediate data of DNN

Prior studies [232, 233] take an attempt to leverage the compute resource of camera executing the first several layers in DNN model; they split inference model (*e.g.*, VGG [252], AlexNet [253]) into two parts and offload feature map to the cloud for further DNN inference. However, we argue that the size of feature map actually is far higher than the input size because the depth dimension of tensor is too high. Moreover, video codec are designed to stream YUV format data; its compression efficiency is quite limited on output of convolutional layer (feature map). There exists some work study video/image compression method for DNN inference [209, 232], but so far, all VAPs from academia and industry choose mature video coding standard. But still, compression on feature map, especially the inter-frame feature map encoding is a promising direction to explore; because deeper layers of a DNN represent coarser features about objects in the frame, there is often more similarity and hence more room for compression.

13.4. Future Direction: Asymmetric encoder-decoder architecture

Current standard video encoding applies symmetric encoder-decoder architecture. However, camera and edge/cloud nodes have heterogeneous computational resource; lightweight encoding but heavyweight decoding is more appropriate for video analytics pipeline. In the era of AI and edge computing, it is a good opportunity to carefully design asymmetric encoder-decoder architecture. Some studies [232] combine the compressive sensing and deep learning to design an asymmetric encoder-decoder architecture, but there is still large room to explore.

Chapter 14

Conclusion & Future Prospects

This dissertation presented an analysis of the UAV Monitoring System including the problems, system goals, and identified the open issues. This dissertation studies several vital open issues in detail and reinforced the heterogeneous and evolving Internet architectures with efficient solutions. The key contributions of MoniSys addressed the issues throughout the whole lifecycle of dataflow (*i.e.*, the sensing, communication, and analysis phase) and has holistically improved the performance, fairness, and optimization challenges.

Contents

14.1. Dissertation summary

We started with analyzing the requirements of each phases as well as the goals of MoniSys, then presented the challenges of each phases and the impacts among each other. After that, we focus on each specific problems.

First, we studied the general and case-by-case challenges associated with waypoint planning of UAVs to sensing high-quality data and proposed a general algorithm design schema which generate approximate algorithm with performance bound. We also applied this schema on three different scenarios/sensing models, 3D directional sensing, 2D anisotropic sensing, and multi-adjustable camera sensing. Each scenario causes different challenges, however, this general schema works always well on various variation of waypoint planning problem. The functionalities facilitate towards realizing the high-quality data sensing in sensing phase of MoniSys.

Second, we studied the characteristics of QoE in mobile video streaming and presented an easy-deployment and high-compatibility end-to-end plug-in module achieving QoE fairness with a shared bottleneck bandwidth. We builded our module beyond DASH framework without affecting original mechanism but help sever take co-ordinated decisions with clients to achieve QoE-fairness. The functionalities facilitate towards realizing the efficient data delivery in communication phase of MoniSys.

Finally, we preliminary studied the video analytics pipeline and developed a novel decoder for video analytics achieving holistic tradeoff among accuracy, bandwidth, and latency in MoniSys. We comprehensively analyzed the limitation of prior work on video analytics as well as argument reality/virtual reality/mixed reality and did experiments to explore the potential improvement. Beyond the experiment results, we took our first step on improving the existing decoder to enhance the performance of video analytics with free codec information. The functionalities facilitate towards realizing the performance optimization in analysis phase and also holistic MoniSys.

All the system work (VSiM and AccDecoder) corresponding source code and platform implementations are/will be[19] open-sourced and made available online.

Part II (VSiM):
`https://github.com/VSiM-QUIC/VSiM-QUIC`

Part III (AccDecoder):
`To Be Determined`

14.2. Dissertation impact

The contents in MoniSys have been published in the following peer-reviewed journals and conference proceedings:

Preliminary versions of Chapters in part I (*i.e.*, Addressing Sensing Challenge in UAV Monitoring System: Waypoint Planning) appear in the paper:

[19] AccDecoder will be released after we clean up our codebase. Now it is a bit messy.

(i) **PANDA** [103]- **Weijun Wang**, Haipeng Dai, Chao Dong, Xiao Cheng, Xiaoyu Wang, Panlong Yang, Guihai Chen, Wanchun Dou. Placement of Unmanned Aerial Vehicles for Directional Coverage in 3D Space. **IEEE/ACM Transactions on Networking (ToN)**, Volume: 28, Issue: 2, 2020.

(ii) **VISIT** [104]-**Weijun Wang**, Haipeng Dai, Chao Dong, Fu Xiao, Jiaqi Zheng, Xiao Cheng, Guihai Chen, Xiaoming Fu. Deployment of Unmanned Aerial Vehicles for Anisotropic Monitoring Tasks. **IEEE Transactions on Mobile Computing (TMC)**, Volume: 21, Issue: 2, 2022.

(iii) **VISIT**- **Weijun Wang**, Haipeng Dai, Yue Zhao, Chao Dong, Bangbang Ren, Guihai Chen, Xiaoming Fu. WiPlan: Waypoints Planning for Adjustable Multi-camera UAVs. *Submit to of the 30th IEEE International Conference on Network Protocols (**ICNP**), Lexington, Kentucky, USA, October 30 - November 2, 2022* [under review].

Preliminary versions of Chapters in part II (*i.e.*, Addressing Sensing Challenge in UAV Monitoring System: Study on Scaling Mobile Clients) appear in the paper:

(vi) **VSiM** [237]- Yali Yuan*, **Weijun Wang***, Yuhan Wang, Sripriya Srikant Adhatarao, Bangbang Ren, Kai Zheng and Xiaoming Fu. VSiM: Improving QoE Fairness for Video Streaming in Mobile Environments. *In Proceedings of IEEE International Conference on Computer Communications (**INFOCOM**), Virtual, May 2-5, 2022. (***Co-first author.** Contribute Equally.).*

(vii) [Extended version of VSiM]- Yali Yuan*, **Weijun Wang***, Yuhan Wang, Sripriya Srikant Adhatarao, Bangbang Ren, Kai Zheng and Xiaoming Fu. VSiM: Improving QoE Fairness for Video Streaming in Mobile Environments. *Submit to IEEE/ACM Transactions on Networking (ToN). (***Co-first author.** Contribute Equally.)* [under review].

Preliminary versions of Chapters in part III (*i.e.*, Addressing Sensing Challenge in UAV Monitoring System: Preliminary study on Video Analytics Pipeline) appear in the paper:

(viii) **AccDecoder**- Tingting Yuan*, **Weijun Wang***, Liang Mi, Haipeng Dai, and Xiaoming Fu. AccDecoder: Accelerated Decoding for Neural-enhanced Video Analytics. *Submit to IEEE International Conference on Computer Communications (**INFOCOM**). (***Co-first author.** Contribute Equally.)* [under review].

14.3. Future prospects

This dissertation has tried to address a few of problems in MoniSys, namely **P1:** waypoint planning in sensing phase, **P2:** mobility impact in communication phase, **P3:** latency-bandwidth-accuracy tradeoff in video analytics, towards our goal of making the system high-accuracy (*i.e.*, high-quality), fast (*i.e.*, low latency), and network-friendly (*i.e.*, low bandwidth cost) in section Sec. 1.3.

14.3.1. Extension to the current work

In MoniSys, we have observed the evolution of UAVs, video streaming architecture, and backend analysis method; then, we provided efficient solutions to address the many identified problems. The development of the camera and photography offers more and more clear monitoring data, however at the same time, it also causes serious requirement of network and analysis techniques. Moreover, the growing number of UAVs makes this issue more serious. These requirements introduce many new challenges like scalability *etc.* The scaling analysis streaming is already a concern and hence there is a growing consensus to utilize correlation graph [254,255] among video data to fuse and reduce the data delivery in UAV monitoring network. However, current work focus on surveillance application (especially on road traffic application) with fixed camera. Our UAVs are moving all the time and hence makes the problem more complex. Fortunately, the trajectories of UAVs are planned by ourselves and this may help us.

14.3.2. Broader Future Directions

The emerging new applications and technologies offers us great design space to improve the performance of MoniSys:

- Edge computing,
- Federated learning,
- Machine learning compiling

With the paradigm of edge computing, MoniSys may view server UAVs as edge node and offload some light-weighted task (*e.g.*, neighboring video data fusion and unimportant data filtering) to UAVs. On one hand, this may reduce the data feeding into network and thus ease the pressure of network; on the other hand, UAVs have not much but enough computation power to complete some tasks, taking full advantage of UAV resource also benefits backend analysis.

Federated learning is one of the best way to overcome the issue of data privacy and long training time of DNN. Its distributed architecture very well fits UAV monitoring system, in particular, offline training model for above light-weighted tasks on server and broadcast to UAVs for online executing. Moreover, continues federated learning on sensing data by UAVs is one of the most attractive way to tackle data drift (its meaning see here 12.1.2) in our opinion.

Machine learning compiling (MLC) [256] exploits many opportunities/tools for UAVs-Server collaborative analysis by cutting DNN into several pieces. Prior approaches (*e.g.*, [233]) cut DNN layer by layer leading to a suboptimal solution, but layer fusion (or called operator fusion) in MLC fuse layer into kernel which optimize the memory allocation as well as latency. We argue that cut DNN over kernel-level and deploying on UAVs and servers is the best choice. In addition, multiple analysis tasks may execute on server at the same time. Their contention of computation resource may occur additional latency and resource waste without careful scheduling [257, 258], while MLC offers us a good way to figure out this.

Bibiography

Bibliography

[1] S. He et al. Full-view area coverage in camera sensor networks: Dimension reduction and near-optimal solutions. *IEEE Transactions on vehicular technology*, 2016.

[2] X. Gao et al. Optimization of full-view barrier coverage with rotatable camera sensors. In *IEEE ICDCS*, pages 870–879, 2017.

[3] Samira Hayat et al. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Communications Surveys & Tutorials*, 2016.

[4] Lav Gupta et al. Survey of important issues in UAV communication networks. *IEEE Communications Surveys & Tutorials*, 2016.

[5] Y. Zeng et al. Wireless communications with unmanned aerial vehicles opportunities and challenges.pdf. *IEEE Communication Magzine*, 2016.

[6] Hamid Menouar et al. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 2017.

[7] https://www.dji.com/phantom-4-adv/info.

[8] Dji matrice 200 and 300 rtk. `https://www.dji.com/products`, 2018.

[9] Skydio r1. https://www.diyphotography.net, 2018.

[10] Waldo xcam system. http://www.waldoair.com/xcam-ultra.html.

[11] Dji waypoint. https://www.dji.com/search?q=waypoint.

[12] Aghahan Hamid et al. *Multi-Camera Networks.* Elsevier, 2009.

[13] C. Shen et al. A multi-camera surveillance system that estimates quality-of-view measurement. In *IEEE ICIP*, 2007.

[14] R Garey Michael et al. Computers and intractability: a guide to the theory of np-completeness. *WH Free. Co., San Fr*, 1979.

[15] Matthew P. Johnson et al. Pan and scan: Configuring cameras for coverage. In *IEEE INFOCOM*, 2011.

[16] Huan Wang, Kui Wu, Jianping Wang, and Guoming Tang. Rldish: Edge-assisted qoe optimization of http live streaming with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020.

[17] Yinjie Zhang, Yuanxing Zhang, Yi Wu, Yu Tao, Kaigui Bian, Pan Zhou, Lingyang Song, and Hu Tuo. Improving quality of experience by adaptive video streaming with super-resolution. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020.

[18] Sa'di Altamimi and Shervin Shirmohammadi. Qoe-fair dash video streaming using server-side reinforcement learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2s):1–21, 2020.

[19] Imen Triki, Majed Haddad, Rachid El-Azouzi, Afef Feki, and Marouen Gachaoui. Context-aware mobility resource allocation for qoe-driven streaming services. In *2016 IEEE Wireless Communications and Networking Conference*, pages 1–6. IEEE, 2016.

[20] Abbas Mehrabi, Matti Siekkinen, and Antti Ylä-Jääski. Joint optimization of qoe and fairness through network assisted adaptive mobile video streaming. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8. IEEE, 2017.

[21] Jian Qiao, Yejun He, and Xuemin Sherman Shen. Proactive caching for mobile video streaming in millimeter wave 5g networks. *IEEE Transactions on Wireless Communications*, 15(10):7187–7198, 2016.

[22] Sergio Cicalo, Nesrine Changuel, Velio Tralli, Bessem Sayadi, Frederic Faucheux, and Sylvaine Kerboeuf. Improving qoe and fairness in http adaptive streaming over lte network. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(12):2284–2298, 2015.

[23] Michael Seufert, Nikolas Wehner, Pedro Casas, and Florian Wamser. A fair share for all: Novel adaptation logic for qoe fairness of http adaptive video streaming. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 19–27. IEEE, 2018.

[24] Li Li, Ke Xu, Tong Li, Kai Zheng, Chunyi Peng, Dan Wang, Xiangxiang Wang, Meng Shen, and Rashid Mijumbi. A measurement study on multi-path tcp with multiple cellular carriers on high speed rails. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 161–175, 2018.

[25] Sheng-Hong Lin, Youyun Xu, and Jin-Yuan Wang. Coverage analysis and optimization for high-speed railway communication systems with narrow-strip-shaped cells. *IEEE Transactions on Vehicular Technology*, 2020.

[26] Joe Yuen, Kam-Yiu Lam, and Edward Chan. A fair and adaptive scheduling protocol for video stream transmission in mobile environment. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 409–412. IEEE, 2002.

[27] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020.

[28] Sanae Rosen, Bo Han, Shuai Hao, Z Morley Mao, and Feng Qian. Push or request: An investigation of http/2 server push for improving mobile performance. In *Proceedings of the 26th International Conference on World Wide Web*, pages 459–468, 2017.

[29] Hung T Le, Thoa Nguyen, Nam Pham Ngoc, Anh T Pham, and Truong Cong Thang. Http/2 push-based low-delay live streaming over mobile networks with stream termination. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9):2423–2427, 2018.

[30] Tiffany Yu-Han Chen, Lenin Ravindranath, Shuo Deng, Paramvir Bahl, and Hari Balakrishnan. Glimpse: Continuous, Real-Time Object Recognition on Mobile Devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems - SenSys '15*, pages 155–168, Seoul, South Korea, 2015. ACM Press.

[31] Yuanqi Li, Arthi Padmanabhan, Pengzhan Zhao, Yufei Wang, Guoqing Harry Xu, and Ravi Netravali. Reducto: On-Camera Filtering for Resource-Efficient Real-Time Video Analytics. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '20, pages 359–376, Virtual Event, USA, July 2020. Association for Computing Machinery.

[32] Ran Xu, Chen-lin Zhang, Pengcheng Wang, Jayoung Lee, Subrata Mitra, Somali Chaterji, Yin Li, and Saurabh Bagchi. ApproxDet: content and contention-aware approximate object detection for mobiles. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 449–462, Virtual Event Japan, November 2020. ACM.

[33] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia. NoScope: optimizing neural network queries over video at scale. *Proceedings of the VLDB Endowment*, 10(11):1586–1597, August 2017.

[34] Focus: Querying Large Video Datasets with Low Latency and Low Cost.

[35] Luyang Liu, Hongyu Li, and Marco Gruteser. Edge Assisted Real-time Object Detection for Mobile Augmented Reality. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, pages 1–16, Los Cabos, Mexico, August 2019. Association for Computing Machinery.

[36] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *NIPS*, pages 91–99, 2015.

[37] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *2010 20th International Conference on Pattern Recognition*, pages 2756–2759, 2010.

[38] Juheon Yi, Sunghyun Choi, and Youngki Lee. EagleEye: wearable camera-based person identification in crowded urban spaces. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, London United Kingdom, April 2020. ACM.

[39] J. Yi, S. Kim, J. Kim, and S. Choi. Supremo: Cloud-Assisted Low-Latency Super-Resolution in Mobile Devices. *IEEE Transactions on Mobile Comput-*

ing, pages 1–1, 2020. Conference Name: IEEE Transactions on Mobile Computing.

[40] Jinrui Zhang, Deyu Zhang, Xiaohui Xu, Fucheng Jia, Yunxin Liu, Xuanzhe Liu, Ju Ren, and Yaoxue Zhang. MobiPose: real-time multi-person pose estimation on mobile devices. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 136–149, Virtual Event Japan, November 2020. ACM.

[41] Wuyang Zhang, Zhezhi He, Luyang Liu, Zhenhua Jia, Yunxin Liu, Marco Gruteser, Dipankar Raychaudhuri, and Yanyong Zhang. Elf: Accelerate high-resolution mobile deep vision with content-aware parallel offloading. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom '21, pages 201–214, New York, NY, USA, 2021. Association for Computing Machinery.

[42] Ben Zhang, Xin Jin, Sylvia Ratnasamy, John Wawrzynek, and Edward A. Lee. AWStream: adaptive wide-area streaming analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 236–252, Budapest Hungary, August 2018. ACM.

[43] Kuntai Du, Ahsan Pervaiz, Xin Yuan, Aakanksha Chowdhery, Qizheng Zhang, Henry Hoffmann, and Junchen Jiang. Server-Driven Video Streaming for Deep Learning Inference. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 557–570, Virtual Event USA, July 2020. ACM.

[44] Kuntai Du, Qizheng Zhang, Anton Arapin, Haodong Wang, Zhengxu Xia, and Junchen Jiang. Accmpeg: Optimizing video encoding for video analytics. In *Fifth Conference on Machine Learning and Systems*, 2022.

[45] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, pages 253–266, Budapest Hungary, August 2018. ACM.

[46] Yiding Wang, Weiyan Wang, Duowen Liu, Xin Jin, Junchen Jiang, and Kai Chen. Enabling Edge-Cloud Video Analytics for Robotics Applications. page 10, 2021.

[47] Yiding Wang, Weiyan Wang, Junxue Zhang, Junchen Jiang, and Kai Chen. Bridging the Edge-Cloud Barrier for Real-time Advanced Vision Analytics. page 7, 2019.

[48] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E.A.Lee. Awstream: Adaptive wide-area streaming analytics. In *Conference of the ACM Special Interest Group on Data Communication*, 2018.

[49] Zhengdong Zhang and Vivienne Sze. FAST: A Framework to Accelerate Super-Resolution Processing on Compressed Videos. *arXiv:1603.08968 [cs]*, August 2017.

[50] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. Compressed Video Action Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, June 2018. IEEE.

[51] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. FrameExit: Conditional Early Exiting for Efficient Video Recognition. April 2021. arXiv: 2104.13400.

[52] Shiyao Wang, Alibaba Group, Hongchao Lu, and Zhidong Deng. Fast Object Detection in Compressed Video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7103–7112, Seoul, Korea (South), October 2019. IEEE.

[53] Hanxiang Hao, Sriram Baireddy, et al. An attention-based system for damage assessment using satellite imagery.

[54] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile networks and Applications*, 10(4):519–528, 2005.

[55] Huadong Ma et al. On coverage problems of directional sensor networks. In *Mobile Ad-hoc and Sensor Networks*, 2005.

[56] Ertan Onur, Cem Ersoy, Hakan Delic, and Lale Akarun. Surveillance wireless sensor networks: Deployment quality analysis. *IEEE Network*, 21(6):48–53, 2007.

[57] Qianqian Yang, Shibo He, Junkun Li, Jiming Chen, and Youxian Sun. Energy-efficient probabilistic area coverage in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 64(1):367–377, 2015.

[58] Guoliang Xing et al. Data fusion improves the coverage of wireless sensor networks. In *ACM MobiHoc*, 2009.

[59] A. Elfes. Occupancy grids: A stochastic spatial representation for active robot perception, 2013.

[60] Ertan Onur, Cem Ersoy, and Hakan Delicc. How many sensors for an acceptable breach detection probability? *Comput. Commun.*, 29(2):173–182, jan 2006.

[61] S. Tang et al. Qute: Quality-of-monitoring Aware Sensing and Routing Strategy in Wireless Sensor Networks. In *ACM MobiHoc*, 2013.

[62] Y. C. Wang et al. Using Rotatable and Directional (R & D) Sensors to Achieve Temporal Coverage of Objects and Its Surveillance Application. *IEEE TMC*, 2012.

[63] Z. Yu et al. Local face-view barrier coverage in camera sensor networks. In *IEEE INFOCOM*, pages 684–692, 2015.

[64] Zhiyin Chen, Xiaofeng Gao, Fan Wu, and Guihai Chen. A ptas to minimize mobile sensor movement for target coverage problem. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE Press, 2016.

[65] Aveek Purohit, Zheng Sun, and Pei Zhang. Sugarmap: Location-less coverage for micro-aerial sensing swarms. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, IPSN'13, pages 253–264, New York, NY, USA, 2013. Association for Computing Machinery.

[66] Ting-Yu Lin, Hendro Agus Santoso, and Kun-Ru Wu. Global sensor deployment and local coverage-aware recovery schemes for smart environments. *IEEE Transactions on Mobile Computing*, 14(7):1382–1396, 2015.

[67] Yi Wang and Guohong Cao. On full-view coverage in camera sensor networks. In *2011 Proceedings IEEE INFOCOM*, pages 1781–1789, 2011.

[68] Zhibo Wang, Jilong Liao, Qing Cao, Hairong Qi, and Zhi Wang. Achieving k-barrier coverage in hybrid directional sensor networks. *IEEE Transactions on Mobile Computing*, 13(7):1443–1455, 2014.

[69] Y. Wang et al. On full-view coverage in camera sensor networks. In *IEEE INFOCOM*, 2011.

[70] Yi Wang et al. Barrier coverage in camera sensor networks. In *ACM MobiHoc*, 2011.

[71] Y. Hu et al. Critical sensing range for mobile heterogeneous camera sensor networks. In *IEEE INFOCOM*, pages 970–978, 2014.

[72] H. Ma et al. Minimum camera barrier coverage in wireless camera sensor networks. In *IEEE INFOCOM*, pages 217–225, 2012.

[73] Q. Zhang et al. Toward optimal orientation scheduling for full-view coverage in camera sensor networks. In *IEEE GLOBECOM*, 2016.

[74] Xiaolan Liu et al. Achieving full-view barrier coverage with mobile camera sensors. In *IEEE NaNA*, 2016.

[75] Y. Wu et al. Achieving full view coverage with randomly-deployed heterogeneous camera sensors. In *IEEE ICDCS*, 2012.

[76] R. Yang et al. Distributed algorithm for full-view barrier coverage with rotatable camera sensors. In *IEEE GLOBECOM*, 2015.

[77] Ertan Onur et al. How many sensors for an acceptable breach detection probability? *Elsevier Comput. Commun.*, 2006.

[78] Q. Yang et al. Energy-efficient probabilistic area coverage in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 2015.

[79] Giordano Fusco et al. Selection and orientation of directional sensors for coverage maximization. In *IEEE SECON*, 2009.

[80] J. Tao et al. A quality-enhancing coverage scheme for camera sensor networks. In *IECON*, 2017.

[81] Ahmed Saeed et al. Argus: realistic target coverage by drones. In *ACM IPSN*, 2017.

[82] Chaoyang Li et al. On k-full-view-coverage-algorithms in camera sensor networks. In *IEEE ICCC*, 2016.

[83] Chien-Fu Cheng et al. Distributed barrier coverage in wireless visual sensor networks with β-QoM. *IEEE Sensors Journal*.

[84] C. F. Cheng et al. Barrier coverage in wireless visual sensor networks with importance of image consideration. In *IEEE ICUFN*, 2015.

[85] Ling Guo et al. Enhancing barrier coverage with β quality of monitoring in wireless camera sensor networks. *Ad Hoc Networks*, 2016.

[86] Xiangmao Chang, Rui Tan, Guoliang Xing, Zhaohui Yuan, Chenyang Lu, Yixin Chen, and Yixian Yang. Sensor placement algorithms for fusion-based surveillance networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(8):1407–1414, 2011.

[87] M. M. et al. 3D human pose estimation model using location-maps for distorted and disconnected images by a wearable omnidirectional camera. *Transactions on Computer Vision and Applications*, 2020.

[88] Fakhreddine Ababsa et al. 3D Human Tracking with Catadioptric Omnidirectional Camera. In *ACM ICMR*, 2019.

[89] Senthil Yogamani et al. WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving. In *IEEE ICCV*, 2019.

[90] Niels Joubert et al. Towards a Drone Cinematographer: Guiding Quadrotor Cameras using Visual Composition Principles. *CoRR*, 2016.

[91] Tobias Nageli et al. Real-time Planning for Automated Multi-view Drone Cinematography. *ACM Transactions on Graphics*, 2017.

[92] Ting-Yu Lin et al. Enhanced deployment algorithms for heterogeneous directional mobile sensors in a bounded monitoring area. *IEEE Transactions on Mobile Computing*, 16, 2017.

[93] H. Ma et al. A coverage-enhancing method for 3d directional sensor networks. In *IEEE INFOCOM*, pages 2791–2795, 2009.

[94] X. Yang and other. 3d visual correlation model for wireless visual sensor networks. In *IEEE ICIS*, pages 75–80, 2017.

[95] C. Han et al. An Energy Efficiency Node Scheduling Model for Spatial-Temporal Coverage Optimization in 3D Directional Sensor Networks. *IEEE Access*, 2016.

[96] X. Yang and other. 3-D Application-Oriented Visual Correlation Model in Wireless Multimedia Sensor Networks. *IEEE Sensors Journal*, 2017.

[97] Pengju Si et al. Barrier coverage for 3d camera sensor networks. *Sensors*, 2017.

[98] M. Hosseini et al. Sensor selection and configuration in visual sensor networks. In *IST*, 2012.

[99] Li Yupeng et al. A virtual potential field based coverage-enhancing algorithm for 3d directional sensor networks. In *ISSDM*, 2012.

[100] J. Peng et al. A coverage detection and re-deployment algorithm in 3d directional sensor networks. In *CCDC*, 2015.

[101] J. Peng et al. A coverage-enhance scheduling algorithm for 3d directional sensor networks. In *CCDC*, 2013.

[102] P. R. et al. Optimal Camera Placement for Motion Capture Systems. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[103] Weijun Wang, Haipeng Dai, Chao Dong, Xiao Cheng, Xiaoyu Wang, Panlong Yang, Guihai Chen, and Wanchun Dou. Placement of unmanned aerial vehicles for directional coverage in 3d space. *IEEE/ACM transactions on networking*, 28(2):888–901, 2020.

[104] Weijun Wang, Haipeng Dai, Chao Dong, Fu Xiao, Jiaqi Zheng, Xiao Cheng, Guihai Chen, and Xiaoming Fu. Deployment of unmanned aerial vehicles for anisotropic monitoring tasks. *IEEE Transactions on Mobile Computing*, 21(2):495–513, 2022.

[105] Giordano Fusco et al. Placement and Orientation of Rotating Directional Sensors. In *IEEE SECON*, 2010.

[106] Vikram P. Munishwar et al. Coverage management for mobile targets in visual sensor networks. In *ACM MSWiM*, 2012.

[107] Fenghua Li et al. HideMe: Privacy-Preserving Photo Sharing on Social Networks. In *IEEE INFOCOM*, 2019.

[108] Mac Schwager et al. Eyes in the Sky: Decentralized Control for the Deployment of Robotic Camera Networks. *Proc. IEEE*, 2011.

[109] S. Aghajanzadeh et al. Camera Placement Meeting Restrictions of Computer Vision. In *IEEE ICIP*, 2020.

[110] S. Jun et al. Camera Placement in Smart Cities for Maximizing Weighted Coverage With Budget Limit. *IEEE Sensors Journal*, 2017.

[111] Ajay Kaushik et al. A Grey Wolf Optimization Based Algorithm for Optimum Camera Placement. *Wireless Personal Communications*, 2019.

[112] A. A. Altahir et al. Visual Sensor Placement Based on Risk Maps. *IEEE Transactions on Instrumentation and Measurement*, 2020.

[113] Xincong Yang et al. Computer-Aided Optimization of Surveillance Cameras Placement on Construction Sites. *Wiley CACIE*, 2018.

[114] F. Jiang et al. Distributed Optimization of Visual Sensor Networks for Coverage of a Large-scale 3-D Scene. *IEEE Transactions on Mechatronics*, 2020.

[115] Aaron Mavrinac et al. Modeling Coverage in Camera Networks: A Survey. *Springer International Journal of Computer Vision*, 2013.

[116] M. Amac Guvensan et al. On coverage issues in directional sensor networks: A survey. *Ad Hoc Networks*, 2011.

[117] E. Koyuncu. Performance Gains of Optimal Antenna Deployment in Massive MIMO Systems. *IEEE Transactions on Wireless Communications*, 2018.

[118] Y. Alsaba et al. Beamforming in Wireless Energy Harvesting Communications Systems: A Survey. *IEEE Communications Surveys Tutorials*, 2018.

[119] R. Zhang et al. MIMO Broadcasting for Simultaneous Wireless Information and Power Transfer. *IEEE Transactions on Wireless Communications*, 2013.

[120] Q. Xu et al. Waveforming: An Overview With Beamforming. *IEEE Communications Surveys Tutorials*, 2018.

[121] Stefan Dobrev et al. strong connectivity in sensor networks with given number of directional antenna of bounded angle. 2012.

[122] T. Tran et al. Symmetric Connectivity Algotirthms in Multiple Directional Antennas Wireless Sensor Networks. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018.

[123] C. Cheng, Y. Adulyasak, and L. M. Rousseau. Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B Methodological*, 139:364–387, 2020.

[124] C. C. Murray and A. G. Chu. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C Emerging Technologies*, 54(may):86–109, 2015.

[125] H. Y. Jeong, B. D. Song, and S. Lee. Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones. *International Journal of Production Economics*, 214(AUG.):220–233, 2019.

[126] Z. Wang and J. B. Sheu. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122(APR.):350–364, 2019.

[127] Dyutimoy Nirupam Das, Rohan Sewani, Junwei Wang, and Manoj Kumar Tiwari. Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–11, 2020.

[128] D. Wang, P. Hu, J. Du, P. Zhou, and M. Hu. Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones. *IEEE Internet of Things Journal*, PP(99):1–1, 2019.

[129] Chuan. Wang and Hongjie. Lan. An expressway based tsp model for vehicle delivery service coordinated with truck + uav. In *IEEE SMC*, pages 307–311, 2019.

[130] H. Ghazzai, H. Menouar, A. Kadri, and Y. Massoud. Future uav-based its: A comprehensive scheduling framework. *IEEE Access*, PP(99):1–1, 2019.

[131] Tatsuaki Kimura and Masaki Ogura. Distributed collaborative 3d-deployment of uav base stations for on-demand coverage. In *IEEE INFOCOM*, pages 1748–1757, 2020.

[132] Chi Harold Liu, Chengzhe Piao, and Jian Tang. Energy-efficient uav crowdsensing with multiple charging stations by deep learning. In *IEEE INFOCOM*, pages 199–208, 2020.

[133] M. Patchou, B. Sliwa, and C. Wietfeld. Unmanned aerial vehicles in logistics: Efficiency gains and communication performance of hybrid combinations of ground and aerial vehicles. In *IEEE VNC*, pages 1–8, 2019.

[134] A. Trotta, F. D. Andreagiovanni, MD Felice, E. Natalizio, and K. R. Chowdhury. When uavs ride a bus: Towards energy-efficient city-scale video surveillance. In *IEEE INFOCOM*, pages 1043–1051, 2018.

[135] http://dlib.net/files/.

[136] https://github.com/yestinsong/text-detection.

[137] L. Catarinucci et al. An IoT-Aware Architecture for Smart Healthcare Systems. *IEEE Internet of Things Journal*, 2015.

[138] Tian He et al. Range-free Localization Schemes for Large Scale Sensor Networks. In *ACM MOBICOM*. ACM, 2003.

[139] K. Akkaya et al. IoT-based occupancy monitoring techniques for energy-efficient smart buildings. In *IEEE WCNC)*, 2015.

[140] Z. Chen et al. A Localization Method for the Internet of Things. *Springer The Journal of Supercomputing*, 2013.

[141] Jeanine D.S. Euler's gem: The polyhedron formula and the birth of topology. *Elsevier*, 6, 2008.

[142] Satoru Fujishige. *Submodular functions and optimization.* 2005.

[143] https://enterprise.dji.com/energy.

[144] Wang Xiaoyu et al. Robust scheduling for wireless charger networks. In *IEEE INFOCOM*, 2019.

[145] Dai Haipeng et al. Radiation constrained scheduling of wireless charging tasks. In *ACM MobiHoc*, 2017.

[146] S. Bhattacharya et al. Multi-application deployment in shared sensor networks based on quality of monitoring. In *IEEE RTAS*, 2010.

[147] Ian F Akyildiz et al. On exploiting spatial and temporal correlation in wireless sensor networks. *Wiopt Modeling* & *Optimization in Mobile Ad Hoc* & *Wireless Networking*, 2004.

[148] Mehmet C. Vuran et al. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Elsevier Computer Networks*, 2004.

[149] T. Cover et al. *Elements of information theory.* Wiley, 1991.

[150] G. Carlos et al. Near-optimal sensor placements in gaussian processes. In *ACM ICML*, 2005.

[151] A. Krause et al. Near-optimal sensor placements: maximizing information while minimizing communication cost. In *ACM IPSN*, 2006.

[152] https://www.dji.com/phantom-4-adv/info#specs.

[153] S. Tang et al. Morello: A quality-of-monitoring oriented sensing scheduling protocol in sensor networks. In *IEEE INFOCOM*, 2012.

[154] S. Tang et al. DAMson: On distributed sensing scheduling to achieve high quality of monitoring. In *IEEE INFOCOM*, 2013.

[155] R. Tan et al. Exploiting data fusion to improve the coverage of wireless sensor networks. *IEEE/ACM Transactions on Networking.*

[156] M. Kan et al. Multi-view deep network for cross-view classification. In *IEEE CVPR*, 2016.

[157] R. Jacobson. The mannual of photography. *Focal Press*, 2000.

[158] Y. Wu et al. Photo crowdsourcing for area coverage in resource constrained environments. In *IEEE INFOCOM*, 2017.

[159] Yi Wang et al. SmartPhoto: a resource-aware crowdsourcing approach for image sensing with smartphones. In *ACM MobiHoc*, 2014.

[160] V. Blanz et al. Face recognition based on frontal views generated from non-frontal images. In *IEEE CVPR*, volume 2, pages 454–461 vol. 2, 2005.

[161] R. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, 1972.

[162] Vasek Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 1979.

[163] Chengjie Wu et al. Submodular game for distributed application allocation in shared sensor networks. In *IEEE INFOCOM*, 2012.

[164] Patricio Ramírez-Correa, Francisco Javier Rondán-Cataluña, Jorge Arenas-Gaitán, and Félix Martín-Velicia. Analysing the acceptation of online games in mobile devices: An application of utaut2. *Journal of Retailing and Consumer Services*, 50:85–93, 2019.

[165] Zhou Su, Qichao Xu, Fen Hou, Qing Yang, and Qifan Qi. Edge caching for layered video contents in mobile social networks. *IEEE Transactions on Multimedia*, 19(10):2210–2221, 2017.

[166] Global Mobile Data Traffic Forecast. Cisco visual networking index: global mobile data traffic forecast update, 2017–2022. *Update*, 2017:2022, 2019.

[167] Haakon Riiser, Tore Endestad, Paul Vigmostad, Carsten Griwodz, and Pâl Halvorsen. Video streaming using a location-based bandwidth-lookup service for bitrate planning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 8(3):1–19, 2012.

[168] Jeffrey G Andrews. Seven ways that hetnets are a cellular paradigm shift. *IEEE communications magazine*, 51(3):136–144, 2013.

[169] Yusuf Sani, Andreas Mauthe, and Christopher Edwards. Adaptive bitrate selection: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2985–3014, 2017.

[170] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: evidence from a large video streaming service. In *In Proc. of ACM SIGCOMM*, pages 187–198. ACM, 2014.

[171] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 325–338, 2015.

[172] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *In Proc. of ACM SIGCOMM*, pages 197–210, 2017.

[173] Mo Dong, Tong Meng, Doron Zarchy, Engin Arslan, Yossi Gilad, Brighten Godfrey, and Michael Schapira. PCC vivace: Online-learning congestion control. In *In Proc. of USENIX NSDI*, 2018.

[174] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS operating systems review*, 42(5):64–74, 2008.

[175] Mark Allman, Vern Paxson, Wright Stevens, et al. Tcp congestion control. 2009.

[176] Junchen Jiang, Vyas Sekar, and Hui Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with FESTIVE. In *In Proc. of CoNext*, pages 97–108, 2012.

[177] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C. Begen, and David Oran. Probe and adapt: Rate adaptation for HTTP video streaming at scale. *IEEE J. Sel. Areas Commun.*, 32(4):719–733, 2014.

[178] Xiaoqi Yin, Mihovil Bartulovic, Vyas Sekar, and Bruno Sinopoli. On the efficiency and fairness of multiplayer http-based adaptive video streaming. In *In Proc. of IEEE ACC*, 2017.

[179] Abdelhak Bentaleb, Ali C. Begen, Saad Harous, and Roger Zimmermann. Want to play dash?: a game theoretic approach for adaptive streaming over

HTTP. In Pablo César, Michael Zink, and Niall Murray, editors, *In Proc. of ACM MMSys*, 2018.

[180] Vikram Nathan, Vibhaalakshmi Sivaraman, Ravichandra Addanki, Mehrdad Khani, Prateesh Goyal, and Mohammad Alizadeh. End-to-end transport for video qoe fairness. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 408–423. 2019.

[181] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hoßfeld, and Phuoc Tran-Gia. A survey on quality of experience of http adaptive streaming. *IEEE Communications Surveys & Tutorials*, 17(1):469–492, 2014.

[182] Jim Summers, Tim Brecht, Derek Eager, and Alex Gutarin. Characterizing the workload of a netflix streaming video server. In *2016 IEEE International Symposium on Workload Characterization (IISWC)*, pages 1–12. IEEE, 2016.

[183] Hyunwoo Nam, Bong Ho Kim, Doru Calin, and Henning Schulzrinne. A mobile video traffic analysis: Badly designed video clients can waste network bandwidth. In *2013 IEEE Globecom Workshops (GC Wkshps)*, pages 506–511. IEEE, 2013.

[184] Kanthi Nagaraj, Dinesh Bharadia, Hongzi Mao, Sandeep Chinchali, Mohammad Alizadeh, and Sachin Katti. Numfabric: Fast and flexible bandwidth allocation in datacenters. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 188–201, 2016.

[185] Xiaohu Ge, Junliang Ye, Yang Yang, and Qiang Li. User mobility evaluation for 5g small cell networks based on individual mobility model. *IEEE Journal on Selected Areas in Communications*, 34(3):528–541, 2016.

[186] Xingqin Lin, Radha Krishna Ganti, Philip J Fleming, and Jeffrey G Andrews. Towards understanding the fundamentals of mobility in cellular networks. *IEEE Transactions on Wireless Communications*, 12(4):1686–1698, 2013.

[187] Bezalel Gavish and Suresh Sridhar. The impact of mobility on cellular network configuration. *Wireless Networks*, 7(2):173–185, 2001.

[188] Waleed Alasmary and Weihua Zhuang. Mobility impact in ieee 802.11 p infrastructureless vehicular networks. *Ad Hoc Networks*, 10(2):222–230, 2012.

[189] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 2, pages 825–835. IEEE, 2003.

[190] Jukka Manner, Alberto Lopéz Toledo, Andrej Mihailovic, Héctor L Velayos Munoz, Eleanor Hepworth, and Youssef Khouaja. Evaluation of mobility and quality of service interaction. *Computer Networks*, 38(2):137–163, 2002.

[191] Jeroen Van Der Hooft, Stefano Petrangeli, Tim Wauters, Rafael Huysegems, Tom Bostoen, and Filip De Turck. An http/2 push-based approach for low-latency live streaming with super-short segments. *Journal of Network and Systems Management*, 26(1):51–78, 2018.

[192] Neural Networks for Multi-Output Regression. `https://machinelearningmastery.com/deep-learning-models-for-multi-output-regression/`.

[193] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[194] dash.js. `https://github.com/Dash-Industry-Forum/dash.js`.

[195] A QUIC implementation in pure Go . `https://github.com/lucas-clemente/quic-go`.

[196] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic adaptive streaming over http dataset. In *Proceedings of the 3rd multimedia systems conference*, pages 89–94, 2012.

[197] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: a new tcp-friendly high-speed tcp variant. *ACM SIGOPS operating systems review*, 42(5):64–74, 2008.

[198] Railway mobility model. `https://en.wikipedia.org/wiki/Metronom_Eisenbahngesellschaft`. Online.

[199] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Ya Xu, and Haobo Yu. Advances in network simulation. *Computer*, 33(5):59–67, May 2000.

[200] Yixue Hao, Min Chen, Long Hu, Jeungeun Song, Mojca Volk, and Iztok Humar. Wireless fractal ultra-dense cellular networks. *Sensors*, 17(4):841, 2017.

[201] One legacy of tiananmen: China's 100 million surveillance cameras. `https://blogs.wsj.com/chinarealtime/2014/06/05/one-legacy-of-tiananmen-chinas-100-million\protect\discretionary{\char\hyphenchar\font}{}{}surveillance-cameras4`.

[202] Can 30,000 cameras help solve chicago's crime problem? `https://www.nytimes.com/2018/05/26/us/chicago-police-surveillance.html`.

[203] One surveillance camera for every 11 people in britain, says cctv survey. `https://www.telegraph.co.uk/technology/10172298/One-surveillance-camera-for-every-11-people\protect\discretionary{\char\hyphenchar\font}{}{}in-Britain-says-CCTV-survey.html`.

[204] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

[205] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet: Real-time video object segmentation. *CoRR*, abs/2102.04604, 2021.

[206] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140, 2017.

[207] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *European Conference on Computer Vision (ECCV)*, 2018.

[208] Tan Zhang, Aakanksha Chowdhery, Paramvir (Victor) Bahl, Kyle Jamieson, and Suman Banerjee. The Design and Implementation of a Wireless Video Surveillance System. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*, pages 426–438, Paris, France, 2015. ACM Press.

[209] Xiufeng Xie and Kyu-Han Kim. Source Compression with Bounded DNN Perception Loss for IoT Edge Computer Vision. In *The 25th Annual International*

Conference on Mobile Computing and Networking, MobiCom '19, pages 1–16, Los Cabos, Mexico, October 2019. Association for Computing Machinery.

[210] Marwa Meddeb. Region-of-interest-based video coding for video conference applications. page 172.

[211] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[212] Zheng Hui, Xiumei Wang, and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[213] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[214] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[215] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object Detection in 20 Years: A Survey. *arXiv:1905.05055 [cs]*, May 2019. arXiv: 1905.05055.

[216] Gpu prices are falling, and you should buy now. `https://venturebeat.com/2018/04/27/gpu-prices-are-falling-and-you-should-buy-now/`.

[217] Hyunho Yeo, Chan Ju Chong, Youngmok Jung, Juncheol Ye, and Dongsu Han. Nemo: Enabling neural-enhanced video streaming on commodity mobile devices. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking, (Mobicom'20)*, New York, NY, USA, 2020. Association for Computing Machinery.

[218] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. Neural adaptive content-aware internet video delivery. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 645–661. USENIX Association, October 2018.

[219] Jaehong Kim, Youngmok Jung, Hyunho Yeo, Juncheol Ye, and Dongsu Han. Neural-Enhanced Live Streaming: Improving Live Video Ingest via Online Learning. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 107–125, Virtual Event USA, July 2020. ACM.

[220] Ganesh Ananthanarayanan, Paramvir Bahl, Peter Bodik, Krishna Chintalapudi, Matthai Philipose, Lenin Ravindranath, and Sudipta Sinha. Real-time video analytics: The killer app for edge computing. *Computer*, 50(10):58–67, 2017.

[221] Xiaochen Liu, Pradipta Ghosh, Oytun Ulutan, B. S. Manjunath, Kevin Chan, and Ramesh Govindan. Caesar: cross-camera complex activity recognition. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, pages 232–244, New York New York, November 2019. ACM.

[222] Apple. Siri. `https://www.apple.com/siri/`.

[223] Dnncam. `https://groupgets.com/campaigns/429-dnncam-ai-camera`.

[224] Nvidea tx2. `https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/`.

[225] Wyze. wyze camera. `https://www.safehome.org/homesecurity-cameras/wyze/`.

[226] Axis for a safety touch at the Grey Cup Festival. Axis. `https://www.germanprotect.com/videoueberwachung/analoge-ueberwachungskameras.html`.

[227] H.264 specification. `https://www.itu.int/rec/T-REC-H.264`.

[228] Webm official website. `https://www.webmproject.org/`.

[229] Av1 specification. `https://aomedia.org/av1-features/get-started/`.

[230] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J. Freedman. Live video analytics at scale with approximation and Delay-Tolerance. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 377–392, Boston, MA, March 2017. USENIX Association.

[231] Xin Dai, Xiangnan Kong, Tian Guo, and Yixian Huang. Cinet: Redesigning deep neural networks for efficient mobile-cloud collaborative inference. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 459–467. SIAM, 2021.

[232] Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Shengzhong Liu, Huajie Shao, and Tarek Abdelzaher. Deep Compressive Offloading: Speeding Up Neural Network Inference by Trading Edge Computation for Network Latency. page 13, 2020.

[233] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge. In *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 615–629, Xi'an China, April 2017. ACM.

[234] Sifeng Xia, Kunchangtai Liang, Wenhan Yang, Ling-Yu Duan, and Jiaying Liu. An emerging coding paradigm vcm: A scalable coding approach beyond feature and signal. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[235] Bhardwaj Romil, Xia Zhengxu, Ananthanarayanan Ganesh, Jiang Junchen, Shu Yuanchao, Karianakis Nikolaos, Hsieh Kevin, Bahl Paramvir, and Stoica Ion. Ekya: Continuous learning of video analytics models on edge compute servers. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, Renton, WA, April 2022. USENIX Association.

[236] Junhyug Noh, Wonho Bae, Wonhee Lee, Jinhwan Seo, and Gunhee Kim. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9724–9733, 2019.

[237] Yali Yuan, Weijun Wang, Yuhan Wang, Sripriya S. Adhatarao, Bangbang Ren, Kai Zheng, and Xiaoming Fu. Vsim: Improving qoe fairness for video stream-

ing in mobile environments. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pages 1309–1318, 2022.

[238] Fcc broadband bandwidth measurement. `https://www.fcc.gov/reports-research/reports/measuring-broadband-america/raw-data-measuring-broadband-america-eighth`.

[239] Gebhardt insurance traffic cam round trip bike shop. `https://www.youtube.com/watch?v=RNi4CKgZVMY`.

[240] Crossroad. `https://yoda.cs.uchicago.edu/videos/crossroad.mp4`.

[241] Traffic. `https://yoda.cs.uchicago.edu/videos/traffic.mp4`.

[242] Edwin E. Catmull and Raphael Rom. A class of local interpolating splines. *Computer Aided Geometric Design*, pages 317–326, 1974.

[243] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2110–2118, 2016.

[244] H.265 specification. `https://www.itu.int/rec/T-REC-H.265`.

[245] Vp8 standard. `https://datatracker.ietf.org/doc/html/rfc6386`.

[246] Hyunho Yeo, Sunghyun Do, and Dongsu Han. How will deep learning change internet video delivery? In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, HotNets-XVI, pages 57–64, New York, NY, USA, 2017. Association for Computing Machinery.

[247] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[248] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv, 2014.

[249] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *CoRR*, abs/1806.00582, 2018.

[250] Mengwei Xu, Mengze Zhu, Yunxin Liu, Felix Xiaozhu Lin, and Xuanzhe Liu. DeepCache: Principled Cache for Mobile Deep Vision. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, MobiCom '18, pages 129–144, New Delhi, India, October 2018. Association for Computing Machinery.

[251] Loc N. Huynh, Youngki Lee, and Rajesh Krishna Balan. DeepMon: Mobile GPU-based Deep Learning Framework for Continuous Vision Applications. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 82–95, Niagara Falls New York USA, June 2017. ACM.

[252] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[253] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, April 2015. arXiv: 1409.1556.

[254] Samvit Jain, Ganesh Ananthanarayanan, Junchen Jiang, Yuanchao Shu, and Joseph E. Gonzalez. Scaling Video Analytics Systems to Large Camera Deployments. *arXiv:1809.02318 [cs]*, July 2019.

[255] Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David G. Andersen, Michael Kaminsky, and Subramanya R. Dulloor. Scaling Video Analytics on Constrained Edge Nodes. *arXiv:1905.13536 [cs, eess, stat]*, May 2019.

[256] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. TVM: An automated End-to-End optimizing compiler for deep learning. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 578–594, Carlsbad, CA, October 2018. USENIX Association.

[257] Juheon Yi and Youngki Lee. Heimdall: mobile GPU coordination platform for augmented reality applications. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, London United Kingdom, 2020. ACM.

[258] Angela H. Jiang, Daniel L.-K. Wong, Christopher Canel, Lilia Tang, Ishan Misra, Michael Kaminsky, Michael A. Kozuch, Padmanabhan Pillai, David G. Andersen, and Gregory R. Ganger. Mainstream: Dynamic Stem-Sharing for Multi-Tenant video processing. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 29–42, Boston, MA, July 2018. USENIX Association.