ENERGY EFFICIENT VIDEO SENSOR NETWORKS FOR SURVEILLANCE APPLICATIONS

by

Bambang Ali Basyah Sarif

B.Sc. Bandung Institute of Technology, 1999

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

August 2016

© Bambang Ali Basyah Sarif, 2016
Abstract

Video sensor networks (VSNs) provide rich sensing information and coverage, both beneficial for applications requiring visual information such as smart homes, traffic control, healthcare systems and monitoring/surveillance systems. Since a VSN-based surveillance application is usually assumed to have limited resources, energy efficiency has become one of the most important design aspects of such networks. However, unlike common sensor network platforms, where power consumption mostly comes from the wireless transmission, the encoding process in a video sensor network contributes to a significant portion of the overall power consumption.

There is a trade-off between encoding complexity and bitrate in a sense that in order to increase compression performance, i.e., achieve a lower bitrate, a more complex encoding process is necessary. The coding complexity and video bitrate determine the overall encoding and transmission power consumption of a VSN. Thus, choosing the right configuration and setting parameters that lead to optimal encoding performance is of primary importance for controlling power consumption in VSNs. The coding complexity and bitrate also depend on the video content complexity, as spatial details and high motion tend to lead to higher computation costs or increased bitrates. In a video surveillance network, each node captures an event from a different point of view, such that each captured video stream has unique spatial and temporal information.

This thesis investigates the trade-off between encoding complexity and communication power consumption in a video surveillance network where the effect of video encoding parameters, content complexity, and network topology are taken into
consideration. In order to take into account the effect of content complexity, we created a video surveillance dataset consisting of a large number of captured videos with different levels of spatial information and motion. Then, we design an algorithm that minimizes the video surveillance network's power consumption for different scene settings. Models that estimate the coding complexity and bitrate were proposed. Finally, these models were used to minimize the video surveillance network's power consumption and estimate the encoding parameters per each node that yield the minimum possible power consumption for the entire network.
Preface

This thesis presents research conducted by Bambang Ali Basyah Sarif, under the guidance of Dr. Victor Leung and Dr. Panos Nasiopoulos. Some of the research results have been published as journal articles, conference proceedings and/or submitted for peer review.

The content of Chapter 2 is from our previous publication in [P1]. The content of Chapter 3 appears in two conference proceedings [P2][P3] and one journal publication [P4]. Portions of Chapter 4 appear in two conference proceedings [P5][P6] and one journal publication [P7], while the main body of Chapter 5 is from [P8]. The research ideas and the work presented in all of these manuscripts, including the literature review, mathematical modeling, algorithm design and development, simulations and implementation, analytical evaluations, result compilation, and manuscript preparation are all the result of my work. Dr. Mahsa T. Pourazad provided high-level consultation and editorial input. Dr. Mahsa T. Pourazad, and Dr. Panos Nasiopoulos provided consultation for experiments conducted in [P7], while Gable Yeung and Dan Miner helped in setting up the equipment for the experiments. My research supervisor, Dr. Victor Leung, has provided research guidance, supervisory comments, and corrections during the process of conducting the studies and writing the manuscripts. My research co-supervisor, Dr. Panos Nasiopoulos, helped with discussions, providing research guidance, supervisory feedback, checking the validity of my analytical results, and proofreading the respective manuscripts. Two of the publications [P4][P5] resulted from this thesis are co-authored by Dr. Amr Mohammed who provided high-level consultation and feedback. The first and last chapters of this thesis were written...
by Bambang Ali Basyah Sarif, with editing assistance from Dr. Panos Nasiopoulos and Dr. Victor Leung.

Over the course of his Ph.D., Bambang Ali Basyah Sarif has participated in other research. However, this thesis exclusively includes work related to energy efficiency of H.264/AVC-based video surveillance networks. Publications resulting from the research presented in this thesis are listed in the following.

[P1]. B.A.B. Sarif, V.C.M. Leung, and P. Nasiopoulos, "Distance Based Heuristic for Power and Rate Allocation of Video Sensor Networks", the IEEE Wireless Communication and Networking Conference (WCNC 2012), pp. 1893-1897, April 2012.


[P8]. Bambang A.B. Sarif, Mahsa T. Pourazad, Panos Nasiopoulos, and Victor C.M. Leung,
# Table of Contents

Abstract .................................................................................................................................................. ii

Preface .................................................................................................................................................. iv

Table of Contents .................................................................................................................................... vii

List of Tables ........................................................................................................................................ xii

List of Figures .......................................................................................................................................... xiv

List of Abbreviations ............................................................................................................................. xviii

Acknowledgments .................................................................................................................................... xx

Dedication ................................................................................................................................................ xxii

**CHAPTER 1: INTRODUCTION** ............................................................................................................. 1

1.1 Video Surveillance Network Overview .......................................................................................... 4

1.2 Overview on Video Surveillance Network Power Consumption Estimation ........................... 8

   1.2.1 *Video Surveillance Network Power Consumption Model* ................................................... 9

   1.2.2 *Trade-off between Encoding and Communication Power Consumption* ...................... 11

1.3 Video Encoding Configuration and Fairness in a Video Surveillance Network ....................... 12

1.4 Coding Complexity and Bitrate Models ....................................................................................... 14

1.5 The Effect of Content Complexity on Video Surveillance Network’s Power Consumption ........ 17

1.6 Research Topic Formulation ......................................................................................................... 18

   1.6.1 *Scope and Limitation of the Thesis* .................................................................................... 19

   1.6.2 *Research Road Map* ............................................................................................................. 21

1.7 Summary of Thesis Contributions ................................................................................................. 24
CHAPTER 2: ENCODING AND COMMUNICATION POWER CONSUMPTION TRADE-OFF
IN A VIDEO SURVEILLANCE NETWORK ................................................................. 30

2.1 Power-rate-distortion Model in Video Surveillance Networks .................................... 31
2.2 Network Lifetime Maximization in Video Surveillance Networks .............................. 31
2.3 Distance Based Heuristics for Energy Efficient Video Surveillance Networks ............ 34
   2.3.1 Proposed Approach .......................................................................................... 35
   2.3.2 Experiments and Results ................................................................................ 39
      2.3.2.1 Performance of the Proposed Technique .................................................... 40
      2.3.2.2 Comparison with Other Techniques ............................................................ 42
2.4 Summary .................................................................................................................. 45

CHAPTER 3: ENCODING PARAMETERS AND SCENE’S CONTENT COMPLEXITY IN VIDEO
SURVEILLANCE NETWORKS ......................................................................................... 47

3.1 Video Encoding Parameters and Node’s Power Consumption ..................................... 47
   3.1.1 H.264/AVC Coding Performance Analysis ...................................................... 48
      3.1.1.1 Encoding Complexity Measure .................................................................. 49
      3.1.1.2 The Effect of GOP Size towards Coding Complexity and Bitrate ............. 50
      3.1.1.3 The Effect of QP towards Coding Complexity and Bitrate ....................... 51
      3.1.1.4 The Effect of SR towards Coding Complexity and Bitrate ....................... 51
   3.1.2 Coding Complexity and Bitrate Trade-off ....................................................... 53
   3.1.3 The Effect of Encoder Parameter Settings towards Node’s Power Consumption..
       ............................................................................................................................. 55
      3.1.3.1 Encoding Power Consumption .................................................................. 55
3.1.3.2 Communication Power Consumption ......................................................... 56
3.1.3.3 Encoding and Communication Power Consumption Trade-off ............... 56
3.1.4 Fairness-based Node’s Encoder Configuration Allocation .......................... 58
3.1.4.1 Simplified Configuration ID Table .......................................................... 59
3.1.4.2 Fairness Constraint ................................................................................. 59
3.1.4.3 Exhaustive Search Fairness-based Algorithm .......................................... 60
3.2 Relation between Video Content, Bitrate and Coding Complexity .......................... 61
3.2.1 Video Surveillance Dataset ......................................................................... 62
3.2.2 Effect of Scene Complexity towards Coding Complexity and Bitrate ............... 65
3.2.3 Scenes’ Complexity Measurement and Classification ................................. 66
3.3 Common Approach ....................................................................................... 73
3.4 Fairness-Based Optimization for Minimum Power Consumption ..................... 74
3.4.1 Fairness-based CID allocation for Minimum Power Video Surveillance Network. .................................................................................................................. 77
3.4.2 Proposed Fairness-based CID Allocation Algorithm for the Test Set ............... 78
3.4.3 Experiments Settings ................................................................................. 81
3.4.4 Performance of the Proposed Technique for the Training Scenes ............... 83
3.4.5 Performance of the Proposed Technique for the Test Scenes ...................... 88
3.5 Summary ........................................................................................................ 89

CHAPTER 4: CODING COMPLEXITY AND BITRATE MODELS FOR H.264/AVC-BASED VIDEO SURVEILLANCE NETWORKS ......................................................... 91
4.1 Proposed Model ............................................................................................. 91
4.1.1 H.264/AVC Coding Complexity Model ................................................................. 92
4.1.2 H.264/AVC Bitrate Model .................................................................................... 95
4.1.3 Analysis of the Model .......................................................................................... 97

4.2 Model Evaluation in a Simulated Video Surveillance Test Environment ................. 98
4.2.1 Model Evaluation for Videos with Changing Content ......................................... 99
4.2.2 Video Surveillance Network’s Power Consumption Estimation ............................ 105
4.2.3 Experiments and Results .................................................................................... 105

4.3 Summary ............................................................................................................... 110

CHAPTER 5: SPATIAL AND TEMPORAL COMPLEXITY OF CONTENT AND VIDEO
SURVEILLANCE NETWORK’S POWER CONSUMPTION .............................................. 111

5.1 Methodology ........................................................................................................... 111
5.1.1 Content Complexity ............................................................................................ 112
5.1.2 Design of Bitrate Model ..................................................................................... 113
5.1.3 Design of Coding Complexity Model .................................................................. 116
5.1.4 Estimation of the Model Parameters .................................................................. 117
5.1.4.1 Video Sequences ............................................................................................ 118
5.1.4.2 Video Encoding Parameters ........................................................................... 118
5.1.4.3 Bitrate Model Parameters Estimation ............................................................ 119
5.1.4.4 Coding Complexity Model Parameters Estimation ....................................... 121
5.2 Experiments and Results ....................................................................................... 124
5.2.1 Performance Evaluation of the Bitrate Model .................................................... 125
5.2.1.1 Comparison with an Existing Bitrate Estimation Model ............................... 127
5.2.2 Performance Evaluation of the Coding Complexity Model ........................................ 128

5.3 Power Consumption Estimation using the Proposed Models ........................................... 130

5.4 Summary ................................................................................................................................. 138

CHAPTER 6: CONCLUSIONS AND FUTURE WORK .............................................................. 140

6.1 Summary of Thesis Contributions ........................................................................................... 140

6.2 Significance and Potential Applications of the Research ......................................................... 143

6.3 Future Work ............................................................................................................................. 144

   6.3.1 The Use of HEVC ........................................................................................................... 144

   6.3.2 Event-triggered Content-Based Video Surveillance Network ........................................ 144

   6.3.3 Content Based Frame Rate Adjustment ............................................................................. 145

   6.3.4 Cloud Based Video Surveillance Network ......................................................................... 146

References ........................................................................................................................................... 147
List of Tables

Table 2.1: Parameters used ..................................................................................................................39
Table 2.2: Average network lifetime and network lifetime offset, normalized to PRO-VSN42 ........42
Table 3.1: Encoding parameters for the low complexity configuration ..............................................48
Table 3.2: Configuration ID ..................................................................................................................54
Table 3.3: Parameters used for encoding and communication trade-off experiments ..............56
Table 3.4: Simplified Configuration ID (CID) ......................................................................................58
Table 3.5: Spatial Information unit (SI) of all videos .........................................................................68
Table 3.6: Temporal Information unit (TI) of all videos .....................................................................69
Table 3.7: Training and test scenes .....................................................................................................72
Table 3.8: Parameters Used for Fairness-based Algorithm Experiments ...........................................82
Table 3.9: Pnet, Pavg and STD(Pi) of the Training Scenes .................................................................86
Table 3.11: Percentage of improvement of the proposed algorithm against the other techniques ..........................................................................................................................88
Table 4.1: ME complexity level (ML) .................................................................................................92
Table 4.2: ME complexity level (ML) and δML ..................................................................................94
Table 4.3: Coding complexity modeling error ......................................................................................97
Table 4.4: Bitrate modeling error .........................................................................................................97
Table 4.5: Coding complexity estimation error for different values of k .........................................101
Table 4.6: Bitrate estimation error for different values of k ..............................................................103
Table 4.7: Parameters used .................................................................................................................105
Table 4.8: Experiment 1 scenarios ................................................................. 106
Table 4.9: Experiment 2 scenarios ................................................................. 108
Table 5.1: Functions of SA and TA to model $f_i$ .............................................. 112
Table 5.2: Functions of SA and TA to model $f_p$ .............................................. 112
Table 5.3: ME complexity level ($M_L$) ........................................................... 114
Table 5.4: Bitrate estimation results ............................................................... 125
Table 5.5: Bitrate estimation results for select videos .................................... 125
Table 5.6: Comparison with existing model ................................................... 127
Table 5.7: Coding complexity estimation results ............................................. 128
Table 5.8: Coding complexity estimation results for select videos ................. 128
Table 5.9: Parameters used ............................................................................ 134
Table 5.10: GOP and $M_L$ values obtained for three scenes selected .......... 135
Table 5.11: Nodes' maximum power consumption ......................................... 137
List of Figures

Figure 1.1: Video Sensor Networks (VSNs)........................................................................................................1
Figure 1.2: Scope of the research ..........................................................................................................................19
Figure 1.3: Research roadmap...............................................................................................................................22
Figure 2.1: Encoding and transmission power consumption..................................................................................30
Figure 2.2: Optimization in PRO-VSN..................................................................................................................33
Figure 2.3: A Video surveillance network’s zones ..................................................................................................35
Figure 2.4: The distance-based optimization ........................................................................................................38
Figure 2.5: Performance of the proposed heuristic. (a) varying distortion, node=4, width=50m; (b) varying distortion, node=9, width=50m..........................................................................................40
Figure 2.6: Performance of the proposed heuristic. (a) varying width, node=4, distortion=400; (b) varying width, node=9, distortion=400........................................................................................................40
Figure 2.7: Performance of the proposed algorithm. (a) maximum network lifetime, (b) the number of zone producing maximum lifetime ($z_{opt}$) ..................................................................................41
Figure 2.8: Network lifetime comparison (a) exact values (b) normalized towards PRO-VSN .........................................43
Figure 2.9: Average power consumption profile.....................................................................................................44
Figure 3.1: Complexity measure of different GOP sizes and QP values (a) encoding time, (b) the number of instruction counts ...........................................................................................................................49
Figure 3.2: Rate distortion plot for different QP values and GOP sizes ......................................................................50
Figure 3.3: The effect of different SR values to bitrate, complexity, and PSNR (a) BQmall 15 fps CIF video, GOP=75, QP=28 (b) Traffic 15 fps CIF video, GOP=75, QP=28........................................................................51
Figure 3.4: Trade-off between coding complexity and bitrate for BQmall 15 fps CIF video, GOP={1,2,4,8,75}, SR={2,4,8,16}, QP=28

Figure 3.5: Node’s power consumption: (a) single hop (b) multi-hops transmission

Figure 3.6: Camera placement

Figure 3.7: Scene settings

Figure 3.8: Snapshots of the “office” scene in the first activity setting from (a) camera2 and (b) camera6; and snapshots the third activity setting: (c) camera2 and (d) camera6

Figure 3.9: Snapshots of the “classroom” scene from (a) camera2 and (b) camera4; and the snapshots of the “party” scene from (c) camera2 and (d) camera4

Figure 3.10: Complexity, bitrate, and video quality trade-off

Figure 3.11: The complexity and bitrate of some videos from the “party” scene (a) different cameras but similar activity setting (b) the same camera but different activity settings

Figure 3.12: The complexity and bitrate of some videos captured by camera2 at similar activity setting but different scenes

Figure 3.13: The complexity and bitrate of some videos captured at different scene settings

Figure 3.14: SI(TI) class of office_act1 and classroom_act1 scenes and their corresponding scene’s class

Figure 3.15: Scene’s classes

Figure 3.16: MaximumFairness CID allocation algorithm

Figure 3.17: Power consumption minimization using CID

Figure 3.18: Power consumption minimization with bounded CID values
Figure 3.19: RecursiveBranchBound procedure .................................................................77
Figure 3.20: Fairness-based minimum energy algorithm ....................................................79
Figure 3.21: Fairness-based with adjustment algorithm ....................................................80
Figure 3.22: Network topology used in the experiments ....................................................81
Figure 3.23: Bitrate allocated per each node in all training scenes obtained by the three
algorithms (a) CommonConfig (b) MaximumFairness (c) Proposed ..................................83
Figure 3.24: Node's power consumption profile for the high SI training scene obtained by
the three algorithms: (a) CommonConfig (b) MaximumFairness (c) Proposed ..................84
Figure 3.25: Comparison of the Pnet, Pavg, and STD(Pi) values obtained from all test scenes
...............................................................................................................................................87
Figure 4.1: Normalized CP of the “BQMall” and “Traffic” videos (15 fps, CIF, GOP=2) ........93
Figure 4.2: Fractional increase of normalized CP of the “BQMall” and “Traffic” videos (15
fps, CIF) ....................................................................................................................................93
Figure 4.3: Bitrate of a P-frame for different M_L of “BQMall” video ..................................95
Figure 4.4: Camera placements ............................................................................................98
Figure 4.5: Content changes during a 10s camera1_shot3 video sequence. (a) frame 1 (b) frame 70 (c) frame 100 .................................................................................................................99
Figure 4.6: Content changes during a 10s camera2_shot2 video sequence. (a) frame 1 (b) frame 60 (c) frame 110 ..................................................................................................................99
Figure 4.7: Flowchart for coding complexity and bitrate estimation error reduction method ........................................................................................................................................100
Figure 4.8: Actual and estimated coding complexity for different values of $k$ and GOP sizes for the following sequences (a) camera4_shot1 (b) camera1_shot3 ...............................102
Figure 4.9: Actual and estimated bitrate for different values of $k$ and GOP sizes for the following video sequences (a) camera4_shot1 (b) camera1_shot3 ........................................104
Figure 4.10: Nodes’ power consumption in experiment 1 (a) scenario 1, (b) scenario 2, (c) scenario 3, (d) scenario 4 ........................................................................................................107
Figure 4.11: Nodes’ power consumption in experiment 2 (a) scenario 1, (b) scenario 2, (c) scenario 3, (d) scenario 4 ........................................................................................................109
Figure 5.1: Actual and estimated bitrate for varying GOP sizes, $\delta_{ML}$ and QP values for the following videos (a) camera1_office_act1, (b) camera4_office_act1, (c) camera5_office_act3, (d) camera2_classroom_act3, (e) camera6_party_act2, and (f) camera3_party_act4. ..........126
Figure 5.2: Actual and estimated coding complexity for varying GOP sizes, $\delta_{ML}$ and QP values for the following videos (a) camera1_office_act1, (b) camera4_office_act1, (c) camera5_office_act3, (d) camera2_classroom_act3, (e) camera6_party_act2, and (f) camera3_party_act4.................................................................129
Figure 5.3: Video surveillance network layout ..................................................................................................131
Figure 5.4: Video surveillance network power consumption minimization .......................................................133
Figure 5.5: Nodes’ video PSNR for the three test scenes analyzed .................................................................136
Figure 5.6: Nodes’ power consumption profile for (a) office_act2 (b) classroom_act3 and (c) party_act4 scenes .........................................................................................................................136
List of Abbreviations

CPI  Cycle per Instruction
DVC  Distributed Video Coding
FPS  Frame Per Second
GOP  Group of Picture
HD   High Definition video (1920×1080 pixels)
IEEE Institute of Electrical and Electronic Engineers
IEC   International Electrotechnical Commission
IoT   Internet of Things
ISO   International Organization for Standardization
ITU-T International Telecommunication Union
JPEG Joint Photographic Expert Group
Kbps  Kilo bit per second
MAC   Medium Access Control
MAD   Mean of Absolute Differences
Mbps  Mega bit per second
MDC   Multiple Descriptions Coding
ME    Motion Estimation
MAPE  Mean Absolute Percentage Error
MPEG  Motion Picture Experts Group
PCC   Pearson's Correlation Coefficient
PRD   Power Rate Distortion
PRA-VSN Power Rate Allocated – Video Surveillance Network
PRE-VSN Power Rate hEuristic – Video Surveillance Network
PRO-VSN Power Rate Optimal – Video Surveillance Network
PSNR  Peak Signal to Noise Ratio
QoS   Quality of Service
QP    Quantization Parameter
RMSE  Root Mean Square Error
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>Spatial Information</td>
</tr>
<tr>
<td>SR</td>
<td>Search Range for motion estimation</td>
</tr>
<tr>
<td>TI</td>
<td>Temporal Information</td>
</tr>
<tr>
<td>UBC</td>
<td>University of British Columbia</td>
</tr>
<tr>
<td>VSN</td>
<td>Video Sensor Network</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
</tbody>
</table>
Acknowledgments

I would like to profoundly thank my supervisors Dr. Victor Leung and Dr. Panos Nasiopoulos for their continuous support, encouragements and guidance throughout the years I spent at UBC. Without their wisdom and valuable suggestions, this work would have not materialized. I am truly honored to be supervised by a pioneer researcher in the field of wireless communications and multimedia. I would also like to thank Dr. Nasiopoulos for his enthusiasm, motivation, and insightful suggestions that not only enlarged my vision of science but also about life. He has always been a great mentor and a role model for me. I am also grateful and thankful for the guidance, help, and support that I received from Dr. Mahsa Pourazad for the last four years of my Ph.D. Finally, I would like to also thank my supervisory committee members for their contribution in this thesis.

I would also like to extend my sincere thanks to my colleagues at the Digital Media Lab, Dr. Di Xu, Mohsen Amiri, Hamid Reza Tohidypour, Dr. Amin Balitalebi, Sima Valizadeh, Basak Oztas, Maryam Azimi, Ahmad Khaldieh, Pedram Mohammadi, Fujun Xie, Stelios Ploumis, Ilya Ganelin, Timothee Bronner and all other lab mates. It was a wonderful experience working in such a friendly but also enriching environment. I also thank my friends in the X310 lab, Dr. Davood Karimi, Hamid Palangi and others. I would also like to thank my colleagues at the Wireless Networks and Mobile Systems Labs for their support in the earlier stage of my research, Dr. Ali Al-Shidani, Dr. Amr Al-Asad and Dr. Sergio Gonzales. I also thank my friends and colleagues I met in Vancouver and also the Indonesian community in Vancouver. Each and every one of them supported me in one way or the other to complete this work.
To my wife for her unconditional support that kept me going whenever everything does not seem to work, and to my kids that brought joy to my life; I cannot describe how thankful I am for their warmth and love that are reminders that my world does not revolve only around my research.

Last but not least, I owe praise and thanks to almighty God, the source of all good, for granting me the wisdom, health, and strength to undertake this task and enable me to complete this long journey.
Dedication

To my parents, words cannot describe how lucky I am to be your son
To my wife for the love and constant support that keep me going
To my kids for the joy that you brought to my life
Chapter 1: Introduction

Recent advances in hardware, computing, and communication technologies have enabled the implementation of pervasive computing applications that share the vision of ubiquitous networked devices that can function and process information gathered from the environment on behalf of users. Wireless sensor networks (WSNs) [1] that are able to provide a diverse set of context data from the monitored environment to interested clients can provide the basis architecture for such technology. Sensor nodes can sense physical features of the environment, perform simple data processing and send the information to a central device that usually has unlimited resources. The information gathered at the central device, usually called the sink node, can be used by a human operator or additional machine/software to perceive the condition of the monitored environment and provide some kind of action if necessary. Due to the ad-hoc nature of the deployment, the information sent to the sink is usually performed in a multi-hop wireless communication fashion.

Figure 1.1: Video Sensor Networks (VSNs)
Although WSN was originally used to gather and process simple physical measurements of the environment such as temperature and humidity, recent trends focus on a wide range of WSN applications that include the following: public structures (such as bridges) monitoring [2], habitat monitoring [3], healthcare [4], access hatch monitoring [5], fire alarm [6], volcanic eruption monitoring [7], and disaster-hit region surveillance [8].

With the availability of more advanced sensor nodes, there is an increasing demand in using rich context information in the form of multimedia streams such as image or video in WSNs [9][10]. This type of sensor networks is equipped with tiny camera sensor nodes, embedded processor, and wireless transceiver. This architecture is called wireless video sensor network (VSN) [11] (see Figure 1.1). Note that some literature uses VSN to denote visual sensor network as WSNs that have the ability to perform image or video processing on their nodes [12][13]. Unlike common WSNs, video sensor networks are able to provide richer sensing information and coverage that are beneficial for applications requiring visual information. Some of these applications include traffic monitoring [14][15], smart home [16][17], fire alarm [18][19], tracking [20], object classification and feature extraction [21][22], video summarization [23], predictive visual analysis [24], intelligent environment [25], Internet of Things [26], and surveillance [11], [27], [28]. In the case of surveillance application, VSNs offer some additional benefits as compared to the existing systems. Firstly, the deployment cost of VSN-based surveillance application is much cheaper than the existing surveillance system, since there is no wiring cost. In addition to that, VSNs have the potential to improve the ability to develop user-centric video surveillance applications to monitor and prevent harmful events [29]–[31].
A video surveillance system that was implemented using VSNs usually has limited resources in terms of energy resources, network bandwidth, and processing capability. Unlike the case with common sensor networks where power consumption mostly comes from the wireless transmission, the video encoding process contributes to a significant portion of the overall power consumption. However, there is a trade-off between coding complexity and bitrate in encoding a video such that higher compression, i.e., lower bitrate, comes at the cost of higher coding complexity. Furthermore, coding complexity and bitrate entailed by the encoding process depend on the encoding parameters used and content features of the captured video. Regarding transmission power consumption, there is a need for considering that the video surveillance node may need to relay information from other nodes. Thus, it is very important to increase the energy efficiency of video encoding and the transmission process performed in the video surveillance network.

This thesis investigates the energy efficiency of video surveillance network where the trade-off between computation and communication power consumption is exploited. In particular, the effect of video encoding parameters, content complexity, and network topology are taken into consideration in the video surveillance network's power consumption minimization. Chapter 2 describes the effect of video encoding configurations on the trade-off between computation and communication power consumption. Chapter 3 analyzes the effect of different video encoding parameter settings towards coding complexity and bitrate. In order to proceed with this analysis, a video surveillance dataset consisting of a large number of captured content with different levels of spatial detail and motion activity was captured. An algorithm that minimizes the video surveillance
network’s power consumption for different scene settings is also proposed in this chapter. In Chapter 4, we analyze the effect of different encoding parameter settings on power consumption by designing models to estimate the coding complexity and bitrate. Chapter 5 discusses how video content features are incorporated into the models proposed in Chapter 4. The optimization method utilizing the coding complexity and bitrate models to minimize the video surveillance network’s power consumption is then explained and analyzed. Summary of the contributions and future work are detailed in Chapter 6.

The following sections in this introductory chapter provide a literature review and an overview of the topics addressed in each of the research chapters. Section 1.1 provides the literature review on video sensor network technology for surveillance application. Existing studies on video surveillance networks’ power consumption measurement, model and analysis are summarized in Section 1.2. Section 1.3 reviews the effect of video encoding parameters on the video surveillance network’s power consumption. In Section 1.4, we provide an overview of coding complexity and bitrate models of H.264/AVC. Section 1.5 discusses the effect of spatial and temporal information of content on power consumption in a video surveillance network. Section 1.6 concludes the introduction with a summary of research contributions presented in this thesis.

1.1 Video Surveillance Network Overview

Research in the VSN for surveillance applications started at the beginning of this century. The idea of capturing multimedia data on a sensor network platform was first introduced in [11][12]. One of the earliest prototypes was reported in [32]. Since then, the field of video surveillance network has received increased attention from researchers
around the world. In [33], a low-end video surveillance network called Cyclops was built using mica2 sensor mote. It uses its own image processing library that can transfer compressed image with a data rate of 40 kbps. Some general purpose video surveillance networks were later developed using the Intel XScale PXA255 board [34] and webcams, e.g., Panoptes [35] and Meerkat [36]. These platforms have the ability to encode captured contents into a stream of JPEG still images. After that, there has been an increasing number of video processing capable sensor nodes reported in the literature, that include the likes of imote2 [36], cmucam3 [38], XYZ sensor mote [39], CITRIC [40], and DSPCam [41]. With higher processing power, these platforms have the computational capability to perform more advanced video encoding techniques. Although most of these nodes operate on the IEEE 802.15.4-compliant Zigbee module that has a data rate of 250 kbps, some of the motes can also use the 802.11b WiFi compliant module that has a maximum data rate of 11 Mbps.

The increase of information level processed by the nodes in a video surveillance network provides a number of research challenges and of course opportunities [9][10][42][43]. Some of the key active research areas are the following: sensor coverage [44]–[48], camera selection [49]–[52], node placement [53]–[55], quality of service (QoS) [56]–[59], energy consumption (and system lifetime) [60]–[67]. However, because of the underlining concern of limited energy resources, the issue of energy efficiency has become one of the most important design aspects in video surveillance networks. In a common sensor network that operates on scalar data, the energy required to process the data is negligible. Therefore, the emphasis on energy efficiency is mostly on the wireless transmission process [68]–[75]. However, video processing requires extensive resources in
both encoding video and transmitting the encoded stream. The techniques that focus on the efficiency of wireless transmission alone, e.g., the work in [76], [77], are not sufficient to guarantee the overall energy efficiency in video surveillance networks.

The amount of information processed by a video surveillance node is at least one order of magnitude higher than the scalar data that was handled in the traditional sensor network. The signal processing and video compression techniques performed on the video surveillance network nodes enforce additional requirements and burden to the computation capability and energy consumption of the nodes. In this regard, researchers have proposed and analyzed different data compression techniques for video surveillance networks. Still image compression algorithms such as JPEG [78] are the earliest choices for low-end video surveillance network platforms. Some researchers have analyzed and modified JPEG to suit the requirements of video surveillance networks [11], [79]–[83]. Another technique considered to be used for video surveillance networks is distributed video coding (DVC) [84][85][86]. The technique is based on information-theoretic bound established by Slepian and Wolf [87] and later by Wyner and Ziv [88]. This technique allows multiple correlated sources to be encoded independently and later jointly decoded. In practice, a regularly spaced subset of frames, called key frames, is coded using a conventional intra-frame coding technique such as JPEG or the one adopted in H.264/AVC. All frames between the key frames are referred to as Wyner-Ziv frames and are intra-frame encoded but inter-frame decoded. Motion compensated frames are generated using key frames as the side information at the decoder. A few examples of DVC-based video surveillance network were reported in the literature [89]–[91]. However, although some
studies considered DVC as the low-power encoder for video surveillance networks, the authors in [92] showed that the power consumption gain of DVC against the standard H.264/AVC encoder in low complexity mode is not significant and may not warrant the use of the feedback channel required in a DVC implementation.

The availability of more advanced sensor nodes allows us to use more advanced video compression techniques such as H.264/AVC [93], [94]. H.264/AVC’s is presently the most widely used ITU-T and MPEG video coding standard. Although this video coding standard is computationally more expensive than the other compression techniques mentioned in the previous paragraphs, it offers better compression performance. In addition to that, the recent advent of integration between sensor networks and Internet of Things (IoT) technologies [96] shows that there are additional benefits in using H.264/AVC. Firstly, H.264/AVC provides an excellent portability across different hardware and software platforms. Secondly, since H.264/AVC is a video coding standard, the issues and solutions for transmitting such a coded stream on the Internet have been well established compared to other codecs. It is worth noting that the performance of H.264/AVC in terms of quality, bitrate, and complexity, is determined by a large number of encoding parameters. In a resource constrained environment such as a video surveillance network, it is very important to choose the right configuration and setting parameters that lead to an optimal coding performance. The encoder configuration settings of a video surveillance node determine not only the node’s power consumption but also the overall power consumption. This will be discussed in more detail in the following sections.
1.2 Overview on Video Surveillance Network Power Consumption Estimation

Video surveillance networks’ computation and communication capabilities and their performance parameters depend on the sensor node platform they use [97]. In addition, video surveillance networks are often powered by batteries, whose replacement may be infrequent, not preferable, or even impossible in many applications. Although there has been a tremendous improvement in the battery technology in the past decades, the capacity of the modern lithium-ion battery is nearing its limit [98]. It is of high importance that a video surveillance network has the capability to perform sophisticated video processing in an energy efficient way.

One of the earliest studies on power consumption needed to encode a video and send the compressed data through a wireless medium is reported in [99]. In that study, it is shown that for even a relatively small picture size, i.e., QCIF (quarter common intermediate format, 176x144 pixels) video, the H.263 encoder consumes about two-thirds of the total power consumption of video transmission in a wireless local area network (WLAN). A characterization of video surveillance network’s power consumption sources in a Meerkat testbed was performed in [61]. The work in [62] highlighted that the two main sources of energy drain in a video surveillance node is the computation and communication. That paper studied the computation and communication power consumption of different video encoders, i.e., Motion JPEG2000, H.264/AVC Intra, H.264/AVC Inter main profile and DVC, implemented on a Stargate platform. The encoding power consumption is calculated as CPU cycles needed to encode the video multiplied by the energy consumption per cycle. It
was shown that the computation and communication power consumption of videos depend on the video encoder used and the encoder configuration settings. Based on these results, the authors claimed that for a single-hop transmission video surveillance network, the inter-based H.264/AVC encoding is not suitable for low energy sensor nodes. The authors in [100] studied the encoding and transmission power consumption of a multi-hop video surveillance network. The video encoders analyzed in [100] include H.264/AVC intra, DISCOVER, a DVC-based encoder presented in [86], and distributed compressive video sensing (DCVS) [101]. In [65], the trade-off between computation and communication energy consumption in a single-hop and multi-hops communication of MPEG-4 and H.264/AVC encoders was compared. It was shown that H.264/AVC is more energy efficient than MPEG-4 in a multi-hop video surveillance network environment.

All the above studies have shown that different encoders and even different profiles of the same encoder lead to different power consumption. In addition, the configuration settings used to encode the video play an important role in controlling the overall power consumption of the video surveillance network. However, although these studies provided some insights about the energy consumption profile of different video encoders in a testbed, the number of video encoder configuration settings used in these studies is limited. In particular, the performance of the H.264/AVC encoder in low encoding mode was not properly investigated.

1.2.1 Video Surveillance Network Power Consumption Model

A video surveillance network node’s total power consumption is due to encoding power consumption and transmission power consumption. Both consumptions depend
largely on video encoding parameters used to encode the video. This is due to the fact that the energy consumed to encode a video depends on the coding complexity of the underlined process. On the other hand, transmission of lower bitrates requires less amount of energy. As a result, in order to be able to estimate the video surveillance network’s power consumption, there is a need to have an in-depth investigation of the effect of encoder parameters settings towards coding complexity and the resulting bitrates.

In order to estimate the nodes’ power consumption, the authors in [62] calculate encoding energy consumption as the multiplication of the encoding elapse time, i.e., the time needed to encode a video, with the node’s energy consumption per cycle. In another study, the encoding energy consumption is calculated as the estimated encoding elapse time multiplied by the assumed voltage and current of a sensor node [100]. From these studies, it can be seen that the encoding power consumption can be estimated from the coding complexity measure for encoding the video and some parameters related to the sensor node hardware platform. On the other hand, the transmission energy consumption depends on the bitrate of the video, the transmission distance and some parameters related to wireless transmission.

However, the nodes in a video surveillance network may need to relay the encoded video data from the other nodes in the network. Therefore, the encoder parameters settings used at a specific node in the video surveillance network affect not only the node’s power consumption but also the distribution of power consumption in the overall system. In order to optimize the video surveillance network performance, i.e., minimize its power consumption, we need to take into consideration the encoding and communication power
consumption of all the nodes in a video surveillance network. This will be explained in more detailed in the following subsection.

1.2.2 Trade-off between Encoding and Communication Power Consumption

Due to the nature of their deployment, the video surveillance nodes farther from the sink may need to send their data through intermediate nodes. This increases the communication power consumption of the nodes that are located closer to the sink [72], [102], [103]. In a battery powered video surveillance network, the amount of initial energy resources available per each node and the node's total power consumption determines its system lifetime, which is an important aspect of its performance [104]. In order to determine the relation between encoding and communication power consumption of a video surveillance network, some researchers tried to relate the nodes’ power consumption with the bitrate of the encoded video. One such model is called the power-rate-distortion (PRD) [60], [105].

According to the PRD model proposed in [60], in order to achieve a specific encoding distortion requirement, one can either increase the encoding efficiency or the bitrate. The power consumption of the encoding process can be controlled by managing three video coding parameters: the frame rate, the number of skipped blocks, and the number of examined blocks for motion estimation [60]. In [105], the authors simplified the PRD model and showed that there is a trade-off between encoding and transmission where optimal power consumption of a source node can be achieved. However, the PRD model includes a parameter called encoding efficiency coefficient that is presumably specific to the hardware platform.
Based on the PRD model, an optimal encoding power, and rate allocation was proposed in [63]. In that study, the node’s encoding power consumption and bitrate were controlled such that the power consumption is minimized. Although the technique can allocate the encoding power consumption and bitrate for each node, the method to determine the encoding parameters settings to get such encoding power consumption and bitrate were not explained. Nevertheless, the nodes’ encoding power consumption can be estimated from the computational complexity of the encoding task. In addition, the coding complexity and bitrate of video depend on the encoder parameters settings used to encode the video. Therefore, in order to find the encoder configuration settings for each node that lead to minimum power consumption, we need to develop models that relate these encoding settings with coding complexity and bitrate.

1.3 Video Encoding Configuration and Fairness in a Video Surveillance Network

H.264/AVC is presently the most widely used ITU-T and ISO/IEC video coding standard. This standard comes with plenty of setting parameters and configurations that can be tailored to suit the application requirements. In order to take full advantage of the flexibility and advanced features of the H.264/AVC, one needs to study the effect that different parameter settings have on H.264/AVC’s performance in terms of computation complexity and bitrate. Note that choosing the right configuration and parameters that lead to optimal encoding performance is of prime importance to video surveillance applications that are constrained in terms of bandwidth and energy resources.
There is a number of published research on H.264/AVC’s performance [106], [107]. However, the focus of the existing studies is to determine the optimal coding configuration without considering the total energy consumed for encoding and transmission. It is worth noting that there is a trade-off between coding complexity and bitrate such that in order to reduce the bitrate (lower transmission power consumption), more advanced encoding techniques are necessary (higher encoding power consumption). The trade-off between encoding and transmission energy consumption for a limited number of H.264/AVC configuration settings were investigated in [62], [65], [100]. However, H.264/AVC comes with a large number of encoding parameters and its low complexity performance needs to be studied in more detail.

Furthermore, video nodes in the surveillance network share the same wireless medium in order to send their encoded video to the sink node. Since the bandwidth allocated to a network is limited, there is an issue of fairness of bitrate allocated per each node. Allocating the same bitrate to each video node guarantees fairness in terms of bitrate and quality of the encoded video, given that each node has the same video and is using the same video encoding parameters settings. However, in many video surveillance deployment scenarios, nodes further from the sink usually need to relay their data through intermediate nodes. Therefore, the total energy consumption of nodes that are closer to the sink will be greater than the nodes that are further. More balanced energy consumption among nodes may be achieved by allocating different fairness ratio for each node in the video surveillance network. However, that needs to be achieved without sacrificing the quality of the transmitted video of any node. This can be done by assigning different
combinations of coding complexity and coding efficiency in terms of bitrate to each node in the video surveillance network so that the nodes produce compressed videos with almost similar quality in terms of peak signal to noise ratio (PSNR). This is in contrast to previous approaches, where all nodes use the same encoding configuration. Note that in this thesis, that approach is called the CommonConfig technique. By assigning different encoding configuration to each node, different fairness ratios are assigned to the nodes. That will affect the distribution of power consumption in a video surveillance network, such that the trade-off between encoding complexity and compression performance can be exploited in order to minimize the overall power consumption. To the best of our knowledge, this idea has not been investigated in any existing literature on video surveillance network.

1.4 Coding Complexity and Bitrate Models

As mentioned in the previous subsections, the energy consumed to encode a video stream depends on the complexity of the encoding process. On the other hand, transmission of lower bitrates requires smaller amounts of energy. One approach to increasing the efficiency of energy consumption is by assigning different fairness ratio per each node, as mentioned in the previous subsection. This fairness ratio can then be mapped towards coding complexity and bitrate tuples that are generated from some combinations of encoding parameters. However, such approach is not flexible and is limited by the number of configurations that can be used per each node. Thus, models that can estimate the coding complexity and bitrate are necessary. Several studies have proposed models that relate the average bitrate of an encoded video to some video encoding parameters, such as the group of picture (GOP) size, frame rate, and/or different video content features.
Ding and Liu proposed one of the earliest rate models in [108] that relate the bitrate to the quantization step-size. In [109], [110], quadratic rate models based on the quantization parameter (QP) were proposed. Another quadratic rate model was proposed in [111] where the bitrate is estimated using QP and the frame complexity measure called mean absolute differences (MAD). MAD is calculated using linear predictions from both the actual MAD and the rough MAD, i.e., the MAD between the current frame and the previous reconstructed frame [112]. Furthermore, Ma et. al., proposed a bitrate model that considers the QP and the frame rate of the video [113]. This model is based on a multiplicative power function where the model parameters are estimated using the following video content features: frame difference, normalized frame difference, motion vectors, displaced frame difference, motion activity intensity, motion vectors normalized by contrast, motion vectors normalized by intensity, and motion vectors normalized by variance. The displaced frame difference is described as the mean absolute difference that is in the top ten percentile between of corresponding pixels in successive frames using an estimated motion vector. For more information on the features used in [113], please see [114]. The multiplicative power function model is later simplified in [115] by using spatial information (SI) and temporal information (TI) indices of video content (these indices are calculated based on [116]). The aforementioned models use the same GOP structure in their analysis. However, the GOP size affects the coding complexity and bitrate trade-off significantly. An elaborate study on the effect of GOP size towards the video surveillance network's power consumption is thus necessary.
In terms of an encoding power consumption model, a power-rate-distortion model was used in [105] where the video encoder power consumption is related to bitrate, video variance and a parameter called encoding efficiency coefficient. Computational complexity management and control strategy of H.264/AVC for low-power devices was proposed in [117], [118]. In these, based on the available processing power of the devices, the number of block size candidates used for motion estimation and the frame rate of the video were controlled. However, these studies only considered the IPPPPP GOP structure. The authors in [119] proposed a coding complexity model for H.264/AVC for different encoding configurations. However, some of the model parameters can only be obtained by encoding the video. Thus, the model cannot be used for new unknown videos.

There are some H.264/AVC encoding parameters that significantly affect the coding complexity and bitrate. These parameters include GOP size, GOP structure (whether the B-frame is used or not), sub-pixel motion estimation, motion estimation algorithms, the type and variety of block size candidates used for motion estimation, QP, and frame rate. In order to develop the coding complexity and bitrate models for video surveillance network, the effect and significance of those parameters need to be investigated. Furthermore, although the results shown in earlier studies indicate that video content affects the coding complexity and bitrate, there has been no in-depth investigation on the use of spatial and temporal information for estimating the coding complexity and bitrate, let alone incorporating this information in the power consumption estimation. The following subsection discusses the need to include the effect of spatial and temporal information in estimating power consumption in a video surveillance network.
1.5 The Effect of Content Complexity on Video Surveillance Network’s Power Consumption

One of the common drawbacks of the existing studies on the video surveillance network’s power consumption is the fact that the analysis is performed with the assumption that the same video content is used in all the nodes [60], [63], [65], [100]. However, this assumption is a very simplistic approach and it does not represent a real video surveillance network deployment, where different nodes capture the scene from different point of views. Therefore, the content complexity of captured video streams from all the node is not the same. Note that the performance of a video encoder in terms of computational complexity and bitrate depends on both encoding parameters and the temporal and spatial complexity of the content. That elevates the importance of determining the trade-off between encoding complexity and data transmission in a video surveillance network that would minimize power consumption.

The coding complexity and bitrate of video determine the video surveillance network’s power consumption. Furthermore, the coding complexity and bitrate of a video depend on the encoding parameters and spatial/temporal information (SI, TI) of content. Therefore, in order to estimate the power consumption of a video surveillance network, one has to also consider the spatial and temporal complexity of the scene at which the video surveillance network will be deployed. In order to do that, it is important to have knowledge of the encoding complexity and efficient bitrate models that take into consideration the complexity level/information of the captured scene. It is important to note that for the same visual quality, the total bitrate generated by the captured scenes that
have high spatial and temporal detail will be higher than the ones obtained from scenes with lower spatial and temporal detail. Considering that the bandwidth of a video surveillance network is limited, the scene’s spatial and temporal detail will affect the visual quality of the encoded video. This means that the compression efficiency will have to change at every scene, depending on its complexity. However, to the best of our knowledge, such condition has not been explored in the existing studies. Furthermore, in order to be able to develop and test models for power consumption, there is a need to have a video surveillance dataset of real-life captured content that represents videos with different spatial and temporal complexities. Unfortunately, such a dataset is not publicly available.

1.6 Research Topic Formulation

The overview provided in Section 1.2 to Section 1.5 presents two important research questions:

1. Can we understand how the encoder parameters settings, the complexity of content and network configuration affect the trade-off between coding complexity and bitrate of videos captured by the nodes in a video surveillance network?

2. Can we exploit the trade-off between the key factors mentioned in point 1 above to minimize the overall power consumption of a video surveillance network?

The above-mentioned research questions require us to perform an in-depth study in both video compression and wireless network subjects. Note that, these two subjects have a diverse active research area, different theories, principles, assumptions and, sometimes, conflicting implementation requirements that are difficult to cover in a thesis. Therefore, in
order to proceed, we need to limit the scope of our research and use some necessary assumptions. The following subsections provide the scope and roadmap of the research presented in this thesis.

1.6.1 Scope and Limitation of the Thesis

A major part of the work presented in this thesis is centered around topics related to the trade-off between encoding complexity and bitrate, based on the analysis of encoder parameter settings and content complexity of videos as shown in Figure 1.2. In exploiting the trade-off of the factors shown in that figure to minimize the overall power consumption, we focus our analysis on the encoding and communication operations on a video surveillance network. In addition to that, we also put an emphasis on taking into account a diverse set of video content to simulate a more practical and realistic deployment of a video surveillance network. In this regard, many aspects of the wireless transmission process of the simulated network were kept simple, to provide a greater depth analysis of the effect of spatial and temporal information content on the video surveillance network’s power consumption. The following are some assumptions that were used in this thesis.
1. All the nodes in the video surveillance network encode and transmit videos of the captured event. This is performed in order to simulate the surveillance network under a heavy load that signifies maximum power consumption and minimum system lifetime.

2. A video surveillance network can have a wide variety of network layouts. We use a grid base topology for the following reasons: (a) Our experiments were conducted using a large number of real-life captured content that we generated; it is easier to deploy and manage the cameras that were deployed in a structured layout such as a grid. (b) Grid layout is one of the common settings used in video surveillance networks. We use nine cameras in our experiments to follow the experiments conducted in [63]. Furthermore, we also assume that each node in the network can only transmit data to its direct neighbor. Note that this assumption has been used in other studies as well [62], [65], [100].

3. The sum of the data rates of the encoded video streams transmitted from all the nodes is constrained to simulate the resource limitation in a video surveillance network. The value of this constraint will be mentioned specifically in the experiment.

4. We assume that all nodes use the same QP so that the encoded videos from different nodes will have almost similar quality. The highest quantization parameter used in this thesis is set equal to 40 to make sure that the encoded video has the minimum acceptable quality. However, different scenes may need to use different QP values.
The minimum QP value, i.e., that gives the highest quality, that can accommodate the network bandwidth for the scene will be used.

1.6.2 Research Road Map

Figure 1.3 shows the research roadmap of the thesis. The first effort in our research involved the use of the existing theoretical model, coming up with a practical implementation and trying to control the bitrate using variable frame rates. This is achieved by assuming that the video surveillance network’s area can be divided into different zones. Nodes at the same zone are using the same frame rate and thus have the same bitrate. This research allowed us to better understand the complexity of the encoding process and that by controlling video encoding parameters used at each node, we could control the trade-off between encoding complexity and bitrate much more efficiently than just using the frame rate. In addition, we established a solid understanding of the relationship between encoding power consumption and that of transmitting the resulting bitstream and the dependence on the network layout. As a result, we focused the next stage of our research on analyzing the effect of different encoder parameters on coding complexity and bitrate. The objective of this effort was to determine the encoder parameters that have the highest impact on coding complexity and bitrate. We learned that the group of pictures (GOP) and search range (SR) for motion estimation significantly affect the coding complexity and bitrate. Furthermore, we also learned that video content affects coding complexity and bitrate and thus it needs to be taken into consideration. However, a video surveillance dataset consisting of a wide variety of content was not available. In fact, every existing method assumed that every camera in the network had the same
## Research Roadmap

### 1. Existing Model
- Practical implementation
- Control nodes' bitrate with variable frame rate

### 2. Encoding Parameters Analysis
- Large number of configuration settings and profiles
- Find parameters with the highest impact on complexity/bitrate

### 3. Dataset Generation
- Generate content with a wide range of spatial and temporal details
- Camera placement
- Video capturing synchronization

### 4. Look Up Table (LUT) Based Approach
- Content captured by each node is unique
- A node needs to relay the video stream through other nodes
- Similar encoded video quality on all nodes

### 5. Model Design
- Effect of GOP, QP & motion estimation
- SI/TI of content
- 108 videos, each with 343 encoding configurations
- Nodes' position in the network

### Assumptions/Setup
- Network can be divided into several zones
- Nodes at each zone use the same frame rate (i.e., bitrate)
- Similar video distortion requirement on all nodes
- Similar video at all nodes

### Results and Findings
- Better performance than the single configuration approach.
- Frame rate is not enough. There are many other encoder parameters that can control the quality and bitrate better.
- Group of picture (GOP) size and search range (SR) for motion estimation have the highest impact on complexity and bitrate of video.
- Encoding complexity and bitrate of video depends on content. Need to have a video dataset with a wide range of content.
- 108 HD videos 60 fps with a wide range of spatial and temporal details were generated. Videos were made publicly available.
- Spatial/temporal information (SI/TI) indices provided by MPEG could be used to classify video content complexity.
- LUT-based method provides better results than the single configuration or common bitrate techniques.
- LUT based method is not practical nor efficient. Need to design coding complexity and bitrate models.
- The bitrate and coding complexity models show very high correlation and low error estimation.
- The proposed technique is able to allocate encoder parameters settings for each node at different scenes that minimize the overall power consumption.

*Figure 1.3: Research roadmap*
video content, something that is far from reality and of course leads to erroneous results. This led to the third roadmap in our research, which was the generation of a comprehensive real-life video sensor dataset with different scenarios and scenes.

In order to generate such a video surveillance dataset, we installed 9 HD cameras which captured different views of the same scene. We developed different scenario settings and simulated different real-life activities on each scene. In addition to that, we also identified that camera placement and video capturing synchronization were two other challenges we face in VSNs. In this regard, we decided to use a grid connected camera layout that provided easier installation and maintenance. We used the default synchronization provided by the surveillance camera system during the video capturing process. This research effort resulted in a video surveillance dataset consisting of 108 unique videos with a wide range of content complexity. Our next effort was to use this dataset in our analysis and development of an overall architecture that minimizes the video surveillance network's power consumption. At the fourth roadmap, we assumed that the tuples of encoding configurations, coding complexity, and bitrate for each video could be stored in a look-up-table. We used the spatial and temporal information unit metrics provided by MPEG to measure the video content complexity. These metrics were also used to classify the videos into different levels of complexity. In this regard, we viewed the nodes’ encoding configuration problem as bitrate allocation such that the bitrate fairness allocation of scenes' with higher content complexity can be used to allocate the nodes' configuration at the other scenes. Although the results of the look-up-table (LUT) based
approach were good, this approach is not practical. This led to the final roadmap in our research.

Using the dataset generated at the third roadmap, and the knowledge about encoding parameters that significantly affect the encoder performance, we developed coding complexity and bitrate models. We performed more than 30000 experiments from 108 unique videos and 343 different encoding configurations for each video to train and evaluate the models. We then combined the developed models with information about the network layout and nodes’ position in the network to design a novel architecture to minimize the video surveillance network's power consumption.

The roadmap explained in the previous paragraphs was used to guide the research presented in this thesis and has resulted in three major contributions that will be explained in the following subsection.

1.7 Summary of Thesis Contributions

Three contributions are presented in this thesis. Each contribution targets specific problems and limitations relevant to the issues described in the previous sections. A summary of this thesis’ main contributions is given below:

1. Determined the H.264/AVC encoder parameters that significantly affect the trade-off between encoding and communication power consumption.

In order to have a practical implementation of the PRD model [105], in Chapter 2, we proposed a method to control the nodes’ bitrate by using variable frame rate. This was achieved by organizing the video surveillance network into different zones based on
the nodes’ distance from the sink. Nodes that are located in the same zone will use the same encoding parameters settings. The encoding power consumption allocated to each node is related to the frame rate and the energy consumption needed to encode one frame that was reported in [62]. The distance-based heuristic was thus proposed [120] and showed to have a performance that although was not as good as the optimal technique that uses the PRD model [63] it was a much higher performance than the single configuration approach. Note that the work described in [63] uses the PRD model to assign different encoding power consumption - and consequently bitrate – to each node in a VSN, without providing any insight how a node can achieve such encoding power consumption and bitrate.

However, frame rate alone is not the best way to control bitrate and coding complexity. There are other encoder parameter settings that can offer better bitrate control and better quality trade-off. Hence, we continued our research by analyzing the effect of different encoder parameters settings on encoding and communication power consumption. The objective was to determine the encoder parameters that have the highest impact on the trade-off between coding complexity and video bitrate. Our analysis, presented in Chapter 3, showed that the group of picture (GOP) size and search range (SR) for motion estimation have the highest impact on coding complexity and bitrate. The tuples of encoding configuration combinations, coding complexity, and bitrate of the video were then represented as a lookup table that can be used to exploit the coding complexity and bitrate trade-off such that the overall power consumption can be minimized. An additional factor that has a direct effect on coding complexity
and bitrate is video complexity in terms of texture, color, and motion. Since all existing approaches make the erroneous assumption of using the same video for each camera node, it was essential to generate a real-life comprehensive video surveillance dataset with a wide variety of content for our analysis. This leads to the next contribution.

2. **Generated a video surveillance dataset that represents different spatial and temporal complexities.**

In Chapter 3, we highlighted the fact that each node in a video surveillance network may capture video from a different angle, thus producing video streams with different content. In addition, the scene layout and setting where the surveillance network is deployed can have different levels of spatial detail and temporal complexity. Unfortunately, earlier studies did not consider the significance of video content in their framework. In fact, all studies we found in the literature assumed that each node uses the same video in their analysis. In order to be able to incorporate the effect of content, we needed to have a video surveillance dataset that has different levels of content complexities that can be used in our study. However, such dataset is not available in the literature. In order to proceed, we set up a complete surveillance system and captured a large number of videos from real-life and simulated events in a couple of public buildings. As a result, we generated two video surveillance datasets. The first dataset was compiled using four cameras capturing real-life events in one of the buildings. Another dataset was obtained by installing nine cameras in one of our labs. Each of these cameras was arranged to have a different point of view. Also, in order to have different levels of spatial and temporal complexities, we arranged a variety of
scene settings and simulated different level of details and activities for each scene setting. In total, we captured 108 unique videos that we used for our analysis. The video surveillance dataset is made available to public for further research [121].

In order to incorporate the effect of content, the spatial and temporal information (SI and TI) metric used by MPEG were utilized. The video surveillance dataset was further divided into different scene categories according to their SI and TI values. In our attempt to provide the nodes' encoding configuration allocation that minimizes the overall power consumption, we first assumed that the tuples of encoding configurations combinations along with its corresponding coding complexity and bitrate could be stored in the form of a lookup table. We then proceeded by assuming that in order to find the optimal encoder parameters settings for the test set, we can use the bitrate ratio information obtained from the training set. Note that, in this contribution, the scenes that have high content complexity were chosen as the training set. This is due to the fact that the scenes with a high complexity of content tend to have a higher bitrate than the scene with lower content information. We addressed the power minimization issue as a fairness allocation problem, where each node needs to be allocated a certain portion of the bitrate of the overall bandwidth such that the power consumption is balanced among all nodes. Comparison with the existing techniques showed a good performance. However, using a look-up-table is not practical nor efficient. We needed to design coding complexity and bitrate models to have a better performance.
3. Designed a novel optimization framework utilizing the models that incorporate video content features to minimize the video surveillance network's power consumption.

To proceed with this contribution, we needed to develop models to predict the coding complexity and bitrate of the videos. In Chapter 4, we considered the effect of GOP size and the number of block sizes candidates used in motion estimation to design the models. We used the first video surveillance dataset that contains 20 videos because at the time we started the work in this contribution, we were still generating the video surveillance dataset described in Chapter 3. The model parameters were obtained by analyzing the effect of these encoding parameters using four training videos. The model was then evaluated against the remaining videos. A power consumption study using the models confirmed that the content of the video affects the video surveillance network’s power consumption [122]. In order to include the effect of spatial and temporal information of video, we made several modifications to the developed models in Chapter 5. In addition, we also incorporated the effect of QP to account the trade-off between video quality and bitrate. The video surveillance dataset explained in Chapter 3 was used to develop and test our new model. Note that, a few scenes that represent the scene with the low, medium and high complexity of content (as classified in Chapter 3) were used to train the models. The models were evaluated against the remaining videos, excluding the ones used in the training. We evaluated the models’ accuracy in terms of Pearson’s correlation coefficient (PCC), root mean square error (RMSE), mean absolute percentage error (MAPE) and R-squared values. Comparison
with the existing techniques showed the superiority of our proposed models. Furthermore, we combined the model with the information about network layout and nodes’ position in the network in a novel optimization framework to minimize surveillance network’s power consumption. The objective of the framework is to find the nodes’ encoder parameters settings for the specific scene’s videos used that minimize the overall power consumption. The performance of the proposed framework for different scenes was shown to be superior to the existing techniques.
Chapter 2: Encoding and Communication Power Consumption Trade-off in a Video Surveillance Network

In this chapter, we propose a distance-based heuristic for energy efficient VSN for surveillance applications. The proposed method exploits the trade-off between encoding and communication power consumption by dividing the deployment area of the surveillance network into several zones. The video surveillance network nodes that are located in the same zone are configured to have the same encoding power consumption and bitrate. The encoding power consumption and bitrate in a zone are controlled by using variable frame rate. In order to estimate the encoding configuration per each node, we use the PRD model [60], [63]. Note that, the objective is to find the optimal zone and correspondingly the frame rate, i.e., bitrate, for each zone that minimize the overall power consumption.

![Figure 2.1: Encoding and transmission power consumption](image-url)
consumption. Section 2.1 and 2.2 describe the PRD model and the corresponding network maximization algorithm that will be used as the reference to evaluate the performance of the proposed heuristic. The proposed distance-based heuristic is provided in Section 2.3. Section 2.4 concludes the chapter.

2.1 Power-rate-distortion Model in Video Surveillance Networks

According to the PRD model proposed in [60], [105], the distortion of a compressed video $D$ depends on the bitrate $R$ and encoding power consumption $P_e$. In this regard, the encoding distortion measured in terms of mean squared error (MSE), is estimated using the following formula:

$$ D = \sigma^2 e^{-\gamma R P_e^{1/3}} $$

(2.1)

where $\sigma^2$ is the average input variance and $\gamma$ is the encoding efficiency coefficient. Hence, to achieve the encoding distortion requirement, one can either increase the encoding power or bitrate. Obviously, there is a trade-off where optimal power consumption of a source node can be achieved, as shown in Figure 2.1. Note that this figure is generated using the parameters mentioned in [63] and video distortion requirement of 100 MSE.

2.2 Network Lifetime Maximization in Video Surveillance Networks

Network lifetime is one of the most important performance measures of a video surveillance network. Depending on the application requirement, different definition of network lifetime exists in the literature [74], [123][104], [124]–[126]. However, for critical applications such as a video surveillance network, it is common to use the minimum node lifetime as the performance measure. The minimum node lifetime assumes that the system
is considered to be in the fail state when at least one of the nodes consumes its entire initial power source.

In order to model the video surveillance network, battery powered sensor nodes are assumed to be statically deployed in a square area characterized by its width. It is assumed that a standard medium access control (MAC) protocol is applied to resolve the link interference problem. The network is modeled as an undirected graph $G(N,L)$ where $N$ is the set of nodes and $L$ is the set of links. Node $i$ can communicate to node $j$ if a link between those nodes ($L_{ij} \in L$) exists. Sensor node $i$, can capture and encode video, and generate data traffic with a source bitrate $R_i$. Furthermore, each node can also relay the traffic from upstream nodes. The flow conservation law at each node is:

$$\sum r_{ij} - \sum r_{ki} = R_i.$$  \hspace{1cm} (2.2)

Here $r_{ij}$ denotes the outgoing rate at $L_{ij}$ while $r_{ki}$ denotes incoming rates at $L_{ki}$, and $L_{ij}, L_{ki} \in L$.

The total power dissipation at node $i$ consists of the transmission power consumption, the reception power consumption, and the encoding power consumption:

$$P_i = P_{ti} + P_{ri} + P_{ei}$$  \hspace{1cm} (2.3)

A general energy consumption model for a wireless communication transmitter and receiver as presented in [127] is used. The total transmission power consumption of node $i$ is the sum of all power consumed to transmit data to other nodes within its transmission range.

$$P_{ti} = \sum (\alpha + \beta \cdot d_{ij}^\eta) \cdot r_{ij}.$$  \hspace{1cm} (2.4)
Here, $\alpha$ and $\beta$ are constant coefficients and $\eta$ is the path loss exponent and $L_{ij} \in L$. Also, the total reception power consumption of node $i$ is the sum of all power consumed to receive data from other nodes, as formulated below, where $\lambda$ is a constant coefficient and $L_{ki} \in L$.

$$P_{ri} = \sum \lambda \cdot r_{ki}$$  \quad (2.5)$$

The encoding power consumption of node $i$ is determined using the PRD model [105]. For a given distortion requirement, the relation between encoding power consumption and bitrate can be stated as follows:

$$P^{2/3}e_i \cdot R_i = \frac{-\log(D_i / \sigma^2)}{\gamma}$$  \quad (2.6)$$

In this equation, $D_i$ is the distortion requirement for node $i$, $\sigma^2$ is the average input variance, and $\gamma$ is the encoding efficiency coefficient.

It is assumed that the node is powered by a battery with an initial amount of energy $B_i$. The lifetime of node $i$ is equal to:

$$T_i = B_i / P_i$$  \quad (2.7)$$
Minimum node lifetime states that $T_{net} = \min T_i, i \in N$. Maximizing $T_{net} = \min T_i$ is equivalent to maximizing $T_{net}$ subject to $T_{net} \leq T_i \forall i$. Therefore, the objective function can be reformulated as a linear program by introducing variables $q_{net} = 1/T_{net}$, $q_i = 1/T_i$ such that $P_i = B_i \cdot q_i$, and $q_i \leq q_{net}$.

The lifetime maximization of the video surveillance network is formulated as a constrained optimization problem as shown in Figure 2.2. This algorithm is similar to the centralized power-rate optimization proposed in [63] and it is called the Power Rate Optimal – Video Surveillance Network (PRO-VSN) algorithm in this thesis.

### 2.3 Distance Based Heuristics for Energy Efficient Video Surveillance Networks

In the intra-coding video mode, the average energy dissipation to encode one frame can be assumed to be constant for a video that has no scene change. In this regard, the encoding power consumption depends mostly on the frame rate of the video. Since the encoding power consumption is known, to achieve the distortion requirement, the bitrate can be calculated using the PRD model.

Furthermore, the nodes that are located closer to the sink usually need to relay the video data from the upstream nodes that are located farther from the sink. Due to this, the energy consumption of the nodes closer to the sink is usually higher than the upstream nodes. However, in order to minimize the overall power consumption and maximize its system lifetime, it is necessary to balance out energy consumption within all nodes [104], [128]. Therefore, the nodes that are closer to the sink should spend less power in encoding.
than the farther ones. In this regard, in order to achieve the target distortion and also maximize the system lifetime, we should control the bitrate and encoding power consumption on each node, estimated based on its position in the network, i.e., their distance from the sink [120].

2.3.1 Proposed Approach

In order to allocate nodes’ encoding power and bitrate based on its distance from the sink, we propose to divide the network deployment area into different zones (z) as shown in Figure 2.3. All nodes located in the same zone will be configured to have the same encoding power consumption and bitrate. A zone is defined as a section of the network covering an area whose distance to the sink is greater than \( r_{m-1} \) and smaller or equal to \( r_m \), \( 1 \leq m \leq Z_{\text{opt}} \). The number of zones (\( Z_{\text{opt}} \), \( 1 \leq Z_{\text{opt}} \leq Z_{\text{max}} \)) should be carefully chosen so that the network lifetime is maximized. On the other hand, the value of \( Z_{\text{max}} \) depends on the nodes’ distortion requirement and the hardware’s capability, as will be explained in the following

Figure 2.3: A Video surveillance network’s zones
paragraphs. Note that in Figure 2.3, sensor node $i$ that is located at a distance $d_i$ from the sink is a member of $Z_m$ if $r_{m-1} < d_i \leq r_m$. Each node has to become a member of exactly one zone. However, a zone may have zero members.

As stated earlier, for the same target video distortion, the encoding power consumption of nodes that are closer to the sink should be lower than the nodes farther from the sink. Consequently, the bitrate of nodes closer to the sink will be higher than the nodes farther from the sink. Therefore, the encoding power consumption and bitrate of any video surveillance network node in zone $m$ ($Z_m$) are dictated by the following rules: $P_1^z < P_2^z < \cdots < P_m^z < \cdots < P_{z_{\text{max}}}^z$ and $R_1^z > R_2^z > \cdots > R_m^z > \cdots > R_{z_{\text{max}}}^z$. Here, $P_m^z$ and $R_m^z$ are the encoding power consumption and bitrate allocated for $Z_m$. Furthermore, we know from the PRD model [105] that:

$$R_m^z \cdot (P_m^z)^{2/3} = R_i^z \cdot (P_i^z)^{2/3}$$  \hspace{1cm} (2.8)

Since the average power dissipation for encoding one frame is assumed to be constant, the encoding power consumption for any node in the first zone can be calculated as follows:

$$P_1^z = P_{1\text{frame}} \cdot f_1^z.$$  \hspace{1cm} (2.9)

Here $P_{1\text{frame}}$ is the average power dissipation to encoding one frame while $f_1^z$ is the frame rate of any node in the first zone. Using (2.8) and (2.9), we can calculate the encoding power dissipation and bitrate in zone $m$:

$$P_m^z = P_{1\text{frame}} \cdot (f_1^z + m - 1)$$  \hspace{1cm} (2.10)
\[ R_m^z = R_1^z \cdot \left( \frac{f_i^z \cdot \left( f_i^z + m - 1 \right)}{m} \right)^{2/3}. \]  

(2.11)

The value of \( f_1^z \) depends on the target distortion and the estimated value of bitrate at zone one. Since \( R_1^z \geq R_m^z \), the bitrate at zone one can be estimated as follows:

\[ \hat{R}_1^z = (1 + \tau) \cdot \frac{W}{V}. \]  

(2.12)

Here \( W \) is the bandwidth (maximum data rate) of the network, \( V \) is the number of video nodes and \( \tau \) is a small number. With \( D_1^z \) as the value of distortion requirement for zone one, the values of \( f_1^z \) and \( R_1^z \) can then be obtained as follows:

\[ f_1^z = \left[ \frac{-V \cdot \log \left( \frac{D_1^z \cdot \sigma^2}{\gamma \cdot (1 + \tau) \cdot W} \right)^{1.5}}{P_{frame}} \right] \]  

(2.13)

\[ R_1^z = -\log \left( \frac{D_1^z \cdot \sigma^2}{\gamma \cdot \left( f_1^z \cdot P_{frame} \right)^{2/3}} \right). \]  

(2.14)

Note that, unless it is specifically mentioned, it is assumed that the distortion requirement for all nodes is the same, \( D_1^z = D_i = D \).

The value of \( Z_{max} \) is obtained by considering the capability of the central processing unit (CPU) to process the captured video. Based on the results shown in [62], the number of cycles required to process one frame is approximately 20 million cycles. Considering that the maximum CPU frequency of the Stargate platform is 400 MHz [129], the maximum number of frames that can be processed in one second is approximately 16. Thus, the value of \( Z_{max} \) can be calculated as \( Z_{max} = 17 - f_1^z \).

Recall that our goal is to find the number of zones that minimizes the overall power consumption. Note that the value of optimal zone (\( Z_{opt} \)) is bound to be integral and solving a mixed integer linear problem is known to be intractable. However, since \( 1 \leq Z_{opt} \leq Z_{max} \) and
the value of $Z_{\text{max}}$ is relatively small, we can evaluate the overall power consumption of the video surveillance network for all possible number of zones values. On the other hand, the distance-based optimization itself is formulated as a constrained optimization problem shown in Figure 2.4. As can be seen from this figure, the bitrate allocated to node $i \in z_m$ is equal to $R_m^z$ that is calculated using (2.15). The value of $R_m^z$ depends on the value of the bitrate in the first zone, i.e., $R_1^z$, that is calculated using (2.16). Furthermore, the encoding power consumption allocated to node $i \in z_m$ is calculated using (2.17).

The algorithm to find the optimal number of zones and at the same time allocate the bitrate and encoding power consumption per each node is then proceed as follow:

1. Based on the distortion requirement, estimate the frame rate to be allocated at the first zone, i.e., $f_1^z$, using (2.18). Set the minimum power consumption ($P_{\text{min}}$) to be equal to infinity.

2. For $Z=1$ until $Z_{\text{max}}$, repeat:
a. Calculate the video surveillance network power consumption \( (P_{\text{net}}) \) using the \texttt{distanceBasedOptimization(Z)} procedure.

b. If \( P_{\text{net}} \) is smaller than \( P_{\text{min}} \), then set the following: \( P_{\text{min}} = P_{\text{net}} \) and \( Z_{\text{opt}} = Z \).

For ease of referencing, this approach is called the Power-Rate-hEuristic Video Surveillance Network (PRE-VSN) algorithm.

### 2.3.2 Experiments and Results

Table 2.1 shows the parameters used in the experiments. It is assumed that the maximum transmission range is 100m and the network bandwidth is 1 Mbps. It is clear from (3.4) that the transmission power consumption depends on the transmission distance. Accordingly, we assume that each node is able to adjust its transmission power to achieve the required performance at the receiver. The term network width denotes the width of the square area in which sensor nodes are statically deployed. The rate adjustment coefficient \( \tau \) is set according to the number of zones in the corresponding

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Initial energy (battery)</td>
<td>100 kJ</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Energy cost for transmitting 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>( \beta )</td>
<td>Transmit amplifier coefficient</td>
<td>( 1.3 \cdot 10^{-8} ) J/Mb/m4</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Energy cost for receiving 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>( \eta )</td>
<td>Path loss exponent</td>
<td>4</td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>Average variance of video (MSE)</td>
<td>3500</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>Encoding efficiency coefficient</td>
<td>55.54 W^{3/2}/Mbps</td>
</tr>
<tr>
<td>( \tau_{\text{max}} )</td>
<td>Max. rate adjustment coefficient</td>
<td>0.1</td>
</tr>
<tr>
<td>( P_{\text{1frame}} )</td>
<td>Power consumption to encode 1 frame</td>
<td>60 mJ/frame</td>
</tr>
</tbody>
</table>
experiment. Hence, $\tau$ is set equal to 0 when the number of zones is 1 and $\tau_{\text{max}}$ when the number of zones is equal to $Z_{\text{max}}$ respectively. In addition to that, the sensor nodes are deployed uniformly in a square grid layout where each node is deployed at the center of each grid.

### 2.3.2.1 Performance of the Proposed Technique

The performance of the proposed heuristic in terms of network lifetime and the optimal zone is shown in Figure 2.5. In Figure 2.5(a), the number of nodes is four and the
network width is 50m. The target distortion is varied from 50 to 800. On the other hand, Figure 2.5(b) depicts network lifetime when the number of nodes is nine, while the network width and the distortion requirement are similar to those in Figure 2.5(a). These graphs show some interesting insights. Firstly, the value of $Z_{\text{max}}$ depends on the distortion requirement and the network size. For example, when the distortion requirement is 800, the value of $Z_{\text{max}}$ for both cases are the same, i.e., $Z_{\text{max}}$ is equal to four. On the other hand, when the distortion requirement is equal to 200, the value of $Z_{\text{max}}$ is five for the four-node video surveillance network and three for the nine-node video surveillance network, respectively. Secondly, the network lifetime graph for lower distortion requirement is rather flat. This means that most of the power is consumed for encoding the captured video to achieve the target distortion.

Furthermore, it can be seen from Figure 2.6 that the value of the optimal number of zones is proportional to the network width. When the network is larger, the bitrate should be reduced to lower the communication power consumption. However, to compensate for the reduction in the bitrate, the video surveillance network’s nodes need to dissipate

![Figure 2.7: Performance of the proposed algorithm. (a) maximum network lifetime, (b) the number of zone producing maximum lifetime ($Z_{\text{opt}}$)](image-url)
higher encoding power to achieve the target distortion. Figure 2.6(a) shows the case when the number of nodes is four and the distortion requirement is equal to 400, while the network width is varied from 10m to 100m. Similar settings are used in Figure 2.6(b) where the number of nodes is increased to nine. In addition, Figure 2.7(a) shows the maximum network lifetime for these experiments, while Figure 2.7(b) shows the optimal number of zones to achieve those results.

2.3.2.2 Comparison with Other Techniques

Common techniques in sensor networks assume that all nodes are configured to use the same processing power consumption [44], [71]. These approaches can be classified as a special case of the proposed distance-based heuristic where the number of zones is equal to one. For ease of referencing, this technique is called Power-Rate-Allocated Video Surveillance Network (PRA-VSN).

Table 2.2: Average network lifetime and network lifetime offset, normalized to PRO-VSN

<table>
<thead>
<tr>
<th>Network width (m)</th>
<th>Normalized Lifetime (%)</th>
<th>Lifetime Offset (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRA-VSN</td>
<td>PRE-VSN</td>
</tr>
<tr>
<td>10</td>
<td>90.85</td>
<td>90.85</td>
</tr>
<tr>
<td>20</td>
<td>88.72</td>
<td>88.72</td>
</tr>
<tr>
<td>30</td>
<td>92.13</td>
<td>92.13</td>
</tr>
<tr>
<td>40</td>
<td>94.65</td>
<td>94.65</td>
</tr>
<tr>
<td>50</td>
<td>93.72</td>
<td>93.72</td>
</tr>
<tr>
<td>60</td>
<td>89.59</td>
<td>92.29</td>
</tr>
<tr>
<td>70</td>
<td>84.58</td>
<td>89.72</td>
</tr>
<tr>
<td>80</td>
<td>78.83</td>
<td>88.51</td>
</tr>
<tr>
<td>90</td>
<td>74.73</td>
<td>89.09</td>
</tr>
<tr>
<td>100</td>
<td>70.19</td>
<td>85.77</td>
</tr>
</tbody>
</table>
In order to compare the proposed approach with the existing techniques, we simulate video surveillance networks whose nodes are randomly deployed within an area whose width is varied between 10m to 100m. The experiments are repeated 10 times and the distortion requirement is set equal to 100. Two performance merits, normalized lifetime and lifetime offset, are used. Normalized lifetime is defined as the ratio of network lifetime obtained from the examined algorithms with the optimal one produced by PRO-VSN. Lifetime offset is the difference between the optimal network lifetime produced by PRO-VSN and the one obtained from the tested algorithms. Results are shown in Table 2.2 and Figure 2.8. Figure 2.8(a) compares the average network lifetime obtained using the proposed PRE-VSN algorithm to the ones obtained using the existing techniques. It can be seen that the proposed PRE-VSN performs better than PRA-VSN when the network width is greater than or equal to 60m. In addition to that, when the network width is smaller than or equal to 30m, the network lifetime obtained by PRE-VSN is at least 4 hours shorter than that obtained by PRO-VSN. However, the lifetime offset of PRE-VSN can be as little as 1.78 hours less than that achieved by PRO-VSN. This happens when the network width is

![Figure 2.8: Network lifetime comparison (a) exact values (b) normalized towards PRO-VSN](image-url)
greater than or equal to 40m. However, when the network width is greater than 40m, the
difference in terms of the average network lifetime obtained by the three algorithms is
small. Hence, in order to get a better understanding of the performance of the proposed
algorithm, the network lifetime values were normalized towards PRO-VSN as shown in
Figure 2.8(b). This figure shows that the network lifetime obtained by the proposed
algorithm is between 0.85% ~ 0.94% of the optimal solution obtained by the PRO-VSN
algorithm. These results show that the proposed distance-based heuristic can achieve a
comparable performance with the optimal solution. Note that the PRO-VSN algorithm can
provide better results because the encoding power is finely optimized. On the other hand,
the proposed PRE-VSN uses the zone-based system on which the encoding power
consumption per each zone is tied to the frame-rate of the video.

Figure 2.9 shows the average power consumption per each node when the network
width is 60m and distortion requirement is 100. It is shown that each algorithm is able to
control power dissipation such that all nodes spend roughly similar total power

![Figure 2.9: Average power consumption profile](image-url)
consumption. It can also be seen that in the PRA-VSN, all nodes dissipate similar encoding power consumption. However, the total power consumption obtained per each node obtained by the PRA-VSN is not the same. Therefore, it is clear that the proposed PRE-VSN performs better than the PRA-VSN, while the PRO-VSN has the best performance in term of minimum total power consumption.

2.4 Summary

In this chapter, the distance-based heuristic for video surveillance network’s encoding power and bitrate allocation is proposed. This is performed by dividing the video surveillance network deployment into several zones. All nodes in the same zone are allocated the same encoding power consumption and bitrate. In order to determine the zones’ encoding power consumption and bitrate, a heuristic to allocate the encoder parameter frame rate for each zone was proposed. The proposed technique shows that by assigning a different frame rate to each zone, the video surveillance network’s system lifetime can be optimized. However, in many video surveillance applications, it is often required that the captured videos are encoded at the same frame rate to guarantee the same video quality across all nodes. In addition, there are other encoder parameters that significantly affect the trade-off between encoding and communication power consumption in video surveillance networks. Furthermore, the videos captured by the surveillance nodes reflect the different point of views of the scene, i.e., each node captures unique video with different content. The different content captured by each node affect the distribution of power consumption in the video surveillance network. The effect of encoder parameters
settings and the complexity of content captured by various nodes in the video surveillance network are the focus of the discussion in the next chapter.
Chapter 3: Encoding Parameters and Scene’s Content Complexity in Video Surveillance Networks

In a resource constrained environment such as video surveillance networks, it is very important to choose the right configuration and setting parameters that lead to the optimal coding performance. On the other hand, the coding complexity and bitrate of videos also depend on the content of the video. This chapter studies the effect of encoding parameter settings and video content towards coding complexity and bitrate. An algorithm to minimize the video surveillance network power consumption considering the above-mentioned factors is also proposed in this chapter.

3.1 Video Encoding Parameters and Node’s Power Consumption

H.264/AVC is a block-based hybrid video coding that reduces the bitrate generated by the encoder by exploiting source statistics. For this purpose, intra prediction technique are used to reduce redundancy within one frame; while inter prediction technique with motion estimation (ME) algorithm is used to exploit redundancies among subsequent frames. In terms of coding complexity, inter prediction is more involved than intra prediction technique. However, inter-coded frames have smaller bitrate than the intra-coded ones. Some initial observations of the encoder’s performance led us to a low complexity encoding configuration as shown in Table 3.1 [130]. This configuration will be used as the basis parameter settings used in this chapter.
Note that due to the limitation in the energy and processing resources of video surveillance networks, less complex encoder configurations are deployed. To this end, the baseline profile of H.264/AVC that is suitable for low complexity applications is used. Therefore, only I and P frames are used. The other encoding settings used in this thesis include the use of context-adaptive variable-length coding (CAVLC) entropy coding, one reference frame, SR equal to eight, while the rate distortion optimization (RDO), rate control, and the deblocking filter are disabled.

### 3.1.1 H.264/AVC Coding Performance Analysis

In order to analyze the relation between encoding configuration and power consumption, there is a need to investigate the effect of different encoder parameter settings towards the encoder’s coding complexity and bitrate. In order to perform our analysis, the H.264/AVC software, JM version 18.2 [131] is used in all experiments. For our

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of reference frame</td>
<td>1</td>
</tr>
<tr>
<td>Smallest motion compensation block size</td>
<td>8x8</td>
</tr>
<tr>
<td>Entropy coding</td>
<td>CAVLC</td>
</tr>
<tr>
<td>B Slices</td>
<td>None</td>
</tr>
<tr>
<td>Subpel motion compensation</td>
<td>Disabled</td>
</tr>
<tr>
<td>Rate Distortion Optimization</td>
<td>Off</td>
</tr>
<tr>
<td>Rate Control</td>
<td>Off</td>
</tr>
<tr>
<td>Deblocking filter</td>
<td>Disabled</td>
</tr>
<tr>
<td>Motion estimation search range (SR)</td>
<td>2</td>
</tr>
<tr>
<td>Group of pictures (GOP) size</td>
<td>1</td>
</tr>
<tr>
<td>Quantization parameter (QP)</td>
<td>34</td>
</tr>
</tbody>
</table>

Note that due to the limitation in the energy and processing resources of video surveillance networks, less complex encoder configurations are deployed. To this end, the baseline profile of H.264/AVC that is suitable for low complexity applications is used. Therefore, only I and P frames are used. The other encoding settings used in this thesis include the use of context-adaptive variable-length coding (CAVLC) entropy coding, one reference frame, SR equal to eight, while the rate distortion optimization (RDO), rate control, and the deblocking filter are disabled.

### 3.1.1 H.264/AVC Coding Performance Analysis

In order to analyze the relation between encoding configuration and power consumption, there is a need to investigate the effect of different encoder parameter settings towards the encoder’s coding complexity and bitrate. In order to perform our analysis, the H.264/AVC software, JM version 18.2 [131] is used in all experiments. For our
initial analysis, two representative test videos are selected from the data set provided by MPEG for HEVC call for proposal [132]. They are the “BQMall” (832x480 pixels, frame rate=60) and “Traffic” (2560x1600 pixels, frame rate = 30). These sequences are downsampled to produce 15 frames per second (fps) videos and resized to obtain the following resolutions: 4CIF (4 times of common intermediate format, 704x576 pixels) and CIF (common intermediate format, 352x288 pixels).

### 3.1.1.1 Encoding Complexity Measure

The coding complexity is usually measured based on the encoder’s computation time, which may not be accurate since it widely depends on the device architecture and the optimization level of the algorithms used (e.g., if GPU or video processors are used) and whether CPU is involved with other processes than encoding. To have an accurate measure of the coding complexity, the instruction level profiler *iprof* [133] that provides the total number of basic instruction counts to execute a task is used. Figure 3.1(a) and Figure 3.1(b) illustrate the complexity measure using encoding time and the instruction counts (IC) provided by *iprof*, respectively. The video sequence used to obtain the plots in the figure is

![Figure 3.1: Complexity measure of different GOP sizes and QP values (a) encoding time, (b) the number of instruction counts](image)
the BQMall 15 fps CIF video. Note that when GOP>1 the inter prediction module in the encoder were set to use SR equal to two as stated in Table 3.1. It can be seen from the figure that the number of instruction count provides a more objective and accurate measure of coding complexity than the encoder’s computation time.

3.1.1.2 The Effect of GOP Size towards Coding Complexity and Bitrate

The GOP size determines the number of successive inter-coded frames between intra-coded frames within a coded video stream. Increasing GOP size will increase the number of inter-frame coded frames. Figure 3.1(b) shows that the IC is increasing as the GOP size increases. It can be seen that the minimum IC happens when GOP size is equal to one. This is expected since the ME algorithm in the encoder is disabled when GOP=1. As it is observed in Figure 3.1(b), the encoder needs around 16 billion of instructions to encode video with GOP=1 configuration. On the other hand, it needs 22 billion of instructions for GOP=4. Furthermore, Figure 3.2 shows the bitrate and the corresponding peak signal to noise ratio (PSNR) of BQmall 15fps CIF video sequence encoded with the configuration

![Figure 3.2: Rate distortion plot for different QP values and GOP sizes](image)

50
settings similar to the one used in Figure 3.1(b). It is observed that the bitrate decreases as the GOP size increases. For example, when QP is equal to 34, the bitrate is around 800 kbps for GOP=1 and 570 kbps for GOP=4. Figure 3.1(b) and Figure 3.2 show that GOP size affects the coding complexity and bitrate significantly.

3.1.1.3 The Effect of QP towards Coding Complexity and Bitrate

The quantization parameter (QP) regulates how much spatial detail is saved, affecting the quality of the encoded video. When QP is very small, almost all of that detail is retained, but at the price of higher complexity and bitrate. Figure 3.1(b) shows the effect of QP towards coding complexity. The figure shows that the coding complexity decreases as the QP increases. A similar observation can be seen for the bitrate (please see Figure 3.2), such that the bitrate decreases as the QP increases.

3.1.1.4 The Effect of SR towards Coding Complexity and Bitrate

The JM 18.2 software provides a number of ME algorithms that can be used in the encoder. These algorithms differ in the method of exploring the search area, whose size is

![Figure 3.3: The effect of different SR values to bitrate, complexity, and PSNR (a) BQmall 15 fps CIF video, GOP=75, QP=28 (b) Traffic 15 fps CIF video, GOP=75, QP=28](image)

51
determined by the SR, for suitable block candidates in the reference frames. The EPZS algorithm that shown to provide the best trade-off in terms of coding complexity and performance is chosen to be used in all experiments [130]. Increasing the SR may provide us with a better prediction, i.e., better compression. However, it comes with the price of increasing coding complexity. Thus, the trade-off between coding complexity and bitrate also depends on the value of SR. Figure 3.3(a) shows the effect of using different values of SR to the bitrate, video quality, and coding complexity of BQMall 15 fps CIF video sequence. On the other hand, Figure 3.3(b) shows the plots obtained from Traffic 15 fps CIF video.

Figure 3.3 shows that video quality in terms of PSNR is almost the same regardless of the value of SR used to encode the video. However, the complexity of the encoder increases with the increase of SR. On the other hand, the bitrate of the encoded video decreases with the increase of SR, especially when SR is increased from 2 to 8. The bitrate reduction obtained from increasing the SR from 8 to 16 is not really significant. However, when the resolution is increased into 4CIF, we notice that increasing SR up to 16 is still beneficial in terms of bitrate saving [130]. Some interesting observations are thus obtained:

1. For the same value of QP, video quality is almost the same, regardless of the size of SR. However, there is a trade-off between coding complexity and bitrate when different values of SR is used;

2. The optimum SR value that should be used for a specific video sequence depends on the resolution of the video; and
3. Video content affects the PSNR of the encoded video. The QPs used in Figure 3.3(a) and Figure 3.3(b) are the same. Hence the total number of instructions used for both video sequences are roughly the same. However, it is observed that the quality of the encoded Traffic sequence is around 1.5 dB lower than the BQmall sequence.

3.1.2 Coding Complexity and Bitrate Trade-off

Figure 3.4 shows coding complexity (shown as IC) and bitrate for different values of GOP and SR of the “BQMall” video. The figure shows that increasing GOP size results in higher IC but reduces the bitrate. For the same GOP size, increasing SR also increases the IC and reduces the bitrate. However, there are some configuration settings that are not optimal as shown by the bold line in the figure, i.e., point A, B, and C, which occurs when GOP>2 and SR=2. It is interesting to note that the configuration settings of point B (GOP=8, SR=4,8,16) results in lower bitrate and higher IC.

Figure 3.4: Trade-off between coding complexity and bitrate for BQmall 15 fps CIF video, GOP={1,2,4,8,75}, SR={2,4,8,16}, QP=28
SR=2) and point D (GOP=4, SR=4) have almost IC. However, the bitrate produced by configuration D is smaller than configuration B. Removing points A, B, and C from the plot produces the dashed-line plot illustrated in Figure 3.4. These configuration settings are translated into a tabular format as shown in Table 3.2. It has to be noted that, these settings are used on top of the configuration detailed in Table 3.1. In addition to that, it is worth noting that the coding complexity and bitrate of video can be used to estimate the video surveillance network’s encoding and communication power consumption. This will be explained in more detail in the following subsections.

Table 3.2: Configuration ID

<table>
<thead>
<tr>
<th>Configuration ID (CID)</th>
<th>GOP</th>
<th>Search Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>75</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>75</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>75</td>
<td>16</td>
</tr>
</tbody>
</table>
3.1.3 The Effect of Encoder Parameter Settings towards Node`s Power Consumption

The total energy dissipation of a video surveillance node consists of encoding power consumption and communication power consumption.

3.1.3.1 Encoding Power Consumption

One of the most important parameter in estimating node's power consumption is the average cycle per instruction (CPI) [134]. If the total number of instructions used to execute a task is known, the energy depleted to execute that task can be estimated as the multiplication of the total number of cycles to execute that task and the average energy depleted per cycle. The number of CPU cycles required to execute a task can be calculated as the elapsed time of the encoding task multiplied by the CPU frequency. However, as Figure 3.1 shows, the time elapse (encoding time) is not as accurate as the instruction count provided by iprof. Therefore, the total number of cycles to execute a task in this thesis is calculated as the multiplication of the instruction count (IC) and CPI. The average energy consumption required to encode a frame is then estimated as:

$$E_{ef} = \frac{IC \cdot CPI \cdot E_{ec}}{N_f}$$

(3.1)

where IC is the total number of instructions to encode a sequence (provided by iprof), CPI is the average number of cycles per instruction of the CPU, $E_{ec}$ is the energy depleted per cycle and $N_f$ is the number of frames. With $F_r$ denotes the video frame rate, the encoding power consumption is then calculated as follows:

$$P_e = E_{ef} \cdot F_r.$$  

(3.2)
### 3.1.3.2 Communication Power Consumption

Node’s communication power consumption consists of transmission and reception power consumption (see Equation (2.4) and (2.5)). In the case of multi-hop transmission, the total communication power consumption dissipated for transmitting the encoded video stream from node $i$ to the sink, separated by $n$-hops is calculated as follow [102]:

$$P_c = n \cdot P_t + (n-1) \cdot P_r.$$  \hspace{1cm} (3.3)

### 3.1.3.3 Encoding and Communication Power Consumption Trade-off

Recall that Table 3.2 shows the configuration IDs (CIDs) producing different pairs of coding complexity and bitrate. The coding complexity and bitrate determine the encoding and communication power consumption of a node in video surveillance networks. Therefore, Table 3.2 can be used to analyze the trade-off between encoding and communication power consumption. The same general parameters that were shown in Table 3.3 is used to analyze the trade-off between computation and communication power consumption.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>Initial energy (battery)</td>
<td>100 kJ</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Energy cost for transmitting 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Transmit amplifier coefficient</td>
<td>$1.3 \cdot 10^{-8}$ J/Mb/m$^4$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Energy cost for receiving 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Path loss exponent</td>
<td>4</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>Average variance of video (MSE)</td>
<td>3500</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Encoding efficiency coefficient</td>
<td>55.54 W$^{3/2}$/Mbps</td>
</tr>
<tr>
<td>$\tau_{\text{max}}$</td>
<td>Max. rate adjustment coefficient</td>
<td>0.1</td>
</tr>
<tr>
<td>$P_{1\text{frame}}$</td>
<td>Power consumption to encode 1 frame</td>
<td>60 mJ/frame</td>
</tr>
</tbody>
</table>
Figure 3.5: Node's power consumption: (a) single hop (b) multi-hops transmission

consumption. In addition, the following parameters were also used: energy depleted per cycle \((E_c)\) is 1.215 nJ [62] while the average CPI is 1.78 [135].

Figure 3.5 (a) shows the encoding and communication power consumption for all CIDs shown in Table 3.2 and QP=28 for a single hop transmission to the sink. In this figure, it is assumed that the transmission distance is 50m. The figure shows that for CID=1, the encoding power consumption of the node is equal to 7.2 Watt while the transmission power consumption is around 0.87 Watt. However, the encoding power consumption is increased to roughly 10.9 Watt while the transmission power consumption is reduced to around 0.49 Watt when the CID=14 configuration is used. This figure shows the effect of using different encoder configuration settings, i.e., different CID, towards node’s power consumption. As it is observed, a node can save transmission power consumption by performing a more complex encoding process, i.e., to get a lower bitrate. On the other hand, when the node is required to save encoding power consumption, it can afford to spend more energy on transmission. Thus, the video surveillance network’s nodes can adjust their encoder parameter settings according to the CID table when the need arises. For example,
as Figure 3.5(b) shows, the optimal configuration for the node that is one hop away from the sink is the CID=1 configuration. However, when the nodes are four hops away from the sink, the node needs to use CID=10 configuration in order to reduce its power consumption. These results show that encoder parameter settings affect the node’s power consumption. Furthermore, it can also be seen from Figure 3.5(b) that there is an intersection point of the different CIDs. This intersection illustrates the condition at which the computation and communication power consumption trade-off, i.e., because of using different CIDs, reach a balance.

In some applications, the video surveillance network’s nodes need to transmit the encoded video stream through a more complex network layout. This creates power consumption and bitrate fairness issue between the nodes in a video surveillance network. This will be discussed in the following sections.

### 3.1.4 Fairness-based Node’s Encoder Configuration Allocation

Video nodes in a surveillance network share the same wireless medium in order to send their encoded video to the sink. Since the bandwidth, i.e., total data rate, allocated for

<table>
<thead>
<tr>
<th>CID</th>
<th>GOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>64</td>
</tr>
</tbody>
</table>
the whole nodes are usually limited, there is an issue of fairness of bandwidth allocated for each node. Furthermore, in many video surveillance’s deployment scenarios, nodes closer to the sink are required to relay the information from the nodes farther from the sink. The total energy consumption of nodes closer to the sink is thus usually greater than the nodes further. This problem can be alleviated by allocating different fairness ratio per each node by exploiting the trade-off between encoding and communication power consumption.

3.1.4.1 Simplified Configuration ID Table

As mentioned in Section 3.1.1.4, the effect of SR towards coding complexity and bitrate depends on the video content. In order to develop a more general model, the configuration ID (CID) table is simplified by setting SR equal to eight as suggested by [106] for all CIDs. In addition to that, since GOP size provides a significant effect on coding complexity and bitrate, the number of GOP size options is increased. The modified CID table is then shown in Table 3.4. Note that, the values used in the table is chosen such that there is a relation between the GOP size and the CID, i.e., \( CID = \log_2(2 \cdot GOP) \) [136].

3.1.4.2 Fairness Constraint

Using the end-to-end fairness constraint (\( \rho \)), the bitrate allocated to a node is controlled using the following relation [137]:

\[
R_i \leq \rho_i \cdot B
\]

where \( B \) denotes the network bandwidth of the video surveillance network. In this equation, \( \rho \) is fairness constraint, i.e., the maximum percentage of the total bitrate that can
be sent to the sink by each node. Furthermore, the sum of transmission rate of all nodes is constrained to be:

$$\sum_{j=1}^{N} R_j \leq B.$$  

(3.5)

Using (3.4), node $i$ can only generate a flow to the sink that is lower than a fraction of $\rho_i$ of the network bandwidth. It has to be noted that when all nodes are using the same fairness constraint, $\rho_i=1/N$, each node will be allocated equal transmission rate. In this condition, the video surveillance network is called to use the maximum fairness scheme.

### 3.1.4.3 Exhaustive Search Fairness-based Algorithm

In [136], the exhaustive fairness based search for minimum power video surveillance network was proposed. The basic idea of the method is to find the fairness allocation per each node such that the power consumption is minimized. The algorithm assumes that the fairness ratio decreases linearly from the node closest to the sink towards the node farthest from the sink. In other words, the nodes closer to the sink is assigned a smaller CID (which entails high bitrate but low coding complexity) than the nodes farther
from the sink. In order to find the best fairness allocation to each node, an exhaustive search of all possible CID combinations were performed. It was reported that for a linear topology with six nodes, the exhaustive fairness-based algorithm manages to reduce the video surveillance network's energy consumption by 11%. However, the algorithm proposed in [136] also assumed that all nodes have the same video content.

3.2 Relation between Video Content, Bitrate and Coding Complexity

Earlier studies on video surveillance network's power consumption assume that all nodes use the same video content. However, this assumption is a very simplistic approach, and it does not represent a real video surveillance deployment where different nodes capture the scene from the different point of views. Therefore, the video content is not consistent over different nodes. Note that the performance of a video encoder in terms of encoding complexity and bitrate also depends on the spatial and temporal complexity of the content. That elevates the importance of determining the trade-off between encoding and communication power consumption in a video surveillance network.
3.2.1 Video Surveillance Dataset

To obtain representative data with different temporal and spatial complexity levels, nine HD (high definition) cameras were installed in one of our labs. The cameras are installed such that each of them has a different field of view (FoV) while there are still overlaps among the cameras’ FoVs. The scene arrangement is such that the motion and activities are not centered in the middle of the field and the complexity level of content captured by each camera is different. With the assumption that important activities in surveillance applications partially occur around the entrance door, the middle camera (see camera 5 in Figure 3.6) is directed towards the door. Furthermore, we modified the layout of the lab to represent three different scenarios, namely “office”, “classroom”, and “party”.

Figure 3.8: Snapshots of the “office” scene in the first activity setting from (a) camera2 and (b) camera6; and snapshots the third activity setting: (c) camera2 and (d) camera6
Figure 3.7 shows the room set up for the different scenarios. To have a representative database with different activity levels, each scene is captured several times and the captured videos for each scene are categorized into four groups as follows:

1. The level of activity of all the people in the room is high, and the total number of people is between six to eight,

2. Three or more people moving around the room, while the total number of people in the room is around six,

3. Couple of people walking around the room, while the total number of people in the room is around five,

4. About four people walking into the room

Figure 3.9: Snapshots of the “classroom” scene from (a) camera2 and (b) camera4; and the snapshots of the “party” scene from (c) camera2 and (d) camera4
Capturing each scenario ("office", "classroom" and "party") at four different activity levels using nine cameras, results in 108 HD video sequences for our study. Each video is 10 seconds long with a frame rate of 60 per seconds (fps). In order to simulate practical low power video surveillance networks applications, the captured videos are downsampled to 416x240 pixels, 15 fps. For ease of referencing, the following video identification is used hereafter: <camera-id_scene-setting_activity-level>. For example, camera2_party_act1 refers to the video captured by camera2 in the "party" scene when the activity level of the scene falls into the first setting as mentioned above. These downsampled videos have been made publically available [121].

Figure 3.8 shows snapshots of the "office" scene from camera2 and camera4 when the activity level of the scene falls into the first and third settings. As it is observed, the video content from the two cameras and the two activity settings are not the same. Figure 3.9 shows snapshots from the "classroom" and "party" scenes when the activity level of the scene falls into the second setting.
3.2.2 Effect of Scene Complexity towards Coding Complexity and Bitrate

For the purpose of analysis, the same CID table shown in Table 3.4 is used. Figure 3.10(a) shows the complexity and bitrate plot of the camera2_party_act2 video encoded with different CIDs and QPs. As it is observed, when CID is small, the compression performance of the encoder is sacrificed such that the bitrate is high. However, this is compensated with a low coding complexity. On the other hand, bigger CID means using higher coding complexity to gain a better compression performance.

Figure 3.10(a) also shows that reducing the QP increases both coding complexity
and bitrate. Therefore, the bitrate and the complexity of the encoding process depend on the CID and QP used. It has to be noted that although the GOP size’s gap between CID=6 (GOP=32) and CID=7 (GOP=64) is high i.e., the GOP size gap is 32; the different in coding complexity (bitrate) between these two CIDs is very small. Furthermore, as mentioned in Section 3.1.4, there is a relation between the GOP size and the CID, i.e., \( CID = \log_2(2 \cdot \text{GOP}) \). This shows that CID values represent an encoder parameter, i.e., the GOP size, whose values affect the coding complexity and bitrate of the encoder. In terms of video quality, Figure 3.10(b) shows that for the same QP, the PSNR is almost the same, i.e., the difference is less than 0.5 dB, regardless of which CID value is used to encode the video.

### 3.2.3 Scenes’ Complexity Measurement and Classification

Figure 3.11(a) shows the complexity and bitrate of videos captured by three different cameras in the “party” scene at the same activity level while Figure 3.11(b) shows the complexity and bitrate of videos captured by camera2 in the “party” scene at different activity levels. Figure 3.11(a) shows that the coding complexity and bitrate of videos depend on the camera point of view at the scene. On the other hand, Figure 3.11(b) illustrates that the coding complexity and bitrate of videos captured at the same scene by the camera depends on the content’s motion activity levels. Furthermore, Figure 3.12 illustrates the coding complexity and bitrate of some videos captured by camera2 at similar activity setting but different scenes. This figure shows that the coding complexity and bitrate trade-off of the videos depends on the scene settings, i.e., spatial detail of the scene. Figure 3.11 and Figure 3.12 show that the coding complexity of the videos captured by each camera depends on the spatial (camera views and scene settings) and temporal (motion
activity levels) complexity of content. Moreover, the videos with a high spatial and temporal complexity tend to have a higher bitrate than the videos with lower content complexity. Consequently, the total bitrate generated by the captured scenes with high spatial and temporal detail will be higher than the ones obtained from the scenes with low spatial and temporal detail. In this regard, we proceed by assuming that the bitrate allocation information at the worst scenario, i.e., when the total bitrate is high, can be used as an initial guide to allocate the configuration settings in the other scenarios. Therefore, the scenes with higher content complexity are used as the training set for the remaining scenes.

Figure 3.13: The complexity and bitrate of some videos captured at different scene settings
Table 3.5: Spatial Information unit (SI) of all videos

<table>
<thead>
<tr>
<th>Scene</th>
<th>cam1</th>
<th>cam2</th>
<th>cam3</th>
<th>cam4</th>
<th>cam5</th>
<th>cam6</th>
<th>cam7</th>
<th>cam8</th>
<th>cam9</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>office_act1</strong></td>
<td>100.52</td>
<td>97.83</td>
<td>88.92</td>
<td>76.17</td>
<td>84.60</td>
<td>89.95</td>
<td>85.73</td>
<td>95.06</td>
<td>94.98</td>
</tr>
<tr>
<td><strong>office_act2</strong></td>
<td>85.83</td>
<td>90.26</td>
<td>85.90</td>
<td>71.24</td>
<td>76.14</td>
<td>84.45</td>
<td>85.20</td>
<td>90.06</td>
<td>88.62</td>
</tr>
<tr>
<td><strong>office_act3</strong></td>
<td>87.78</td>
<td>88.68</td>
<td>89.78</td>
<td>74.30</td>
<td>67.62</td>
<td>91.16</td>
<td>89.93</td>
<td>90.80</td>
<td>92.75</td>
</tr>
<tr>
<td><strong>office_act4</strong></td>
<td>84.97</td>
<td>87.01</td>
<td>87.96</td>
<td>71.40</td>
<td>61.17</td>
<td>89.97</td>
<td>83.85</td>
<td>87.36</td>
<td>87.43</td>
</tr>
<tr>
<td><strong>classroom_act1</strong></td>
<td>94.62</td>
<td>99.18</td>
<td>96.73</td>
<td>89.56</td>
<td>97.46</td>
<td>96.04</td>
<td>92.03</td>
<td>88.12</td>
<td>83.59</td>
</tr>
<tr>
<td><strong>classroom_act2</strong></td>
<td>96.30</td>
<td>94.46</td>
<td>96.30</td>
<td>91.46</td>
<td>74.07</td>
<td>88.83</td>
<td>92.74</td>
<td>85.52</td>
<td>84.99</td>
</tr>
<tr>
<td><strong>classroom_act3</strong></td>
<td>89.67</td>
<td>94.11</td>
<td>90.04</td>
<td>85.71</td>
<td>77.42</td>
<td>87.03</td>
<td>88.94</td>
<td>85.85</td>
<td>82.55</td>
</tr>
<tr>
<td><strong>classroom_act4</strong></td>
<td>101.69</td>
<td>101.21</td>
<td>94.74</td>
<td>92.68</td>
<td>89.08</td>
<td>94.57</td>
<td>91.86</td>
<td>89.19</td>
<td>83.64</td>
</tr>
<tr>
<td><strong>party_act1</strong></td>
<td>114.85</td>
<td>112.10</td>
<td>116.85</td>
<td>99.60</td>
<td>81.27</td>
<td>104.73</td>
<td>104.60</td>
<td>110.79</td>
<td>105.97</td>
</tr>
<tr>
<td><strong>party_act2</strong></td>
<td>115.40</td>
<td>114.49</td>
<td>107.53</td>
<td>90.61</td>
<td>95.06</td>
<td>101.31</td>
<td>104.56</td>
<td>111.56</td>
<td>101.90</td>
</tr>
<tr>
<td><strong>party_act3</strong></td>
<td>113.43</td>
<td>112.04</td>
<td>113.87</td>
<td>99.02</td>
<td>76.15</td>
<td>96.36</td>
<td>107.40</td>
<td>111.86</td>
<td>107.08</td>
</tr>
<tr>
<td><strong>party_act4</strong></td>
<td>106.64</td>
<td>103.51</td>
<td>111.48</td>
<td>89.54</td>
<td>93.97</td>
<td>95.22</td>
<td>101.28</td>
<td>102.44</td>
<td>105.30</td>
</tr>
<tr>
<td>Scene</td>
<td>cam1</td>
<td>cam2</td>
<td>cam3</td>
<td>cam4</td>
<td>cam5</td>
<td>cam6</td>
<td>cam7</td>
<td>cam8</td>
<td>cam9</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>office_act2</td>
<td>14.59</td>
<td>19.21</td>
<td>18.46</td>
<td>22.00</td>
<td>19.43</td>
<td>23.26</td>
<td>17.33</td>
<td>15.10</td>
<td>15.47</td>
</tr>
<tr>
<td>office_act3</td>
<td>13.10</td>
<td>18.34</td>
<td>18.77</td>
<td>20.77</td>
<td>13.73</td>
<td>20.90</td>
<td>17.04</td>
<td>17.20</td>
<td>16.51</td>
</tr>
<tr>
<td>office_act4</td>
<td>14.05</td>
<td>16.76</td>
<td>16.21</td>
<td>17.74</td>
<td>1.74</td>
<td>15.87</td>
<td>14.55</td>
<td>18.08</td>
<td>18.60</td>
</tr>
<tr>
<td>classroom_act1</td>
<td>18.05</td>
<td>18.39</td>
<td>18.42</td>
<td>17.47</td>
<td>19.47</td>
<td>19.56</td>
<td>11.04</td>
<td>11.21</td>
<td>7.38</td>
</tr>
<tr>
<td>classroom_act2</td>
<td>12.04</td>
<td>16.78</td>
<td>12.75</td>
<td>15.14</td>
<td>12.25</td>
<td>13.82</td>
<td>13.85</td>
<td>7.60</td>
<td>11.58</td>
</tr>
<tr>
<td>classroom_act3</td>
<td>15.18</td>
<td>15.88</td>
<td>11.79</td>
<td>13.76</td>
<td>17.64</td>
<td>14.98</td>
<td>9.65</td>
<td>11.07</td>
<td>4.75</td>
</tr>
<tr>
<td>party_act1</td>
<td>22.45</td>
<td>27.82</td>
<td>21.44</td>
<td>27.15</td>
<td>19.59</td>
<td>21.93</td>
<td>17.15</td>
<td>18.27</td>
<td>17.98</td>
</tr>
<tr>
<td>party_act3</td>
<td>13.75</td>
<td>17.83</td>
<td>14.74</td>
<td>17.05</td>
<td>10.92</td>
<td>15.01</td>
<td>13.93</td>
<td>13.50</td>
<td>11.72</td>
</tr>
<tr>
<td>party_act4</td>
<td>16.59</td>
<td>16.78</td>
<td>17.00</td>
<td>13.17</td>
<td>18.25</td>
<td>15.32</td>
<td>11.10</td>
<td>13.31</td>
<td>10.83</td>
</tr>
</tbody>
</table>
In order to find the scenes with higher activity content, a methodology to classify each scene into different content complexity level is needed. For that purpose, the ITU-T recommendation that includes the use of spatial information (SI) and temporal information (TI) defined in [138] is used:

\[
SI = \max_{time} \left\{ std_{space} \left[ Sobel(F_n) \right] \right\} \\
TI = \max_{time} \left\{ std_{space} \left[ F_n - F_{n-1} \right] \right\}
\]

(3.6) \hspace{2cm} (3.7)

SI and TI measure the spatial and temporal activity level of videos. In this regard, Figure 3.13 shows the SI and TI values of camera1, camera2, camera5, and camera9. It can be seen from this figure that for the same scene, each camera has different spatial and temporal activity level. Table 3.5 shows the SI values of all videos while Table 3.6 shows the TI values, respectively. Furthermore, in order to classify the scenes into different content complexity level, the following procedure is used:

1. Classify each video from a scene into different SI and TI classes using the following threshold:

\[
t_L^{CC} = \text{mean}(CC) - 0.5 \cdot \text{std}(CC) \\
t_H^{CC} = \text{mean}(CC) + 0.5 \cdot \text{std}(CC)
\]

(3.8) \hspace{2cm} \hspace{2cm} \hspace{2cm} CC = \{SI, TI\}

2. The value of \( t_L^{CC} \) and \( t_H^{CC} \) are used to classify whether a specific video has a low, medium or high SI or TI. For example, if a specific video’s SI is less than \( t_L^{SI} \), the video is classified as low-SI_video. If the video’s SI is higher than \( t_H^{SI} \), it is classified as high-SI_video. If the video’s SI is between \( t_L^{SI} \) and \( t_H^{SI} \), the video is classified as medium-
Using (3.8) on the data in Table 3.4 and Table 3.5, the following threshold values are obtained: \( t_{L}^{SI} = 87.85 \), \( t_{H}^{SI} = 98.95 \), \( t_{L}^{TI} = 14.32 \), and \( t_{H}^{TI} = 19.04 \).

3. Based on the SI(TI) classes of the videos, we can then classify the scene into different SI(TI) classes using the following rules:

   a. The SI(TI) class of a scene is equal to the majority of SI(TI) classes of all videos from that scene

   b. If no majority is found, the scene is classified as medium SI(TI) scene

   Figure 3.14 shows an example of how to determine the \textit{office\_act1} and \textit{classroom\_act1} scene class. Using the rules mentioned in the previous paragraphs, \textit{office\_act1} is classified as \textit{medium-SI\_high-TI\_scene}. On the other hand, scene \textit{classroom\_act1} is classified as \textit{medium-SI\_medium-TI\_scene}.

   Recall that the scenes with high SI produce videos with higher bitrate than the scenes with lower SI. This is especially true for CID configuration one, i.e., the video is coded with intra prediction only. Therefore, the configuration setting that is suitable for a specific SI class may not be suitable to be used for the other SI classes. Based on this assumption, the scenes are arranged into three different sets, namely scenes that have high,
medium, and low SI. In each of these sets, the scene with the highest TI will be selected as the training scene. For example, in Figure 3.15, the scenes with medium SI are the office_act1, office_act3, classroom_act1, classroom_act2, and classroom_act4. Out of these five scenes, the scene that has the highest TI class, i.e., office_act1, is selected as the training scene for the medium SI scenes. If there is more than one candidate for the training scene, the scene that has the highest average TI will be selected as the training scene. Henceforth, the training set for the high SI scenes is party_act2, the training set for the medium SI scenes is party_act2, classroom_act1, classroom_act2, classroom_act4, and classroom_act3 is selected as the training set for the low SI scenes.
scenes is *office_act1*, and the training scene for the low SI scenes is *office_act2*. Note that the training scenes are shown in bold in Figure 3.15.

Table 3.7 shows the training scenes and their corresponding test scenes. We will use the training scenes to find the fairness ratio allocation per each node in the video surveillance network that minimizes the overall power consumption. The fairness ratio obtained from the training scenes will be used as an initial guess for the CID allocation in the test scenes. The following sections explain the algorithms proposed in this thesis.

### 3.3 Common Approach

A common approach to allocating the encoding configurations per each video surveillance network node is by assigning the same settings for all nodes. We call this approach *CommonConfig* algorithm. This approach was used to analyze the video surveillance network's power consumption of Intra only configuration and Inter Main Profile with GOP size of 6 and frame-type sequence of I-P-B-P-B-P-I in [62][65]. The authors in [100] also used *CommonConfig* algorithm for Intra-only configuration in their analysis. In order to implement the *CommonConfig* algorithm while still being fair with the implementation reported in the literature, we try to assign the same CID to all nodes such

```plaintext
Algorithm MaximumFairness(f\text{ratio})

% N: number of node

for i := 1: N
    CID ← getCID(p, 0)
end

minimizePower(CID)
```

**Figure 3.16: MaximumFairness CID allocation algorithm**
that the bandwidth constraint is not violated.

Another common approach to assign nodes’ encoding configuration is by allocating the same fairness constraints to each node, that is equal to \( \rho_{\text{fair}} = \rho_i = 1/N \). This algorithm is called the MaximumFairness scheme, as shown in Figure 3.16. In this figure, the procedure getCID is a procedure to assign a node with a specific CID, where \( R(CID_i, QP)/B<\rho_i \) and \( R \) is the bitrate allocated for node \( i \) when using the corresponding CID. Note that, \( f_{\text{ratio}} = \{\rho_l, \rho_2, ..., \rho_N\} \), \( N \) is the number of nodes, and \( \rho = 1/N \).

### 3.4 Fairness-Based Optimization for Minimum Power Consumption

The encoding power consumption of a video surveillance node depends on the CID value assigned to that node. In this regard, by merging (3.1) and (3.2), the encoding power consumption is estimated as:

\[
P_{ei} = \frac{\kappa_i \cdot CPI \cdot E_{ec} \cdot F_r}{N_f}
\]

where, \( \kappa_i \) is the total number of instructions to encode the video for node \( i \), \( CPI \) is the average number of cycles per instruction of the CPU, \( E_{ec} \) is the energy depleted per cycle, \( N_f \) is the number of frames and \( F_r \) denotes the frame rate of the video sequence. The value of \( \kappa_i \) is obtained using the following relation \( \kappa_i = IC(CID_i, QP) \), where \( IC \) is the instruction count provided by iprof for the pair of CID and QP values used, \( 1 \leq CID_i \leq 7 \) and \( QP_{\min} \leq QP_i \leq QP_{\max} \). Since we want each node to produce a video with almost similar quality, all nodes have to use the same QP, thus, \( QP_i = QP, \forall i \in N \).
On the other hand, some nodes need to relay their data through intermediate nodes. Therefore, the node's communication power consumption depends on both the CID value assigned to that node and the way the encoded data is relayed in the network. Combining the transmission, reception and nodes' power consumption model formulation shown in (2.3), (2.4), and (2.5), the rate flow conservation law (2.2), the bandwidth constraint formula (3.5) and nodes' encoding power consumption model (3.9); the task to assign nodes' CID allocation for minimum power consumption video surveillance network is formulated as an optimization problem shown in Figure 3.17.

The objective of the optimization in Figure 3.17 is to find the allocation of transmission rate on each link and CID at each node that minimize the total power consumption. The coding complexity and bitrate of node i, i.e., $\kappa_i$ and $R_i$ respectively, in the

![Optimization minimizePowerCID](image)

**Figure 3.17: Power consumption minimization using CID**
optimization shown in Figure 3.17 are obtained from the look-up table. Note that, the value of CID is constrained to be integral. On the other hand, the value of $r_{ij}$ and $r_{ki}$ that determine the routing of data from and to node $i$ in (2.4) and (2.5) are rational numbers. However, although the CID values represent encoder configuration labels, it is also related with GOP size as explained in Section 3.1.4. Also, the value of bitrate (coding complexity) is monotonously decreasing (increasing) with the increase of CID value (see Figure 3.10). An optimization problem involving mixed linear and integer variables is NP-complete, where some of the solutions are intractable. However, using the branch and bound algorithm [139], the optimization procedure can be terminated early and as long as a solution that satisfies the stopping criteria is found. Therefore, a feasible, not necessarily optimal solution can be obtained. The following section explains the proposed approach.

\[
\begin{align*}
\text{minimize} & \quad P_{\text{net}} \\
\text{subject to:} & \quad P_i \leq P_{\text{net}}, \forall i \in N \\
& \quad P_i = P_{a_i} + P_{n_i} + P_{d_i} \\
& \quad P_{n_i} = \frac{\kappa_i \cdot CPI \cdot E_{\text{op}} \cdot F_i}{N_f} \\
& \quad \kappa_i = IC\{\text{CID}_i, \text{QP}\} \\
& \quad \sum r_{ij} - \sum r_{ji} = R_i, \forall i \in N, \forall j \in N, j \neq i \forall k \in N, k \neq i \\
& \quad R_i = IC\{\text{CID}_i, \text{QP}\} \\
& \quad P_{a_i} = \sum (\alpha + \beta \cdot d_{ij}^g) \cdot r_{ij} \\
& \quad P_{n_i} = \sum \gamma \cdot r_{ji}, \forall k \in N, k \neq i \\
& \quad \sum R_i \leq B \\
& \quad \text{CID}_i \leq \text{CID}^{\text{lower bound}}_i \\
& \quad \text{CID}_i \geq \text{CID}^{\text{upper bound}}_i
\end{align*}
\]

Figure 3.18: Power consumption minimization with bounded CID values
3.4.1 Fairness-based CID allocation for Minimum Power Video Surveillance Network

In this thesis, the branch and bound approach is implemented by using the following steps: 1) solve the bounded optimization problem 2) call a recursive procedure to perform branch and bound until a solution is found or termination criteria are satisfied. The bounded optimization problem is shown in Figure 3.18. The difference between this algorithm and minimizePowerCID optimization described in the previous section is the fact...

```
Procedure RecursiveBranchBound (Power, CID_{\text{lower bound}}, CID_{\text{upper bound}})

% $\xi$: integrality constraint

[TPower, status, TCID] $\leftarrow$ minimizePowerBounded (CID_{\text{lower bound}}, CID_{\text{upper bound}})

if status $<$ 0 or Tpower $>$ Power
    return [TPower, status, TCID]
end

% if TCID satisfy integrality constraint
if (TCID - round(TCID)) $\leq$ $\xi$
    status = 1
    if TPower $<$ Power
        CID $\leftarrow$ round(TCID)
        Power $\leftarrow$ TPower
    end
    return [TPower, status, CID]
end

% solution is not feasible $\Rightarrow$ solve the first branch
CID_{\text{lower bound}} = floor(TCID)
[TPower, status, CID] $\leftarrow$ RecursiveBranchBound (Power, CID_{\text{lower bound}}, CID_{\text{upper bound}})

% solution is not feasible $\Rightarrow$ solve the second branch
CID_{\text{upper bound}} = ceil(TCID)
[TPower, status, CID] $\leftarrow$ RecursiveBranchBound (Power, CID_{\text{upper bound}}, CID_{\text{lower bound}})
```

Figure 3.19: RecursiveBranchBound procedure
that the CID values are given as upper and lower bounds. Note that, since the CID value represents the configuration label, whenever the optimization procedure needs to look up the values of the coding complexity and bitrate, the CID value needs be rounded to the nearest integer. If the CID provided by the bounded optimization does not satisfy the integrality constraint, the RecursiveBranchBound procedure will be called to perform branch and bound approach to find the solution. Figure 3.19 shows how the branch and bound algorithm is implemented to allocate the rate on each transmission link and CID per each node that minimizes the total power consumption. The algorithm proceeds by finding the CID allocation using the optimization procedure shown in Figure 3.18. If the obtained solution does not satisfy the integrality constraint, the optimization will be divided into two sub-problems by defining new upper and lower bounds followed by a call to the recursive functions. The integrality constraint $\epsilon$ is small number that signify the error between the CID and the rounded integral value of the CID, such that $0 < \epsilon < |CID_i - \text{round}(CID_i)|$.

3.4.2 Proposed Fairness-based CID Allocation Algorithm for the Test Set

As mentioned in the previous sections, the optimal fairness ratio allocation obtained from the training sets will be used as an initial guess to allocate the CID for the test videos. However, since the content of the video of the training set and the test set are not exactly the same, we may need to perform some adjustment procedure while assigning the nodes’ CID in the test sets. Hence, the algorithm fairness based CID allocation for the test set is shown in Figure 3.20. In this algorithm, the getCID procedure returns the highest possible CID option that can be allocated to node $i$ with fairness ratio equal to $\rho_i$. For example, if the possible CIDs that can be allocated to node $i$ are either six or seven, the getCID procedure
Algorithm FairnessBased
\[\text{Algorithm FairnessBased}(\text{fratio})\]
\%
\frac{\text{fratio} = \{\rho_1, \rho_2, .., \rho_N\}}{\text{fratio} = \{\rho_1, \rho_2, .., \rho_N\}}
\%
\frac{\text{B: Bandwidth of the network}}{\text{B: Bandwidth of the network}}
\%
\begin{align*}
\text{overflow} &= 0 \\
\text{for } i &\equiv N/1:1 \\
\text{CID}_i &\leftarrow \text{getCID}(\rho_i, 0) \\
\rho'_i &\leftarrow \frac{R\{\text{CID}_i, QP\}/B}{\rho'_i \leftarrow \frac{R\{\text{CID}_i, QP\}/B}{\text{if } \text{CID}_i \text{ is } \emptyset} \\
\text{overflow} &\leftarrow \text{overflow} + (R\{\text{CID}_i, QP\} - \rho'_i B) \\
\text{else} &\leftarrow \text{overflow} + (R\{\text{CID}_i, QP\} - \rho'_i B) \\
\text{CID}_i &\leftarrow \text{getCID}(\rho'_i, 0) \\
\text{overflow} &\leftarrow \text{overflow} - (\rho'_i B - R\{\text{CID}_i, QP\}) \\
\text{end} &\leftarrow \text{overflow} - (\rho'_i B - R\{\text{CID}_i, QP\}) \\
\text{end} &\leftarrow \text{overflow} - (\rho'_i B - R\{\text{CID}_i, QP\}) \\
\text{checkBandwidthConstraint}(&\text{CID}) \\
\text{minimizePower}(&\text{CID}) \\
\end{align*}

Figure 3.20: Fairness-based minimum energy algorithm

will return CID equal to six. However, in some cases, the getCID procedure may not be able to find a suitable CID with fairness ratio allocation $\rho_j$ to be allocated to node $j$. In this regard, the node will be assigned the highest CID that entails the lowest possible bitrate in our configuration. A variable named overflow is then updated with the difference between the allocated bitrate from the look-up table $R\{\text{CID}_j, QP\}$ with the supposed maximum bitrate for that node, i.e., $\rho_j B$. In this regard, the variable overflow is used to quantity the accumulative bitrate borrowed from the other nodes.

On the other hand, if an appropriate CID is available while the value of overflow variable is positive, another call to procedure getCID with a lower fairness ratio is performed to get another CID. This is performed so that we can ‘pay back’ the outstanding bitrate ‘debt’. The overflow variable is then updated accordingly. In the chance that the
overflow variable is still positive after the CID allocation for all nodes have been performed, a procedure `checkBandwidthConstraint` is then called to adjust the CID allocation per each node. Starting from the node furthest from the sink, the procedure checks whether assigning a higher CID to that node can reduce variable overflow to be less than or equal to zero. After that, nodes’ power consumption is calculated using `minimizePower` procedure shown in Figure 3.17.

```
Algorithm FairnessBased_withAdjustment(f_ratio)

% f_ratio = {ρ_1, ρ_2, ..., ρ_N}
% N: number of node

{Power, CID} ← FairnessBased(f_ratio)
TCID ← CID

% adjust the last node
for i = CID_N + 1:1:7
    TPower ← minimizePower(TCID)
    if TPower < Power
        Power ← TPower
        CID ← TCID
    end
    TCID_N = i
end

% adjust the first node
for i = CID_1 - 1:1:1
    TPower ← minimizePower(TCID)
    if TPower < Power
        Power ← TPower
        CID ← TCID
    end
    TCID_1 = i
end
```

Figure 3.21: Fairness-based with adjustment algorithm
As mentioned in the previous paragraphs, an adjustment procedure can be performed to reduce the power consumption further. Figure 3.21 shows the adjustment procedure performed on two nodes, i.e., the last node and the first node. These two nodes are chosen because of the following reasons. Firstly, using a lower bitrate configuration at the last node will reduce the transmission power consumption on all the nodes in the video sensor network. On the other hand, the extra bitrate available can be allocated to the first node so that the encoding power consumption at the first node is reduced.

3.4.3 Experiments Settings

Figure 3.22 shows the network topology used in the simulations. In this figure, the dark node is the sink node while the blank nodes are the video node. Each node is given an identification number according to its distance to the sink. Therefore, the distance between node1 to the sink is smaller than the distance between node2 to the sink. It is assumed that each video node located at a specific location in the topology illustrated in Figure 3.22 is attached to the camera located at the same location shown in Figure 3.6. Therefore, node1
will be using the video captured by camera1; node2 will be using the video captured by camera4, and so forth. The H.264/AVC software, JM version 18.2 is used to generate the CID lookup table of encoding complexity and bitrate of all videos. The QP value used in the experiment ranges from 28 until 36.

Two separate sets of experiments were performed. The first set of experiment is conducted on the training set. The objective is to compare the results obtained by the proposed optimization technique with the ones obtained using *CommonConfig* and *MaximumFairness* approaches. From this experiment, we obtain the fairness ratios of the training set that minimize the node’s maximum energy consumption. The energy consumption obtained using the proposed approach will be compared to the one obtained using the *CommonConfig* and *MaximumFairness* approaches. To this end, the parameters shown in Table 3.8 are used.

**Table 3.8: Parameters Used for Fairness-based Algorithm Experiments**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>Energy cost for transmitting 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>(\beta)</td>
<td>Transmit amplifier coefficient</td>
<td>1.3 \times 10^{-8} J/Mb/m^4</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>Energy cost for receiving 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>(\eta)</td>
<td>Path loss exponent</td>
<td>4</td>
</tr>
<tr>
<td>CPI</td>
<td>average cycle per instruction</td>
<td>1.78</td>
</tr>
<tr>
<td>(E_c)</td>
<td>Energy depleted per cycle</td>
<td>1.215 nJ</td>
</tr>
<tr>
<td>B</td>
<td>Network Bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>d</td>
<td>Distance between node</td>
<td>5m</td>
</tr>
<tr>
<td>(\varepsilon)</td>
<td>Integrality constraints</td>
<td>0.2</td>
</tr>
</tbody>
</table>
3.4.4 Performance of the Proposed Technique for the Training Scenes

In order to find the minimum power consumption with the highest possible video quality, we need to find the minimum QP for each content complexity class. To do this, starting from QP equal to 28, a procedure to check the possibility to allocate CIDs to all the nodes is performed using the MaximumFairness approach. The minimum QP suitable for each scene may not be the same because each scene has different SI (TI) complexity class. For example, the minimum QP for scene party_act2 is 36. However, the minimum QP for scene office_act2 is 28. The optimization-based approach can then be performed on the corresponding training scene. The initial solution provided for the optimization algorithm in each run is different, controlled using the following setting. For each setting, the

![Figure 3.23: Bitrate allocated per each node in all training scenes obtained by the three algorithms (a) CommonConfig (b) MaximumFairness (c) Proposed](image)

83
algorithm is was run twice.

1. All nodes are initiated to use the lowest CID
2. All nodes are initiated to use the highest CID
3. The CID is allocated in increasing order, starting from the first node
4. The CID is allocated in decreasing order, starting from the first node

For the purpose of the analysis, the performance of the algorithm is compared against CommonConfig and MaximumFairness approaches mentioned in Section 3.3. Figure 3.23 shows the bitrate allocated using the compared techniques. Figure 3.23(a) shows the bitrate allocation for the training scenes obtained using the CommonConfig algorithm. The figure shows that the difference between the highest bitrate and the lowest bitrate

Figure 3.24: Node’s power consumption profile for the high SI training scene obtained by the three algorithms: (a) CommonConfig (b) MaximumFairness (c) Proposed
allocated in each scene is as follow: 129.85 kbps for the high SI training scene \((party\_act2)\), 115.87 kbps for the medium SI training scene \((office\_act1)\) and 115.08 kbps for the low SI training scene \((office\_act2)\), respectively. The algorithm \textit{CommonConfig} does not regulate the bitrate assigned per each node since the algorithm assign the same configuration per each node. Thus, the bitrate assigned to each node does not follow any trend. However, the \textit{MaximumFairness} approach allocates roughly the same bitrate per each node in any training scene used as shown in Figure 3.23(b). Given that the content captured by each camera in each training scene is not the same, there are some variations in the bitrate assigned to each node. However, the difference between the highest bitrate and the lowest bitrate allocated in each scene is not significant, i.e., 50.37 kbps for the high SI training scene \((party\_act2)\), 63.85 kbps for the medium SI training scene \((office\_act1)\) and 41.64 kbps for the low SI training scene \((office\_act2)\), respectively.

On the other hand, as Figure 3.23(c) shows, the proposed technique allocates different bitrate to each node such that the nodes closer to the sink have generally higher bitrate than the nodes that are farther from the sink. The difference between the maximum and minimum bitrate allocated in each scene has become more significant, i.e., equal to 327.73 kbps for the high SI scene, 471.56 kbps for the medium SI scene and 476.59 kbps

<table>
<thead>
<tr>
<th>Training sequence</th>
<th>Node</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>party1_act2</td>
<td>0.243</td>
</tr>
<tr>
<td>office_act1</td>
<td>0.296</td>
</tr>
<tr>
<td>office_act2</td>
<td>0.307</td>
</tr>
</tbody>
</table>
for the low SI training scene. Figure 3.23 also shows that node2 is allocated with a lower bitrate than the other nodes. This is due to the fact that node2 corresponds to camera4 (see Figure 3.6), which according to Table 3.5 and Table 3.6 has a lower content complexity level than the other cameras.

Figure 3.24 shows power consumption profile of the high SI scene. The communication power consumption shown in this figure is the sum of transmission and reception power consumption. Figure 3.24(a) shows that by using algorithm CommonConfig, each node consumes almost the same encoding power consumption. However, in the MaximumFairness approach (see Figure 3.24(b)), each node was assigned roughly the same bitrate. However, the nodes that are closer to the sink consume more energy because they need to relay data from the other nodes. On the other hand, Figure 3.24(c) shows that the proposed optimization-based approach manages to provide a more

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Training sequence</th>
<th>party1_act2</th>
<th>office_act1</th>
<th>office_act2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P_{net}</strong></td>
<td>CommonConfig</td>
<td>10.73</td>
<td>10.25</td>
<td>9.97</td>
</tr>
<tr>
<td></td>
<td>MaximumFairness</td>
<td>10.95</td>
<td>10.45</td>
<td>10.06</td>
</tr>
<tr>
<td></td>
<td>proposed</td>
<td>9.73</td>
<td>9.53</td>
<td>9.35</td>
</tr>
<tr>
<td><strong>P_{avg}</strong></td>
<td>CommonConfig</td>
<td>9.4</td>
<td>9.03</td>
<td>8.8</td>
</tr>
<tr>
<td></td>
<td>MaximumFairness</td>
<td>9.39</td>
<td>9.03</td>
<td>8.83</td>
</tr>
<tr>
<td></td>
<td>proposed</td>
<td>9.31</td>
<td>9.38</td>
<td>9.19</td>
</tr>
<tr>
<td><strong>STD(P_{i})</strong></td>
<td>CommonConfig</td>
<td>0.54</td>
<td>0.49</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>MaximumFairness</td>
<td>0.64</td>
<td>0.57</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>proposed</td>
<td>0.18</td>
<td>0.09</td>
<td>0.09</td>
</tr>
</tbody>
</table>
balance power consumption among all nodes in the video surveillance network. This trend is also observed in the medium SI and low SI training scenes.

Table 3.10 shows the $P_{\text{net}}$ (nodes’ maximum power consumption), $P_{\text{avg}}$ (average maximum power consumption) and STD($P_i$) (standard deviation of nodes’ power consumption) of the three algorithms. It is interesting to see that CommonConfig algorithm manages to perform better than the MaximumFairness algorithm. This shows that assigning the same bitrate to each node does not help in reducing the video surveillance network’s power consumption. On the other hand, the table also shows that the optimization-based approach manage to have lower $P_{\text{net}}$, $P_{\text{avg}}$ and STD($P_i$) than the other algorithms. This shows

![Bar chart comparison of Pnet, Pavg, and STD(Pi) values](image)

**Figure 3.25:** Comparison of the Pnet, Pavg, and STD(Pi) values obtained from all test scenes
that by regulating the bitrate and assigning the corresponding CID to a node, video surveillance network's power consumption is reduced.

3.4.5 Performance of the Proposed Technique for the Test Scenes

Using the fairness ratio obtained from the training scenes (see Table 3.9), the fairness based algorithm explained in Section 3.4.2 is used to allocate the nodes’ CID for all test scenes. We noticed that the fairness-based adjustment algorithm shown in Figure 3.21 provides better results than the proposed fairness-based allocation algorithm [140]. Therefore, from this point forward, the comparison will show only the proposed fairness-based with adjustment with the existing techniques mention in Section 3.3. Correspondingly, Figure 3.25 compares the value of the $P_{\text{net}}$, $P_{\text{avg}}$ and $\text{STD}(P_i)$ obtained by

<table>
<thead>
<tr>
<th>Test Scenes</th>
<th>Improvement against CommonConfig (%)</th>
<th>Improvement against MaximumFairness (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$P_{\text{net}}$</td>
<td>$P_{\text{avg}}$</td>
</tr>
<tr>
<td>VS1</td>
<td>8.33</td>
<td>-0.29</td>
</tr>
<tr>
<td>VS2</td>
<td>4.24</td>
<td>0.41</td>
</tr>
<tr>
<td>VS3</td>
<td>9.67</td>
<td>0.63</td>
</tr>
<tr>
<td>VS4</td>
<td>7.01</td>
<td>-1.29</td>
</tr>
<tr>
<td>VS5</td>
<td>7.88</td>
<td>0.83</td>
</tr>
<tr>
<td>VS6</td>
<td>7.50</td>
<td>0.41</td>
</tr>
<tr>
<td>VS7</td>
<td>4.60</td>
<td>0.63</td>
</tr>
<tr>
<td>VS8</td>
<td>5.40</td>
<td>0.96</td>
</tr>
<tr>
<td>VS9</td>
<td>8.12</td>
<td>0.73</td>
</tr>
</tbody>
</table>
the three algorithms in all test cases. Furthermore, Table 3.11 shows the percentage of improvement in terms of $P_{\text{net}}$, $P_{\text{avg}}$ and STD($P_i$) obtained by the proposed techniques against the common approach for all test cases used. It can be seen that the amount of power consumption reduction obtained by the proposed fairness-based with adjustment technique is in the range of 5.06% to 10.48%, averaging into 8.18% improvement against MaximumFairness algorithm. On the other hand, the percentage of $P_{\text{net}}$ reduction against CommonConfig algorithm is in the range of 4.24% to 9.67%, averaging into 6.97% improvement. The average improvement of the proposed algorithm against common approaches is thus around 7.58%. This result shows that by using the fairness ratio obtained from the training scenes to allocate the nodes’ CID, the proposed algorithm reduces the video surveillance network’s power consumption by around 7.58%. Since video surveillance network nodes’ energy resource is usually limited, 7.58% reduction in power consumption equates to increasing the video surveillance network lifetime by 7.58%. In addition to that, except for test scene VS1 and VS2, the proposed algorithm also manages to slightly reduce the video surveillance network’s average power consumption. The standard deviation of nodes’ power consumption is also reduced by more than 40% on average.

### 3.5 Summary

This chapter analyzed the effect of encoder parameter settings and content complexity to video surveillance network’s power consumption. A large number of real-life captured videos of simulated video surveillance network settings with different activity levels were developed and used in the analysis. The scenes were classified according to its
content complexity on which the higher activity level scenes were used as the training set. It was shown that the fairness ratio allocated per each node affects the distribution of power consumption in a video surveillance network. The fairness ratio obtained from the training sets were then used as the initial guide to allocate the nodes’ encoder configuration for the test scenes. The results show that the proposed techniques manage to reduce the power consumption by 7.58% as compared to the results obtained by the other techniques. The proposed method depends on the assumption that the nodes have the information about the coding complexity and bitrate of the videos in the form of a look up CID table. Models to estimate the coding complexity and bitrate of videos can alleviate the need for such a table. This is what will be studied in the next chapter.
Chapter 4: Coding Complexity and Bitrate Models for H.264/AVC-based Video Surveillance Networks

In this chapter, models to estimate coding complexity and bitrate of videos are proposed. The models were developed based on the analysis of the effect of some encoding parameters to coding complexity and bitrate on a set of training video, and tested against a set of unseen test video. An approach to reduce the models’ estimation error for videos whose content changes within a specific period of time is also proposed in this chapter. A study to estimate video surveillance networks’ total power consumption using the proposed models was also discussed.

4.1 Proposed Model

In Chapter 3, it was shown that the GOP size that controls the number of the inter-coded pictures in successive frames is a parameter that significantly affects the coding complexity and bitrate of the video. The other factor that controls the coding complexity and bitrate is the type and variety of block sizes used in the inter prediction process [141]. Increasing the number of block sizes results in better prediction and consequently higher compression performance at the expense of increased complexity. In general, there are seven block sizes defined for inter-prediction in H.264/AVC. In this thesis, the complexity of motion estimation (ME) is classified into different levels of complexity based on the variety of block size candidates used (see Table 4.1) used to encode a video.
The complexity of encoding process of a video sequence (Cₜ) is formulated as follows:

\[ C_τ = C_I \cdot n_I + C_P \cdot n_P \quad (4.1) \]

where \( C_I \) is the complexity to encode an I-frame, \( C_P \) is the complexity to encode a P-frame, \( n_I \) is the number of I-frames in the sequence and \( n_P \) is the number of P-frames in the sequence. For a video sequence with no scene change, the value of \( C_I \) can be considered constant. On the other hand, \( C_P \) depends on the complexity level of the ME process. The complexity level of ME process (called \( M_L \)) is classified based on the used block size candidates in the encoding process as shown in Table 4.1.

The GOP size does not affect the normalized coding complexity of P frames at each \( M_L \) [141]. Note that the normalized coding complexity is calculated by dividing the coding complexity at certain \( M_L \) with the coding complexity when \( M_L \) is equal to one. Furthermore, Figure 4.1 shows the plot of normalized \( C_P \) (NCₚ) for the video “BQMall” and “Traffic”. This figure shows that the plots have the same slope but the length is scaled by a constant. The

<table>
<thead>
<tr>
<th>( M_L )</th>
<th>Block Size Candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SKIP, 16x16</td>
</tr>
<tr>
<td>2</td>
<td>SKIP, 16x16, 16x8</td>
</tr>
<tr>
<td>3</td>
<td>SKIP, 16x16, 16x8, 8x16</td>
</tr>
<tr>
<td>4</td>
<td>SKIP, 16x16, 16x8, 8x16, 8x8</td>
</tr>
<tr>
<td>5</td>
<td>SKIP, 16x16, 16x8, 8x16, 8x8, 8x4</td>
</tr>
<tr>
<td>6</td>
<td>SKIP, 16x16, 16x8, 8x16, 8x8, 8x4, 4x8</td>
</tr>
<tr>
<td>7</td>
<td>SKIP, 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4</td>
</tr>
</tbody>
</table>

4.1.1 H.264/AVC Coding Complexity Model

The complexity of encoding process of a video sequence (Cₜ) is formulated as follows:

\[ C_τ = C_I \cdot n_I + C_P \cdot n_P \quad (4.1) \]

where \( C_I \) is the complexity to encode an I-frame, \( C_P \) is the complexity to encode a P-frame, \( n_I \) is the number of I-frames in the sequence and \( n_P \) is the number of P-frames in the sequence. For a video sequence with no scene change, the value of \( C_I \) can be considered constant. On the other hand, \( C_P \) depends on the complexity level of the ME process. The complexity level of ME process (called \( M_L \)) is classified based on the used block size candidates in the encoding process as shown in Table 4.1.

The GOP size does not affect the normalized coding complexity of P frames at each \( M_L \) [141]. Note that the normalized coding complexity is calculated by dividing the coding complexity at certain \( M_L \) with the coding complexity when \( M_L \) is equal to one. Furthermore, Figure 4.1 shows the plot of normalized \( C_P \) (NCₚ) for the video “BQMall” and “Traffic”. This figure shows that the plots have the same slope but the length is scaled by a constant. The
normalized $C_P$ range is then defined as the range of normalized $C_P$ values for a specific video. As it is observed, the value of normalized $C_P$ for the “Traffic” video is between 1 and 1.485. Hence, the normalized $C_P$ range for “Traffic” video is thus equal to 0.485 and . On the other hand, the normalized $C_P$ range for the “BQMall” video is equal to 0.66. In this regard, Figure 4.1 also shows that the normalized $C_P$ range of a video is proportional to the value of $C_P$ when $M_L$ is equal to one ($C_P(M_L=1)$). Therefore, the range of normalized $C_P$ of a specific video sequence is modeled as:
Furthermore, the fractional increase of normalized $C_P$ is calculated by scaling the value of normalized $C_P$ using the following formula:

$$\delta_{ML}(M_L = i) = \frac{NC_P(M_L = i) - 1}{NC_P(M_L = 7) - 1},$$

where $NC_P(M_L = i)$ denotes the normalized $C_P$ when $M_L$ is equal to $i$.

Figure 4.2 shows the $\delta_{ML}$ plot of “Traffic” and “BQMall” videos. It is interesting to see that the increase of normalized $C_P$ with respect to $M_L$ is almost similar to both videos. By averaging the values of the fractional increase of normalized $C_P$ from both videos, we obtain the value of $\delta_{ML}$ shown in Table 4.2.

Using $\omega_l$, the complexity to encode a P-frame for a specific $M_L$ is:

$$C_{P(M_L = i)} = C_{P(M_L = i)} \cdot (1 + \delta_{ML}(i) \cdot \omega_l)$$

(4.4)

Considering that $n_I = N/GOP$, where $N$ is the total number of frames and $n_P = N - N/GOP$, then the average coding complexity per frame is:

<table>
<thead>
<tr>
<th>$M_L$</th>
<th>$\delta_{ML}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0.13</td>
</tr>
<tr>
<td>3</td>
<td>0.26</td>
</tr>
<tr>
<td>4</td>
<td>0.54</td>
</tr>
<tr>
<td>5</td>
<td>0.67</td>
</tr>
<tr>
<td>6</td>
<td>0.81</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.2: ME complexity level ($M_L$) and $\delta_{ML}$
\[ C_f = \left( C_i + C_{P(M_L=1)} \cdot (1 + \delta_{ML}(i) \cdot (a \cdot C_{P(M_L=1)} + b)) \cdot (GOP-1) \right) / GOP \]  

(4.5)

4.1.2 H.264/AVC Bitrate Model

Following the same approach to develop coding complexity model, the size of encoded video sequence (in bit) is modeled as:

\[ R_S = R_I \cdot n_I + R_P \cdot n_P \]  

(4.6)

where \( R_I \) is the average size of an I-frame and \( R_P \) is the average size of a P-frame. Note that the value of \( R_P \) depends on the \( M_L \) and \( GOP \) used by the encoder. As it is observed in Figure 4.3, the value of \( R_P \) decreases as \( M_L \) increases. Therefore, for a certain \( GOP \) value, the \( R_P \) is modeled as:

\[ R_{P(GOP=1)} = R_{P(GOP=1, M_L=1)} \cdot f(M_L) \]  

(4.7)

where \( R_{P(GOP=1, M_L=1)} \) is the bitrate of a P-frame when \( GOP=1 \) and \( M_L=1 \), and \( f(M_L) \) is modeled as a decay function with respect to \( M_L \) that match the plot in Figure 4.3. We have analyzed different decay functions and decided to use the generalized logistic function. The

![Figure 4.3: Bitrate of a P-frame for different M_L of “BQMall” video](image)
logistic function is a widely used sigmoid function for growth/decay modeling where the growth/decay is exponential at first, but eventually slower and then levels off. This matches the way $R_P$ is reduced with the increase of $M_L$. The logistic function $f(M_L)$ used in our study is as follows:

$$f(M_L) = p + \frac{q - p}{1 + e^{-r(s_{ML} - s)}}$$  \hspace{1cm} (4.8)

where $p$ and $q$ indicate the minimum and maximum asymptote of the plot respectively, $r$ is the growth rate, while $s$ signify the time for maximum growth.

Figure 4.3 also shows that the slope of the $R_P$ plot for different GOP sizes is the same. Therefore, $R_P$ is modeled equal to:

$$R_P = R_{P(GOP=2,ML=1)} \cdot f(M_L) + f(GOP)$$  \hspace{1cm} (4.9)

where $f(GOP)$ is a function that reflects the effect of GOP towards $R_P$. To obtain the parameters for $f(M_L)$, we used the least mean square regression technique on the normalized $R_P$ using the data from the training video sequences that were encoded with GOP=2 and $M_L=1$. We used $\omega_2 \cdot \ln(GOP)$ to estimate the $f(GOP)$ [141]. The value of $\omega_2$ is estimated using the least square regression technique from the $R_P$ values of the training video sequences that are encoded with different GOP settings. Assuming that the average bitrate of an I-frame is equal to $R_I$, the average bitrate of a frame ($R_f$) is estimated as:

$$R_f = \frac{R_I}{GOP} + \left( R_{P(GOP=2,ML=1)} \cdot \left( p + \frac{q - p}{1 + e^{-r(s_{ML} - s)}} \right) + \omega_2 \cdot \ln(GOP) \right) \cdot \left( \frac{GOP - 1}{GOP} \right)$$  \hspace{1cm} (4.10)
4.1.3 Analysis of the Model

In order to obtain the parameters for the model, we encode the first two frames of each video sequence. Assuming that there is no scene change in the video sequence, the bitrate of each I-frame will be almost similar. Therefore, for the bitrate model, $R_I$ is assumed to be equal to the bitrate of the encoded first frame, while $R_{P(GOP=2, ML=1)}$ is equal to the bitrate of the second frame. The parameters for (4.8) are obtained from the training videos, and the values are as follow: $p=0.92$, $q=1$, $r=-21.36$ and $s=0.14$.

For the complexity modeling, the iprof provides us with the complexity of encoding the first two frames of the video sequences, i.e., $C_{2frames} = C_I + C_{P(ML=1)}$. In order to obtain the value of $C_{P(ML=1)}$, we need to estimate the value of $C_I$. In this regard, we assume that for the I-frame, the value of $C_I$ can be estimated from the value of $R_I$ using a linear regression of the training videos [141], i.e., $C_I = 0.09 \cdot R_I + 216.97$. Furthermore, the value of $\omega_I$ is calculated using (4.3) with the following parameters: $a=0.0135$ and $b=-2.13$.

<table>
<thead>
<tr>
<th>Test video</th>
<th>Percentage Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RaceHorses</td>
<td>2.79</td>
</tr>
<tr>
<td>PeopleOnStreet</td>
<td>2.05</td>
</tr>
<tr>
<td>Vidyo1</td>
<td>3.45</td>
</tr>
</tbody>
</table>

Table 4.3: Coding complexity modeling error

<table>
<thead>
<tr>
<th>Test video</th>
<th>Percentage Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RaceHorses</td>
<td>11.6</td>
</tr>
<tr>
<td>PeopleOnStreet</td>
<td>8.57</td>
</tr>
<tr>
<td>Vidyo1</td>
<td>9.55</td>
</tr>
</tbody>
</table>

Table 4.4: Bitrate modeling error
The average percentage error of coding complexity and bitrate for GOP={1, 2, 4, 8, 16, 32, 64} and M={1, 2, 3, 4, 5, 6, 7} for videos “RaceHorse”, “PeopleOnStreet” and “Vidyo1” [132] are calculated. As Table 4.3 shows, the average error in complexity modeling is less than or equal to 3.45% for the abovementioned video sequences, while the average error in bitrate modeling is less than or equal to 11.6% as reported in Table 4.4.

4.2 Model Evaluation in a Simulated Video Surveillance Test Environment

In order to mimic realistic video surveillance network applications, we have captured real-life videos using four cameras in the atrium of a public building. The cameras were installed so that each of them had a different point of view as shown in Figure 4.4. The views of some cameras were overlapping with one another. The scene arrangement was such that each camera point of view was different. In order to mimic a practical application, the videos were downsampled to 416x240 pixels resolution and 15 frames per second (fps). Five shots of videos were captured using the four cameras, resulting in a total of 20

![Camera placements](image)

*Figure 4.4: Camera placements*
different videos. These videos were named using the following convention <camera-id_shot-id>. Therefore, camera1_shot1 is the video obtained by camera1 in the first shot. The four videos of the fifth shot were selected as the training set for the model, while the remaining videos were used as the test set.

4.2.1 Model Evaluation for Videos with Changing Content

In many real-life captured videos, content may change during a 10s video shot. For example, Figure 4.5 shows the snapshots of camera1_shot3 video sequence at frame number 1, 70, and 100. It can be seen that the content at the start of the video (frame number 1) differs significantly from the content towards the end of the video (frame number 100). On the other hand, Figure 4.6 shows the snapshots at frames number 1, 60 and 110 of the camera2_shot2 video sequence. Again, we can see that the content at the

![Figure 4.5: Content changes during a 10s camera1_shot3 video sequence. (a) frame 1 (b) frame 70 (c) frame 100](image1)

![Figure 4.6: Content changes during a 10s camera2_shot2 video sequence. (a) frame 1 (b) frame 60 (c) frame 110](image2)
beginning of the video differs significantly from the ones captured at a later time, i.e., frames 60 and 110. From this observation, it is clear that utilizing model parameters from the first two frames at the beginning of video may lead to a large estimation error. In order to tackle this problem, we divide the 10s video into a number of sub-shots. In each sub-shot, coding complexity and bitrate estimation were performed.

Figure 4.7 shows the flowchart of the proposed method to reduce coding complexity and bitrate estimation error. In this figure, the variable frame_num is the current frame number, while \( k \) denotes the length of a sub-shot in terms of the number of frames. Note that, since the video is divided into \( \lceil N/k \rceil \) sub-shots, the first two frames of each sub-shots are encoded to obtain the required parameters for the model [142]. The estimation error is calculated from the complexity per second \( (C_{ps}) \) and average bitrate \( (R_{av}) \), that is defined as follows [122]:

\[
C_{ps} = \frac{F_r}{N} \sum_{i=1}^{\left\lceil N/k \right\rceil} C_{j}^i \cdot K_i
\]

\[
R_{av} = \frac{F_r}{N} \sum_{i=1}^{\left\lceil N/k \right\rceil} R_{j}^i \cdot K_i
\]

Here, \( F_r \) is the frame rate and \( K_i \) is calculated as follows:

\[
K_i = \begin{cases} 
  k & 1 \leq i \leq \left\lfloor N/k \right\rfloor \\
  \text{mod}(N,k) & \text{otherwise}
\end{cases}
\]
In order to estimate the modeling error, the root mean square error (RMSE) of the coding complexity and bitrate for $\text{GOP}={1, 2, 4, 8, 16, 32, 64}$, $M_L={1, 2, 3, 4, 5, 6, 7}$ and $k={150, 75, 60, 45}$ are calculated.

Recall that in order to implement our model using the complexity and bitrate modeling, we need to obtain several variables from each video sequence. Therefore, the first two frames of each sub-shots of the video sequence were encoded. For the bitrate model, $R_l$ is assumed to be equal to the bitrate of the encoded first frame of each sub-shot, while $R_P = R_P(\text{GOP}=2, \ M_L=1)$ is equal to the bitrate of the second frame. The following parameters for (4.8) are used: $p=0.92$, $q=1$, $r=-21.36$ and $s=0.14$.

<table>
<thead>
<tr>
<th>Test sequence</th>
<th>$k=150$</th>
<th>$k=75$</th>
<th>$k=60$</th>
<th>$k=45$</th>
</tr>
</thead>
<tbody>
<tr>
<td>camera1_shot1</td>
<td>34.966</td>
<td>35.533</td>
<td>28.523</td>
<td>27.890</td>
</tr>
<tr>
<td>camera3_shot1</td>
<td>48.005</td>
<td>38.997</td>
<td>45.258</td>
<td>35.861</td>
</tr>
<tr>
<td>camera4_shot1</td>
<td>45.437</td>
<td>36.435</td>
<td>34.667</td>
<td>32.790</td>
</tr>
<tr>
<td>camera1_shot2</td>
<td>33.850</td>
<td>32.615</td>
<td>37.589</td>
<td>34.662</td>
</tr>
<tr>
<td>camera2_shot2</td>
<td>37.247</td>
<td>29.967</td>
<td>26.934</td>
<td>26.985</td>
</tr>
<tr>
<td>camera3_shot2</td>
<td>28.769</td>
<td>36.088</td>
<td>28.459</td>
<td>32.256</td>
</tr>
<tr>
<td>camera4_shot2</td>
<td>33.145</td>
<td>27.086</td>
<td>27.538</td>
<td>27.052</td>
</tr>
<tr>
<td>camera1_shot3</td>
<td>69.666</td>
<td>30.149</td>
<td>37.266</td>
<td>39.555</td>
</tr>
<tr>
<td>camera2_shot3</td>
<td>59.759</td>
<td>29.830</td>
<td>39.279</td>
<td>30.596</td>
</tr>
<tr>
<td>camera3_shot3</td>
<td>47.022</td>
<td>37.739</td>
<td>35.961</td>
<td>39.236</td>
</tr>
<tr>
<td>camera4_shot3</td>
<td>41.304</td>
<td>33.905</td>
<td>35.581</td>
<td>31.479</td>
</tr>
<tr>
<td>camera1_shot4</td>
<td>27.858</td>
<td>32.782</td>
<td>30.906</td>
<td>32.127</td>
</tr>
<tr>
<td>camera2_shot4</td>
<td>38.642</td>
<td>38.363</td>
<td>31.962</td>
<td>33.426</td>
</tr>
<tr>
<td>camera3_shot4</td>
<td>36.970</td>
<td>36.930</td>
<td>36.914</td>
<td>36.860</td>
</tr>
<tr>
<td>camera4_shot4</td>
<td>39.797</td>
<td>39.986</td>
<td>32.818</td>
<td>39.890</td>
</tr>
</tbody>
</table>
For complexity modeling, the coding complexity of the first two frames, i.e., $C_{2frames} = C_i + C_{P(ML=1)}$ of each sub-shots is provided by the iprof tool. In order to obtain the value of $C_{P(ML=1)}$ the value of $C_i$ is estimated from the value of $R_i$ using a linear regression of the training videos using the following formula $C_i = 0.0637 \cdot R_i + 214.56$. Furthermore, the value of $\omega_i$ is calculated using (4.5) and the following parameters: $a=0.009$, and $b=-1.178 [122]$.

Table 4.5 shows the coding complexity estimation error of all test sequences and different values of $k$. The table shows that, in general, the coding complexity estimation error decreases as we use a larger number of sub-shots, i.e., using smaller $k$ values. We can also see that the proposed method manages to reduce coding complexity estimation error in 11 out of 16 cases when $k$ is set equal to 45. On the other hand, using $k=60$ frames, the coding complexity estimation error is reduced in 13 out of 16 cases. In particular, in the case of video camera1_shot3, coding complexity estimation error for $k=150$ and $k=60$ is equal to 69.666 and 37.266, respectively. This is equal to 46.5% reduction in estimation.
error. Figure 4.8 compares the measured coding complexity and the estimated coding complexity per second ($C_{ps}$) for different values of $k$ and varying GOP sizes. Note that, in this figure, the value of $M_L$ is set to four.

Furthermore, Table 4.6 shows the bitrate estimation of all test sequences and different values of $k$. Similar to the coding complexity case; the table shows that, in general, the bitrate estimation error decreases as smaller $k$ values are used. We can also see that the proposed method manages to reduce the bitrate estimation error in 12 out of 16 cases when $k$ is set equal to 45. However, when $k$ is set equal to 60, the bitrate estimation error is reduced in 13 out of 16 cases. The highest error reduction is obtained in the case of the

<table>
<thead>
<tr>
<th>Test sequence</th>
<th>k=150</th>
<th>k=75</th>
<th>k=60</th>
<th>k=45</th>
</tr>
</thead>
<tbody>
<tr>
<td>camera1_shot1</td>
<td>34.518</td>
<td>25.418</td>
<td>25.172</td>
<td>13.673</td>
</tr>
<tr>
<td>camera2_shot1</td>
<td>19.536</td>
<td>15.413</td>
<td>15.931</td>
<td>7.517</td>
</tr>
<tr>
<td>camera3_shot1</td>
<td>16.299</td>
<td>8.806</td>
<td>9.189</td>
<td>8.517</td>
</tr>
<tr>
<td>camera4_shot1</td>
<td>6.920</td>
<td>6.256</td>
<td>6.201</td>
<td>4.397</td>
</tr>
<tr>
<td>camera1_shot2</td>
<td>8.105</td>
<td>4.326</td>
<td>10.246</td>
<td>8.994</td>
</tr>
<tr>
<td>camera2_shot2</td>
<td>1.729</td>
<td>3.986</td>
<td>5.002</td>
<td>8.189</td>
</tr>
<tr>
<td>camera3_shot2</td>
<td>16.081</td>
<td>14.566</td>
<td>13.806</td>
<td>12.671</td>
</tr>
<tr>
<td>camera4_shot2</td>
<td>23.886</td>
<td>5.123</td>
<td>11.233</td>
<td>11.414</td>
</tr>
<tr>
<td>camera1_shot3</td>
<td>75.219</td>
<td>39.312</td>
<td>33.851</td>
<td>33.323</td>
</tr>
<tr>
<td>camera2_shot3</td>
<td>46.566</td>
<td>27.678</td>
<td>28.663</td>
<td>19.118</td>
</tr>
<tr>
<td>camera4_shot3</td>
<td>12.459</td>
<td>5.488</td>
<td>1.949</td>
<td>3.270</td>
</tr>
<tr>
<td>camera1_shot4</td>
<td>10.359</td>
<td>15.375</td>
<td>20.977</td>
<td>19.560</td>
</tr>
<tr>
<td>camera2_shot4</td>
<td>18.760</td>
<td>22.383</td>
<td>17.658</td>
<td>15.693</td>
</tr>
<tr>
<td>camera4_shot4</td>
<td>6.576</td>
<td>7.030</td>
<td>6.678</td>
<td>7.041</td>
</tr>
</tbody>
</table>
camera1_shot3 video sequence. In this particular video, the RMSE of the bitrate model for 
k=150 is equal to 75.219 kbps. However, when \( k \) is set equal to 60 frames, the RMSE of the bitrate model is reduced to 33.851 kbps. This is equal to 55.7% reduction in the estimation error. Figure 4.9 compares the measured bitrate and the estimated average bitrate (\( R_{av} \)) for different values of \( k \) and varying GOP sizes. Note that, in this figure, the value of \( M_L \) is set equal to four.

The results analyzed in the previous paragraphs show that by dividing a video sequence into a number of sub-shots, the model estimation error is reduced. The results also show that the estimation error reduction varies from one video to another. However, it is observed that the setting \( k=60 \) provides the smallest estimation error. From this point onward, the analysis of video surveillance network’s power consumption is performed under the assumption that the value of \( k \) is set equal to 60 frames.
4.2.2 Video Surveillance Network’s Power Consumption Estimation

The power consumption of a node in a video surveillance network consists of encoding energy consumption and communication power consumption. The power consumption for encoding is estimated as follows:

\[ P_e = C_{ps} \cdot CPI \cdot E_c \] (4.13)

where CPI is the number of CPU cycles to perform one basic instruction and \( E_c \) is the energy depletion per cycle. On the other hand, for a direct transmission to the sink, the transmission power consumption is calculated as:

\[ P_t = \sum (\alpha + \beta \cdot d^n) \cdot R_{av} \] (4.14)

where \( \alpha \) is a constant coefficient related to coding and modulation, \( \beta \) is the amplifier energy coefficient, \( d \) is the transmission distance, and \( \eta \) is path loss exponent.

4.2.3 Experiments and Results

For our analysis, we use the topology shown in Figure 4.4, consisting of four video

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_r )</td>
<td>Frame rate</td>
<td>15 fps</td>
</tr>
<tr>
<td>( N )</td>
<td>Number of frames</td>
<td>150 frames</td>
</tr>
<tr>
<td>( k )</td>
<td>The length of sub-shot</td>
<td>60 frames</td>
</tr>
<tr>
<td>CPI</td>
<td>Average cycle per instruction</td>
<td>1.78</td>
</tr>
<tr>
<td>( E_c )</td>
<td>Energy consumption per cycle</td>
<td>1.215e-9 J/cycle</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Energy cost for transmitting 1 bit</td>
<td>1e-9 J/b/m^4</td>
</tr>
<tr>
<td>( \beta )</td>
<td>Transmit amplifier coefficient</td>
<td>5e-8 J/b</td>
</tr>
<tr>
<td>( \eta )</td>
<td>Path loss exponent</td>
<td>3.5</td>
</tr>
</tbody>
</table>
nodes and a sink. The parameters shown in Table 4.7 are used for the experiments. In order to analyze the effect of different video sources and encoding configurations, two sets of experiments are conducted. In the first experiment, the nodes’ encoder parameter settings are set to be the same in all scenarios. However, the video sources used in each scenario vary. On the other hand, in the second experiment, the nodes are configured to use the same set of video sources in all scenarios, while the nodes’ encoding parameter settings and the nodes’ distance to the sink are varied.

The scenarios’ configuration for the first experiment is shown in Table 4.8. In the first scenario, the nodes are using the videos obtained from the first shot: camera1_shot1, camera2_shot1, camera3_shot1, and camera4_shot1. On the other hand, in the second scenario of the first experiment, the videos used are the videos obtained from the second shot, and so on. Note that, for this experiment, the $M_L$ value is set equal to six.

Figure 4.10 shows the estimated nodes’ power consumption in the first experiment. The figure shows that the nodes’ power consumption in each scenario is not the same. We

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Test sequences used</th>
<th>Distance to the sink</th>
<th>GOP size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>camera1_shot1 (node 1), camera2_shot1 (node 2), camera3_shot1 (node 3), camera4_shot1 (node 4)</td>
<td>3m</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>camera1_shot2 (node 1), camera2_shot2 (node 2), camera3_shot2 (node 3), camera4_shot2 (node 4)</td>
<td>3m</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>camera1_shot3 (node 1), camera2_shot3 (node 2), camera3_shot3 (node 3), camera4_shot3 (node 4)</td>
<td>3m</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>camera1_shot4 (node 1), camera2_shot4 (node 2), camera3_shot4 (node 3), camera4_shot4 (node 4)</td>
<td>3m</td>
<td>8</td>
</tr>
</tbody>
</table>
can also observe that the trend of nodes’ power consumption profile for each scenario varies. For example, Figure 4.10(a) that corresponds to scenario 1 shows that the node that consumes the highest power consumption is node 3. However, the difference in terms of total power consumption between node 3 and the other nodes in this scenario is not significant. In the other scenarios (i.e., scenario 2, scenario 3 and scenario 4), however, the node that has the highest power consumption is node 1. It is interesting to see that the encoding power consumption of each node in each scenario is not the same, even though all nodes are using the same encoding parameter settings in this experiment. In addition, the variance of nodes’ total power consumption varies between one scenario and the other. In terms of the video surveillance network’s average power consumption, we obtained the

![Figure 4.10: Nodes’ power consumption in experiment 1 (a) scenario 1, (b) scenario 2, (c) scenario 3, (d) scenario 4](image-url)
following values: 7.756 W (scenario 1), 7.843 W (scenario 2), 7.787 W (scenario 3) and 7.824 W (scenario 4). These results show that the content captured by each camera node affect not only the node’s power consumption but also the video surveillance network’s average power consumption.

In the second set of experiments, the nodes are set to use the videos from the first shot. However, the nodes’ distance to the sink and the GOP size are varied. The $M_L$ is set equal to six, similar to the first experiment. The configuration used in the second experiment is summarized in Table 4.9. Figure 4.11 shows the estimated nodes’ power consumption in this experiment. Figure 4.11(a) and Figure 4.11(b) show the nodes’ power consumption when the nodes’ distance to the sink is equal to 1.5m for scenario 1 and scenario 2, respectively. However, the GOP size is set equal to 2 (scenario 1) and 16 (scenario 2). It can be seen from these figures that when the distance to the sink is small, using smaller GOP size will reduce the node's power consumption. The video surveillance network's average power consumptions shown by these figures are 7.388 W (scenario 1, Table 4.9: Experiment 2 scenarios

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Test sequences used</th>
<th>Distance to the sink</th>
<th>GOP size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>camera1_shot1 (node 1), camera2_shot1 (node 2), camera3_shot1 (node 3), camera4_shot1 (node 4)</td>
<td>1.5m</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>camera1_shot1 (node 1), camera2_shot1 (node 2), camera3_shot1 (node 3), camera4_shot1 (node 4)</td>
<td>1.5m</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>camera1_shot1 (node 1), camera2_shot1 (node 2), camera3_shot1 (node 3), camera4_shot1 (node 4)</td>
<td>5m</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>camera1_shot1 (node 1), camera2_shot1 (node 2), camera3_shot1 (node 3), camera4_shot1 (node 4)</td>
<td>5m</td>
<td>16</td>
</tr>
</tbody>
</table>
GOP=2), and 7.451 W (scenario 2, GOP=16). The node’s power consumption can be further reduced if the nodes’ are configured to use GOP equal to one. In this case, the video surveillance network’s average power consumption will be equal to 7.316 W.

Furthermore, Figure 4.11(c) and Figure 4.11(d) show the nodes’ power consumption when $d$ is equal to 5 m. Similar to the previous case, the GOP size is set equal to 2 and 16 for scenario 3 and scenario 4, respectively. It can be seen clearly from these figures that the cost of transmitting the encoded video increased tremendously as compared with the first two scenarios when $d$ is smaller. Therefore, when the nodes’ distance from the sink is large, the node’s power consumption can be reduced if bigger GOP sizes are used. Comparing Figure 4.11(c) and Figure 4.11(d) we observe that the video

![Figure 4.11: Nodes’ power consumption in experiment 2 (a) scenario 1, (b) scenario 2, (c) scenario 3, (d) scenario 4](image)
surveillance network’s average power consumptions for these scenarios are 14.607 W (scenario 3) and 8.616 W (scenario 4). These results show that the node’s power consumption depends on the encoding configuration used and the distance between the node and the sink.

4.3 Summary

In this chapter, coding complexity and bitrate models were proposed. The models were developed by considering the video encoding parameters that significantly affect the coding complexity and bitrate. Through an adaptive scheme in adjusting model parameters, it was shown that the model estimation error can be reduced. Using the proposed models, a study on the video surveillance network node’s power consumption under different scenarios that involved the use of various video content, encoding configurations, and nodes’ distance from the sink were analyzed. It is shown that the nodes’ power consumption depends on the encoding parameter settings and video content captured by the node. In the next chapter, in addition to encoding parameters, the spatial and temporal complexity of the content will be incorporated into the bitrate and coding complexity models. Furthermore, in order to comply with the bandwidth constraint, the effect of different QP settings will also be considered.
Chapter 5: Spatial and Temporal Complexity of Content and Video Surveillance Network’s Power Consumption

In Chapter 3, we show that coding complexity and bitrate of videos depend on the encoder parameter settings and the video content. In order to spend energy efficiently, the nodes in a video surveillance network need to be able to estimate the coding complexity and bitrate for the captured content and utilize the models to minimize its power consumption. In this chapter, we extend the models designed in Chapter 4 by incorporating more encoding parameters, i.e., the quantization parameter, and the scene’s complexity of content. An optimization framework that utilizes the proposed coding complexity and bitrate models to minimize the video surveillance network’s power consumption is also proposed.

5.1 Methodology

Our objective here is to model the effect of content complexity and encoding parameters on bitrate and coding complexity. To this end, we consider the effect of spatial/temporal information of video, GOP size, complexity level of motion estimation, and QP. The following subsections explain how the models were developed.
5.1.1 Content Complexity

In order to incorporate the effect of content complexity in our models, the average spatial complexity (SA) and temporal complexity (TA) are used. To this end, we calculate SA and TA according to [115], as follows:

\[
SA = \text{mean}_{\text{frame}} \{ \text{std}_{\text{space}}[\text{Sobel}(F_n)] \} \tag{5.1}
\]

\[
TA = \text{mean}_{\text{frame}} \{ \text{std}_{\text{space}}[F_n - F_{n-1}] \} \tag{5.2}
\]

where \text{std} stands for the mean of standard deviation, \text{Sobel} refers to Sobel edge detection filter and \( F_n \) denotes the \( n \)th frame of the video sequence. The overall complexity of the content is a function of SA and TA (i.e., \( f(SA, TA) \)). Considering that the intra prediction technique considers only the redundancy of spatial information within each frame while the inter prediction technique exploits both the temporal and spatial redundancy, the effects of content complexity on intra and inter prediction are investigated separately. For
the intra-coded frames, \( f(.) \) depends only on SA. However, in the case of inter-coded frames, \( f(.) \) depends on both SA and TA. To estimate \( f(.) \) for inter and intra-coded frames, a set of possible functions are presumed as suggested in [115] (see Table 5.1 and Table 5.2). Subsection 5.1.4 discusses the process for estimating the \( f(.) \) and finding its parameters \( a_i \).

### 5.1.2 Design of Bitrate Model

The average bitrate of a frame in a video sequence can be estimated as follows:

\[
\overline{R} = \frac{n_I \cdot \overline{R}_I + n_P \cdot \overline{R}_P}{N}
\]

\[n_I \propto \frac{N}{GOP}
\]

\[n_P \propto \frac{N \cdot (GOP - 1)}{GOP}
\]

where \( \overline{R}_I \) is the estimated average bitrate of I-frames, \( \overline{R}_P \) is the estimated average bitrate of P-frames, \( N \) is the total number of frames, \( n_I \) is the number of I-frames, \( n_P \) is the number of P-frames, and \( GOP \) refers to group of pictures (GOP) size. In general, in order to obtain a higher compression performance, a more complex and computationally expensive encoding scheme is required. For instance, increasing the \( GOP \) size reduces the bitrate at the cost of higher coding complexity [141]. To model \( \overline{R}_I \) in (5.3), we need to take into account that the bitrate of an I-frame depends on the \( QP \) as well as the content complexity. Thus, \( \overline{R}_I \) is modeled as follows:

\[
\overline{R}_I = \overline{R}_I \cdot g_I(QP)
\]

where \( \overline{R}_I \) is the bitrate function related to content complexity of I-frames and \( g_I \) is a bitrate function related to \( QP \). To estimate \( \overline{R}_I \), we follow the approach proposed in [115] and
consider two possible functions based on the average spatial complexity (SA) unit as shown in Table 5.1. The model parameters $a_1$ and $a_0$ for $\bar{R}_i$ are estimated using least square regression (see subsection 5.1.4). For $g(.)$, we use a slightly modified version of the power function used in [115] as follows:

$$g_i(QP) = w \cdot \left( \frac{QP}{QP_{\min}} \right)^{f_i}$$

(5.5)

where $QP$ stands for the quantization parameter used for video compression, $w$ is a model parameter that is estimated using least mean square regression, $QP_{\min}$ refers to the smallest $QP$ value that can result in a decent quality compressed video in a surveillance system with limited recourses, and $f_i$ is a function of SA unit. The parameter $w$ is restricted such that $1-\zeta \leq w \leq 1-\zeta$, where $\zeta$ is a small number (the value of $\zeta$ is set equal to 0.1 throughout this chapter). This restriction is made to ensure that when $QP$ is equal to $QP_{\min}$, the value of $w$ is close to one. In this study, $QP_{\min}$ is set to 28 because we consider this as the smallest $QP$ value that can deliver a decent quality video in a low-cost surveillance system. For $f_i$, two possible functions based on the SA unit as shown in Table 5.1, are considered.
To model $\bar{R}_P$ in (5.3), we need to consider that the bitrate of the P-frames depends on temporal content complexity, spatial content complexity, and motion estimation as well as $QP$. Thus we model $\bar{R}_P$ as follows:

$$\bar{R}_P = \bar{R}_P \cdot h_p(\delta_{ML}) \cdot g_p(QP)$$  \hspace{1cm} (5.6)$$

where $\bar{R}_P$ is the bitrate function related to temporal and spatial complexity of P-frames, $g_p$ is a bitrate function related to $QP$, and $h_p$ is a bitrate function related to motion estimation. $\bar{R}_P$ is estimated by considering the six possible functions shown in Table 5.2. The bitrate function related to $QP$, $g_p$, is modeled similar to the one used for I-frames:

$$g_p(QP) = w \cdot \left( \frac{QP}{QP_{\text{min}}} \right)^{f_p}$$  \hspace{1cm} (5.7)$$

where $f_p$ is a function of temporal and spatial complexity (i.e., TA and SA). Six possible functions based on TA and SA as shown in Table 5.2, can be considered for $f_p$. To model $h_p$ in (5.6), we need to investigate the effect of motion estimation on bitrate. Using more variety of block-size candidates for motion estimation results in a better prediction and consequently higher compression performance at the expense of increased complexity [141]. There are seven block sizes defined for inter-prediction in H.264/AVC. Depending on the block-size candidates used in the encoding process, motion estimation (ME) can be classified into different levels of complexity (called $M_L$) as shown in Table 5.3 [141]. In this Table, $\delta_{ML}$ is defined as the ratio of normalized complexity increase for different $M_L$ with respect to the case where $M_L$ is equal to one (see [141] for more details). According to the study in [141], $h_p$ can be modeled using a logistic function as follows:
\[ h_p = f_p + \frac{(1 - f_p)}{1 + \exp\left(w_2 \cdot (\delta_{ML} - w_3)\right)} \]  (5.8)

where \( f_p \) denotes the effect of content complexity that can be represented by the functions shown in Table 5.2, \( w_2 \) and \( w_3 \) are model parameters that will be obtained using least square regression.

### 5.1.3 Design of Coding Complexity Model

In a similar way to bitrate modeling, the coding complexity is modeled as follows:

\[
\bar{C} = \frac{n_I \cdot \tilde{C}_I + n_P \cdot \tilde{C}_P}{N} \\
\tilde{C}_I = \bar{C}_I \cdot v_I(QP) \\
\tilde{C}_P = \bar{C}_P \cdot u_P(\delta_{ML}) \cdot v_P(QP) \tag{5.9}
\]

where \( \tilde{C}_I \) and \( \tilde{C}_P \) are the average coding complexity of I-frames and P-frames respectively, \( u(.) \) is a coding complexity function related to the ME process, and \( v(.) \) is a coding complexity function related to \( QP. \) \( \bar{C}_I \) and \( \bar{C}_P \) are the complexity functions related to the content complexity of I-frames and P-frames, respectively. \( \bar{C}_I \) will be estimated from content complexity functions shown in Table 5.1, while function \( v_I \) is modeled as:

\[
v_I(QP) = w \cdot \left( \frac{QP}{QP_{min}} \right)^{\nu_I} \tag{5.10}
\]

Furthermore, the \( \bar{C}_P \) will be estimated using content complexity functions shown in Table 5.2, while function \( v_P \) is modeled as:

\[
v_P(QP) = w \cdot \left( \frac{QP}{QP_{min}} \right)^{\nu_P} \tag{5.11}
\]
Our previous work has shown that coding complexity increases linearly with the increase of $\delta_{ML}$ [141]. However, the coding complexity of P-frames also depends on the content complexity. Therefore, the coding complexity function related to the ME process, i.e., $u_p$, is modeled as follows:

$$u_p(\delta_{ML}) = (1 + \delta_{ML} \cdot f_p)$$

(5.12)

### 5.1.4 Estimation of the Model Parameters

In order to find the model parameters shown in (5.1) to (5.12), we utilize a set of videos for training and use the least square regression technique to estimate the parameters. To this end, the training videos are encoded with different GOP, $M_L$ values and $QP$ as it is explained in the following subsections. In order to select the best functions for content complexity for our models, we calculate the root mean square error (RMSE) and Pearson’s Correlation coefficient (PC). The function providing the least RMSE and highest PC is selected to be integrated into the model. In the training process, we take into account that some data points might be outliers and have extreme/illegitimate values during model fitting. The presence of these points can lead to inflated error rates and substantial distortion in parameter estimation [143]. There are different approaches in dealing with extreme data in the training sample, such as removal, transformation, truncation and robust method [143]. In this study, we use one of the robust methods proposed in [144] by replacing extreme data with the next closest valid point in the dataset.
5.1.4.1 Video Sequences

To obtain representative data with different temporal and spatial complexity levels, we use the video dataset developed in Section 3.2.1. In our study, we require selecting a representative set of videos with different spatial and temporal complexity (SA and TA) levels from the captured database for training our model. In this regard, the following scenes are used as the training set in this study:

1. scene *office_act4* that represents scene with low SA and low TA,
2. scene *classroom_act2* that represents scene with medium SA and medium TA, and
3. scene *party_act1* that represents scene with high SA and high TA.

Each of the above three scenes consists of nine videos. Thus, 27 videos are used as the training set. The remaining 81 videos from the other scenes and activity levels are used as the test set for the models.

5.1.4.2 Video Encoding Parameters

In our study, due to the limitation in the energy and processing resources of the video surveillance network application, less complex encoder configurations are deployed. For instance, the baseline profile of H.264/AVC, which only utilizes I and P frames (no B-frame) and is suitable for low complexity applications is used. The other encoding settings in this study include using context-adaptive variable-length coding (CAVLC) entropy coding, one reference frame, and SR (search range) equal to eight, while the rate distortion optimization (RDO), rate control, and the deblocking filter are disabled. In addition, the QP of the P-frames is set equal to the QP of the I-frames minus one. The H.264/AVC reference encoder software (JM 18.2) is used in our implementation. To measure coding complexity,
the instruction level profiler iprof [133] is utilized. The iprof gives us the number of basic instruction counts (IC) to perform an encoding task and has been used to evaluate the performance of different ME algorithms in MPEG video coding standardization activities. The instruction count is proven to be an objective and accurate measure of coding complexity [106], [133], [141].

5.1.4.3 Bitrate Model Parameters Estimation

If we assume the GOP size is equal to one, \( n_P \) in (5.3) will be equal to zero and thus \( \bar{R} = \bar{R}_I \cdot g_I(QP) \). Furthermore, if \( QP \) is set to 28, the value \( g_I(QP) \) will be close to one. Therefore, the average bitrate of intra coded frames \( \bar{R}_I \), which is represented by one of the functions shown in Table 5.1, is approximately equal to the average bitrate of the video frames, when \( GOP=1 \) and \( QP=28 \). To estimate the model for I-frame bitrate, the two functions shown in Table 5.1 are used to estimate the average frame bitrate of training videos encoded with \( GOP=1 \) and \( QP=28 \). The parameters of the functions are estimated using least square regression. Then, the RMSE and PC obtained from each function are calculated. The function with the least RMSE and highest PC is selected for the model. With this approach, we found that the function \( a_1 \cdot SA + a_0 \) provides the best estimation with RMSE=5.5769 and PC=0.9493. The model parameters obtained are \( a_1=1.3557 \), and \( a_0=-40.093 \). Therefore, \( \bar{R}_I \) is modeled as:

\[
\bar{R}_I = 1.3557 \cdot SA - 40.093
\]  

(5.13)

Having the estimate for \( \bar{R}_I, g_I(QP) \) can be estimated using (5.3), (5.4), and (5.5). To this end, we assume that the GOP size is equal to 1, thus \( \bar{R} = \bar{R}_I \cdot g_I(QP) \). Using the average
frame bitrate of training videos (compressed with the following configuration: $GOP=1$, $QP=28$ to 40 with step size of 2), we use (5.5) to estimate $g_i(QP)$. The least square regression technique shows that the function $a_1 \cdot \log(SA) + a_0$ from Table 5.1 provides the best result with RMSE=0.0476 and PC=0.9801 when the model parameters are equal to $w=1.0226$, $a_1=-0.2451$, and $a_0=-2.1131$. Thus, $g_i(QP)$ is formulated as follows:

$$g_i(QP) = 1.0226 \cdot \left( \frac{QP}{Q_{mn}} \right)^{-0.2451 \cdot \log(SA) - 2.1131} \tag{5.14}$$

To estimate the model for $R_P$, we first assume $M_L=1$ and $QP=28$, thus $h_p(\delta_{ML}) = 1$ and $g_p(QP) \approx 1$, respectively. Then using the estimated models for $R_i$ and $g_i(QP)$, $R_P$ can be estimated based on (5.3) and (5.6). Here the frame bitrate of training videos (encoded with $GOP=\{2, 4, 8, 16, 32, 64\}$ and $QP=28$) are used to identify which function shown in Table 5.2 results in the highest PC and lowest RMSE. The least square regression shows that the best estimation for $R_P$ is as follows:

$$R_P = 0.0223 \cdot SA \cdot TA - 1.8088 \tag{5.15}$$

resulting in minimum RMSE of 2.9867 and the highest PC of 0.9606.

The function $h_P(\delta_{ML})$ quantifies the effect of using different $M_L$ values on the bitrate. Assuming that $GOP=1$ and $QP=28$, $h_p(\delta_{ML})$ can be estimated using (5.3), (5.6), (5.8) and (5.15). To this end, the frame bitrate of training videos (encoded with $GOP=\{2, 4, 8, 16, 32, 64\}$, $M_L=\{1, 2, 3, 4, 5, 6, 7\}$, $QP=28$) is used to identify which function shown in Table 5.2 results in the highest PC and lowest RMSE. Note that the value of $\bar{R}_i$ in (5.3) is calculated using (5.4), (5.13) and (5.14). Using least square regression, we obtain the lowest RMSE of
0.2251 and highest PC of 0.6823 when the function $a_2 \cdot \log(SA) + a_1 \cdot \log(TA) + a_0$ is used with $a_2=0.012$, $a_1=0.1539$, and $a_0=0.5518$. Therefore:

$$h_p(\delta_{ML}) = b + \frac{(1-b)}{1 + \exp(5.404 \cdot (\delta_{ML} + 0.0394))}$$

(5.16)

$$b = 0.012 \cdot \log(SA) + 0.1539 \cdot \log(TA) + 0.5518$$

Note that, we have also used different functions to model $h_p(\delta_{ML})$ and found that the logistic function provide the best estimate.

Finally, $g_p(QP)$ is estimated using (5.3) and (5.13) to (5.16). Here the frame bitrate of training videos (encoded with the following configuration: $GOP= \{2, 4, 8, 16, 32, 64\}$, $M_L=\{1, 2, 3, 4, 5, 6, 7\}$, $QP= \{28, 30, 32, 34, 36, 38, 40\}$) is used. Using least square regression, we find that the function shown in Table 5.2 that has the lowest RMSE and highest PC is $a_2 \cdot SA + a_1 \cdot TA + a_0$ with the following model parameters: $w=1.1$, $a_2=0.0483$, $a_1=-0.1492$, and $a_0=-6.4872$. This provide us an estimation with RMSE=0.4704 and PC=0.6242. Thus $g_p(QP)$ is formulated as:

$$g_p(QP) = 1.1 \left( \frac{QP}{QP_{\text{min}}} \right)^{0.0483 \cdot SA - 0.1492 \cdot TA - 6.4872}$$

(5.17)

5.1.4.4 Coding Complexity Model Parameters Estimation

Similar with the bitrate model parameters estimation case, if we assume that the $GOP$ is equal to one, $n_P$ in (9) will be equal to zero and thus $\bar{C} = \bar{C} \cdot v_t(QP)$. Note that, when $QP$ is set to 28, the value $v_t(QP)$ will be close to one. Therefore, $\bar{C}_t$ can be estimated from the frame coding complexity of training videos when $GOP=1$ and $QP=28$. Recall that $\bar{C}_t$ can
be represented by any function shown in Table 5.1. Using least square regression, we noticed that function $a_1 \cdot SA + a_0$ provides the best estimate with RMSE of 0.6725 and PC of 0.8542. The model parameters for this estimation are: $a_1=-0.0889$, and $a_0=208.6018$. Therefore, $\bar{C}_I$ is modeled as follows:

$$\bar{C}_I = 0.0889 \cdot SA + 208.6018$$  \hspace{1cm} (5.18)

Having $\bar{C}_I$ modeled using (5.18), $v_I(QP)$ can then be estimated from the frame coding complexity of training videos when compressed with $GOP=1$ and $QP=\{28, 30, 32, 34, 36, 38, 40\}$. The least square regression technique shows that function $a_1 \cdot SA + a_0$ from Table 5.1 provides the best result with RMSE=0.0031 and PC=0.9048. The model parameters obtained are: $w=0.9976$, $a_1=-0.0011$, and $a_0=0.0421$. Therefore:

$$v_I(QP) = 0.9976 \cdot \left( \frac{QP}{QP_{\text{min}}} \right)^{-0.0011 \cdot SA + 0.0421}$$  \hspace{1cm} (5.19)

Now that $\bar{C}_I$ and $v_I(QP)$ are formulated, using (5.9), we can estimate the coding complexity of the video, if we know $\bar{C}_P$, $u_P(\delta_{ML})$ and $v_P(QP)$. Note that, according to Table 5.1, the value of $\delta_{ML}$ is equal to zero when $M_l$ is set to one. In this setting, the value of $u_P(\delta_{ML})$ will be equal to one (see (5.12)). Furthermore, by setting $QP$ equal to 28, the value of $v_P(QP)$ is also equal to one. Following this approach, the $\bar{C}_P$ that can be represented by the functions shown in Table 5.2 is estimated from the frame coding complexity of training videos that are compressed with the following configuration: $M_l=1$, $QP=28$ and $GOP=\{2, 4, 8, 16, 32, 64\}$. Using least square regression, we found that the best function that can represent $\bar{C}_P$ is $a_2 \cdot SA + a_1 \cdot TA + a_0$. This function provided an estimation with RMSE=0.975
and PC=0.9741. The model parameters are: $a_2=0.0639$, $a_1=0.8404$, and $a_0=168.1378$. Thus, $C_P$ is:

$$C_P = 0.0639 \cdot SA + 0.8404 \cdot TA + 168.1378$$  \hspace{1cm} (5.20)

In order to find the model parameters for $u_P(\delta_{ML})$, we use (5.18) to (5.20) in addition to setting $QP$ equal to 28, i.e., this will make the value of $v_P(QP)$ close to one. To this end, we use the frame coding complexity from training videos that are encoded with the following configuration: $GOP=\{2, 4, 8, 16, 32, 64\}$, $M_L=\{1, 2, 3, 4, 5, 6, 7\}$, and $QP=28$. The best result with RMSE=0.0151 and PC=0.9956 is obtained by function $a_2 \cdot SA + a_1 \cdot TA + a_0$ from Table 5.2. The model parameters for this function are $a_2=7.2515 \cdot 10^{-4}$, $a_1=0.0084$, and $a_0=0.2936$. Thus, $u_P(\delta_{ML})$ is formulated as:

$$u_P(\delta_{ML}) = 1 + \delta_{ML} \cdot \left(7.25 \cdot 10^{-4} \cdot SA + 0.0084 \cdot TA + 0.2936\right)$$  \hspace{1cm} (5.21)

Next, we need to find the model parameters for $v_P(QP)$. This can be done by using equations (5.18) to (5.21) and coding complexity data of the training videos with $GOP=\{2, 4, 8, 16, 32, 64\}$, $M_L=\{1, 2, 3, 4, 5, 6, 7\}$ and $QP=\{28, 30, 32, 34, 36, 38, 40\}$. We found that by using least regression, the following model parameters $w=0.9913$ $a_2=0.0011$, $a_1=-0.0077$, and $a_0=0.1335$ for function $a_2 \cdot SA + a_1 \cdot TA + a_0$ provide us with the best estimation (RMSE=0.2728 and PC=0.76). Thus, $v_P(QP)$ is then modeled as:

$$v_P(QP) = 0.9913 \left(\frac{QP}{QP_{\text{min}}}ight)^{0.0011 \cdot SA - 0.0077 \cdot TA + 0.1335}$$  \hspace{1cm} (5.22)
5.2 Experiments and Results

The performance of the proposed model is analyzed using the 81 test video sequences (note that the video sequences used for training are excluded). To obtain actual values for bitrate and coding complexity, the test video sequences are first encoded using different configurations: \( GOP = \{1, 2, 4, 8, 16, 32, 64\} \), \( M_L = \{1, 2, 3, 4, 5, 6, 7\} \) and \( QP = \{28, 30, 32, 34, 36, 38, 40\} \). Then, using the proposed models the bitrate and encoding complexity of the videos for each coding configurations are estimated. The estimated values are then compared with actual values using RMSE and mean absolute percentage error (MAPE). MAPE is formulated as follow [145]:

\[
MAPE = mean \left( \frac{|y' - y|}{y} \right) \times 100\%
\]

(5.23)

where \( y \) is the actual value and \( y' \) is the estimated one. Furthermore, the correlation between the estimated and the actual results is analyzed using Pearson’s coefficient correlation and R-squared values. The R-squared is calculated as:

\[
R - squared = 1 - \frac{\sum (y' - y)^2}{\sum (y' - mean(y))^2}
\]

(5.24)

We also compared the results of the proposed bitrate model with those of an existing scheme, which also takes into account content complexity in addition to coding parameters (similar to our scheme). Note that for estimating the coding complexity, to the best of our knowledge, there is no existing model in the literature that, as our model, takes into account content complexity in addition to coding parameters. The closest possible model that can be related to our coding complexity model is the one proposed in [119],
which measures coding complexity using the statistical information of compressed content. In other words, to obtain the required parameters for this model, video content needs to be compressed first. Hence, this approach cannot be used on unknown videos or for real-time applications. Due to these shortcomings, we have excluded this approach from our comparison study. The following subsections elaborate on our proposed models’ performance evaluation results.

### 5.2.1 Performance Evaluation of the Bitrate Model

To compare the bitrate values estimated by our model with the actual ones, the

<table>
<thead>
<tr>
<th>Scene</th>
<th>PC</th>
<th>RMSE (kbps)</th>
<th>MAPE (%)</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>office_act1</td>
<td>0.982</td>
<td>33.126</td>
<td>12.481</td>
<td>0.941</td>
</tr>
<tr>
<td>office_act2</td>
<td>0.989</td>
<td>24.155</td>
<td>16.254</td>
<td>0.964</td>
</tr>
<tr>
<td>office_act3</td>
<td>0.993</td>
<td>20.935</td>
<td>10.156</td>
<td>0.978</td>
</tr>
<tr>
<td>classroom_act1</td>
<td>0.983</td>
<td>35.596</td>
<td>13.120</td>
<td>0.949</td>
</tr>
<tr>
<td>classroom_act3</td>
<td>0.983</td>
<td>32.172</td>
<td>20.068</td>
<td>0.952</td>
</tr>
<tr>
<td>classroom_act4</td>
<td>0.986</td>
<td>31.849</td>
<td>20.026</td>
<td>0.958</td>
</tr>
<tr>
<td>party_act2</td>
<td>0.983</td>
<td>44.700</td>
<td>10.896</td>
<td>0.949</td>
</tr>
<tr>
<td>party_act3</td>
<td>0.986</td>
<td>54.229</td>
<td>16.332</td>
<td>0.920</td>
</tr>
<tr>
<td>party_act4</td>
<td>0.983</td>
<td>54.978</td>
<td>13.867</td>
<td>0.904</td>
</tr>
<tr>
<td>all sequences (81 videos)</td>
<td>0.984</td>
<td>38.582</td>
<td>14.800</td>
<td>0.949</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Video sequence</th>
<th>PC</th>
<th>RMSE (kbps)</th>
<th>MAPE (%)</th>
<th>R-squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>camera5_office_act3</td>
<td>0.999</td>
<td>4.919</td>
<td>3.842</td>
<td>0.997</td>
</tr>
<tr>
<td>camera3_party_act4</td>
<td>0.987</td>
<td>86.345</td>
<td>17.374</td>
<td>0.798</td>
</tr>
</tbody>
</table>
estimated values are multiplied by the sequences’ frame rates, as our model estimates the average bitrate per frame (see (5.3)).

Table 5.4 shows the bitrate estimation results for each scene. Note that, each scene contains videos from all nine cameras. As it is observed, the performance of the bitrate model varies over the scenes. The reported results for all the test sequences confirm high correlation between the estimated values and the actual ones (PC=0.984, R-squared 0.949) and small estimation error (RMSE=35.582, MAPE=14.8%).

Figure 5.1 shows the estimated and actual bitrate for some test videos (captured by different cameras in different scenarios) when GOP size, \( M_L \) and \( QP \) values are changing. As it is observed, there is a linear relationship between the estimated and actual bitrate values with a negligible error in some cases. To evaluate the performance of the model for a single video, the performance of our proposed model is objectively evaluated for

![Figure 5.1: Actual and estimated bitrate for varying GOP sizes, \( M_L \) and \( QP \) values for the following videos (a) camera1_office_act1, (b) camera4_office_act1, (c) camera5_office_act3, (d) camera2_classroom_act3, (e) camera6_party_act2, and (f) camera3_party_act4.](image)
camera5_office_act3 and camera3_party_act4, where the performance of our model is the best and the worst, respectively. As Table 5.5 shows, in the case of camera5_office_act3 test sequence, the highest correlation between the estimated and actual bitrate values (PC=0.99954, R-squared=0.99741) is obtained and the estimation error is quite negligible (RMSE=4.9189, MAPE=3.8417%), however for camera3_party_act4 video, the results are not as good as camera5_office_act3 video with PC=0.98679, RMSE=86.345, MAPE=17.374% and R-squared=0.79784.

5.2.1.1 Comparison with an Existing Bitrate Estimation Model

To investigate the performance of the proposed bitrate model, the estimated bitrates using our model are compared with those estimated by the model presented in [115] that utilizes the spatial and temporal information. Considering that the proposed model in [115] uses a fixed GOP size which is equal to the frame rate, for our experiments we used GOP = 16 which is quite close to the frame rate of the test videos (15 fps). Furthermore, since the technique in [115] does not take into account the effect of block size candidates used for motion estimation, we assume that the encoder uses the configuration setting which is equal to using ML=7. Since GOP size, ML and frame rate are fixed, QP is the only varying parameter (i.e., QP = {28, 30, 32, 34, 36, 38, 40}).

<table>
<thead>
<tr>
<th>Performance metric</th>
<th>Proposed bitrate model</th>
<th>Lotterman et. al. [113]</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>0.966</td>
<td>0.922</td>
</tr>
<tr>
<td>RMSE (kbps)</td>
<td>29.6453</td>
<td>45.676</td>
</tr>
<tr>
<td>MAPE (%)</td>
<td>17.819</td>
<td>19.201</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.915</td>
<td>0.6709</td>
</tr>
</tbody>
</table>
5.2.2 Performance Evaluation of the Coding Complexity Model

To compare the complexity values estimated by our model with the actual ones, in order to obtain the coding complexity per second, (5.4) is multiplied by the frame rate. Table 5.7 shows the coding complexity estimation results for each scene. Note that, each sequence.
scene contains videos from all nine cameras. As it is observed, there is a high correlation between the estimated and actual coding complexity values (PC=0.996, R-squared 0.978) and the estimation error is low (RMSE=53.368, MAPE=1.224%). Figure 5.2 shows the estimated and actual bitrates for some test videos (captured via different cameras in different scenes) when GOP size, ML and QP values are changing. As it is observed, there is a linear relationship between the estimated and actual coding complexity values with a negligible error in some cases. To evaluate the performance of the model for a single video, the performance of our proposed model is objectively evaluated for camera4_office1_act1 and camera1_office1_act1, where the performance of our model is the best and the worst, respectively. As Table 5.8 shows, in the case of camera4_office1_act1 test sequence, the highest correlation between the estimated and actual bitrate values (PC=0.99858, R-squared=0.99086) is obtained and the estimation error is quite negligible (RMSE=33.344, MAPE=1.224%).

![Figure 5.2: Actual and estimated coding complexity for varying GOP sizes, ML and QP values for the following videos (a) camera1_office_act1, (b) camera4_office_act1, (c) camera5_office_act3, (d) camera2_classroom_act3, (e) camera6_party_act2, and (f) camera3_party_act4.](image-url)
MAPE=0.84909%). However for camera1_office1_act1 video, the results are not as good as for camera4_office1_act1, with PC=0.99722, RMSE=87.27, MAPE=1.8634% and R-squared=0.93303.

5.3 Power Consumption Estimation using the Proposed Models

In order to estimate the video surveillance network’s power consumption, we consider the network layout shown in Figure 5.3. In order to perform the experiments, the network is modeled as a graph $G(V,L)$, where $V$ is the set of nodes and $L$ is the set of links. It is assumed that a standard medium access control (MAC) protocol is applied to resolve the link interference problem. Note that $i$ and $j$ are two nodes in our network. Node $i$ can communicate with node $j$ if a link between those nodes ($L_{ij} \in L$) exists. In addition to that, we also assume that the link $L_{ij}$ exists if $i$ and $j$ are direct neighbors. Node $i$ can capture and encode video, and then generate video traffic with bitrate $\overline{R_i}$. Note that, $\overline{R_i}$ is estimated using our proposed model. Each node can also relay the traffic from the upstream nodes. The flow conservation law at each node is then:

$$\sum r_{ij} - \sum r_{ki} = \overline{R}_i \cdot F_r$$  \hspace{1cm} (5.25)

where $F_r$ is the video frame rate, $r_{ij}$ denotes the outgoing bitrate at $L_{ij}$, $r_{ki}$ is the incoming bitrate at $L_{ki}$, and $L_{ij}, L_{ki} \in L$. The sum of transmission rates of all the nodes is constrained to be equal to the available bandwidth ($B$):

$$F_r \cdot \sum \overline{R}_i \leq B$$  \hspace{1cm} (5.26)
As reported in [140], the network’s power consumption is reduced if the nodes that are further from the sink are allocated with smaller bitrate than the nodes’ that are closer to the sink. However, due to the difference in scene complexity, video captured by node $i$ may have less bitrate than node $j$ that is located further from the sink that node $i$. On the other hand, all data transmissions are assumed to go through the first node. Therefore, we assume that the node closest to the sink has a higher bitrate than the other nodes:

$$\overline{R}_i \leq \overline{R}_i \tag{5.27}$$

The general model suggested by [127] for energy consumption of a wireless communication transmitter and receiver is used in this study. Thus, the total transmission power consumption of node $i$ is equal to the sum of all power consumed to transmit data to other nodes within its transmission range as follows:

$$P_{ti} = \sum (\alpha + \beta \cdot d_{ij}^{-\eta}) \cdot r_{ij} \tag{5.28}$$

where, $P_{ti}$ is the transmission power consumption of node $i$, $\alpha$ and $\beta$ are constant coefficients, $\eta$ is the path loss exponent, and $d_{ij}$ is the distance between node $i$ and node $j$. 

![Figure 5.3: Video surveillance network layout](image)
The total reception power consumption of node $i$ is equal to the sum of all the power consumed to receive data from other nodes, as formulated below, where $\lambda$ is a constant coefficient:

$$P_r = \sum \lambda \cdot r_i$$  \hspace{1cm} (5.29)

To measure the required energy for encoding at each node, we need to multiply the total number of required cycles for encoding with the average energy depleted per cycle. The total number of cycles to execute encoding can be estimated as the multiplication of the total number instruction count with a parameter called cycle per instruction (CPI). The average encoding power consumption is then estimated as [141]:

$$P_{ei} = \bar{C}_i \cdot F_r \cdot CPI \cdot E_c$$  \hspace{1cm} (5.30)

where, $\bar{C}_i$ is the coding complexity of node $i$ estimated using the proposed model, $F_r$ is the frame rate, $CPI$ is the average number of cycles per instruction of the CPU, and $E_c$ is the energy depleted per cycle.

Furthermore, the total energy dissipation at a sensor node consists of the encoding power consumption ($P_{ei}$), the transmission power consumption ($P_t$) and the reception power consumption ($P_r$):

$$P_i = P_{ei} + P_t + P_r$$  \hspace{1cm} (5.31)

The objective is to minimize the maximum energy consumption among all nodes, i.e., minimize $P_{net}$ where $P_i \leq P_{net}$ $\forall i \in N$. This optimization problem is shown in Figure 5.4. Our goal is to find the encoding parameter settings, i.e., GOP and ML, for each node that minimize the video surveillance network's power consumption. Note that we assume that
Figure 5.4: Video surveillance network power consumption minimization
all nodes in the surveillance network use the same $QP$. However, different scenes may need to use different $QP$ values. The minimum $QP$ values that can accommodate the network bandwidth for the scene will be used. As can be seen from the figure, the encoding power consumption of node $i \in V$ is calculated from the coding complexity at node $i$. The coding complexity of node $i$ itself is estimated using the coding complexity model described in the previous sections. In order to estimate the coding complexity of node $i$, the spatial and temporal information of the video captured on the node, i.e., $SA_i$ and $TA_i$, as well as the variable GOP size and ML value are needed. On the other hand, the node’s transmission power consumption depends on the bitrate of the encoded video. Similar to the coding complexity, the spatial and temporal information of video as well as the variable GOP size and ML value of node $i$ will be used to estimate the bitrate of the node. Note that, it is assumed that instead of the real value $\delta_{ML}$, we will use the $ML$ value in allocating the encoder configuration at the node in the surveillance network. Therefore, in the optimization procedure shown in Figure 5.4, an additional equation to calculate $\delta_{ML}$ is used,

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>Energy cost for transmitting 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Transmit amplifier coefficient</td>
<td>$1.3\times10^{-8}$ J/Mb/m$^4$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Energy cost for receiving 1 bit</td>
<td>0.5 J/Mb</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Path loss exponent</td>
<td>4</td>
</tr>
<tr>
<td>CPI</td>
<td>XScale average cycle per instruction</td>
<td>1.78</td>
</tr>
<tr>
<td>$E_c$</td>
<td>Energy depleted per cycle</td>
<td>1.215 nJ</td>
</tr>
<tr>
<td>$B$</td>
<td>Network Bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>$d$</td>
<td>Distance between node</td>
<td>5m</td>
</tr>
</tbody>
</table>
\[ \delta_{ML} = 0.1704 \cdot M_L - 0.1943. \]

At the beginning of the optimization procedure, each node is assigned to use the initial GOP value of 16 while \( M_L \) is set equal to seven. Since the GOP and \( M_L \) values obtained from the optimization procedure are in the form of floating number, the GOP and \( M_L \) values were rounded to the nearest integer. Then the optimization algorithm is re-run with the values of the GOP and \( M_L \) considered as constant, i.e., not variables. If the result from the rounded values violates the bandwidth constraint, we use the ceil function to round the GOP and \( M_L \) values obtained from the first run of the optimization, followed by a re-run of the optimization with a newly ceiled constant of the GOP and \( M_L \) values. Table 5.9 shows the parameters used in our experiments.

Table 5.10 shows the GOP and \( M_L \) values of each node for office_act2, classroom_act3 and party_act4 obtained from the optimization procedure. In terms of video quality, each

<table>
<thead>
<tr>
<th>Node</th>
<th>Test sequence</th>
<th>Test sequence</th>
<th>Test sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>office_act2</td>
<td>classroom_act3</td>
<td>party_act4</td>
</tr>
<tr>
<td></td>
<td>GOP</td>
<td>ML</td>
<td>GOP</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>22</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>1</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>26</td>
<td>1</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>23</td>
<td>1</td>
<td>36</td>
</tr>
<tr>
<td>9</td>
<td>35</td>
<td>2</td>
<td>64</td>
</tr>
</tbody>
</table>
A video node is assumed to use the same QP. However, the QP used for each scene may not be the same due to bandwidth restrictions. In this regard, our experiment suggests that the QP for the three scenes analyzed, i.e., office_act2, classroom_act3, and party_act4, as mentioned in the previous paragraphs, are using the following QPs: 29, 30 and 37, respectively.

Furthermore, Figure 5.6 shows the nodes’ power consumption profile for the three scenes analyzed. It can be seen from this figure that the nodes’ maximum power consumption for the three scenes are not the same. This is because the content captured by each camera node in each scene is not the same. Note that, the communication power consumption shown in Figure 5.6 includes the transmission and reception power consumption. It can be seen from the figure that the nodes’ maximum power consumption...
(P_{\text{net}}) in the office\_act2 scene are 7.55 W. On the other hand, the P_{\text{net}} for the classroom\_act3 and party\_act4 scene were 7.14W and 7.36W, respectively. As it is observed from the figure, the encoding power consumption per each node in each scene is not the same.

The nodes’ video quality encoded with the above-mentioned QPs and encoding parameters shown in Table 5.10 in terms of PSNR is shown in Figure 5.5. It can be seen that the PSNR of the encoded video per each node of the same scene is not the same. This is due to the fact that the videos captured by each node are not the same and the difference in the GOP and M_{L} values used to encode the video. However, the PSNR difference of these videos is not significant, except for the party\_act4 scene. This is because of the variation of SA and TA in that particular scene where the video captured by node 4, 5 and 6 have lower SA and TA values than the other videos captured at the same scene setting.

Table 5.11 shows the P_{\text{net}} of each test scene obtained by 1) the proposed optimization approach utilizing the coding complexity and bitrate models (Proposed -
estimated), 2) the proposed optimization approach and the coding complexity and bitrate of the videos (Proposed - actual), and 3) using the technique proposed [140]. We observe that the proposed algorithm manages to reduce the video surveillance network power consumption obtained by the technique proposed in [140] by 1.15W on average. In addition, the table shows that the results obtained by the proposed algorithm utilizing the models described in Section 5.1 are very close to the results obtained using the measured coding complexity and bitrate data. The range of error is between 0.002W and 0.36W. Overall, the proposed method manages to estimate the video surveillance network’s power consumption within a marginal error of RMSE=0.195 and MAPE=2.23%.

5.4 Summary

In this chapter, we proposed models for estimating the coding complexity and bitrate for H.264/AVC-based video surveillance applications. The proposed models incorporate content’s spatial and temporal complexity as well as encoder parameter settings that significantly affect compression performance. For the purpose of the analysis, a large number of real-life content with diverse spatial and temporal information was captured. Some of the captured videos were used as the training set while the remaining videos were used to test our models. It was shown that the proposed models are able to estimate the coding complexity and bitrate with high accuracy. It was also shown that the proposed models outperform existing techniques in all performance measurements, i.e., Pearson’s correlation coefficient, root mean square error, mean absolute percentage error, and R-squared values.
We also proposed an optimization procedure utilizing the proposed models. The objective is to find the nodes’ encoding parameter settings that minimize the video surveillance network’s power consumption, while considering the following factors: 1) that videos captured by each node have different spatial and temporal complexity, and 2) the nodes are located at different distance from the sink and thus may need to relay information from the other nodes. Overall, the proposed method manages to estimate the video surveillance network’s power consumption within a marginal error of RMSE=0.195 and MAPE=2.23%.
Chapter 6: Conclusions and Future Work

6.1 Summary of Thesis Contributions

This thesis studies the issues related to the trade-off between video encoding complexity and communication energy consumption for an energy efficient VSN for surveillance applications. The main goal of the thesis is to investigate the factors affecting the video surveillance networks’ power consumption and propose a methodology that exploits the trade-off between these factors in order to minimize the VSN’s overall power consumption. The methods proposed in this thesis incorporate various aspects of video encoding and the video surveillance network’s deployment settings to address this challenge. These include video encoding parameters used at each node, the spatial and temporal complexity of content and the network topology. The performance of the proposed methods has been verified through large-scale experiments using a dataset comprised of real-life captured content.

We started our investigation by proposing a practical implementation of the existing power-rate-distortion model such that the nodes’ bitrate in the video surveillance network is controlled using variable frame rate. This was achieved by assuming that the video surveillance networks’ deployment area can be divided into several zones, where the encoding configuration of nodes in each zone is related to the video frame rate. This allowed us to regulate the overall power consumption by controlling the video frame rate. It was shown that in terms of system lifetime, the proposed approach is better than the common single configuration setting, but inferior than the optimal PRD-based solution.
reported in [63]. However, controlling the bitrate alone is not enough since there are other encoding parameters that provide better bitrate control and video quality trade-off. For this purpose, we investigated the effect of various encoder parameters towards coding complexity and bitrate in Chapter 3. The objective was to determine the encoding parameters that have the highest impact on the coding complexity and bitrate of the video. This is because the coding complexity and bitrate of videos determine the encoding and communication power consumption of a video surveillance network. We showed that the GOP size significantly affects coding complexity and bitrate. In addition, we also showed that the tuples containing encoding configuration settings, coding complexity and bitrate of the videos that were stored in a form of a lookup table could be used to exploit the trade-off between encoding and communication power consumption. Furthermore, different nodes in a video surveillance network may capture a scene from different angles such that each view is unique. Video content affects coding complexity and bitrate and thus it needs to be taken into consideration in minimizing the video surveillance network's power consumption. For that purpose, we needed to have a video surveillance dataset that has a wide range of content complexity. Unfortunately, such video surveillance dataset was not available.

In order to consider the effect of content, we generated a video surveillance dataset containing a diverse set of content. The dataset was generated by installing nine HD cameras in one of our labs to capture simulated events with varying degree of spatial details and motion activity levels. The cameras were arranged such that their point of views are different. This arrangement provided us with a video dataset consisting of a large
number of unique videos that have a wide variety of content complexity. The dataset was made available for public use to facilitate further research in this area [121]. In our first attempt to account the effect of content complexities on power consumption minimization, we proposed a classification technique to divide the videos into training and test sets. Furthermore, we also proposed a look-up table denoting different encoding configuration ID that can be used by each node in the video surveillance network. This look-up table was then used in the proposed fairness-based power consumption minimization for each scene setting. Although the look-up table approach showed a good performance, it was not practical nor efficient for a real-life implementation of video surveillance networks. In this regard, we believed that models that can estimate the coding complexity and bitrate of videos will have a higher performance than the look-up table approach.

In Chapter 4, models to estimate the coding complexity and bitrate were proposed. These models considered the effect of GOP size and the type and variety of block size candidates used in motion estimation on coding complexity and bitrate. In Chapter 4, the effect of content complexities was represented by using data from encoding the first two frames of videos of each scene. In order to improve the estimation accuracy, in Chapter 5, we improved our models by incorporating the effect of quantization parameter and content complexity in terms of spatial and temporal information. The models were trained and evaluated using the dataset created in Chapter 3. A comprehensive examination of our model using test videos with different error measurement metrics was performed. The comparison of the proposed model and an existing state-of-the-art technique reconfirmed the superiority of our approach. Finally, we combined the proposed models that consider
encoding parameters and content complexities, network layout and nodes’ position in the network in an optimization procedure to minimize the overall power consumption. The objective was to find the nodes’ encoding configuration that leads to minimal power consumption. The results obtained from the proposed optimization technique and the work described in Chapter 3 were compared. It was shown that, compared to existing techniques, the proposed optimization technique managed to significantly reduce the video surveillance network's power consumption.

6.2 Significance and Potential Applications of the Research

The research presented in this thesis aims at determining the encoder parameter settings per each video surveillance network’s node that lead to minimum power consumption. Our findings in Chapter 3 are helpful in defining encoder parameter settings that significantly affect coding complexity and bitrate of the video. The same chapter also highlighted the fact that the complexity of spatial and temporal content have a direct impact on coding efficiency and bitrate, something that unfortunately has been neglected by existing works in the video surveillance network field. An approach to classifying videos and events based on content complexity was suggested in Chapter 3. In addition to that, a simple look-up table method for minimizing video surveillance networks’ power consumption was proposed.

The models proposed in Chapter 4 and defined in Chapter 5 have various potential applications, especially for devices or infrastructures with limited resources. Such applications include wireless surveillance, ad-hoc video monitoring, habitat monitoring, video-based Internet of Things, wearable MPEG [146], green MPEG [147] and any
technology that allows the proliferation of video application on pervasive computing platforms. All the above-mentioned applications and technologies could benefit from the findings investigated in this thesis.

6.3 Future Work

6.3.1 The Use of HEVC

HEVC is the latest MPEG ITU-T video coding standard that comes with an increased video compression performance at the cost of almost double coding complexity than H.264/AVC [148]. Although HEVC is currently not widely used in many low-power devices, future adoption of HEVC is inevitable. There has been some effort to analyze HEVC coding complexity and bitrate performance in the literature. Unfortunately, the focus of these efforts was mostly on the decoding end [149]. Note that, HEVC comes with plenty of coding tools and modes that were not available in H.264/AVC. However, we can follow the same approach described in Chapter 3 and extend it to apply to the new HEVC coding standard in the future.

6.3.2 Event-triggered Content-Based Video Surveillance Network

The model developed in this thesis assumes that all the nodes in the video surveillance network encode and transmit videos of the captured event. This is performed in order to simulate the video surveillance network under a heavy load that signifies maximum power consumption and minimum system lifetime. Nevertheless, in order to prolong video surveillance networks’ system lifetime, we can utilize content information captured by the video nodes to regulate the video encoding process in the whole system.
Note that, the calculation of SI and TI of videos is performed on raw video data, i.e., we do not need to encode the video to get content information. Therefore, based on the value of content complexity, i.e., SI and TI values, we can decide the following:

1. Whether or not the video surveillance network needs to capture an event. This could be done by using a simple thresholding mechanism, such that if there is an increase in the SI or TI values, the video surveillance network can start or stop encoding the event.

2. The nodes that should encode or transmit video. Based on the captured content complexity at each node, we can build a heuristic or an optimization approach to decide the number of nodes needed to capture an event and assign the specific nodes to encode and transmit the encoded video. As our experiments suggest, some nodes can have a lower SI or TI than the other nodes during an event. In addition, the number of nodes with high SI or TI increases as the event’s content complexity increases, i.e., party_act1 event has more spatial and temporal information than the office_act4 event. By considering the event’s content complexity, nodes’ SI and TI values, nodes’ remaining energy resources, we can formulate a method to find the most optimum video capturing process per each event.

### 6.3.3 Content Based Frame Rate Adjustment

In Chapter 2, the effect of the frame rate on video surveillance networks’ power consumption was investigated. On the other hand, the effect of content complexity of videos with the same frame rate was investigated in Chapter 3, 4 and 5. We can extend the
work in this thesis by incorporating both the video frame rate and content complexity into our model. For example, instead of assigning each video node to encode video with the same frame rate, we can assign the video frame rate at each node based on the SI or TI values of the node. It is worth noting that assigning low video frame rate results to video jerkiness. Nevertheless, the energy saved from using a lower video frame rate is significant. Therefore, a study to find the correlation between video perception quality, frame rate and content complexity need to be performed.

### 6.3.4 Cloud Based Video Surveillance Network

We have shown that the video encoding process at a video node contributes to a significant portion of the overall power consumption. Furthermore, it was also shown that it is more beneficial in terms of power consumption to use a simple encoding approach, i.e., intra coding only, for a short distance communication. However, the use of intra only coding entails high bandwidth or storage. Therefore, there is a need to transcode the intra-only coded video into another format that is more efficient for storage and later distribution. This can be done by establishing a cloud-based video surveillance network, where the sink is a server that is placed at the edge of a cloud platform. Assuming that there is no issue with bandwidth availability with the coming UWB standard for a video surveillance sensor network, we can shift the burden of the video encoding process from the node to the sink or the cloud. The cloud will transcode the video from the video surveillance sensor network nodes from intra-coded into another video format for efficient distribution. The cloud can also perform further processing such as video analysis and feature extraction to obtain a much more detailed information from the captured event.
References


[144] K. Lane, “What is Robust Regression and How Do You Do It?,” in The annual meeting of the Southwest Educational Research Association, Austin, TX, 2002.


