Vision-based Resource Constrained Event Detection for Medical Smart Homes

VISION-BASED RESOURCE CONSTRAINED EVENT DETECTION FOR MEDICAL SMART HOMES

BY

MAHDY NABAEE, B.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2010) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Vision-based Resource Constrained Event Detection for Medical Smart Homes
AUTHOR:	Mahdy Nabaee B.Sc., (Electrical Engineering) University of Tehran, Tehran, Iran
SUPERVISOR:	Prof. Shahram Shirani
CO-SUPERVISOR:	Prof. M. Jamal Deen
NUMBER OF PAGES:	xiii, 95

To my beloved parents

Abstract

As the number of elderly persons as well as their fraction of the total population continues to rise, especially in the developed countries, providing an appropriate living environment for them using smart home technology is rapidly gaining attention. Two important tasks of a smart home technology are monitoring the daily activities and the vital signs of the elderly to improve their quality of life and to monitor existing or the onset of health abnormalities. In this thesis, we focus on the monitoring of taking medicine by the elderly person using vision sensors (low-cost cameras). This task is important since it helps both the person and the doctor in the treatment of illnesses of elderly persons. The allocated resources of communication bandwidth between the sensor nodes and the computational power, used for this task, affect the implementation cost. Therefore, it is desired to develop an effective scheme which efficiently allocates bandwidth and computational resources to achieve a high reliability (detection performance) at low cost.

In this thesis, we have proposed two different approaches to solve this detection and monitoring problem. As the input data are video frames, captured by cameras from the same scene, the frames have inter-view redundancy. Taking advantage of this inter-view redundancy, we proposed a video coding classification scheme based on separate encoding and joint decoding, and have obtained significant compression improvement compared to existing techniques. In the second approach, we studied different parts of the detection and monitoring system to find an efficient design for distribution of different event detection parts between the nodes and the central processing unit so that the allocated resources are reduced. In this scheme, the useful information of the frames are extracted in the form of their main features such that decision making based on these features is the same as decision making

based on the raw frames. As a result, we could propose a new scheme which requires significantly less bandwidth and computational resources while achieving the same detection performance.

Acknowledgements

I would like to express my thanks to my supervisor, Prof. Shahram Shirani for his guidance, encouragement and unwavering support. I am thankful to him for affording me the excellent conditions to do graduate studies at research at McMaster and also for the opportunity for academic advancement. I would also like to express my gratitude to my co-supervisor, Prof. M. Jamal Deen for his support, encouragement and honesty. It is impossible to put in the words the level of his commitment to me and my work. He helped me become a better scholar both in my studies and research work. I must thank him again for his valuable comments and guidance which has a great effect in my life.

I must thank Dr. Thomas Doyle and Dr. Sorina Dumitrescu for their productive comments which helped me improve this thesis. I wish to thank Dr. Nicola Nicolici for his great help and guidance in developing the tools that I need for my research during my graduate studies. I am grateful with Prof. Jim Reilly for helping me in the teaching assistantship duties. I would also like to thank Drs. Ali Olfat, Gholam-Ali Hossein-Zadeh, Alireza Nasiri Avanaki, and Saeid Sanei for teaching me a lot of valuable courses in my life and also my career.

I also thank my colleagues, Mohammadreza Dadkhah, Ramin Mafi, Reza Pournaghi, Ali Gorji, Amin Behnad, Mohammad Mehdi Korjani and Peyman Setoudeh for learning me a lot of valuable things which really helped me in my study and work. I must acknowledge my old friends in the University of Tehran, Pedram Ataee, Ali Pooyafard, and Abbas Rahimi who helped me have a productive and fun time during my undergraduate studies.

Last but not the least, my greatest attitude goes to my mother, my father and my brother, Mojtaba, who supported and helped me during my education.

Notation and abbreviations

$a(n)$	Occurrence of event at time n
$\hat{a}_m(n)$	m'th sensor marginal decision about $a(n)$
$F_{n,m}$	Captured frame at time n from camera m
$ST_m(n)$	Transmitted stream from m'th sensor to the central node
$\mathbb{P}^{det}$	Probability of detection
$\mathbb{P}^{fa}$	Probability of false alarm
$\mathbf{C}_P(\cdot)$	Cost corresponding to the detection performance
$\mathbf{C}_{BW}(\cdot)$	Cost corresponding to the required bandwidth
$\mathbf{C}_{CO}(\cdot)$	Cost corresponding to the computational complexity of sensors
$\mathbf{f}_{DE}(\cdot,\cdot)$	Disparity estimation operator
$R_{n,m}$	Decoded frame of time n and view m
$CB_n^m(u,v)$	Classification bit of (u,v)'th block $F_{n,m}$
$\tilde{F}_{n,m}$	Approximate description of $F_{n,m}$
$p\{\cdot\}$	Probability density function
$P\{\cdot\}$	Probability
$HD(n)$	Occurrence of Hand-Drug approaching at time n
$HM(n)$	Occurrence of Hand-Mouth approaching at time n
$\mathbf{F}_{n,m}(i,j)$	RGB vector of (i,j)'th pixel value of n'th frame of m'th view

Contents

Abstract iv

Acknowledgements vi

Notation and abbreviations vii

1 Introduction 1

1.1 Problem Description and Formulation 5

1.2 Thesis Objectives and Organization 8

2 Literature Review 11

2.1 Medical Smart Home 11

2.2 Multi-Terminal Video Coding 15

2.3 Automated Event Detection 17

3 Multi-Terminal Video Coding 22

3.1 Proposed Classification based Multi-Terminal Video Coding 25

3.1.1 Proposed I frame coding scheme 26

3.1.2 Classification of I Frame Blocks 28

3.1.3 Approximate Description Generation 30

3.1.4 Proposed P/B frame Coding Scheme 31
3.1.5 SW Coding of Frame Blocks 34
3.1.6 Disparity Estimation 35
3.2 Experimental Results 37
3.2.1 Rate-Distortion Evaluation of the Proposed Method 38
3.2.2 Analysis of Encoder Computational Complexity 42

4 Automated Vision-based Event Detection **46**
4.1 Optimal Vision based Event Detection 46
4.1.1 Marginal Decision Making 50
4.1.2 Multi-Sensor Decision Making 51
4.2 Human Body and Drug Bottle Tracking 53
4.2.1 Human Body Tracking 54
4.2.2 Drug Bottle Recognition and Tracking 58
4.3 Proposed Constrained based Event Detection System 60
4.3.1 Human Body Tracking 61
4.3.2 SIFT Drug Bottle Tracking 63
4.3.3 Color-based Drug Bottle Tracking 67
4.4 Experimental Results 72
4.4.1 Human Body Tracking 73
4.4.2 SIFT Drug Bottle Tracking 75
4.4.3 Color based Drug Bottle Tracking 76
4.4.4 Overall Analysis of Experimental Results 78
4.4.5 Color Pattern for Drug Bottle Tracking 81

5 Conclusions and Future Work **83**

5.1 Recommendations for Future Work 85

A Mathematical Relation between the number of SW bits and classification threshold, $\Delta_{DC_{TH}}$ **87**

List of Figures

1.1 Medical smart home 2

1.2 Multi-Camera Event Detection System 4

1.3 Multi-view event detection scenario 6

2.1 PlaceLab research facility 12

2.2 Smart in-home monitoring system 13

2.3 Automated emergency and falls detection system 14

2.4 Automated analysis of functional balance 15

2.5 Temporal/inter-view prediction structure for MVC 16

2.6 *Protector* system 18

2.7 Overview of semi-HMM event detection system 19

2.8 An example of the symbolic streams 20

2.9 Traffic event detection using HMM and MPEG data 21

3.1 Proposed Classification based Multi-Terminal Video Coding scheme . 24

3.2 Proposed I frame coding using feedback channel 27

3.3 Disparity compensation and block classification by using approximate description 29

3.4 Calculating approximate description for I frames 30

3.5 Rate-distortion curves for *tunnel* 38

3.6 Rate-distortion curves for videos with several views 45
4.1 Sub-events of Taking Medicine 47
4.2 Marginal Decision Making 50
4.3 Multi-Sensor Decision Making 52
4.4 Human body recognition. 54
4.5 A sample for human body tracking 55
4.6 Foreground Extraction for Human Body Tracking 56
4.7 Articulated human body model 57
4.8 SIFT Object Recognition 58
4.9 SIFT drug bottle recognition and tracking. 59
4.10 Overview of Event Detection System 61
4.11 Proposed resource constrained event detection design. 62
4.12 Proposed Constrained based Human Body Tracking Structure 63
4.13 Proposed Constrained based Drug Bottle Tracking Structure using SIFT 64
4.14 Overview of SIFT Feature Extraction 66
4.15 Shrinking SIFT descriptors' vector. 67
4.16 Proposed resource constrained SIFT drug bottle tracking. 68
4.17 Used Color Pattern for Drug Bottle Recognition 68
4.18 Color-based Drug Bottle Recognition 69
4.19 An Example of Color-based Drug Bottle Tracking 70
4.20 A three-view sample frame of test video sequence 73
4.21 An example of human body part recognition 74

List of Tables

3.1 Coding parameters for classification based coding of *tunnel* 37

3.2 Coding parameters for separate H.264/AVC coding of *tunnel* 38

3.3 Summary of experimental results for *tunnel* sequence 39

3.4 Separate H.264 coding parameters for multi-view video sequences . . 41

3.5 Joint coding parameters for *Akko&Kayo* and *Rena* sequences 41

3.6 Summary of experimental results and parameters for multi-view videos 42

3.7 Average execution times of encoder 44

4.1 Experimental Results on Human Body Tracking 74

4.2 Experimental Results on SIFT Drug Bottle Tracking 76

4.3 Experimental Results on Color-based Drug Bottle Tracking 77

4.4 Analytical Experimental Results in color Mode 80

4.5 Analytical Experimental Results in Gray Mode 80

Chapter 1

Introduction

Industrialized countries are experiencing a transition from younger populations to a much larger proportion of older people. According to the UN[1], "Population ageing is unprecedented, without parallel in the history of humanity. . . . By 2050, the number of older persons in the world will exceed the number of young for the first time in history." The number of elderly people in the world who are 60 years or older will increase from 10 percent currently to around 20 percent in 2050 [1]. In some countries, 16 to 18 percent of the population has already turned into 65 or older [2]. For instance, by the year 2025, Japan is expected to have twice as many old people as children [2]. With a larger proportion of the population over age 65, healthcare systems should be enhanced to better deal with the disorders and diseases of elderly people. Medical professionals should obtain special types of training for elderly people. Furthermore, traditional methods of caring elderly people need to be improved, as the portion of elderly people exceeds the portion of young skilled people.

Aging in the world and especially industrialized countries has caused different health issues. Each year, around half a million of elderly people in the world are hospitalized as a result of injuries caused from a fall [3]. Mental disorders caused by Alzheimer influences the daily activities of elderly people and the affects the treatment of their diseases. Dehydration is also a common problem among elderly people, because their thirst center may not function as well as that in younger people [4]. Finally, elderly people who live alone fail to follow their prescribed schedule for taking

[1]United Nations organization

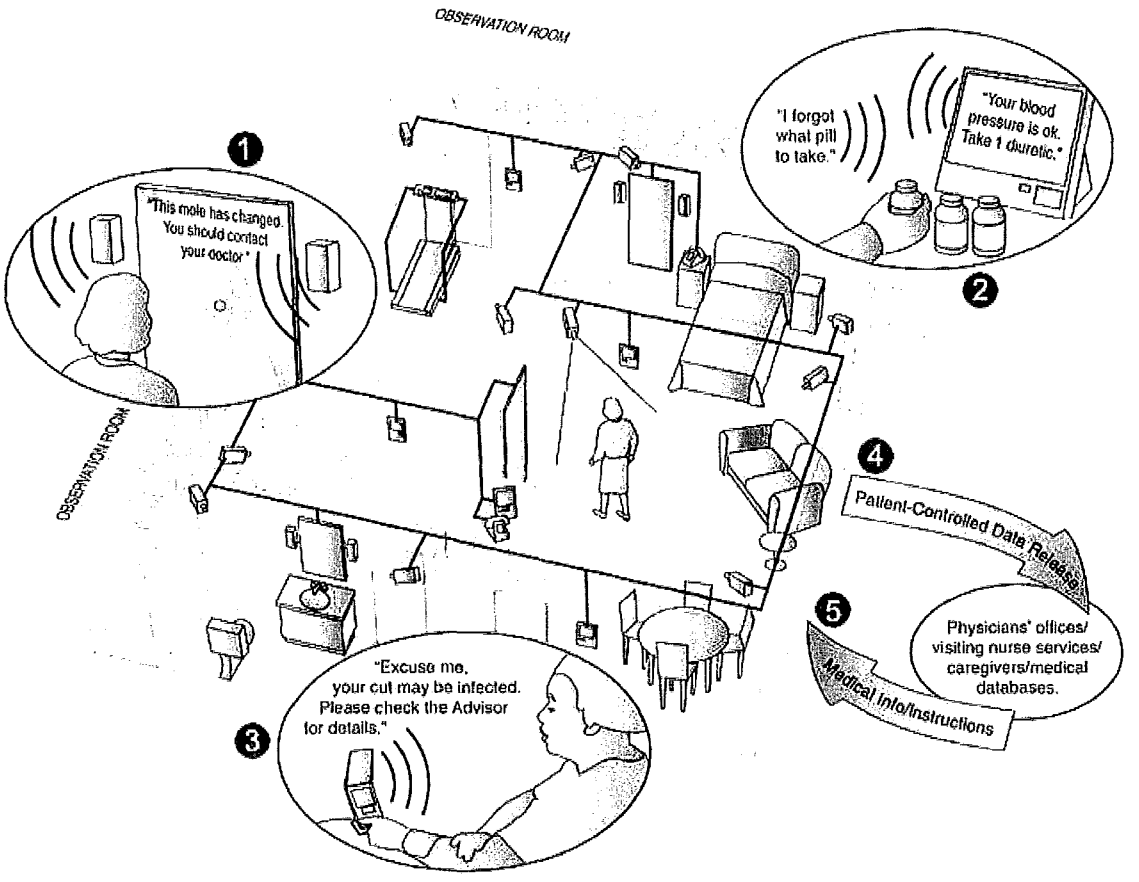


Figure 1.1: Medical smart home [6].

their medicine(s) both unintentionally and intentionally [5].

A key proposed solution for independent living of elderly people is the smart home technology. The smart homes with medical applications, called *medical smart homes*, help elderly people live in technology-assisted environments instead of living in human-assisted environments. In addition to the cost reduction caused by not employing humans to care for elderly people, the medical smart home are useful for the diagnosis and monitoring of the diseases, treatment of them, and assisting the occupants in their daily activities.

In Fig. 1.1, a sample smart home is shown in which different tasks are illustrated.

For instance, vision sensors are used in tasks number 1,4 and 5 to monitor the treatment procedure and provide the patient with some recommendations made by the system or a second party advisor (*e.g.* a doctor). In tasks 2 and 3, different types of sensors are used to monitor different body parameters and remind/suggest the patient his/her tasks (*e.g.* taking medicine). All these sensors build up a network of sensor nodes in which a lot of data processing and decision makings is involved.

Much effort has been made to build sensors which can be placed on the body, clothing or living space of the elderly to monitor their vital signs or daily activities. It may be in the form a simple ECG[2] signal sensor or a very complex video sensing and processing unit. These sensors are used to measure information or obtain a feedback from the smart home occupants and perform a special process which helps the doctor and the occupants.

Easy installation and maintenance of vision sensors (*i.e.* cameras) in the living environment of elderly people makes them a favorite option in medical smart homes. However, the system which processes the captured raw frames from different camera frames and extracts useful information for healthcare monitoring is challenging both in terms of its reliability and its allocated resources. Therefore, having an automated system which is capable of monitoring elderly people and reliably and efficiently detecting events of interest, is an important goal of most smart medical homes.

The previous works on medical smart home are mostly concentrated on development of different tasks used for diagnosis of diseases by analyzing the activities of the occupants [7–10]. This activity monitoring is also useful in detection of dangerous or important tasks in the smart home. It should be noted that their key challenge was on improving the reliability of such systems.

In addition to the reliability of the system, there are other important parameters which need to be taken care in design of a special task for smart homes. One of the important factors is the mass-production cost associated with putting a new task in the smart home. In other words, it is essential to design the smart home systems in a cost and performance efficient way by considering our priorities. The mass production of the smart home systems for elderly people require a low implementation cost. The allocated resources used by the system is one of the key parameters which define the

[2]Electrocardiography

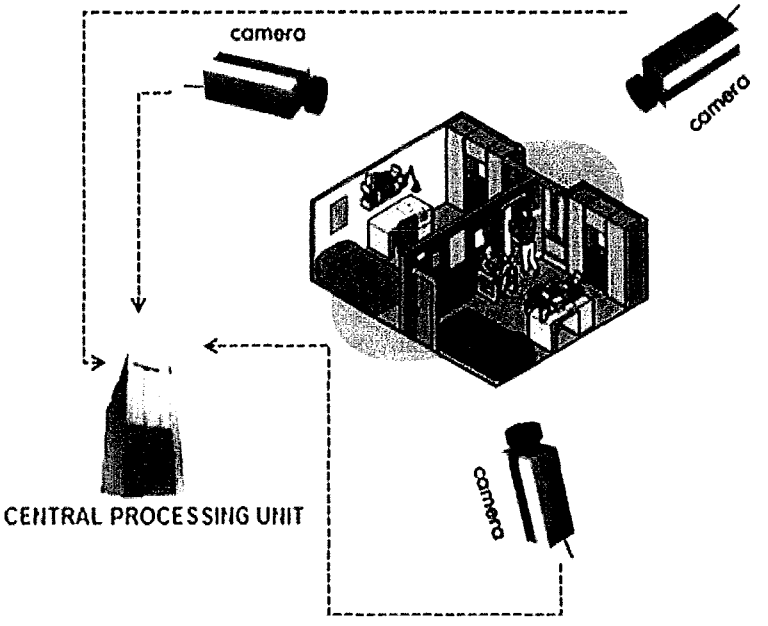


Figure 1.2: Multi-camera sensory system for event detection.

total implementation cost of the smart home.

Most of the previous works in medical smart homes [7–10] are concentrated on developing algorithms and softwares which need a high computational power. Such systems could be implemented with the aid of a strong computational and storage unit for the real-time applications. Moreover, requiring a high communication bandwidth for transmitting raw data from the sensor nodes to the central processing unit is the other drawback of such systems. Therefore, it is very important to minimize the allocated resources during design and development of automated systems for healthcare monitoring. At the same time, it is not desired to sacrifice the reliability of such medical systems in order to get a saving in the cost of allocated resources.

In this thesis, we study the processes which an automated multi-camera system should perform in order to make a reliable decision about the detection of an event of interest. Specifically, our work is concentrated on reducing the allocated resources used for detection of the event of *taking medicine*. The input of this system is the set of frames captured by a series of cameras located in the living space of the elderly

person. These camera sensor nodes (shown in Fig. 1.2) are capturing and processing the real time video sequences from different angles and positions in a room. As shown in Fig. 1.2, a computationally powerful central processing unit is used as the central node to perform computationally heavy processes. Using the captured video sequences, it is desired to find an optimal binary decision about the occurrence of the event (*i.e.* taking medicine) at the central node.

In this thesis, we examine two different approaches for improvements in the required resources for the event detection system. First, the multiple camera system is considered as a multi-view video compression system in which inter-view similarity of the view frames is used to decrease the transmission bandwidth. Second, different parts of event detection are distributed between the sensor and central nodes in a way such that the computational load of sensor nodes as well as the required sensor-central node bandwidth is significantly decreased.

1.1 Problem Description and Formulation

In this section, a mathematical formulation for the problem of event detection is provided. As shown in Fig. 1.3, it is assumed that a set of M cameras are located in different positions and angles of a room, monitoring the same scene. The set of captured video frames are represented by $[F_{n,m}]$ where n and m represent the corresponding frame and view (camera) numbers. Having sensed $\underline{F}_m(n) = \{F_{n',m},\ n' = 1, ..., n\}$ in the m'th sensor node, the sensor will process $\underline{F}_m(n)$ and transmit a set of information, called $ST_m(n)$, to the central node. The central node then processes the received information, $\mathbf{ST}(n) = \{ST_m(n'),\ n' = 1, ..., n\ , m = 1, ..., M\}$, and makes a binary decision $\hat{a}(n)$ about the occurrence of the event at time n ($\hat{a}(n) = 1$ when the center node decides the event has been occurred at time n). The parameters used for the evaluation of the whole system include:

1. The event detection performance of the system which is measured in terms of the probabilities of detection and false alarm.

2. The communication bandwidth, required between the sensor and central nodes.

3. The average computational power, required in each sensor node.

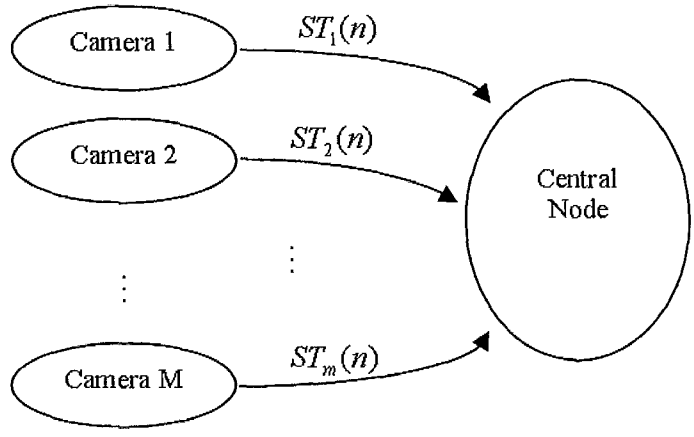


Figure 1.3: Multi-view event detection scenario.

In our scenario, no communication exists between the cameras; however a very low rate feedback channel is available from the central node to each sensor node.

The primary goal of the event detection is to have a high performance which is measured in terms of the probabilities of detection and false alarm, $\mathbb{P}^{det}$ and $\mathbb{P}^{fa}$, of the system:

$$\mathbb{P}^{det} = P\{\hat{a}(n) = 1 | a(n) = 1\} \tag{1.1}$$

$$\mathbb{P}^{fa} = P\{\hat{a}(n) = 1 | a(n) = 0\} \tag{1.2}$$

Representing the probabilities of detection and false alarm, corresponding to $\mathbf{ST}(n)$, by $\mathbb{P}^{det}(\mathbf{ST}(n))$ and $\mathbb{P}^{fa}(\mathbf{ST}(n))$, respectively, the cost corresponding to the detection performance (reliability) of the system can be defined according to:

$$\mathbf{C}_P(\mathbf{ST}(n)) = -\lambda_{det} \cdot \mathbb{P}^{det}(\mathbf{ST}(n)) + \lambda_{fa} \cdot \mathbb{P}^{fa}(\mathbf{ST}(n)) \tag{1.3}$$

where λ_{det} and λ_{fa} are the positive importance factors corresponding to the probabilities of detection and false alarm. These import factors determine which of the detection or false alarm probabilities is more important in our evaluation.

As the second goal, it is desired to decrease the rate required for transmitting the extracted information in the sensor nodes, $ST_m(n)$, so much that it does not affect the detection performance cost of the system, $\mathbf{C}_P(\mathbf{ST}(n))$, significantly. In other

words, having a maximum allowable performance cost, $\mathbf{C}_P^{Goal}$, we need to minimize the cost, corresponding to bandwidth allocation of Eq. 1.4:

$$\mathbf{C}_{BW}(\mathbf{ST}(n)) = \sum_{m=1}^{M} \bar{\mathbf{R}}_m \tag{1.4}$$

In Eq. 1.4, $\bar{\mathbf{R}}_m$ is the average number of bits required for transmission of $ST_m(n)$. The optimum choice of $ST_m(n)$ should be selected in a way such that the bandwidth cost, $\mathbf{C}_{BW}(\mathbf{ST}(n))$, is minimized conditional upon having $\mathbf{C}_P(\mathbf{ST}(n))$ smaller than the desired value, $\mathbf{C}_P^{Goal}$:

$$\begin{aligned} &\min_{\mathbf{ST}(n)} \mathbf{C}_{BW}(\mathbf{ST}(n)) \\ &s.t.\ \mathbf{C}_P(\mathbf{ST}(n)) \leq \mathbf{C}_P^{Goal} \end{aligned} \tag{1.5}$$

The sensor node computational power is another parameter which affects the implementation cost of the whole system. As a result, it is desired to decrease the computational complexity of the process which is required to be performed at sensor nodes. To formulate the computational complexity of the adopted information extraction (calculating $ST_m(n)$ from $\underline{F}_m(n)$) method used in m'th sensor node, we represent it by CO_m which is obtained by time averaging. This computational complexity is usually measured in terms of the number of mathematical operations (*i.e.* multiplications and additions) that the sensor information extraction algorithm requires. However, in some of the cases (*e.g.* H.264 video encoding which includes complex motion estimation and compensation blocks), it is not easy to calculate the number of required operations analytically. Therefore, we are using the execution times of different algorithms on a Central Processing Unit (CPU)[3] to evaluate their computational complexities. The mentioned time averaging is done by measuring the execution time of each algorithm for all the input frames and then calculating their average value to be used as the corresponding computational cost.

Finally, the resource-constrained event detection problem can be formulated in the form of minimization criteria of Eq. 1.6:

$$\begin{aligned} &\min_{\mathbf{ST}(n)} \mathbf{C}_{total}(\mathbf{ST}(n)) \\ &s.t.\ \mathbf{C}_P(\mathbf{ST}(n)) \leq \mathbf{C}_P^{Goal} \end{aligned} \tag{1.6}$$

[3]A 2.4 GHz Pentium 4 Xeon CPU with 1 GBytes of memory

where $\mathbf{C}_{total}(\mathbf{ST}(n))$ is defined according to Eq. 1.7.

$$\mathbf{C}_{total}(\mathbf{ST}(n)) = \lambda_{BW} \cdot \mathbf{C}_{BW}(\mathbf{ST}(n)) + \lambda_{CO} \cdot \mathbf{C}_{CO}(\mathbf{ST}(n)) \tag{1.7}$$

In Eq. 1.7, $\mathbf{C}_{CO}(\mathbf{ST}(n))$ is the total cost corresponding to the required computational powers of all sensor nodes, calculated according to:

$$\mathbf{C}_{CO}(\mathbf{ST}(n)) = \sum_{m=1}^{M} CO_m \tag{1.8}$$

The positive import factors, λ_{BW} and λ_{CO}, determines importance of bandwidth and computational resources relative to each other. These factors usually depend on the real constraints in mass production of the system.

Although we can achieve a constrained optimization problem for the choice of the extracted information, $\mathbf{ST}(n)$, it is not easy to find a closed form expression for the costs corresponding to the sensor-center communication rate, $\mathbf{C}_{BW}(\mathbf{ST}(n))$, and the computational power of the sensor nodes, $\mathbf{C}_{CO}(\mathbf{ST}(n))$. In the other words, the optimum choice of $\mathbf{ST}(n)$ can not be easily obtained by solving a closed form optimization. Different choices are investigated to find the best choice. It should be noted that the choice of $\mathbf{ST}(n)$ may be different in cases where different events of interests are supposed to be detected by the system. For example, the appropriate choice of $\mathbf{ST}(n)$ when taking medicine is interested to be detected is different from the case where happening of a fall is detected.

1.2 Thesis Objectives and Organization

As the input data captured by the sensor nodes is the video sequence, efficient video encoding and decoding will decrease the average number of bits required for transmission from the sender nodes to the central node. This has been addressed well in the literature under the topic of video coding [11]. Furthermore, as the video sequence of different sensors (cameras) are captured from the same scene (*e.g.* room), there is a high correlation among different view video sequences. As a result, taking advantage of inter-view redundancy, the sensor nodes can encode more efficiently

through compression. The study of separate encoding and joint decoding of so called *multi-view video* is done under the topic of *Multi-Terminal Video Coding* (MTVC) in the literature [12, 13]. Specifically, we adopt multi-terminal video coding to improve the rate-distortion performance of the system. Therefore, the event detection performance will not be affected, conditional upon keeping the average quality (measured in terms of PSNR[4]) of the received frames in a fixed range. However, the rate required for transmission of the frames is decreased by the proposed multi-terminal video coding schemes.

Distributing the parts of the event detection system between the sensor and central nodes in an efficient way can enhance the required bandwidth and computational complexity of the whole system. A multi-sensor event detection system is composed of different blocks which are in charge of: recognition and tracking of the objects of interest, marginal decision making, and fusion of multiple sensor data. Considering the bandwidth required for the transmission of the output information of each block as well as the computational complexity of calculating them is the key to designing an improved resource-constrained event detection system. We have studied different blocks of event detection in this thesis and proposed a new scheme which achieves significant improvements in the allocated resources.

In chapter 2 of this thesis, a comprehensive review of the previous and current works on medical smart homes is presented. Furthermore, a literature review on multi-terminal video coding and vision-based event detection systems is provided in this chapter. As a theoretical framework which has a significant effect on the communication load of sensor-center node transmission, we study multi-terminal video coding in a general case. The discussion of our study and the proposed feedback-assisted multi-terminal video coding scheme as well as the performance evaluations are presented in chapter 3. In this chapter, the theoretical background required for multi-terminal video coding including distributed source coding and disparity estimation and compensation are described. In chapter 4, different parts of the event detection are studied and a new scheme is adopted to satisfy the sensor-central node bandwidth and sensor node computational complexity. Specifically, it is desired to extract high entropy frame features which does not require a high computational power

[4]Peak Signal to Noise Ratio: $10 \log_{10} \frac{255^2}{MSE}$

at the sensor nodes. The proposed scheme is compared with different event detection scenarios in terms of the required sensor-central node bandwidth and the sensor node computational complexity. Finally, the conclusions and some recommendations for the future work are presented in chapter 5.

Chapter 2

Literature Review

A review of the current research on smart homes with medical applications, called *medical smart homes*, nationally and internationally is presented in section 2.1. In section 2.2, a brief review of the previous work done on multi-terminal video coding is provided. Finally, the previous work on automated event detection and object tracking by using vision data are summarized in section 2.3.

2.1 Medical Smart Home

Recently, medical smart home has been paid attention by the researchers with different fields of interests, including rehabilitation, hardware implementation and computer engineering. In this section, a description of the previous and current work done on medical home automation, tele-care and tele-medicine systems is provided.

A group of researchers at MIT[1] are currently working on different techniques and design strategies which can be applied on three different environments: home, workspace and city [14]. Their attempt is to address different objectives in the design and installation of required digital infrastructures, developing new sensing and interfacing instruments between the environment and the its occupant(s), and design of proactive health displays for heath assessment and self-reflection of elderly people. Moreover, their *Place Lab*[2] is a residential condominium used for highly flexible and

[1]Massachusetts Institute of Technology

[2]The PlaceLab is a joint MIT and TIAX, LLC initiative in Cambridge, Massachusetts, USA

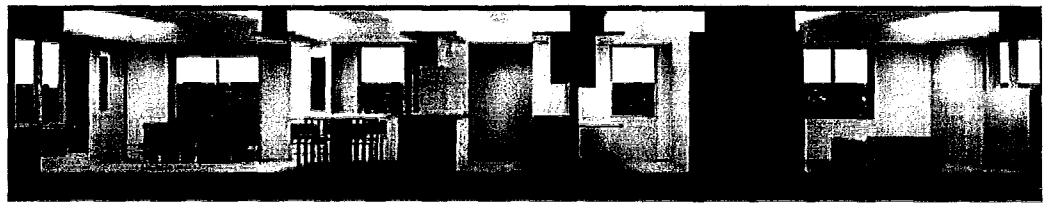


Figure 2.1: PlaceLab research facility [15].

multi-disciplinary observations to study the people and their interactions with the environments (shown in Fig. 2.1).

A variety of different projects are currently being done at the University of Virginia [16], including smart in-home monitoring, sleep monitoring, gait monitoring and eldercare robotics. The objective of smart in-home monitoring (shown in Fig. 2.2) is to use different sensor data and provide different feedback for both the patient and the doctor about the possibility of a sudden change of an activity, *e.g.;* fall. Sleep monitoring is also a well known research topic in which the sleep patterns of the patient, captured by different types of sensors, are classified in order to detect abnormalities or sleep related illnesses.

The gait signal is widely used for different applications including security and healthcare monitoring. Moreover, it is claimed that the walking pattern of the patient indicates the body health status [17]. Monitoring the gait pattern is a critical task in most of smart self-assisted environments as it lets us detect the occurrence of fall. Alwan *et al.* [18] have developed a new system which relies on the vibration of the floor in order to detect the happening of the falls which makes the patients free of wearing any kind of sensor.

In Canada, the TAFETA[3] group is working on multi-disciplinary projects related to medical smart homes [19, 20]. They have finished their work on magnetic fridge sensor development and blood pressure monitoring. Currently, their work is concentrated on pressure sensitive sensors for mattresses which is used for different purposes,

[3]Technology Assisted Friendly Environment for the Third Age

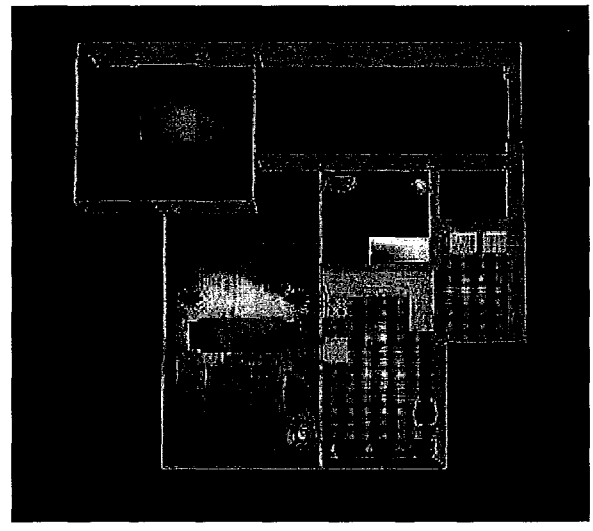


Figure 2.2: Smart in-home monitoring system [16].

including the identification of changes in bed pressure patterns in the aging. Two important goals which they pursue in this project is identifying the end of life breathing patterns and identifying elderly adults with hip fractures. Furthermore, with the aid of pressure sensitive mat technology, they are studying the correlation of the sensed data with the clinical sleep evaluations. They have also conducted a research to find out the feasibility of evaluating the health level of occupants' activities by using the smart fridge sensors which remind the occupants about the status of fridge door.

In IATSL[4], the researchers are working on different projects related to the fall [8]. As shown in Fig. 2.3, they are developing automated systems in order to detect the happening of the falls and also provide the patient with a help system to inform the clinicians or the family about it. Furthermore, an automated system is being developed to analyze the functional balance of the patients by tracking the position of different body parts of the patient, as shown in Fig. 2.4. In a parallel project, the researchers in IATSL are working to develop an automated vision based system which is capable of detection of abnormal events on the stairs [8]. Their system uses computer vision and tracking algorithms to track the feet and distinguish the normal

[4]Intelligent Assistive Technology and Systems Lab, Toronto Rehab

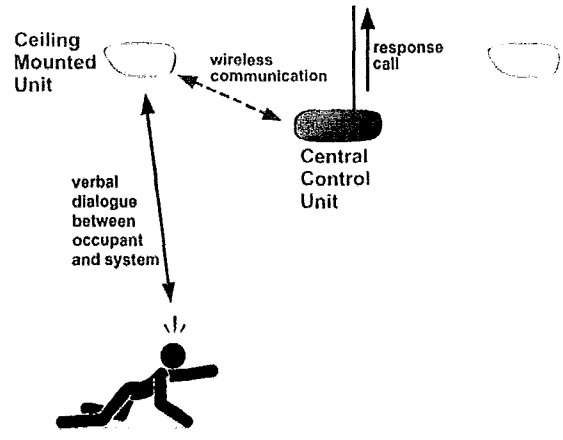


Figure 2.3: Automated emergency and falls detection system [8].

stair traversals from abnormal ones, like dangerous stepping of the stairs.

In Canada, there are other healthcare monitoring projects currently running at the SFU[5] Living Lab [10] and the University of Sherbrooke [9]. Some of the projects, currently being done in SFU Living lab [10], include design of more elder friendly acute hospital environments, studying the behavioural and physiological effects of technology on elderly people, and providing workspaces which satisfy the requirements of aging employees. The researchers in the University of Sherbrooke [9] are also working on projects helpful for the elderly people with mental problems. One of the major goals they pursue is to design and implement different interfacing and assisting systems and tools for alzheimer's people. For example, they have worked on localization of the patients for human-machine interfacing purposes.

In [21] and [7], a detailed review of previous works and future challenges on smart homes is provided. In [21], it is mentioned that different needs of the elderly people should be taken care by the decentralized system of smart homes (*i.e.;* a distributed system of smart homes instead of a centralized hospital). Moreover, the health monitoring and assisting system has to be non-invasive and acceptable by the users which needs a lot of research in the area of smart homes and the corporation of different standardization agencies. Currently, researchers are concerned about different issues

[5]Simon Fraser University, BC, Canada

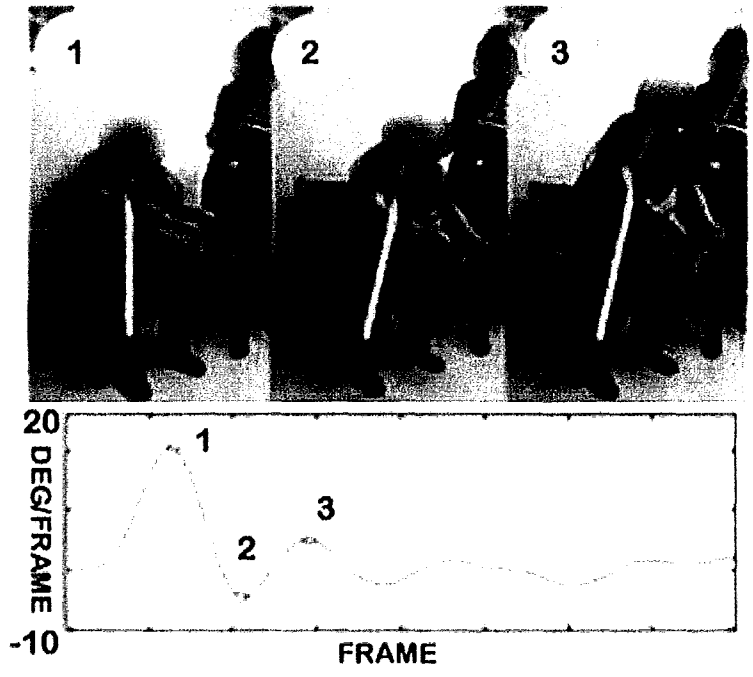


Figure 2.4: A balance impaired individual being tracked in three dimensions as she rises from a chair. At the base of the figure is the angle of the individuals torso, reconstructed using USB camera data, versus time. This angle has been smoothed and the time points corresponding to images indicated with the dots [8].

in smart homes, including the acceptability of the technologies by the user, reliability and efficiency of the sensing and data processing system, standardization of information processing and communication units, and the cost of the system [7]. There are also other legal and ethical concerns, like protection of privacy of the elderly people, which are beyond the scope of our study.

2.2 Multi-Terminal Video Coding

Multi-view video can be considered as a two-dimensional matrix of pictures, including the frames in temporal (time) and view directions. In such a matrix, the neighboring frames are correlated in both temporal and view directions. Therefore, the inter-view redundancy can let us achieve better compression efficiencies when different views

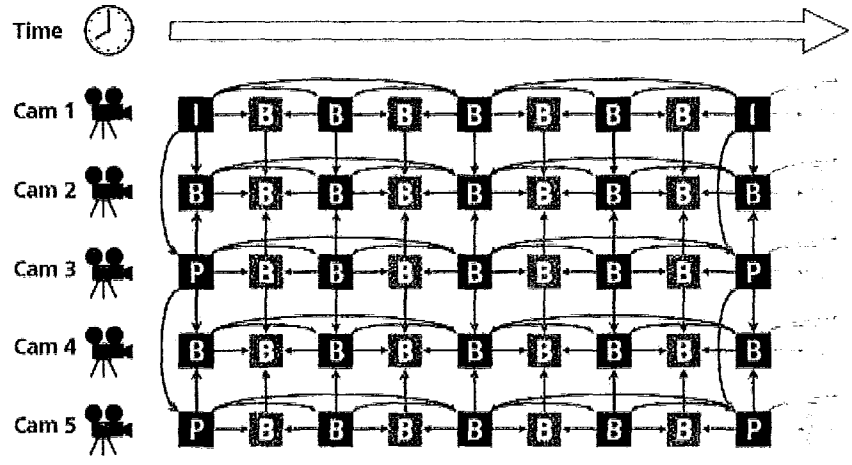


Figure 2.5: Temporal/inter-view prediction structure for MVC [23].

are jointly encoded/decoded. Joint Encoding and joint decoding of multi-view video, called *Multi-View Video Coding* (MVC), was previously studied in the literature [11]. Moreover, MVC is an extension to H.264/AVC video coding standard which enables the compression of multi-view video sequences.

In joint encoding of multi-view video, temporal prediction and residue coding techniques are applied for both temporal and view directions. In other words, non-reference frames are motion or disparity compensated and then the residue frames corresponding to the best reference frame are encoded. The set of residue frame coefficients, motion and/or disparity vectors are sent to the joint decoder. At the decoder, the motion or disparity compensation is performed by using the received motion/disparity vectors and the reference frame. The prediction structure mentioned in the MPEG working documents [22] is shown in Fig. 2.5.

Unlike the MVC extension of H.264/AVC standard in which the rate-distortion performance of multi-view video coding is improved over separate H.264/AVC encoding/decoding, there are other works which aimed to decrease the computational load of the encoders. As a practical solution for the cases where the processing power of the sensors and their encoders is limited (*e.g.;* handset devices), Girod *et al.* [24] have proposed new coding methods which decrease the computational complexity of the

encoders. They have used the idea of distributed source coding to encode consecutive temporal frames of a single view as the correlated sources with the aid of Slepian-Wolf [25] and Wyner-Ziv [26] coding. As a result, the computational load of motion estimation and compensation is moved from the encoder side to the decoder side. Although the computational complexity of the encoders is improved, the rate-distortion performance of the coding is decreased relative to conventional H.264/AVC coding. Moreover, Guo *et al.* [27] have used spatial domain wavelet transform in WZ coding of multi-view video sequences using a low-complexity encoder.

In some cases, the encoders cannot communicate with each other which makes joint encoding of multi-view video impossible. In terms of rate-distortion performance, separate encoding and joint decoding of multi-view video is a better alternative to separate H.264/AVC coding. This is an applicable version of multi-terminal source coding theory which was first proposed by Yang *et al.* [13] for a stereoscopic video sequence. To the best of our knowledge, no multi-terminal video coding scheme has been proposed and used for multi-view video with more than two views.

2.3 Automated Event Detection

Automated event detection based on vision data has been well studied in the literature for different applications [28–31]. In a vision-based event detection system, first the features of the input video frames are extracted and then the objects of interest are recognized and tracked in time. The extracted trajectories of the objects of interst (for the event of interest) are fed into a set of conditions, usually obtained from a probabilitstic model, to detect the occurrence of the event. As a combination of computer vision, object tracking and decision making, there are different variety of works done on vision-based event detection.

The detection of unusual activities for surveillance applications, especially in crowded backgrounds of people has been studied in [28] and [29]. As shown in Fig. 2.6, a so-called *protector* system was developed to avoid any collision in the road by detecting the pedestrians and bicyclers [28]. In order to detect *unusual* activities in a living or working environment, the authors in [29] have proposed a hidden Markov model based system to distinguish the usual activities from the unusual activities and

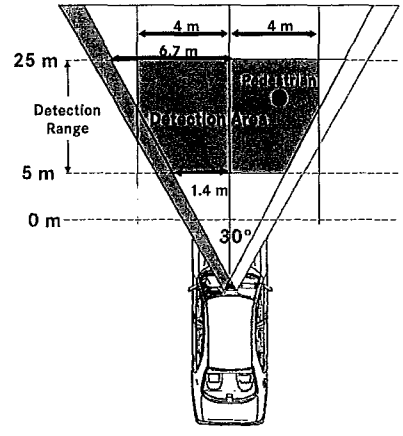


Figure 2.6: *Protector* system [28].

alarm the occurrence of any abnormal activity. In contrast to the traditional surveillance systems, they used multiple video streams simultaneously in order to extract the features and detect the event probabilistically.

Hidden Markov Model (HMM) is widely used in the detection of the occurrence of unusual events based on the vision data. All of such systems require a supervised/semi-supervised training step and/or adapting step. In [30], the authors have proposed a semi-supervised HMM in which the usual events are firstly learned by using a big training set. Then, the unusual events are learned in an unsupervised manner. In [32], the event detection is performed in two steps, as shown in Fig. 2.7. In Fig. 2.7 the first the primitive events are detected using the extracted trajectories and shapes of the objects in the scene. Then, the extracted primitive events are fed into a finite state HMM model in order to detect the occurrence of composite events.

In [33], the authors have used data mining in order to extract the visual and audio features of tennis play, to be used for event detection. More specifically, the extracted features of the visual and audio data are transformed to a set of symbolic streams (a sample is shown in Fig. 2.8) to be used in mining. After data mining, the extracted patterns are categorized into different events. Li *et al.* [31] have proposed an HMM based framework for detection of traffic event using the video streams. As

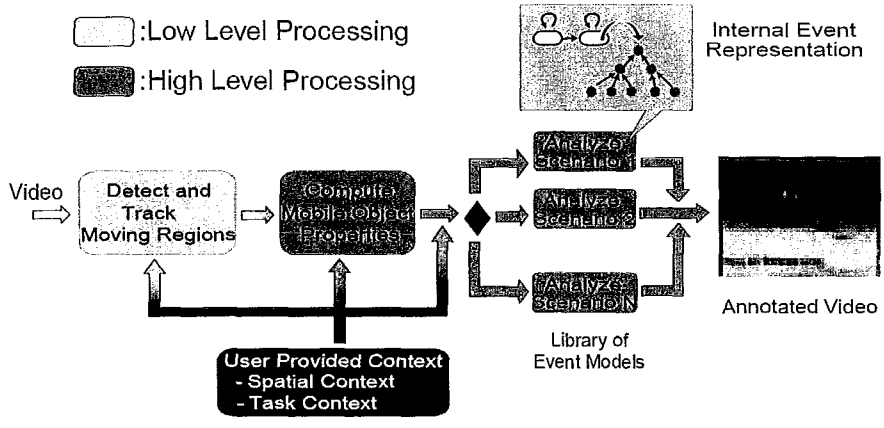


Figure 2.7: Overview of semi-HMM event detection system [32].

a computational advantage, they extract the features directly from the compressed video stream. As shown in Fig. 2.9, a feature vector is extracted from the DCT coefficients and macro-block motion vectors. Then, having different Gaussian-Mixture HMMs (GMHMM) for traffic events (offline mode), a Viterbi algorithm is used (online mode) to find the most probable event.

In [34], the planar constraint on the location of the feet of people in a multi-camera system are used to improve the tracking performance. In the multi-camera system, the planar constraint is applied by knowing the intrinsic and extrinsic parameters of cameras relative to a reference. As a result, a calibration step should be performed on the camera setup before the online stage of tracking. Mohammadi *et al.* [35] have solved this issue by proposing an object tracking system which works in un-calibrated multi-camera setups. Specifically, a single-camera object tracking is performed and a consistent object labeling is applied on the resulting extracted objects. Using a Homography transform on the extracted objects, the corresponding objects of different views are obtained. The resulting correspondence among the extracted objects of different views is then used to find the region mapping among different views by forcing the region mapping of different views. This mapping is finally used to find the best object matching between the cameras. The authors in [36] have proposed a different multi-view tracking and surveillance system which works in calibrated camera system.

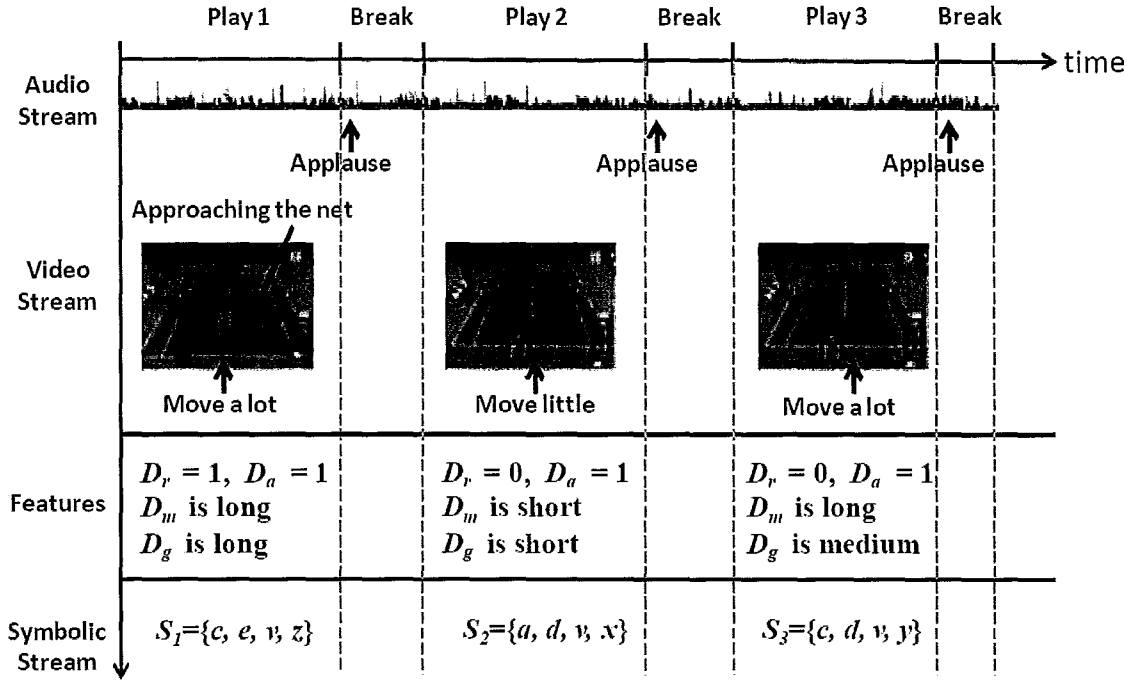


Figure 2.8: An example of the symbolic streams, introduced in [33].

In addition to the constraint on different views of the system, they have used temporal correlation to improve the tracking for both overlapping and non-overlapping situations. Specifically, temporal alignment is performed to compensate for different processing rates of cameras.

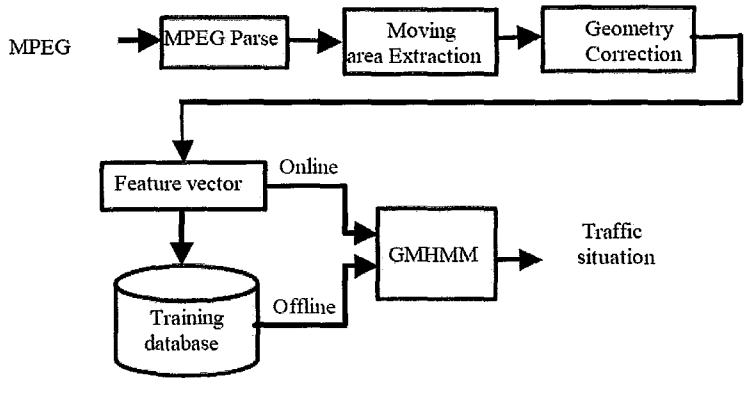


Figure 2.9: The algorithm flow chart of traffic event detection, proposed in [31].

Chapter 3

Multi-Terminal Video Coding

Separate encoding and joint decoding of correlated sources, called multi-terminal source coding [37], has attracted attention because of its application in sensor networks and multi-view video coding. Theoretical rate-distortion regions for separate encoding and joint decoding of quadratic Gaussian correlated sources were studied in [38, 39]. In [40–42], design of practical codes for distributed source coding of correlated sources was studied. In [41], the authors provide a practical framework based on trellis codes which can be applicable in different cases. Practical code designs for multi-terminal source coding based on generalized coset codes were discussed [42]. Xiong *et al.* [43,44] have developed practical codes for lossless multi-terminal coding of uniform memoryless binary sources. Yang *et al.* [45] proposed symmetric and asymmetric multi-terminal source coding schemes, assuming a known correlation model for the sources. In [45], Slepian-Wolf (SW) coding [25] using turbo codes can achieve near theoretical rate-distortion boundaries for quadratic Gaussian sources.

Multi-view video coding has recently attracted interest because of its application in 3D-TV, free viewpoint TV and smart homes. In a multi-view video system, there is an array of cameras which are capturing correlated video sequences. As a result, the captured frames are correlated both in time and view, especially in a case with tight camera angles. By using the inter-view similarity of multi-view video, and having a joint encoder, the compression efficiency is increased. Inter-view redundancy can also be used in Multi-Terminal Video Coding (MTVC) where the encoders do not communicate, but there is a joint decoder, to increase the compression efficiency over

separate H.264/AVC coding. In some cases, the communication between the encoders is not possible or in other cases not desired because of decreasing the computational load of the encoders. As a result, joint Multi-View Video Coding (MVC) [46] cannot be employed. Multi-terminal video coding is the best alternative for coding of correlated video sequences when the lack of communication between the encoders makes us unable to use joint multi-view video coding.

Girod *et al.* [24] used distributed source coding to decrease the *complexity of the encoders* for compression of single view video. They considered consecutive time frames of video as correlated sources and Wyner-Ziv (WZ) encoded ([26]) them in the encoder. Earlier, a Power-efficient, Robust, hIgh compression, Syndrome based Multimedia coding (PRISM) paradigm had been introduced by Puri *et al.* [47]. Similarly, their attempt was to move the computational burden from the encoder to the decoder by using the principles of distributed source coding. Flierl *et al.* have proposed lifting motion and disparity compensated wavelet transforms to remove intra-view and inter-view redundancy of multi-view video frames [48, 49]. In [27], Guo *et al.* have used spatial domain wavelet transform in WZ coding of multi-view video sequences using a low-complexity encoder.

Yang *et al.* [13] proposed two two-terminal video coding schemes to increase the compression efficiency of separate encoding. Assuming 3D camera parameters to be known, inter-view similarity was used for distributed coding of DCT coefficients of I frames, residual frames of P frames and motion vectors [13]. The authors firstly associated an efficient stereo matching algorithm with WZ coding to improve separate encoding efficiency of stereoscopic (two-view) video. In their second design, the idea of source splitting was used to achieve flexible rate allocation for I frame coding of different views. As a result of using inter-view redundancy in separate encoding, their compression schemes could outperform separate H.264/AVC encoding in terms of the rate-distortion.

In our work, we consider *feedback assisted* multi-terminal video coding of M different views, including N frames in each view. The very first frame of a Group of Pictures (GOP), called I frame, can only be coded by using intra-frame and inter-view redundancies as there is no temporal reference for its coding. The remaining frames of the GOP (with N frames in each GOP), called P/B frames, are coded

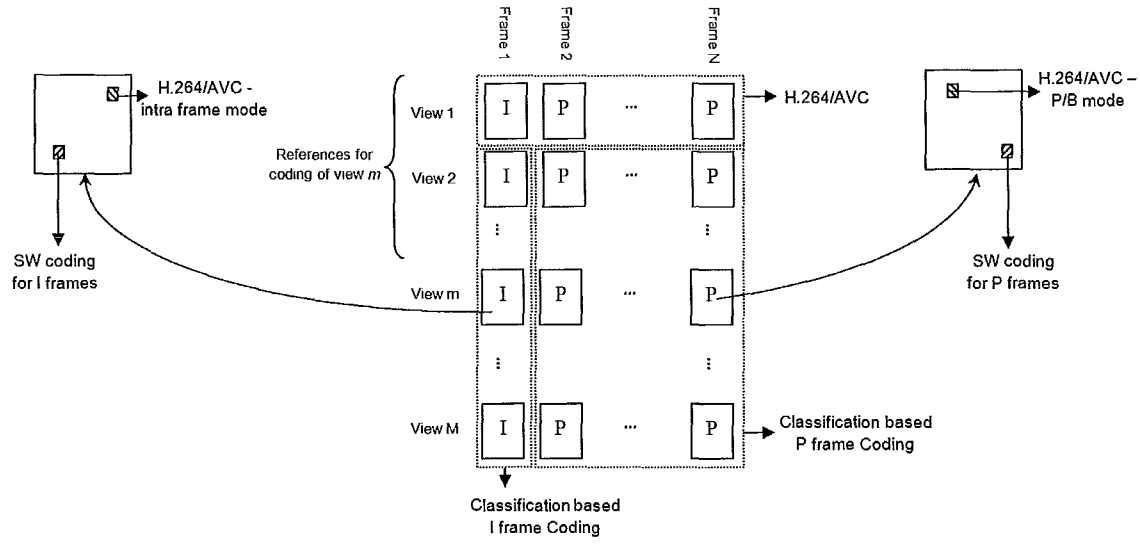


Figure 3.1: Proposed Classification based Multi-Terminal Video Coding scheme.

by using intra-frame, temporal and inter-view redundancies. Having understood the statistical characteristics of I and P/B frames, we proposed a novel feedback assisted multi-terminal video coding where encoding of each frame is adaptive. Specifically, the joint decoder classifies different blocks of frames of multi-view video and determines the best encoding/decoding option for each block. In other words, the classification will choose the best option among conventional H.264/AVC coding and SW coding, for each block of each frame. The encoder is informed about the classification results with the aid of a very low rate feedback channel from the joint decoder to the encoders. The phrase *feedback assisted* multi-terminal video coding is chosen for the method for two reasons. Firstly, it should be noted that the proposed method is different from multi-terminal video coding as the feedback channel enables the encoders to communicate with each other. Secondly, the communication rate required for the feedback channel is much lower than the total rate which makes the encoders practically separate. The proposed method is described for a general case of M views including one GOP in each view.

In section 3.1, a general overview of the proposed classification based multi-terminal video coding scheme is provided. In section 3.1.1, the proposed classification

based I frame coding scheme is described in detail. Classification of P/B frames is discussed in section 3.1.4, and this is followed by the description of the proposed block based SW coding scheme in section 3.1.5. In section 3.2, the proposed classification based method was applied on sets of two-view and multi-view video sequences to evaluate the achieved performance in comparison to Yang *et al.* method, separate H.264/AVC coding and joint coding.

3.1 Proposed Classification based Multi-Terminal Video Coding

In this section, the proposed scheme for M different views of multi-view video, each including one GOP of N frames, is described. The frames are represented by $F_{n,m}$ where m and n are the corresponding view and frame number in the view, respectively. As shown in Fig. 3.1, a reference view (view 1) is chosen and its frames, $\{F_{n,1},\ n = 1, ..., N\}$, are encoded/decoded by using a conventional H.264/AVC video coder. The remaining frames are coded by using the proposed method based on their type (*e.g.;* I or P/B frame). More specifically, the very first frame of each view, $F_{1,m}$, called I frame, is coded by using the proposed classification based I frame coding scheme, described in section 3.1.1. The remaining frames of each view, called P/B frames, are coded by using the proposed classification based P/B frames coding scheme, explained in section 3.1.4.

After the first view is transmitted to the joint decoder by using H.264/AVC, the second view frames, $\{F_{n,2},\ n = 1, ..., N\}$, are encoded/decoded by using the proposed classification based method. In this process, the decoded (reconstructed) frames of the first view, $\{R_{n,1},\ n = 1, ..., N\}$, are used as the reference frames for disparity compensation of the second view frames. Similarly, the third view frames, $\{F_{n,3},\ n = 1, ..., N\}$, are coded by using the first and second decoded view frames, $\{R_{n,m},\ n = 1, ..., N\ , m = 1, 2\}$, as a reference. Generally speaking, the m'th view frames, $\{F_{n,m},\ n = 1, ..., N\}$, are coded by using previously decoded view frames, $\{R_{n,m'},\ n = 1, ..., N,\ m' = 1, ..., m-1\}$, as reference. More specifically, the best choice among $R_{n,1}, ...R_{n,m-1}$ will be used as the reference for coding of $R_{n,m}$.

As shown previously in Fig. 3.1, the blocks of each I frame are classified into intra-frame and inter-view coded categories. The blocks in the first category are encoded by using intra-frame mode of H.264/AVC in which only intra-frame redundancy is exploited. The blocks of the second category are coded by using both intra-frame and inter-view redundancies with the proposed SW coding method, described in section 3.1.5. The classification of blocks is performed by evaluating the similarity of the block and its corresponding side information, as described in section 3.1.1. Similarly, every block in the so called P/B frames is classified into intra-view and inter-view coded categories, as described in section 3.1.4. The first category blocks are transmitted by using P/B mode of H.264/AVC where the appropriate choice of P or B is already determined by the user. The remaining blocks are encoded/decoded by applying the proposed SW coding of section 3.1.5.

3.1.1 Proposed I frame coding scheme

In a mono-view video, the very first frame of GOP, *i.e.;* the I frame, is coded only using its spatial redundancy. In the case of multi-terminal video coding, inter-view redundancy can also be used in coding of an I frame as it enhances the rate-distortion performance in joint decoding. This is why distributed coding of I frames of different views can increase the compression efficiency compared to separate H.264/AVC coding of these frames. However, in terms of rate-distortion distributed coding of I frames fails to perform as efficiently as intra coding when the side information (*i.e.,* disparity compensated frame) is not sufficiently correlated with the I frame; *e.g.,* for occluded regions. Therefore, we proposed to encode/decode each block of the I frame in its appropriate mode adaptively. In other words, each block will be coded by distributed coding (SW mode) or H.264/AVC Intra Frame coding (IF mode), based on the correlation between the block and corresponding side information.

To take advantage of inter-view similarity of frames in the decoder and to generate the side information, a disparity map between two views has to be available. Usually, this mapping is defined by disparity vectors, D_n^{pq}, between $F_{n,p}$ and $F_{n,q}$. Since there is no inter camera communication, we have to estimate disparity vectors and obtain $\hat{D}_n^{pq}$, with the aid of some references. Therefore, the estimation can be performed in the decoder where references of different views are available ($R_{n,m'}$, $m' = 1, ..., (m-1)$ are

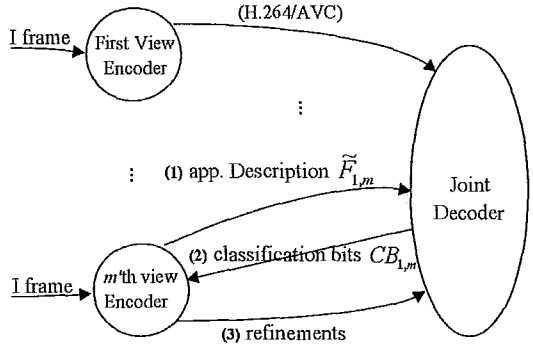


Figure 3.2: Proposed I frame coding using feedback channel.

already decoded). Furthermore, it is reasonable to use the previously decoded frames of m'th view, $R_{n',m}$, $n' = 1, ..., n-1$, as the reference for disparity estimation of current frame, $F_{n,m}$. However, this can not be applied to the I frames (i.e.; $F_{1,m}$, $m = 2, ..., M$) as there is no intra-view reference frame for their coding.

To handle this issue, the current view I frame, $F_{1,m}$, is coded/transmitted in two steps: *Approximate transmission* and *refinement coding*. In approximate transmission, an approximate description of I frame, $\tilde{F}_{1,m}$, is calculated (described in section 3.1.3) and transmitted to the decoder. This approximate description, $\tilde{F}_{1,m}$, is used to calculate a rough estimate of the disparity vectors between m'th view I frame and p'th view I frame, called $\hat{D}_1^{mp}$:

$$(\hat{D}_1^{mp}, c_1^{mp}) = \mathrm{f}_{DE}(\tilde{F}_{1,m}, \tilde{R}_{1,p}) \tag{3.1}$$

where f_{DE} represents the disparity estimation operator. In Eq. 3.1, $\hat{D}_1^{mp}$ is the estimated disparity vector which warps $\tilde{R}_{1,p}$ to $\tilde{F}_{1,m}$ with the minimum cost, as described in section 3.1.6. The associated cost for assigning $\hat{D}_1^{mp}$ as their disparity vector is also represented by c_1^{mp}, as in Eq. 3.1. $\tilde{R}_{1,p}$ is the approximate description of $R_{1,p}$ which is obtained by applying the method described in section 3.1.3.

To have the best choice for disparity compensation and side information generation, we calculate the associated cost for each pair of $(\tilde{F}_{1,m}, \tilde{R}_{1,p}), p = 1, ..., (m-1)$. The I frame with the minimum cost among all the reference I frames will be selected

according to Eq. 3.2:

$$\hat{t}_1^m = \arg \min_{p=1,\ldots,m-1} c_1^{mp} \tag{3.2}$$

where $\hat{t}_1^m$ is the index of view used for disparity compensation of $\tilde{F}_{1,m}$. The corresponding reference frame, $R_{1,p}$, $p = \hat{t}_1^m$, is then warped according to the selected disparity vector, $\hat{D}_1^m = \hat{D}_1^{m,\hat{t}_1^m}$, and the disparity compensated frame, $R_{1,m}^{DC}$, is obtained for side information generation:

$$R_{1,p} \xrightarrow{\hat{D}_1^m} R_{1,m}^{DC}, \quad \begin{matrix} p = \hat{t}_1^m \\ \hat{D}_1^m = \hat{D}_1^{m,p} \end{matrix} \tag{3.3}$$

After obtaining $\hat{D}_1^m$ at the decoder, the blocks of $F_{1,m}$ will be classified into IF and SW categories, as described in section 3.1.2. As shown in Fig. 3.2, the resulting classification bits, representing the category of each block, are sent back to the m'th view encoder through a very low rate feedback communication channel. These classification bits will determine the mode in which each I frame block is encoded in the corresponding view encoder. If a block is marked to be IF-coded, it is encoded/decoded by applying I mode of H.264/AVC. Otherwise, it is encoded/decoded by using SW coding method described in section 3.1.5, having $R_{1,p}$, $p = \hat{t}_1^m$ for side information generation. Finally, the encoded sequence, including both IF and SW coded blocks, are transmitted to the decoder as the refinements for $\tilde{F}_{1,m}$.

3.1.2 Classification of I Frame Blocks

As shown in Fig. 3.3, after $\tilde{F}_{1,m}$ is received at the decoder and $\hat{D}_1^m$ is calculated, each block of I frame is classified into one of IF or SW categories. This can be done by obtaining the mean difference between the block and the corresponding generated side information. Since the original gray values of $F_{1,m}$ are not available at the decoder, the approximate descriptions of the m'th view, $\tilde{F}_{1,m}$, and the corresponding reference frame, $\tilde{R}_{1,p}$, $p = \hat{t}_1^m$, are used for classification of $F_{1,m}$ blocks. The approximate description of $R_{1,p}$, $\tilde{R}_{1,p}$, is calculated at the decoder by using the same rule that was used to generate $\tilde{F}_{1,m}$ in the encoder. Then, $\tilde{R}_{1,p}$ is warped according to $\hat{D}_1^m$ to obtain

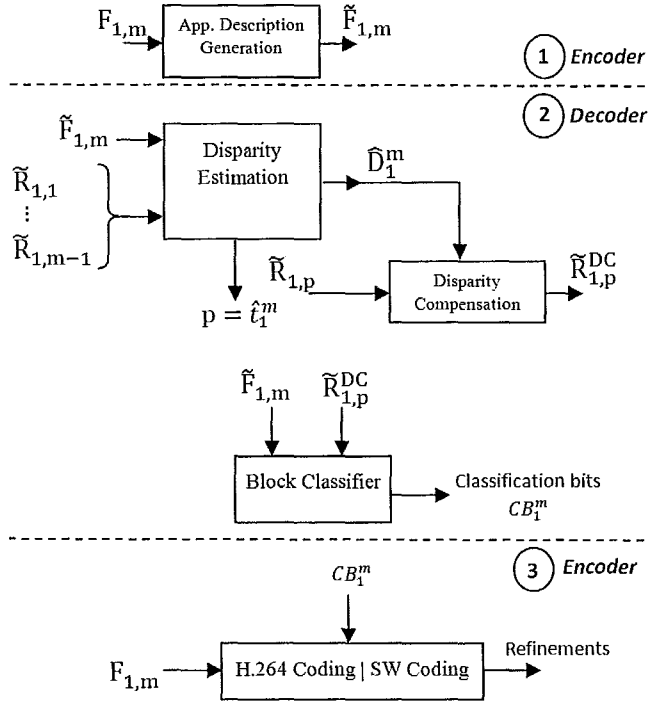


Figure 3.3: Disparity compensation and block classification by using approximate description.

$\tilde{R}^{DC}_{1,m}$:

$$\tilde{R}_{1,p} \xrightarrow{\hat{D}^m_1} \tilde{R}^{DC}_{1,m},\ p = \hat{t}^m_1 \tag{3.4}$$

The mean difference of $\tilde{R}^{DC}_{1,m}$ and $\tilde{F}_{1,m}$ is used as a measure for evaluating the similarity of $F_{1,m}$ and its side information, $R_{1,p}$, $p = \hat{t}^m_1$. More specifically, $\Delta_{DC1,m}(u,v)$ is firstly calculated for each block of the I frame ((u,v) represents the index of the block to be classified):

$$\Delta_{DC1,m}(u,v) = \frac{1}{64} \sum_{(i,j)\epsilon B_{uv}} \left[\tilde{R}^{DC}_{1,m}(i,j) - \tilde{F}_{1,m}(i,j)\right]^2 \tag{3.5}$$

where $B_{uv} = \{(i,j)|8(u-1) < i \leq 8u, 8(v-1) < j \leq 8v\}$ (Classification is performed on blocks of 8×8 for both I and P frames). Then, (u,v)'th block of $F_{1,m}$ is classified and a classification bit, $CB^m_1(u,v)$, (1 for SW mode and 0 for IF mode) is assigned

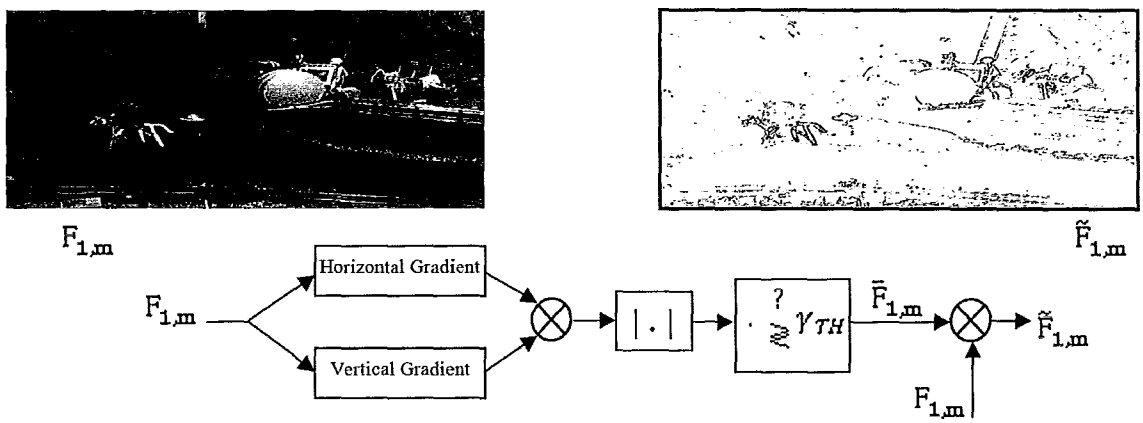


Figure 3.4: Calculating approximate description for I frames.

to it:

$$CB_1^m(u,v) = \begin{cases} 1 & ,\Delta_{DC_{1,m}}(u,v) \leq \Delta_{DC_{TH}} \\ 0 & ,\Delta_{DC_{1,m}}(u,v) > \Delta_{DC_{TH}} \end{cases} \tag{3.6}$$

In Eq. 3.6, $\Delta_{DC_{TH}}$ is a predefined threshold which depends on the number of allocated SW bits, sb, in the compression. The mathematical derivations expressing the relation between the allocated SW bits, sb, and the classification threshold, $\Delta_{DC_{TH}}$, is presented in the Appendix.

3.1.3 Approximate Description Generation

To provide the decoder with an approximate description, different choices were considered. Yang. *et al.* proposed to overquantize DCT coefficients and transmit the resulting coefficients [13] as the approximate description. Using such approximate description may provide a blurred version of the frame which is not suitable for estimating disparity vectors, $\hat{D}_1^m$. Thus, an approximate description of the image which yields the disparity vectors more accurately and needs less bits for transmission is a better choice.

In this work, the approximate description is obtained by first finding the corner edges of the I frame. As shown in Fig. 3.4, the horizontal and vertical gradients of the I frame are calculated and multiplied together. The gray value of the corner edge points

in the resulting image will be significant in terms of absolute value. After passing this image through the absolute value block and thresholding the image by a predefined value, γ_{TH}, a binary image of the corner points, $\bar{F}_{1,m}$, is obtained. This binary image is multiplied by the original I frame, $F_{1,m}$, to achieve $\tilde{F}_{1,m}$. In $\tilde{F}_{1,m}$, the pixels with non-zero value are the corner points used for describing the I frame. In order to efficiently transmit $\tilde{F}_{1,m}$, the binary map of the corner points, $\bar{F}_{1,m}$, and their gray values are encoded separately. More specifically, $\bar{F}_{1,m}$ is run-length coded and then the gray values of corner points (*i.e.;* $\{F_{1,m}(i,j),\ \bar{F}_{1,m}(i,j) = 1\}$) are entropy coded, using arithmetic coding. Using the corner edge points is a good choice to estimate the disparity vectors accurately because of their ability in defining the disparity vectors. Furthermore, the obtained binary map of corner points can be transmitted with few bits. However, using more complex approximate descriptions may result in minor improvements in the total performance.

3.1.4 Proposed P/B frame Coding Scheme

In [50] and [51], fusion of motion and disparity compensated frames for side information generation is described. This side information is used to decode the encoded sequence generated by distributed coding (SW/WZ coding). However, in cases where motion compensated frame is the better side information, conventional H.264/AVC coding outperforms distributed coding of the frame in terms of rate-distortion. On the other hand, since there is no communication between the encoders, we cannot find the better coding option at the encoder. We solve this issue by finding the better option at the decoder and transmitting the resulting classification bits to the encoder through the feedback channel. Moreover, a block classification is utilized instead of a frame classification in order to get a better rate-distortion performance.

The classification of blocks of current P/B frame, $F_{n,m}$, $n \geq 2$, is done at the decoder by using the previously decoded frame of the same view, $R_{n-1,m}$, as its reference. In cases where $R_{n-1,m}$ is not decoded yet (like B frames), the last decoded frame is used as its reference. More specifically, the pixel values of each block of $R_{n-1,m}$ are compared with motion and disparity compensated frames, $R^{MC}_{n,m}$ and $R^{DC}_{n,m}$, and the better choice is used for coding of current P/B frame. If the disparity compensated frame is a better choice, the encoder will encode the block in SW mode. Otherwise,

the block will be encoded in Intra View (IV) mode (*i.e.*, P/B mode of H.264/AVC). The encoder is informed about the category of each block with the aid of the feedback channel introduced in section 3.1.1. The classification procedure of P/B frame blocks is described below.

To classify the blocks of $F_{n,m}$, firstly, the mean square difference of $R_{n-1,m}$ (or the reference frame) with motion and disparity compensated frames, $\sigma_{MC_{n,m}}(u,v)$ and $\sigma_{DC_{n,m}}(u,v)$, are calculated as follows:

$$\sigma_{MC_{n,m}}(u,v) = \frac{1}{64} \sum_{(i,j)\epsilon B_{uv}} \left[R_{n-1,m}(i,j) - R^{MC}_{n,m}(i,j)\right]^2 \tag{3.7}$$

$$\sigma_{DC_{n,m}}(u,v) = \frac{1}{64} \sum_{(i,j)\epsilon B_{uv}} \left[R_{n-1,m}(i,j) - R^{DC}_{n,m}(i,j)\right]^2 \tag{3.8}$$

Unlike the I frames where an approximate description of the frame is needed for disparity estimation at the decoder, we can use the disparity vectors corresponding to the previously decoded frame for generating $R^{DC}_{n,m}$:

$$R^{\hat{t}^m_{n-1}}_{n-1} \xrightarrow{\hat{D}^m_{n-1}} R^{DC}_{n,m}, \qquad \begin{aligned} \hat{D}^m_{n-1} &= \hat{D}^{mp}_{n-1} \\ p &= \hat{t}^m_{n-1} \end{aligned} \tag{3.9}$$

where $\hat{t}^m_{n-1}$ is the view index which minimizes the following criteria:

$$\hat{t}^m_{n-1} = \arg\min_p c^{mp}_{n-1} \tag{3.10}$$

In Eq. 3.10, c^{mp}_{n-1} is the associated disparity estimation cost, calculated as in Eq. 3.11.

$$(\hat{D}^{mp}_{n-1}, c^{mp}_{n-1}) = \mathrm{f}_{DE}(R_{n-1,m}, R_{n-1,p}) \tag{3.11}$$

As it was mentioned, the choice of reference frame is not necessarily the $(n-1)$'th frame since sometimes the $(n-1)$'th frame is decoded after the n'th frame. Motion compensation is also performed by using the motion vectors of previously decoded frame.

As an approximation to the number of bits required to encode the motion vectors corresponding to the (u,v)'th block in frame n, $[F_{n,m}(i,j)]$, $(i,j)\epsilon B_{uv}$, we use $s^{MV}_{n-1,m}(u,v)$, the number of bits used to encode the motion vectors of (u,v)'th block in frame $n-1$ of m'th view, $F_{n-1,m}$. Therefore, the total number of bits required for encoding of current frame in each mode can be approximated according to Eqs. 3.12 and 3.13.

$$\begin{aligned} S_{MC_{n,m}}(u,v) \approx\ & [\tfrac{1}{2}\log_2(2\pi e\sigma_{MC_{n,m}}(u,v))]\cdot 64 \\ & +s^{MV}_{n-1,m}(u,v) \end{aligned} \tag{3.12}$$

$$S_{DC_{n,m}}(u,v) \approx [\tfrac{1}{2}\log_2(2\pi e\sigma_{DC_{n,m}}(u,v))]\cdot 64 \tag{3.13}$$

In Eqs. 3.12 and 3.13, $\frac{1}{2}\log_2(2\pi e\sigma^2)$ is the entropy of a Gaussian source with a variance of σ^2 [52]. As a result, $\frac{1}{2}\log_2(2\pi e\sigma^2)\times 64$ is approximately the required number of bits for encoding one block, assuming a Gaussian distribution for motion and disparity compensated residues (Gaussian distribution for motion and disparity compensated frames is also used in [49]).

If (u,v)'th block is assigned to be coded by using motion compensation, the required number of bits for coding of MC residue, can be approximated by the entropy of the residue frames times the size of the block which is formulated in the form of $[\frac{1}{2}\log_2(2\pi e\sigma^2{}_{DC_{n,m}}(u,v))]\cdot 64$. In order to be able to perform the motion compensation in the decoder, H.264/AVC will also transmit the motion vectors. Therefore, the approximate size of motion vectors of the corresponding block, $s^{MV}_{n-1,m}(u,v)$, is also added to obtain the approximate size of Eq. 3.12. Similarly, if the block is supposed to be coded by disparity compensation, the residue block can be coded by approximately $[\frac{1}{2}\log_2(2\pi e\sigma^2{}_{MC_{n,m}}(u,v))]\cdot 64$ bits. In contrast to the case of motion compensation, it is not required to transmit disparity vectors and the total approximate size will be as in Eq. 3.13.

These approximate sizes, $S_{MC_{n,m}}(u,v)$ and $S_{DC_{n,m}}(u,v)$, are finally used in Eq. 3.14 for classifying each block of P/B frame. Similar to the case of I frames, these classification bits (1 for SW and 0 for IV) are sent back to the encoder to determine the mode in which each block should be encoded.

$$CB^m_n(u,v) = \begin{cases} 0 & , S_{MC_{n,m}}(u,v) \le S_{DC_{n,m}}(u,v) \\ 1 & , S_{MC_{n,m}}(u,v) > S_{DC_{n,m}}(u,v) \end{cases} \tag{3.14}$$

According to Eq. 3.14, the mode which needs less bits for transmission of the block is assigned for coding of that block.

3.1.5 SW Coding of Frame Blocks

Assuming the (u,v)'th block , $[F_{n,m}(i,j)], (i,j)\epsilon B_{uv}$, to be SW encoded, its coding is described in this section. Having a noise free channel, the encoded sequence is decoded with the aid of the generated side information, $[R^{DC}_{n,m}(i,j)], (i,j)\epsilon B_{uv}$, at the decoder and $[R_{n,m}(i,j)], (i,j)\epsilon B_{uv}$ is reconstructed. Now, we consider SW coding of $[F_{n,m}(i,j)]$ having the side information, $[R^{DC}_{n,m}(i,j)]$, in the decoder where $(i,j)\epsilon B_{uv}$.

Applying DCT on $[F_{n,m}(i,j)]$, the DCT coefficients, $[F^{DCT}_{n,m}(i,j)]$, are obtained. These coefficients are then divided by a rate-adapted 8×8 quantization matrix, $[Q(i',j')]$, and the results are quantized to obtain $[F^{DCT,Q}_{n,m}(i,j)]$ as follows,

$$\begin{aligned} F^{DCT,Q}_{n,m}(i,j) &= \lfloor \frac{F^{DCT}_{n,m}(i,j)}{Q(i',j')} \rfloor, \\ where \quad & i' = i \ mod \ 8, j' = j \ mod \ 8 \end{aligned} \tag{3.15}$$

where $\lfloor . \rfloor$ stands for rounding to the nearest integer. We used 2^{sb} (sb is the predefined number of SW bits) bins, b_x, where $x = 1, .., 2^{sb}$. In each bin, there are the same number of integers, $z_{x,y}$, so that the union of bins spans the integer numbers between -256 and 255:

$$\begin{aligned} b_x &= \{z_{x,y} | y = 1, ..., \tfrac{512}{2^{sb}}\}, \\ z_{x,y} &= -256 + (x-1) + (y-1) \cdot 2^{sb} \end{aligned} \tag{3.16}$$

As represented in Eq. 3.17, SW coding will map $F^{DCT,Q}_{n,m}(i,j)$ to the index, $F^{SW}_{n,m}(i,j) = x$, where $F^{DCT,Q}_{n,m}(i,j) \in b_x$:

$$F^{SW}_{n,m}(i,j) = x \Leftrightarrow F^{DCT,Q}_{n,m}(i,j) \in b_x \tag{3.17}$$

Although using advanced channel codes (*e.g.,* turbo and LDPC codes) may be more efficient for SW coding, simple binning is preferred as it has the flexibility of being combined with zigzag zero-run coding. Specifically, the set of indices, $[F^{SW}_{n,m}(i,j)], (i,j)\epsilon B_{uv}$, is zigzag scanned and the resulting stream is zero-run coded (similar to JPEG entropy coding, described in [52]). As a result, we have simultaneously used inter-view and

spatial redundancies.

At the decoder, $[F^{SW}_{n,m}(i,j)]$ is used along the side information, $[R^{DC}_{n,m}(i,j)]$, to obtain $[R_{n,m}(i,j)]$. By applying DCT and quantization of Eq. 3.15 on $[R^{DC}_{n,m}(i,j)]$, we obtain $[\breve{R}^{DC}_{n}(i,j)]$ to be used in decoding of $F^{SW}_{n,m}(i,j)$'s:

$$\begin{aligned} R^{DCT,Q}_{n,m}(i,j) = \ & \arg\min_{z_{x,y}\epsilon b_x} |z_{x,y} - \breve{R}^{DC}_{n,m}(i,j)| \\ & where \quad x = F^{SW}_{n,m}(i,j) \end{aligned} \tag{3.18}$$

The decoded quantized DCT coefficients, $[R^{DCT,Q}_{n,m}(i,j)]$, are dequantized using the quantization matrix of the encoder, $[Q(i',j')]$:

$$\begin{aligned} R^{DCT}_{n,m}(i,j) \ &= (R^{DCT,Q}_{n,m}(i,j) + 0.5) \times Q(i',j'), \\ & i' = i \ mod \ 8, j' = j \ mod \ 8 \end{aligned} \tag{3.19}$$

Then, inverse blockwise DCT is applied on the reconstructed DCT coefficients, $[R^{DCT}_{n,m}(i,j)]$, and gray values of the block, $[R_{n,m}(i,j)]$, are finally obtained.

3.1.6 Disparity Estimation

Assume F_A is the frame, for which we calculate the disparity vectors, $\hat{D}$, and the associated cost, c, with respect to a reference frame, F_B:

$$(\hat{D}, c) \ = \mathrm{f}_{DE}(F_A, F_B) \tag{3.20}$$

We represent the horizontal and vertical disparity vectors corresponding to the (i,j)'th pixel of F_A by $D(i,j) = (D_H(i,j), D_V(i,j))$. Similarly, the set of disparity vectors for all the pixels of F_A is represented by $\hat{D} = (\hat{D}_H, \hat{D}_V)$ where $\hat{D}_H = [D_H(i,j)]$ and $\hat{D}_V = [D_V(i,j)]$. In this section, a brief explanation of the method used for calculating $\hat{D} = (\hat{D}_H, \hat{D}_V)$ is presented. As used in [13], a Belief Propagation (BP) based stereo matching algorithm is used for estimating the disparity vectors.

The disparity estimation among two view frames can be formulated in the following form:

$$\hat{D} = \arg\min_{D\epsilon\mathfrak{D}} E(D) \tag{3.21}$$

In Eq. 3.21, $E(D)$ is the cost associated with the frame disparity vectors, $D \epsilon \mathfrak{D}$ (defined in Eq. 3.23), where $\mathfrak{D}$ is the predefined set of possible integer frame disparity vectors. In our case, this set was defined according to:

$$\begin{aligned}\mathfrak{D} = \; & \{D = (D_H, D_V),\ D_H = [D_H(i,j)],\ D_V = [D_V(i,j)] |\ D_H \in \mathrm{I}_D,\ D_V \in \mathrm{I}_D\} \\ & , \mathrm{I}_D = \{(I_{D,step} \cdot I) \in \mathbb{Z} |\ |I_{D,step} \cdot I| \leq I_{D,max}\}\end{aligned} \tag{3.22}$$

where $I_{D,max}$ and $I_{D,step}$ are the predefined positive integers which determines the biggest possible value and the step size (precision) of disparity vectors, respectively.

$$E(D) = \sum_{(i,j)} e_1(D(i,j)) + \sum_{\langle (i,j),(i',j') \rangle \epsilon \mathfrak{N}} e_2(D(i,j) - D(i',j')) \tag{3.23}$$

In Eq. 3.23, $e_1(D(i,j))$ is the cost of assigning $D(i,j)$ to (i,j)'th pixel, and $e_2(D(i,j) - D(i',j'))$ is the cost of assigning $D(i,j)$ and $D(i',j')$ to the neighboring pixels, (i,j) and (i',j'). $\mathfrak{N}$ is the set of neighboring pixel pairs in the frame as defined in the following:

$$\mathfrak{N} = \{\langle (i,j),(i',j') \rangle |\ (i-i')^2 + (j-j')^2 < B_{size}^2\} \tag{3.24}$$

where B_{size} is the block size used for disparity compensation (*e.g.;* $B_{size} = 8$ in this case).

To solve the minimization problem, the iterative method, described in [53], is adopted. The aforementioned costs, $e_1(D(i,j))$ and $e_2(D(i,j) - D(i',j'))$ are also defined in Eqs. 3.25 and 3.26:

$$e_1(D(i,j)) = |F_A(i,j) - F_B^{DC}(i,j)| \tag{3.25}$$

$$e_2(D(i,j) - D(i',j')) = \frac{\|D(i,j) - D(i',j')\|}{\sqrt{(i-i')^2 + (j-j')^2}} \tag{3.26}$$

where $\|.\|$ stands for Euclidean norm operator. In Eq. 3.25, F_B^{DC} is the result of warping F_B^{DC} according to D (*i.e.;* $F_B \overset{D}{\rightarrow} F_B^{DC}$):

$$F_B^{DC}(i,j) = F_B(i + D_H(i,j), j + D_V(i,j)) \tag{3.27}$$

and the marginal pixels are handled by zero padding. Finally, the associated cost for

Table 3.1: Coding parameters for classification based coding (proposed method) of *tunnel*

Parameter	Value
Frame Size	720×288
Frame Rate	30 fps
No of Frames	20
No of Views	2
Classification Block Size	8×8
Intra-View Coding Structure	IPP..P (GOP=20)
No of SW Bins (2^{sb})	8
Disparity Vector Precision	1

assigning $\hat{D}$ as the frame disparity vectors, c, is calculated as follows:

$$c = E(\hat{D}) \tag{3.28}$$

where $E(.)$ is the energy function introduced in Eq. 3.23.

3.2 Experimental Results

The evaluation of our proposed multi-view video coding scheme is done in two steps. First, the proposed method is compared with: **a)** the asymmetric coding scheme of Yang *et al.* [45], and **b)** with separate H.264/AVC encoding and **c)** joint encoding (JMVM [46]), for a two-view video sequence. Furthermore, our proposed method is applied on two multi-view video sets (with 15 and 16 views) and its efficiency is compared with separate encoding and joint encoding in terms of rate-distortion. Second, the proposed method is compared with separate H.264/AVC coding in terms of the computational complexity of the encoder, in section 3.2.2. The set of samples used for our experiments can be downloaded at http://grads.ece.mcmaster.ca/~nabaeem/MultiViewdata,

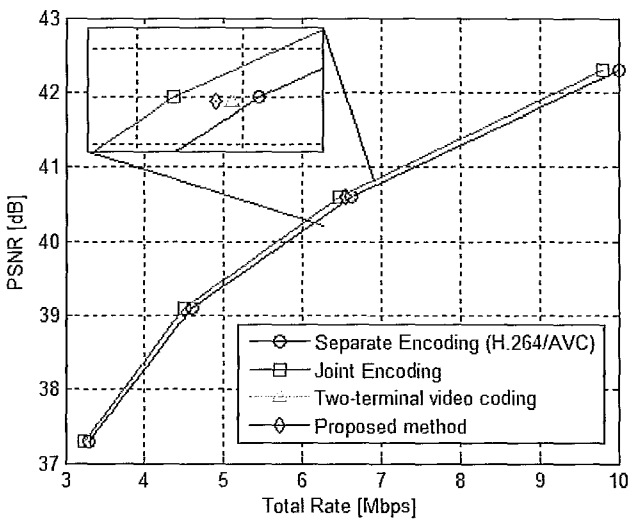


Figure 3.5: Rate-distortion curves for different coding schemes on *tunnel* (*Mbps* stands for Megabits per second).

Table 3.2: Coding parameters for separate H.264/AVC coding of *tunnel*

Parameter	Low Rate	High Rate
Coding Structure	IPP..P (GOP=20)	IPP..P (GOP=20)
I Frame QP	35	22
P Frame QP	33	20
Motion Search Range	16	16
Entropy Coding Mode	CABAC	CABAC

3.2.1 Rate-Distortion Evaluation of the Proposed Method

A stereoscopic (two-view) video sequence, called *tunnel*, is used to compare the proposed method with Yang *et al.* two-terminal video coding method [13]. We applied our classification based scheme on Y-components of 288×720 *tunnel* sequence which has 20 frames in each view. More specifically, the left view frames are coded by using H.264/AVC mono-view coder and the right view was coded by using the proposed method. The resulting rate-distortion curve for our method is depicted along the reported rate-distortion points of Yang *et al.* work in Fig. 3.5. The adopted coding

Table 3.3: Summary of experimental results for *tunnel* sequence

Parameter	Low Rate	High Rate
App. Description Rate [%]	8.41	1.03
Feedback Channel Rate [%]	1.14	0.21
I Refinements Rate [%]	27.46	10.52
P Refinements Rate [%]	62.98	88.24
Total Rate	856.94 kbps	6.568 Mbps
IF Classified Blocks (I frame)	39.75%	36.40%
IV Classified Blocks (P frame)	91.23%	89.78%
PSNR	31.15 dB	40.59 dB
Separate Encoding [54]	866.3 kbps	6.63 Mbps
Yang et al. [13]	860.7 kbps	6.58 Mbps
Joint Encoding [46]	848.97 kbps	6.50 Mbps

structure for separate coding, joint coding and the proposed method includes a single I frame followed by P frames which is the same as the structure used in [13]. Lists of coding parameters for the proposed method and separate H.264/AVC coding are presented in Tables 3.1 and 3.2, respectively. As reported in [13], the disparity maps and motion fields for two-terminal video coding of *tunnel* are generated in half-pel precision by using the BP stereo matching algorithm. To compare the rates corresponding to each compression method, we interpolated the rate-distortion curves and obtained the corresponding rates for the same PSNR, as presented in Table 3.3. Also, the rate-distortion curves for separate encoding using H.264/AVC reference software, [54], and joint encoding (JMVM [46]), are shown in Fig. 3.5.

As shown, joint encoding is the upper bound of the curves in terms of rate-distortion as no multi-terminal video coding scheme can outperform it. In Fig. 3.5, the horizontal axis represents the average total rate, required for transmission of one view. This is calculated by accumulating the number of bits required to transmit approximate descriptions, classification bits and refinements of all frames (captured in one second) of different views and then dividing the result by the number of views. The vertical axis also represents the average PSNR, calculated for all frames (both I and P) of the views. As presented in Table 3.3, the proposed method has a better compression efficiency than both separate encoding and two-terminal video coding work by Yang *et. al.* Our proposed method achieves 1.08% and 0.65% bit saving for

low and high rates respectively, compared to separate encoding. Although these values may seem small, it should be noted that the upper bound, achieved by joint encoding, requires only 2.00% and 1.96% less bits for low and high rates respectively, compared to separate encoding (there is a tiny gap between separate and joint encoding where our results and Yang *et al.* results lay in). However, this tiny gap can be expanded if the multi-view video sequence has more than two views, as discussed below. Moreover, it should be highlighted that the feedback channel bits is less than one percent of the total bits required for the entire transmission. Therefore, it is reasonable to say that the proposed method has very limited communication among the encoders and can be compared with multi-terminal video coding schemes.

We evaluate the proposed method for cases where several views are included in the multi-view video sequence. Therefore, we use two commonly used multi-view video sequences, called *Akko&Kayo* and *Rena*, to prove the potential improvement of our method. Similar to the first step, the Y-component of *Akko&Kayo* and *Rena* are calculated and different methods are applied on them. The adopted coding uses IBPBP...IBPBP structure for both separate H.264/AVC coding and the proposed method. The list of coding parameters for separate H.264/AVC coding and joint encoding are provided in Tables 3.4 and 3.5. In Fig. 3.6, the rate-distortion curves for separate encoding, joint encoding and the proposed method are shown. To see the drawback of not using the classification, the proposed method was used with having all the blocks classified as SW coded and the resulting rate-distortion curve is depicted in Fig. 3.6. Furthermore, the proposed method was evaluated with the ideal classification where the original blocks are used rather than their references (approximate description for I frames, and previously decoded frame of the view for P/B frames) and the resulting rate-distortion curve is presented in Fig. 3.6. To the best of our knowledge, there is no other experiments on multi-terminal video coding (with several views) in the literature, to compare our method with.

As shown in Fig. 3.6, the proposed classification based multi-terminal video coding scheme has gained around 1 dB improvement over separate H.264/AVC encoding which is significant. However, there is still around 0.7 dB room to reach the upper boundary (where the views are jointly encoded). It is shown in Fig. 3.6 that having the

Table 3.4: Separate H.264/AVC coding parameters for *Akko&Kayo* and *Rena* sequences

Parameter	Akko&Kayo	Rena
Coding Structure	IBP...IBP..	IBP...IBP..
Entropy Coding Mode	CABAC	CABAC
GOP Size	12	12
Motion Search Range	32	32
I Frame QP Range	17-36	17-36
P Frame QP Range	15-34	15-34
B Frame QP Range	15-34	15-34

Table 3.5: Joint coding parameters for *Akko&Kayo* and *Rena* sequences

Parameter	Akko&Kayo	Rena
View Order	26 27 28 29 30 46 47 48 49 50 66 67 68 69 70	38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
Entropy Coding Mode	CABAC	CABAC
GOP Size	15	15
Motion Skip Mode	Off	Off
Basic QP Range	17-36	17-36

classification not being used causes the rate-distortion performance to be worse than separate H.264/AVC coding. Moreover, it is experimentally proven that the choice of references used for classification (approximate description for I frames and previously decoded intra-view frame for P/B frames) is fairly appropriate as the rate-distortion curve of the proposed method nearly lays on the same rate-distortion curve of the ideal classification with the original frames. Parameters of *Akko&Kayo* and *Rena* and the experimental results are presented in Table 3.6. In this table, $\Delta PSNR_{C-S}$ is defined as the average improvement of the proposed classification based method over separate H.264/AVC coding in terms of PSNR. This average was obtained by interpolating the rate-distortion curves for all the rates in the obtained range and then calculating the average PSNR improvement over those rates. Similarly, $\Delta PSNR_{J-S}$ is the average PSNR improvement of joint encoding over separate H.264/AVC coding. By interpolating the corresponding rate-distortion curves, we calculated the average bit rate savings for different cases and they are reported in Table 3.6. Specifically, *C-S*

Table 3.6: Summary of experimental results and parameters for *Akko&Kayo* and *Rena* sequences

	Akko&Kayo	Rena
Frame Size	640 × 480	640 × 480
Frame Rate	30 fps	30 fps
No of Frames	300	300
No of Views	15	16
GOP Size	12	12
Classification Block Size	8 × 8	8 × 8
Intra-View Coding Structure	IBP...IBP..	IBP...IBP..
No of SW Bins (2^{sb})	8	8
Disparity Vector Precision	1	1
Average Total Rate [kbps]	1310.4	551.4930
App. Description Bits [%]	6.57	4.71
Feedbach Channel Bits [%]	0.95	1.88
I Refinements Bits [%]	38.36	43.50
P/B Refinements Bits [%]	54.12	49.91
IF Classified Blocks (I frame)	33.94%	24.67%
IV Classified Blocks (P/B frame)	87.41%	83.56%
$\Delta PSNR_{C-S}$	0.84 dB	1.22 dB
$\Delta PSNR_{J-S}$	1.64 dB	2.15 dB
C-S Bit Saving	14.90%	19.89%
J-S Bit Saving	27.89%	33.98%

Bit Saving is defined as the average percentage of bit saving which is achieved by the proposed classification based method over separate H.264/AVC coding. Similar to the case of $\Delta PSNR_{C-S}$, this average is obtained by interpolating the rate-distortion curves for different PSNRs and then calculating the average percentage of bit saving improvement. *J-S Bit Saving* is also the average percentage of bit saving which is achieved by joint encoding over separate H.264/AVC coding.

3.2.2 Analysis of Encoder Computational Complexity

In this section, the computational complexity of the encoder is investigated to find out any possible changes in the computational efficiency of proposed method compared to separate H.264/AVC encoding. In the case of I frames, the encoder is in

charge of generating the approximate description and its coding. We represent its computational complexity order by CO_{app}. After the classification bits are received in the decoder, a fraction γ_I of the blocks are marked to be H.264 encoded and the remaining fraction $(1-\gamma_I)$ of the blocks are SW encoded. Representing the computational complexity of H.264/AVC (I mode) and SW encoding by CO_{H264-I} and CO_{SW}, the average computational complexity of the encoder for I frames can be obtained according to:

$$CO_I = CO_{app} + \gamma_I.CO_{H264-I} + (1-\gamma_I).CO_{SW} \tag{3.29}$$

where CO_{H264-I} and CO_{SW} are the average computational complexities for an entire frame.

For P/B frames, the encoder is only in charge of H.264 encoding of a fraction γ_P/γ_B of blocks and SW encoding of the remaining blocks. Representing the average computational complexities of H.264 in P and B mode by CO_{H264-P} and CO_{H264-B}, the average complexity of the encoder for P and B frames will be formulated as follows:

$$CO_P = \gamma_P.CO_{H264-P} + (1-\gamma_P).CO_{SW} \tag{3.30}$$

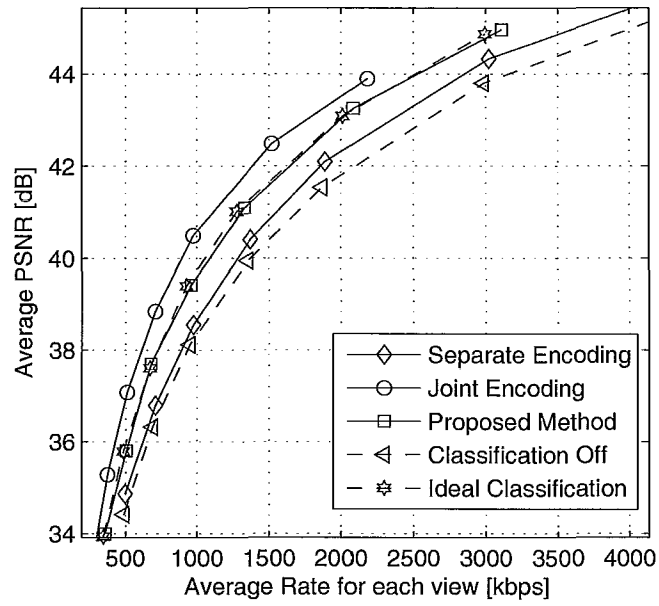
$$CO_B = \gamma_B.CO_{H264-B} + (1-\gamma_B).CO_{SW} \tag{3.31}$$

The average computational complexities of H.264 encoding in I, P and B modes in addition to SW encoding and approximate description generation and encoding is measured in terms of their average execution times on the same processor. More specifically, we use a 2.4 GHz Central Processing Unit with 1 GBytes of memory for our our experiments. Shown in Table 3.7, these execution times are obtained by averaging over views, and for different rates/distortions, for a particular frame type (*i.e.;* I, P, B). In this table, CO_{total} and $CO_{H264-total}$ are the average execution times for encoding all the frames of a single view GOP using the proposed method and separate H.264/AVC. These values are calculated by taking into account the adopted GOP structure (*i.e.;* IBP..IBP..) and the values obtained for CO_{H264-I}, CO_{H264-P}, CO_{H264-B}, CO_{app} and CO_{SW}. calculated by considering the adopted coding structure (*i.e.;* IBP..IBP..). It is shown in Table 3.7 that the average execution times for encoding one GOP of test sequences is almost the same for both H.264/AVC and the proposed method (around $4.2ksec$ and $5.0ksec$ for the proposed method and separate

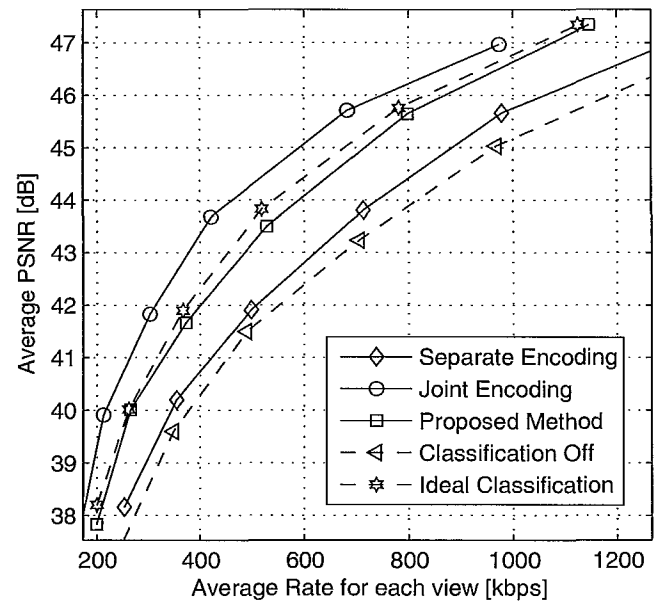
H.264/AVC encoding). The proposed method is better than separate H.264 coding in terms of its rate-distortion performance (around 17% bit saving) and almost the same order in terms of encoder complexity, which is achieved by a small cost in having an additional feedback channel in our scenario.

Table 3.7: Average execution times of encoder blocks for different sequences

	Akka&Kayo	Rena
CO_{app} [ms]	12,897	13,561
CO_{SW} [ms]	5,106	4,763
CO_{H264-I} [ms]	4,311	4,103
CO_{H264-P} [ms]	211,154	205,323
CO_{H264-B} [ms]	653,763	637,271
CO_{I} [ms]	17,733	18,159
CO_{P} [ms]	188,489	157,189
CO_{B} [ms]	556,464	574,020
CO_{total} [ms]	4,299,002	4,248,224
$CO_{H264-total}$ [ms]	4,982,659	4,854,344



(a)



(b)

Figure 3.6: Rate-distortion curves for different coding schemes of (a) *Akko&Kayo* and (b) *Rena* samples.

Chapter 4

Automated Vision-based Event Detection

In this chapter, the process of event detection using the vision data, in the form of different camera frames, $\mathrm{F} = \{F_{n,m},\ n = 1, ..., N,\ m = 1, ..., M\}$, is described. We will describe the process of optimal decision making using the original measurements, *i.e.;* F. In section 4.1, we discuss optimal decision making in which bandwidth and computational constraints of the resources are not considered. In this section, we derive a *sub-optimal* decision making process which is practical for our application of interest, *i.e.;* detection of taking medicine. In sections 4.2.1 and 4.2.2, the details of two key blocks in the sub-optimal decision making system are described. The adopted human body and object tracking methods, described in sections 4.2.1 and 4.2.2, are modified versions of previously proposed methods in the literature, compatible with our application.

4.1 Optimal Vision based Event Detection

We represent the occurrence of the event at time n and its estimated (detected) value by $a(n) = a$ and $\hat{a}(n) = a$, respectively, where $a = 1$ and $a = 0$ for occurrence and not occurrence of the event. Moreover, the estimated value of $a(n)$, given the captured frames of m'th camera ($\underline{F}_m$), is represented by $\hat{a}_m(n)$.

In the optimal decision making, the joint probability density function of the original measurements and $a(n)$ is used for decision making. Specifically, representing the probability density functions and the probability mass functions by $p(\cdot)$ and $P(\cdot)$, respectively, the decision making criteria can be written in the form of Eq. 4.1:

$$P\{a(n)=1|\mathrm{F}\} \underset{\hat{a}(n)=1}{\overset{\hat{a}(n)=0}{\lessgtr}} P\{a(n)=0|\mathrm{F}\} \tag{4.1}$$

which compares a posteriori conditional probabilities. Using the Bayes' equality in Eq. 4.2, the detection criteria of Eq. 4.3 is obtained. In Eq. 4.3, the likelihood ratio of $\frac{p\{\mathrm{F}|a(n)=1\}}{p\{\mathrm{F}|a(n)=0\}}$ is compared with the optimal threshold of $\frac{P\{a(n)=0\}}{P\{a(n)=1\}}$.

$$P\{a(n)=a|\mathrm{F}\} = \frac{P\{a(n)=a\}\cdot p\{\mathrm{F}|a(n)=a\}}{p\{\mathrm{F}\}},\ a=1,0 \tag{4.2}$$

$$\frac{p\{\mathrm{F}|a(n)=1\}}{p\{\mathrm{F}|a(n)=0\}} \underset{\hat{a}(n)=1}{\overset{\hat{a}(n)=0}{\lessgtr}} \frac{P\{a(n)=0\}}{P\{a(n)=1\}} \tag{4.3}$$

Although the detection criteria of Eq. 4.3 is easy to implement (a simple comparison), it needs a significant amount of data for obtaining the probability functions. Therefore, using the criteria of Eq. 4.3 is not feasible for event detection from the original captured frames, F. To solve this issue, a set of operations are applied on the original measurements, F, to obtain a new set of measurements for which their conditional distributions are easier to calculate. To determine the best choice for new measurements and the corresponding operation, the special event that we are dealing with shall be discussed.

(a) (b)

Figure 4.1: The sub-events which compose the event of taking medicine: (a) hand approaches the drug bottle (b) hand approaches the mouth.

An event is composed of a series of motions (relative distance changes) of objects. As the motion can only be tracked by temporal data, it is essential to have a number of consecutive frames of the same view, $\underline{F}_m$. Understanding the nature of an event and breaking it into sub-events can makes it easier to derive a mathematical representation for the detection of the event. For instance, taking medicine is composed of two sub-events (shown in Fig. 4.1): *hand approaching the drug bottle* and *hand approaching the mouth*. We represent these two sub-events by HD ($HD(n) = 1$ when it occurs) and HM ($HM(n) = 1$ when it occurs), respectively. Likewise, their estimated value is represented by $\hat{HD}(n)$ and $\hat{HM}(n)$, respectively. Although this decomposition of taking medicine into these two sub-events is not always correct, it is useful for most practical cases.

Now, having these two sub-events, one reasonable condition to decide the event has occurred is that both HD and HM happen within a reasonable time interval. More specifically:

$$\{\hat{a}(n) = 1\} \Leftrightarrow \{\hat{HM}(n) = 1 \ \ \& \ \ \hat{HD}(n - n_{TD}) = 1\} \tag{4.4}$$

where n_{TD} is in a predefined reasonable interval.

The sub-events *hand approaching the drug bottle* and *hand approaching the mouth* can be detected by using optimal decision making based on the original measurements, *i.e.;* F, and their a posteriori probabilities. Similarly, this is not practical because it requires a significant amount of training data. Therefore, we developed a computer vision based algorithm for detection of these sub-events. More specifically, the motion of objects of interest, which in this case are *hand*, *mouth* and *drug bottle*, are used as the intermediate measurements for detection of $HD(n)$ and $HM(n)$. Therefore, we have to track the trajectory of these three objects. Then, the extracted trajectories can be fed into trained decision making systems for the occurrence of HD and HM. The process of tracking the objects of interest (hand, mouth and drug bottle) is discussed in section 4.2.

It should also be noted that the number of views in our scenario is more than one (*i.e.;* three views in our experiments) and this can help us have a better detection performance. In other words, fusing the information, which are captured and processed by different sensors, can enhance our total performance. For instance, having

multiple cameras let us capture pieces of the scene which are not in field of view of one camera and assure all objects of interest are tracked (*i.e.;* occluded regions). As a result, the detection performance of the system will be increased because of using different view information in making decision about the occurrence of an event, sensed by different sensors.

The performance of a detection criteria is usually evaluated in terms of its detection and false alarm probabilities, $\mathbb{P}^{det}$ and $\mathbb{P}^{fa}$.

$$\begin{aligned} \mathbb{P}^{det} &= P\{\hat{a}(n) = 1 | a(n) = 1\} \\ &= P\{\hat{HD}(n - n_{TD}) = 1 \ \& \ \hat{HM}(n) = 1 | a(n) = 1\} \end{aligned} \tag{4.5}$$

As you will see in the following sections, we adopted the process of detecting HD and HM, independently. Therefore:

$$\begin{aligned} \mathbb{P}^{det} &= P\{\hat{a}(n) = 1 | a(n) = 1\} \\ &= P\{\hat{HD}(n - n_{TD}) = 1 \ \& \ \hat{HM}(n) = 1 | a(n) = 1\} \\ &= P\{\hat{HD}(n - n_{TD}) = 1 | a(n) = 1\} \times P\{\hat{HM}(n) = 1 | a(n) = 1\} \\ &= \mathbb{P}^{det}_{HD} \times \mathbb{P}^{det}_{HM} \end{aligned} \tag{4.6}$$

Similarly, the probability of false alarm, $\mathbb{P}^{fa}$, is as in Eq. 4.7:

$$\begin{aligned} \mathbb{P}^{fa} &= P\{\hat{a}(n) = 1 | a(n) = 0\} \\ &= P\{\hat{HD}(n - n_{TD}) = 1, \ \hat{HM}(n) = 1 | a(n) = 0\} \\ &= P\{\hat{HD}(n - n_{TD}) = 1, \ \hat{HM}(n) = 1 | HD(n - n_{TD}) = 1, HM(n) = 0\} \\ &\times \tfrac{P\{HD(n-n_{TD})=1, HM(n)=0\}}{P\{a(n)=0\}} \\ &+ P\{\hat{HD}(n - n_{TD}) = 1, \ \hat{HM}(n) = 1 | HD(n - n_{TD}) = 0, HM(n) = 1\} \\ &\times \tfrac{P\{HD(n-n_{TD})=0, HM(n)=1\}}{P\{a(n)=0\}} \\ &+ P\{\hat{HD}(n - n_{TD}) = 1, \ \hat{HM}(n) = 1 | HD(n - n_{TD}) = 0, HM(n) = 0\} \\ &\times \tfrac{P\{HD(n-n_{TD})=0, HM(n)=0\}}{P\{a(n)=0\}} \\ &= \tfrac{P\{HD(n-n_{TD})=1, HM(n)=0\}}{P\{a(n)=0\}} \cdot \mathbb{P}^{det}_{HD} \cdot \mathbb{P}^{fa}_{HM} \\ &+ \tfrac{P\{HD(n-n_{TD})=0, HM(n)=1\}}{P\{a(n)=0\}} \cdot \mathbb{P}^{fa}_{HD} \cdot \mathbb{P}^{det}_{HM} \\ &+ \tfrac{P\{HD(n-n_{TD})=0, HM(n)=0\}}{P\{a(n)=0\}} \cdot \mathbb{P}^{fa}_{HD} \cdot \mathbb{P}^{fa}_{HM} \end{aligned} \tag{4.7}$$

In order to discuss the role of multi-sensor information fusion, we discuss optimal

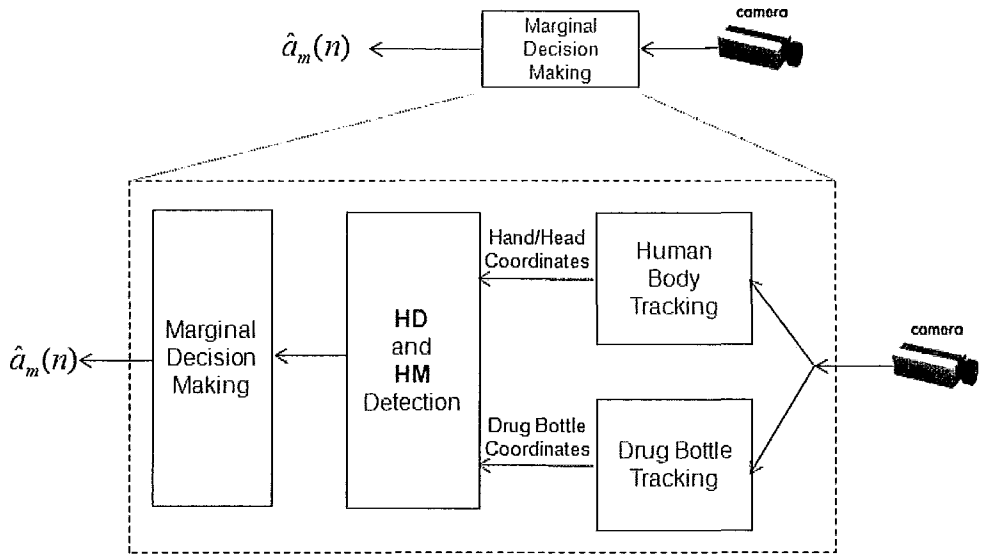


Figure 4.2: Marginal Decision Making.

decision making in two different scenarios. First, we discuss the process of marginal decision making ($\hat{HD}_m(n)$ =?, $\hat{HM}_m(n)$ =?) based on the data captured by a single sensor, $\underline{F}_m$, in section 4.1.1. Then, in section 4.1.2, we look at a multi-sensor system, where the information extracted from several sensors are used jointly to make a decision about the occurrence of an event.

4.1.1 Marginal Decision Making

In this work, we use the position of each object of interest as the intermediate measurements for the sub-event detection step. Specifically, we consider the position of each object in two-dimensional coordinate of each frame, which is obtained by the algorithm detailed in section 4.2. We use $\mathbf{O}^H_{n,m}$, $\mathbf{O}^D_{n,m}$ and $\mathbf{O}^M_{n,m}$ for the extracted coordinates of *hand*, *drug bottle* and *mouth* in the n'th frame of m'th view.

As shown in Fig. 4.2, the marginal decision making uses a single sensor data, $\underline{F}_m$, to make a marginal decision, $\hat{a}_m(n)$. As it was mentioned in section 4.1, we consider the event, as a composition of HD and HM sub-events. Using Eq. 4.4, the marginal

event detection rule from the detected marginal sub-events is according to:

$$\{\hat{a}_m(n) = 1\} \Leftrightarrow \{\hat{HM}_m(n) = 1 \ \ \& \ \ \hat{HD}_m(n - n_{TD}) = 1\} \tag{4.8}$$

where n_{TD} is a time delay, introduced in Eq. 4.4.

The optimal sub-event detection from the intermediate measurements, *i.e.;* the position of the objects of interest in each frame, $\mathbf{O}^H_{n,m}$, $\mathbf{O}^D_{n,m}$ and $\mathbf{O}^M_{n,m}$, can be performed by using the likelihood ratio test (similar to the criteria of Eq. 4.3). However, this would require the knowledge of conditional probability functions, which may not be desired for feasibility reasons. In the literature, a wide range of different applications have used Hidden Markov Model (HMM) to understand the characteristic of audio and visual events through time changes. More specifically, using the relative distance of the objects of interest during the time as the input of the HMM, the likelihood of occurrence of the sub-event can be obtained.

In [55], the application of hidden Markov model in speech recognition was firstly discussed. Following the work in [55], other people have used HMM for modeling the events and making decision about their occurrence. In a special case, one may use an HMM, composed of states, each representing a single event. The mixture of Gaussians is also a good choice for modeling the transition probabilities from one state to another. In such an HMM, after training the transition probabilities (*e.g.;* using BaumWelch algorithm [56]), the likelihood of happening of each event can be obtained from the trained HMM. Having these likelihood values, we can make a decision about the occurrence of each event (*i.e.;* two states in our case).

4.1.2 Multi-Sensor Decision Making

In a multi-sensor system, the information extracted from the streams of different sensors can be jointly used to make a decision about the occurrence of an event in the region covered by all or some of the sensors. As shown in Fig. 4.3, the fusion of extracted information can be in the highest level for which the marginal decisions, $\hat{a}_m(n)$, are fused together to estimate $\hat{a}(n)$. In other words, the marginal decisions, $\mathrm{A}(n) = (\hat{a}_1(n), ..., \hat{a}_M(n))$ can be considered as the measurements to the high level detection block which finds the optimum value of $\hat{a}(n)$. This decision making can be

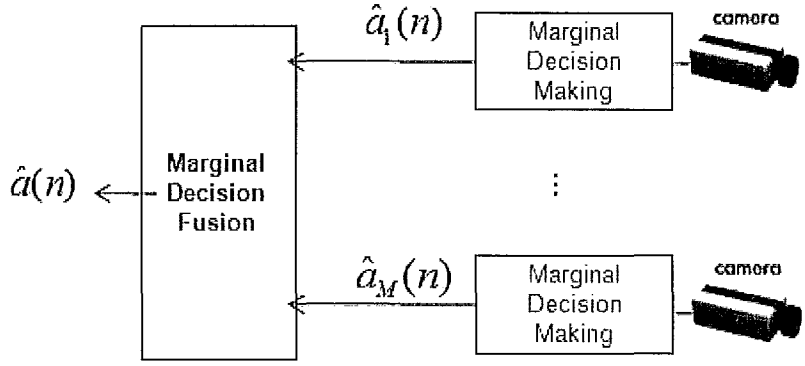


Figure 4.3: Multi-sensory decision making system.

done by using the Bayesian hypothesis test of Eq. 4.9:

$$\frac{p\{\mathrm{A}(n)|a(n)=1\}}{p\{\mathrm{A}(n)|a(n)=0\}} \underset{\hat{a}(n)=1}{\overset{\hat{a}(n)=0}{\lessgtr}} \frac{P\{a(n)=0\}}{P\{a(n)=1\}} \tag{4.9}$$

In this case, obtaining the set of posteriori probabilities of $a(n)$, conditioned on $\mathrm{A}(n)$, is feasible.

As $\hat{a}_m(n)$'s are estimated independently, the likelihood ratio of Eq. 4.9 can be expressed in the form of:

$$\frac{p\{\mathrm{A}(n)|a(n)=1\}}{p\{\mathrm{A}(n)|a(n)=0\}} = \frac{p\{\hat{a}_1(n)|a(n)=1\}\cdot \ \ldots \ \cdot p\{\hat{a}_M(n)|a(n)=1\}}{p\{\hat{a}_1(n)|a(n)=0\}\cdot \ \ldots \ \cdot p\{\hat{a}_M(n)|a(n)=0\}} \tag{4.10}$$

In Eq. 4.10, the conditional probabilities of $p\{\hat{a}_1(n)|a(n)=1\}$ and $p\{\hat{a}_1(n)|a(n)=0\}$ are according to Eqs. 4.11 and 4.12:

$$\begin{aligned} p\{\hat{a}_m(n)=\mathrm{a}|a(n)=1\} &= P\{\hat{a}_m(n)=1|a(n)=1\}\cdot\delta(\mathrm{a}-1) \\ &\quad + P\{\hat{a}_m(n)=0|a(n)=1\}\cdot\delta(\mathrm{a}-0) \\ &= \mathbb{P}_m^{det}\cdot\delta(\mathrm{a}-1)+(1-\mathbb{P}_m^{det})\cdot\delta(\mathrm{a}-0) \end{aligned} \tag{4.11}$$

$$\begin{aligned} p\{\hat{a}_m(n)=\mathrm{a}|a(n)=0\} &= P\{\hat{a}_m(n)=1|a(n)=0\}\cdot\delta(\mathrm{a}-1) \\ &\quad + P\{\hat{a}_m(n)=0|a(n)=0\}\cdot\delta(\mathrm{a}-0) \\ &= \mathbb{P}_m^{fa}\cdot\delta(\mathrm{a}-1)+(1-\mathbb{P}_m^{fa})\cdot\delta(\mathrm{a}-0) \end{aligned} \tag{4.12}$$

where $\delta(\cdot)$ is the dirac delta function with $\delta(a) = 1$ for $a = 0$ and $\delta(a) = 0$, otherwise. In Eqs. 4.11 and 4.12, $\mathbb{P}_m^{det}$ and $\mathbb{P}_m^{fa}$ are the probabilities of detection and false alarm corresponding to the decision making of the m'th sensor. Representing the ratio of $\frac{p\{\hat{a}_m(n)=a_m|a(n)=1\}}{p\{\hat{a}_m(n)=a_m|a(n)=0\}}$ by $\xi_m(a_m)$, the likelihood ratio of Eq. 4.10 is simplified to:

$$\frac{p\{\mathrm{A}(n) = (a_1, ..., a_M)|a(n) = 1\}}{p\{\mathrm{A}(n) = (a_1, ..., a_M)|a(n) = 0\}} = \Pi_{m=1}^{M} \xi_m(a_m) \tag{4.13}$$

where:

$$\begin{aligned} \xi_m(a_m) &= \frac{p\{\hat{a}_m(n)=a_m|a(n)=1\}}{p\{\hat{a}_m(n)=a_m|a(n)=0\}} \\ &= \frac{\mathbb{P}_m^{det}.\delta(a_m-1)+(1-\mathbb{P}_m^{det})\cdot\delta(a_m-0)}{\mathbb{P}_m^{fa}.\delta(a_m-1)+(1-\mathbb{P}_m^{fa})\cdot\delta(a_m-0)} \end{aligned} \tag{4.14}$$

Finally, the optimal decision making based on the marginal decisions of each sensor is according to the criteria of Eq. 4.15:

$$\Pi_{m=1}^{M} \xi_m(a_m) \underset{\hat{a}(n)=1}{\overset{\hat{a}(n)=0}{\lessgtr}} \frac{P\{a(n) = 0\}}{P\{a(n) = 1\}} \tag{4.15}$$

where $\xi_m(\cdot)$'s are as defined in Eq. 4.14.

4.2 Human Body and Drug Bottle Tracking

The tracking of our objects of interest based on vision data processing is discussed in this section. The vision based tracking systems include two major steps of extracting the descriptors (features) and then using them for detection and tracking of objects. In [57], a complete list of object descriptors (like point, geometrical shape, silhouette and contour) suitable for different objects and the suitable approaches for their recognition are provided. The study of multiple camera object tracking by using geometrical constraints on the camera parameters (*i.e.;* their relative position and angle) is discussed in [58]. However, as the fusion of multiple sensor data in our work is considered to be done at the level of marginal decisions, we focus on the detection and tracking of objects by using single camera data. Specifically, we studied previously proposed methods of object tracking and modified them in order to obtain the best tracking performance for our case.

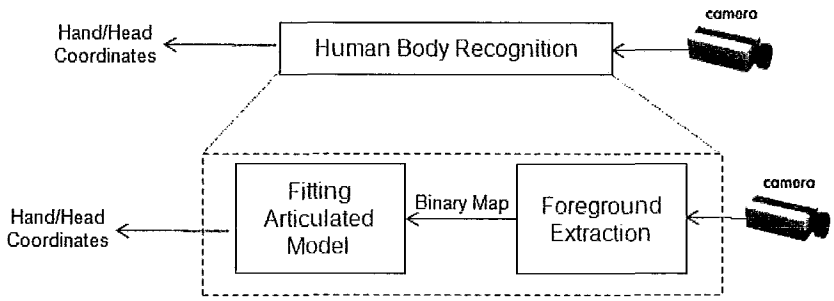
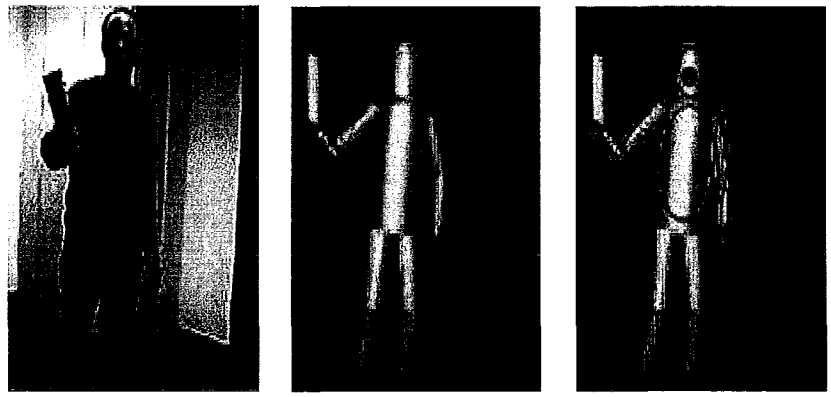


Figure 4.4: Human body recognition.

4.2.1 Human Body Tracking

Tracking of human body and segmenting each part of it (*e.g.;* hands and head) is well studied in the literature for different applications [59]. In our work, the process of extracting the human body from the content of a frame, $[F_{n,m}(i,j)]$, is done in two major steps (shown in Fig. 4.4). First, a *foreground extraction* method is applied on the input frames (having the background model) and the binary map of the foreground, $[FG_{n,m}(i,j)]$, is obtained. Then, the extracted foreground is fed into the *silhouette matching* step which matched an articulated model to the extracted binary map. The matched articulated model has the center positions corresponding to the hands, arms and head. The ending points of the hands are assigned as the position of hands, $\mathbf{O}^H_{m,n}$, used for event detection. The center of the head part is also assigned as the position of the mouth, $\mathbf{O}^M_{m,n}$. In Fig. 4.5(b), the extracted binary map corresponding to the input frame of Fig. 4.5(a), is shown. The extracted binary map is used to find the ellipses of human body and the coordinates of hands and mouth, as shown in Fig. 4.5(c).

The foreground extraction step is studied in the literature for a very wide variety of applications. Most of foreground extraction algorithms work based on the idea of motion detection and modeling the background. Inspired by the work in [60], we calculate a background model and then each frame is compared with the background model to distinguish the moving parts of the frame from the rest. As shown in Fig. 4.6, the difference between the background model and the frame, $[F_{n,m}(i,j) - F^{BG}(i,j)]$,



(a) (b) (c)

Figure 4.5: Human body recognition: (a) Input frame (b) The extracted foreground of input frame (c) The fitted articulated model to the extracted foreground map.

is calculated and compared to a threshold to obtain a binary frame;

$$|F_{n,m}(i,j) - F^{BG}(i,j)| \lesssim_{F^{FG}_{n,m}(i,j)=1}^{F^{FG}_{n,m}(i,j)=0} F^{TH}(i,j).$$

This binary frame, $[F^{FG}_{n,m}(i,j)]$, is passed through a segmentation step to merge disconnected parts and removing the noisy parts from the binary map. Finally, $[FG_{n,m}(i,j)]$ is the binary frame, including a single (or more depending on the predefined threshold) connected part as the silhouette of the human body. The segmentation step is the series combination of a few morphological blocks which first connect the disconnected parts by a closing operator and then remove the small noisy parts in the frame. In the resulting binary frame, each connected part is processed to obtain its geometrical characteristics. We used the size and the ratio of filled pixels to the product of width and height of the part in order to remove noisy parts of the binary map.

The background model extraction is a critical task as it affects the performance of foreground extraction. An adaptive mixture of Gaussians ([61]) is used to characterize the background model. More specifically, the gray value of each pixel, $f_m(i,j)$, is modeled by the mixture model of Eq. 4.16:

$$f_m(i,j) = \sum_{\upsilon=1}^{V} \sigma^{\upsilon}_m(i,j) \cdot (\mathbf{X} + \zeta^{\upsilon}_m(i,j)) \tag{4.16}$$

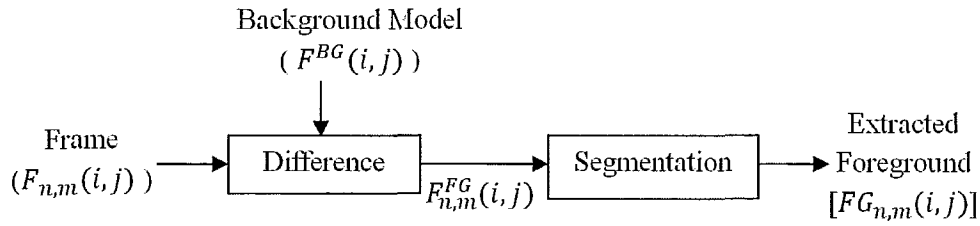


Figure 4.6: Foreground Extraction for Human Body Tracking.

where $\mathbf{X}$ is the normalized zero-mean Gaussian variable. To train the parameters of Eq. 4.16, *i.e.;* $\zeta^{\upsilon}_{m}(i,j)$ and $\sigma^{\upsilon}_{m}(i,j)$, a number of frames in the beginning of the stream are used which causes a delay in the tracking process in the beginning. After the set of mixture model parameters, *i.e.;* σ_{υ} and ζ_{υ}, is obtained for each pixel, the pixels of each frame are categorized into foreground and background regions, according to:

$$\begin{aligned} F^{FG}_{n,m}(i,j) = 1 \quad & \textbf{iff} \ \ \exists \upsilon = 1, ..., V \\ & s.t. \ \ |F_{n,m}(i,j) - \zeta_{\upsilon}| \leq \lambda_{FG} \cdot \sigma_{\upsilon} \end{aligned} \tag{4.17}$$

where λ_{FG} is a predefined parameter. After the pixels of current frame, $F_{n,m}(i,j)$, are categorized based on this rule, the parameters of the adaptive mixture model are modified based on these categorizations.

After obtaining a single silhouette binary map of the foreground, $[FG_{n,m}(i,j)]$, an articulated model is fitted to the binary map. As shown in Fig. 4.7, we have used an articulated human body model which included six connected ellipses: Two for each hand, one for head and one for trunk. Generally, each one has three freedom parameters as well as one parameter for the center of the body. The length of the ellipse (major diameter), the ratio of the major diameter to the minor diameter, and the angle of the major diameter with the horizontal axis are the freedom parameters of each ellipse in our model.

However, having these number of freedom parameters makes it computationally complex to find the best match for each binary map. Therefore, we usually decrease the number of used parameters by putting some reasonable constraints on different ellipse parameters, *e.g.;* the length of each hand are almost the same. Moreover, we

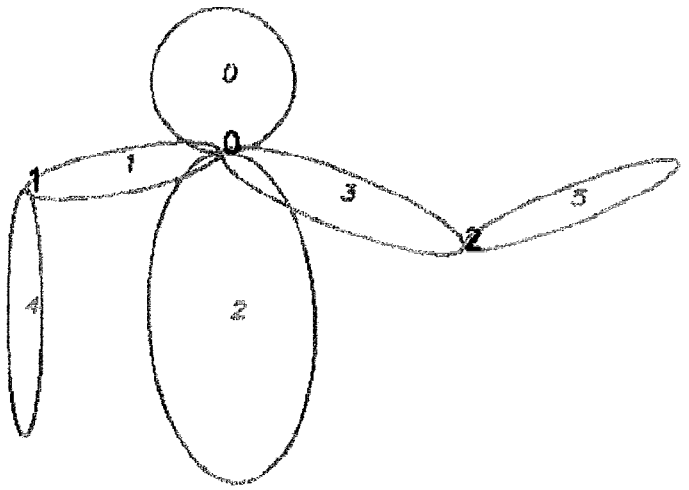


Figure 4.7: Articulated human body model [62].

perform the ellipse matching in a more computationally efficient way, as described in the following:

- Find the top of the binary map, called *top point.*
- Fit a circle to top part of the binary map, forcing it to pass the top point, called *head* ellipse.
- Find the bottom of the head ellipse, called *center point.*
- Perform the optimization on the freedom parameters of the remaining ellipses, having the ellipses number 1,2 and 3 (shown in Fig. 4.7) forced to pass the center point.

The last step, optimization on the freedom parameters, is also done by considering other constraints on the parameters of each ellipse to decrease the range of each parameter and the computation time, as a result. Moreover, temporal correlation of frames can help us limit the range of each ellipse parameters and gain a great advantage in the computational complexity.

4.2.2 Drug Bottle Recognition and Tracking

Detection and tracking of the drug bottle is more challenging than the human body. There are different descriptors and techniques, developed for such task and still there is no promising method with full reliability. In our work, we used a combination of point descriptors and color filtering for our aim. So far, the best point descriptors, widely used in the literature, is the *Scale Invariant Feature Transform (SIFT)*, developed by Lowe [63]. In Lowe's method, the input image is processed to find a set of candidate points with significant local features. The descriptor of each point is a vector of 128 elements which let us distinguish different points from each other, theoretically.

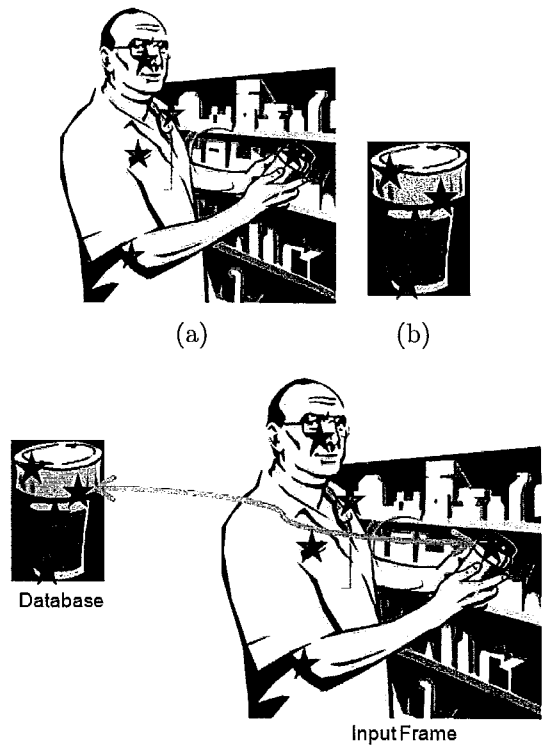


Figure 4.8: SIFT Object Recognition: (a) Input image and its SIFT feature points (b) One of the database image and its SIFT feature points (c) Matched SIFT points and their correspondence

In our work, we first have to develop a method to detect the center of the drug

bottle in an image. Then, we can develop our tracking method based on such a detection scheme. More specifically, temporal correlation of the frames can be used to limit the size of the image, fed into the bottle detector block, as the position of the drug bottle in the previous frame is known. As a result, a significant gain in our computational complexity can be achieved (since the number of SIFT feature points is decreased) in addition to an improvement in the accuracy of the bottle tracking. In the following, a brief explanation of SIFT object recognition [64] is provided. As

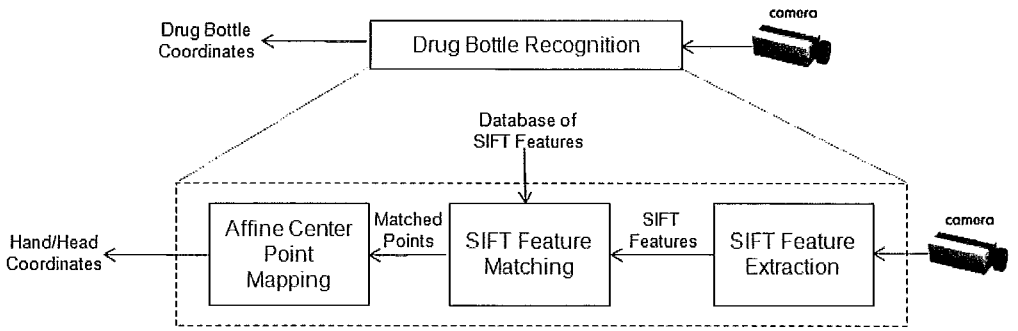


Figure 4.9: SIFT drug bottle recognition and tracking.

shown in Fig. 4.9, the process of SIFT object detection can be summarized in three major steps:

1. First, the SIFT features of the image are obtained, using the method described in [63].

2. Each SIFT feature point is compared with all the SIFT feature points in the database to find matches.

3. Finally, the matched points are processed in order to filter out mistaken points and find the corresponding center point of the object.

For the second step, a database of SIFT feature points and their descriptors is required which usually is provided by offline training with the aid of manual or semi-manual recognition. Obviously, the more feature points we store in our database, the better the detection performance would be. On the contrary, having more stored features

points will require more comparisons for finding matched SIFT points between the input frame and templates. As a result, the computational load is increased.

In the last step, the matched feature points are past-processed in order to find an affine transform among the feature points of the input image and those of database, as described in section 4.3. An example of SIFT feature matching is illustrated in Fig. 4.8. In Fig. 4.8(a), the SIFT feature points of the input image, used for SIFT tracking, are extracted and marked with bold stars. The template, used for SIFT tracking, and its SIFT feature points are shown in Fig. 4.8(b). Finally, a matched SIFT point between the input image and the template is shown in Fig. 4.8(c).

4.3 Proposed Constrained based Event Detection System

The human body and SIFT object tracking methods, described in section 4.2, are the conventional approach pursued in the previous human and SIFT object tracking systems. However, there is not any work done to adapt these methods with the bandwidth and computational constraints of our multi-sensory system. In this section, we introduce our proposed scheme for multi-camera event detection considering the bandwidth and computational complexity constraints. As shown in Fig. 4.10, the frames of each sensor, $\underline{F}_m$, are processed separately to find the position of objects of interest, $\mathbf{O}^H_{n,m}$, $\mathbf{O}^D_{n,m}$ and $\mathbf{O}^M_{n,m}$. These coordinates are then fed into HD and HM detection blocks and the detected values, $\widehat{HD}_m(n)$ and $\widehat{HM}_m(n)$, are obtained. These results are then used to find the sub-optimally detected values of $\widehat{a}_m(n)$.

In the hierarchy of Fig. 4.10, it is reasonable not to transmit the raw frames or even their compressed stream to the central node for processing, as it requires a lot of bits for its representation. This proposed resource-constrained design for event detection is shown in Fig. 4.11 where the distribution of different blocks between the sensor and central nodes is presented. Assume each frame includes a typical number of 640×480 pixels, each represented by eight bits. This will require around $2400kb$ per frame for its transmission without any coding. By using a conventional H.264/AVC coder, this is decreased to a bit rate of around $100kb$ per frame. Interestingly, if we transmit the extracted coordinates of objects of interest, *i.e.;* $\mathbf{O}^H_{n,m}$, $\mathbf{O}^D_{n,m}$ and $\mathbf{O}^M_{n,m}$, only a

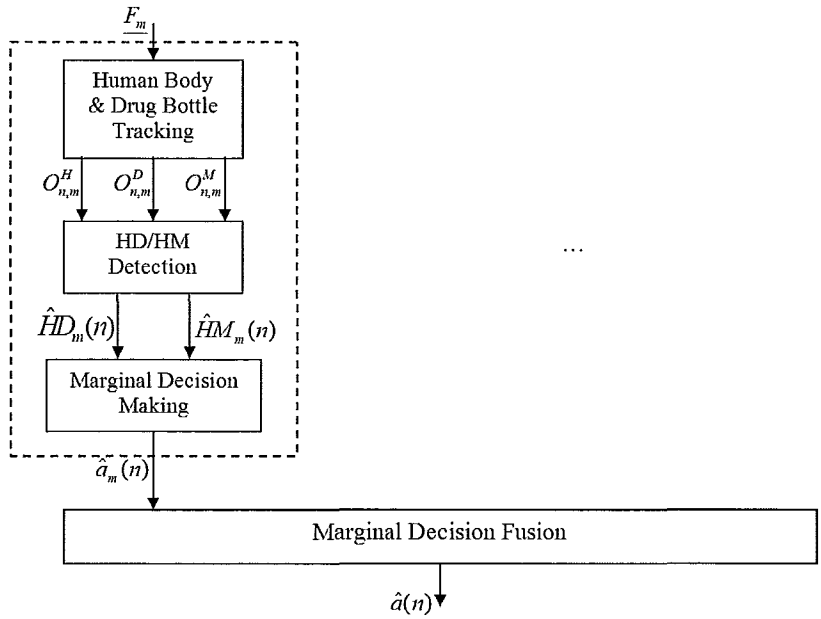


Figure 4.10: Overview of Event Detection System.

small rate, around $80b$ per frame, is necessary. Therefore, there will be no bandwidth related issue if we put the blocks of HD and HM detection, event detection in the central node. However, extracting these coordinates may be computationally complex and needs a computationally powerful sensor node. In the following sections, we investigate this issue and propose suitable solutions for each case of human body and drug bottle.

4.3.1 Human Body Tracking

As described in section 4.2.1, the recognition of the human body is performed in two major steps. First, the foreground of each frame is extracted (shown in Fig. 4.6) and then the extracted foreground is used to segment different body parts and obtain the position of hands and head. Unfortunately, the computational load of fitting the articulated model of ellipses to the foreground may be very high depending on the range and precision of different ellipse parameters.

To solve this issue, we assign the articulated model fitting to the central node

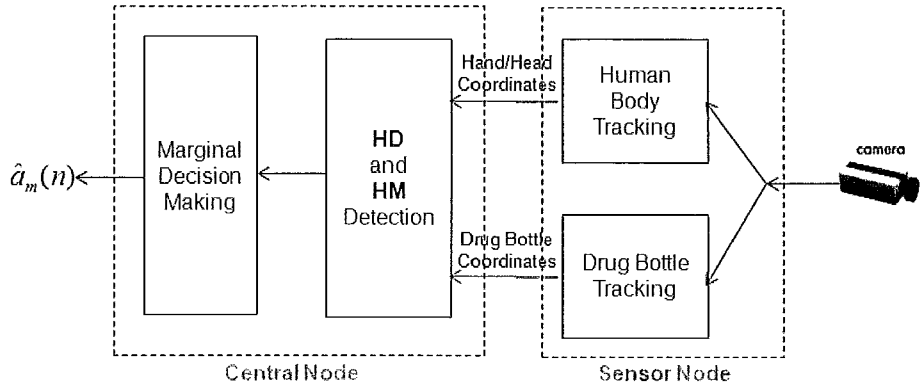


Figure 4.11: Proposed resource constrained event detection design.

where a high computational power is available. As a result, we need to transmit only the binary foreground map, $[FG_{n,m}(i,j)]$, from the sensor node to the central node. Fortunately, this binary map can be coded by a simple run-length encoding and transmitted to the central node with a very low rate. Moreover, the computational complexity of the sensor nodes is decreased as the sensor nodes are now only in charge of extracting the binary map of the foreground. As described in section 4.2.1, the binary foreground extraction is composed of a small number (V) of comparisons for each pixel and a morphological operation used for removing noisy parts of binary foreground map.

To achieve a computational efficiency than the aforementioned design, we will transmit $[F^{FG}_{n,m}(i,j)]$'s instead of $[FG_{n,m}(i,j)]$'s. As a result, the morphological operations, described in section 4.2, are not required to be performed at the sensor nodes while the required bandwidth for transmission of $[F^{FG}_{n,m}(i,j)]$ instead of $[FG_{n,m}(i,j)]$ is not increased significantly. A flow chart for the proposed human tracking scheme is shown in Fig. 4.12. In Fig. 4.12, the blocks of human body tracking are divided between sensor and central nodes in an efficient way, both in terms of communication bandwidth and computational load.

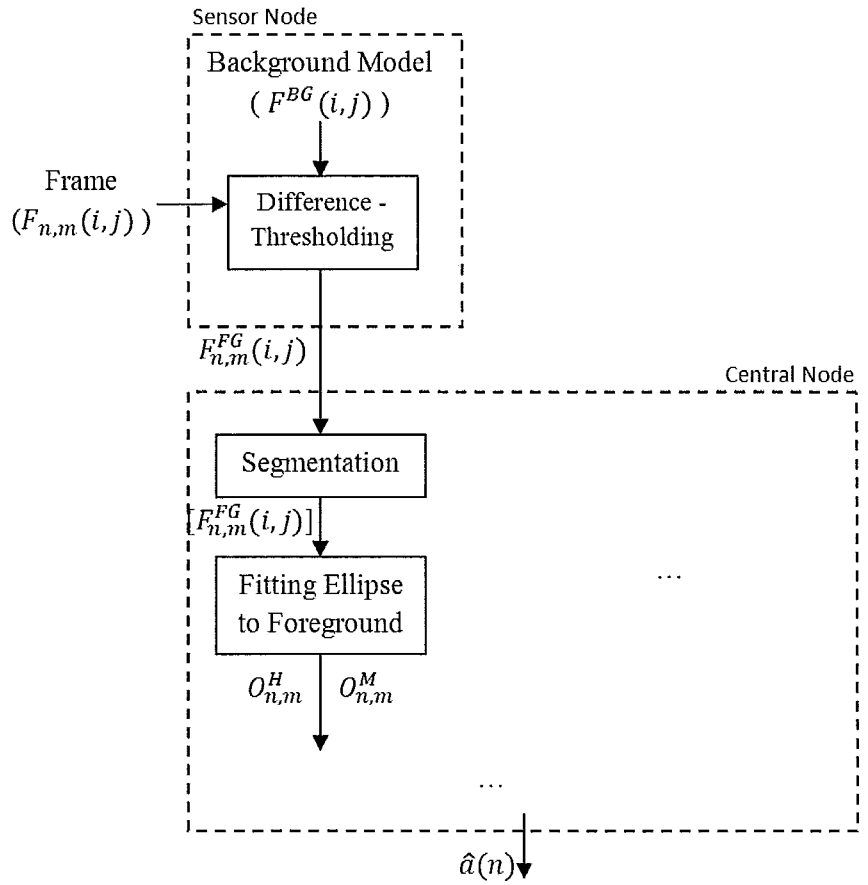


Figure 4.12: Proposed Constrained based Human Body Tracking Structure.

4.3.2 SIFT Drug Bottle Tracking

Tracking of the drug bottle is more challenging than human body tracking as discussed in this section. As described in section 4.2.2, the input image is processed to find its SIFT feature points and the corresponding SIFT descriptors, each a 128 element vector, are calculated. Then, each SIFT descriptor vector is compared with the descriptor vectors in the database to find matched pairs. These matched pairs are then processed to find an affine transformed object mapping between the corresponding object in the database and that of the input image. Temporal correlation of consecutive frames let us confine the window in which the object recognition is

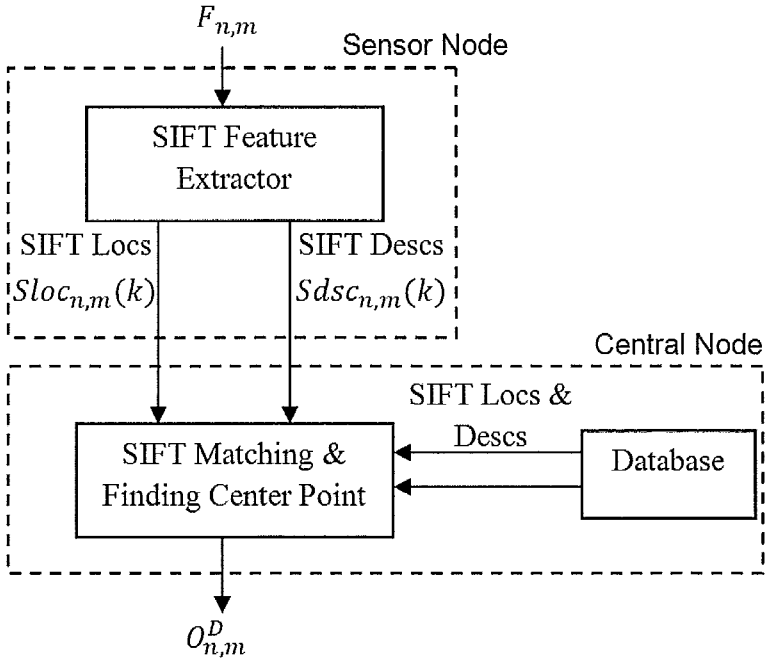


Figure 4.13: Proposed Constrained based Drug Bottle Tracking Structure using SIFT.

performed.

As shown in Fig. 4.13, we put the SIFT feature point extraction in the sensor node and the remaining parts in the central node in order to decrease the computational complexity of the sensor nodes. However, it should be determined that the bandwidth, required to transmit the SIFT feature points (*i.e.;* coordinates, $Sloc_{n,m}(k)$'s, and descriptors, $Sdsc_{n,m}(k)$'s), is not high compared to the bandwidth required for transmitting raw/compressed spatial domain data.

To find out whether the transmission of SIFT coordinates and descriptors is efficient in terms of the required bandwidth, we performed a series of experiments. Specifically, we used different input images and applied SIFT feature extraction algorithm of Lowe, [63] on them to find the required number of bits for transmission of the resulting $Sloc_{n,m}(k)$'s and $Sdsc_{n,m}(k)$'s. The experimental results showed that in some cases we require even more bits for transmission of the SIFT coordinates and descriptors than transmitting the raw/compressed spatial domain data. This issue can

be addressed by performing a KLT[1] on the SIFT descriptors to decrease the number of required bits with a sacrifice of little precision of $Sdsc_{n,m}(k)$'s. Specifically, we can consider $Sdsc_{n,m}(k) = [Sdsc^1_{n,m}(k)...Sdsc^{128}_{n,m}(k)]$'s as a random vector and perform KLT on the vectors to transform them onto a sparse or near sparse space. Then, the elements of the transformed vector with near zero values are not transmitted. We represent the transformed vector of SIFT descriptors by $Sdsc^T_{n,m}(k)$.

The KLT is a linear transform from $Sdsc_{n,m}(k)$'s into the transform domain of $Sdsc^T_{n,m}(k)$'s. Therefore, the value of each element of $Sdsc^T_{n,m}(k)$ is a linear combination of all elements of $Sdsc_{n,m}(k)$. In other words, all of the elements of $Sdsc_{n,m}(k)$ should be calculated to be able to perform KLT and transmit the high energy elements. Although this has no consequences in terms of the required sensor-central node bandwidth, it is associated with some computational load. To understand this issue and find a solution for that, a brief description of SIFT feature extraction is provided below.

As shown in Fig. 4.14, the input image is passed through a series of Gaussian filters with different scales, s, to obtain $G_s(i,j)$'s. Then, the resulting Gaussian filtered images are subtracted from the adjacent ones to obtain $\triangle G_s(i,j)$'s. In order to find the candidate keypoints in the image, each $\triangle G_s(i,j)$ is processed to find its local minima and maxima. These points are marked as the candidate points for SIFT keypoints and some of them are removed. The remaining points are the SIFT feature points with coordinates, $Sloc_{n,m}(k)$. For every $Sloc_{n,m}(k)$, the descriptor vector, $Sdsc_{n,m}(k) = [Sdsc^1_{n,m}(k)...Sdsc^{128}_{n,m}(k)]$, is then calculated, as described in [63] in detail. Specifically, each element of descriptor vector, $Sdsc^l_{n,m}(k)$, is calculated independently. In fact, the computational complexity of calculating all $Sdsc^l_{n,m}(k)$'s is almost 128 times of calculating a single $Sdsc^l_{n,m}(k)$. Therefore, using KLT for having a bandwidth efficient feature transmission, requires a heavy computational power.

To address bandwidth and computational power issues simultaneously, some form of feature selection should be done. Specifically, we pick a number L of $Sdsc^l_{n,m}(k)$'s to represent each $Sdsc_{n,m}(k)$, as shown in Fig. 4.15. As a result, it is not required to calculate the $Sdsc^l_{n,m}(k)$'s which are not selected and the total computational

[1]Karhunen-Loeve Transform

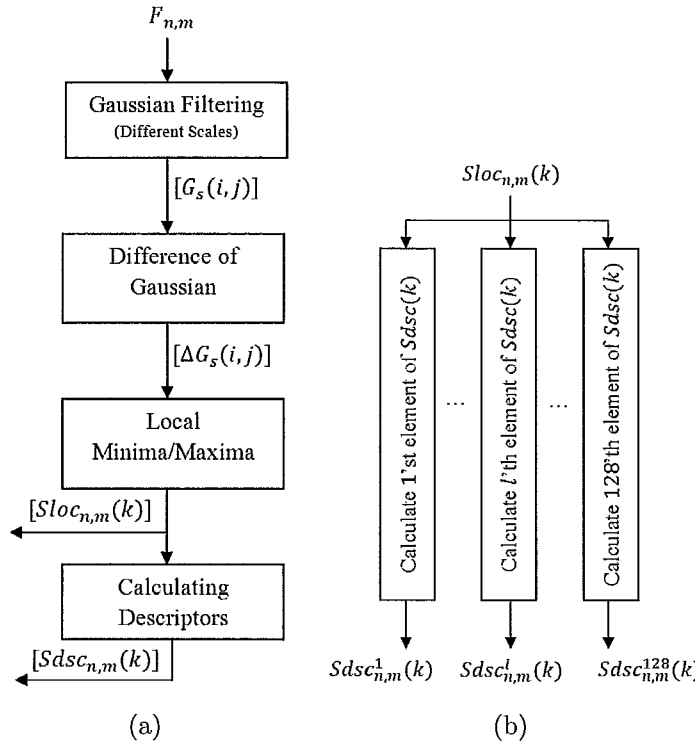


Figure 4.14: Overview of SIFT Feature Extraction: (a) Different blocks to extract the SIFT points and their descriptors (b) Calculating SIFT descriptor vector for each feature point.

complexity of the sensor nodes is decreased proportional to the number of unchosen $Sdsc^{l}_{n,m}(k)$'s. Moreover, the required sensor-central node bandwidth is decreased.

Ignoring some of the $Sdsc^{l}_{n,m}(k)$'s from the descriptor vector, $Sdsc_{n,m}(k)$, may cause mismatching of SIFT feature points of the input image and those in the database. For example, one of the SIFT points from a region in the input image which includes our object of interest may not be matched with any of SIFT points in the database. On the contrary, some of the SIFT points from background regions of the input image may be matched with some of SIFT points in the database.

Ignoring some of the $Sdsc^{l}_{n,m}(k)$'s from the descriptor vector can also be interpreted as a codeword reduction in a source coding. Specifically, having N^{SIFT}_{bit} bits to represent each of $Sdsc^{l}_{n,m}(k)$'s, the number of codewords (different SIFT feature descriptors) is $(2^{N^{SIFT}_{bit}})^{128}$. After selecting L number (out of 128) of $Sdsc^{l}_{n,m}(k)$'s,

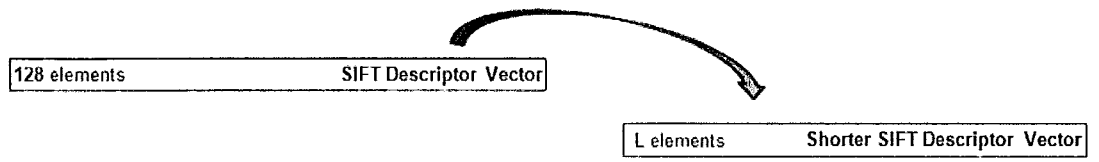


Figure 4.15: Shrinking SIFT descriptors' vector.

the number of possible codewords (different possible SIFT feature descriptors) is reduced to $(2^{N_{bit}^{SIFT}})^{L}$. Therefore, the selection of the $Sdsc_{n,m}^{l}(k)$'s should be done in an efficient way so that the SIFT object recognition and tracking is not affected significantly.

To select $Sdsc_{n,m}^{l}(k)$'s in an efficient way which does not affect the SIFT object recognition significantly, we have used the variance of each $Sdsc_{n,m}^{l}(k)$. Specifically, we calculate $\sigma_{SIFT_l^2}$ according to:

$$\begin{aligned} \sigma_{SIFT_l^2} &= E\{(\mathsf{Sdsc}^l - \eta_{Sdsc}^l)^2\} \\ \eta_{Sdsc}^l &= E\{\mathsf{Sdsc}^l\} \end{aligned} \tag{4.18}$$

where Sdsc^l is the random variable, representing all different outcomes of $Sdsc_{n,m}^{l}(k)$ for different n's, m's and k's. Using a small set of training samples (it can be done by calculating all $Sdsc_{n,m}^{l}(k)$'s for a small number of frames), the variance of each element of descriptor vector is obtained. This variance is a good criteria to select the Sdsc^l's with a better classification potential. More specifically, we select a number of $Sdsc_{n,m}^{l}(k)$'s with bigger $\sigma_{SIFT_l^2}$ and transmit them using a KLT and lossy encoding. The number of selected $Sdsc_{n,m}^{l}(k)$'s is also chosen in a way such that the SIFT object recognition is not affected significantly. An overview of the proposed drug bottle tracking using the modified SIFT features is shown in Fig. 4.16.

4.3.3 Color-based Drug Bottle Tracking

The need for a large database of different SIFT feature points to achieve a moderate tracking performance motivated us to develop a different approach for drug bottle tracking. Therefore, a color based object recognition and tracking method is

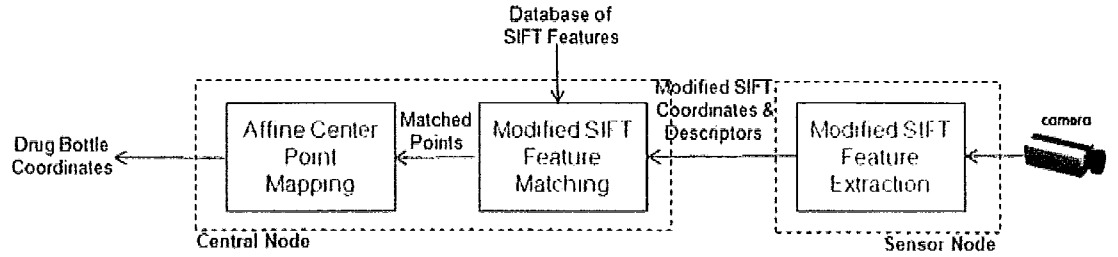


Figure 4.16: Proposed resource constrained SIFT drug bottle tracking.

adopted for drug bottles which does not require a large database. In this method, a special color pattern (shown in Fig. 4.17) is attached on the drug bottle to makes its recognition easier and more robust, although it may not be favorable in some cases. The process of extracting this color pattern with respect to the bandwidth and computational complexity constraints is described below.

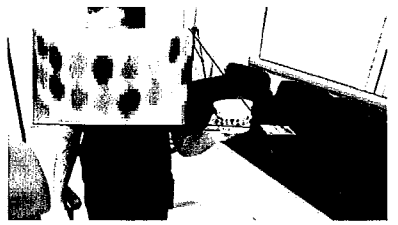


Figure 4.17: Used Color Pattern for Drug Bottle Recognition.

As shown in Fig. 4.17, the used pattern includes a combination of different color circles, *i.e.;* red-green-blue-..., which requires color filtering for their separation. However, this color filtering may be affected and one or more of the adopted color spots are not extracted, for some reason, *e.g.;* the shadowing effect. Therefore, a series of post-processing steps is done on the color filtered images to filter out the noisy connected pieces and mark the verified ones. A flow chart diagram of the proposed algorithm for extracting the color circles of the pattern is shown in Fig. 4.18(a).

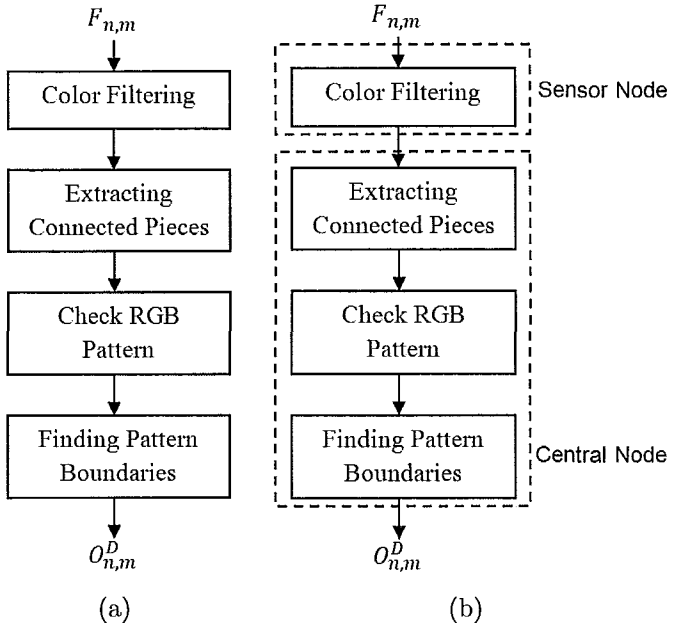


Figure 4.18: Color based Drug Bottle Recognition: (a) General Block Diagram (b) Resource Constrained Block Diagram.

The color filtering step, shown in Fig. 4.18, is a simple subtraction and thresholding. Specifically, each pixel RGB value of the input image,

$$\mathbf{F}_{n,m}(i,j) = (F^R_{n,m}(i,j), F^G_{n,m}(i,j), F^B_{n,m}(i,j)) \ ,$$

is compared with three different reference RGB values, each representing one color of the pattern. Representing each reference color RGB value by

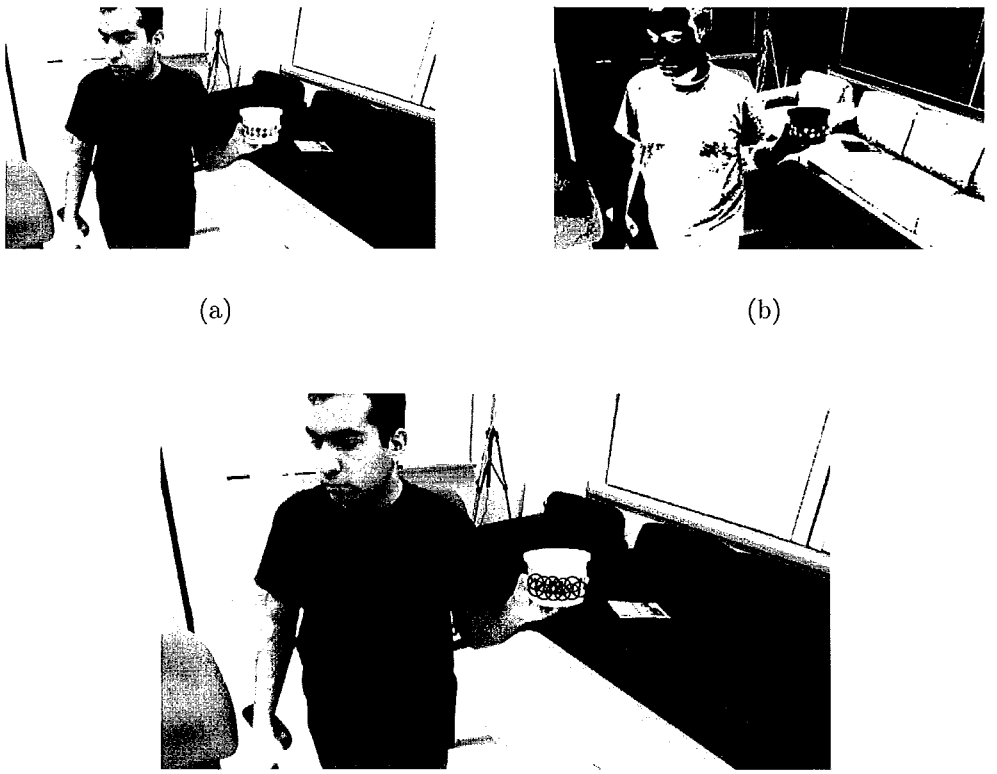
$$RGB_c = (RGB^R_c, RGB^G_c, RGB^B_c), \ c = R, G, B \ ,$$

the color filtered pixels, $F^{CF,c}_{n,m}(i,j)$, $c = R, G, B$, are calculated according to Eq. 4.19. Then, the reference color with the maximum filtered value is chosen for each pixel, according to Eq. 4.20. This selected color is then compared with a predetermined threshold to distinguish it from the background of the image. After thresholding, a gray image with only *four* different values, each representing one color (*i.e.;* three reference colors and one for the background), is obtained. This gray image is shown

in Fig. 4.19(b) which is obtained by applying the aforementioned color filtering on the input image of Fig. 4.19(a).

$$F_{n,m}^{CF,c}(i,j) = \frac{1}{\|\mathbf{F}_{n,m}(i,j) - RGB_c\|_2}, \quad c = R,G,B \tag{4.19}$$

$$F_{n,m}^{CF}(i,j) = max_c F_{n,m}^{CF,c}(i,j) \tag{4.20}$$



(a) (b)

(c)

Figure 4.19: An Example of Color based Drug Bottle Tracking: (a) Input Image (b) Color Filtered Image (c) Extracted Color Circles of the Input Image (black circles).

The color filtered images are then passed through a morphological step to calculate some geometrical properties of each connected part. This morphological step includes partitioning of the image into connected parts whose corresponding color does not

represent background. Then, each connected part is processed to obtain the following parameters:

- Real Surface: The number of pixels in the connected piece.
- Outer Surface: The multiplication of Width and Height of the connected piece.
- Center Coordinates: The center of mass of the pixels in the connected piece.

Having obtained this set of parameters for each connected piece, a number of geometrical constraints are forced to find out candidates of the color pattern circles. Specifically, the Width and Height of each connected piece should be almost the same, as they are circles of the same size. Moreover, we will remove some of the connected pieces with an unusual ratio of Real Surface and Outer Surface. Therefore, providing a noise margin for these set of constraints, we can remove pieces which do not belong to the pattern.

To verify that all of the remaining pieces belong to the color pattern, another set of geometrical constraints are examined between each triple of the connected pieces. Specifically, we pick two connected pieces of the same color, called CP_A and CP_B, and one from a different color, CP_C, as a triple. In each triple, the ratio of the distance between CP_C and each of CP_A and CP_B is compared with the distance between CP_A and CP_B as one of the refinement rules. The real surfaces of these three pieces are also compared with each other, as a different rule to verify pattern pieces. After this refinement step, the remaining pieces represent the circles of the color pattern and are used to obtain an outer boundary for the color pattern. The center of this boundary is used as the location of the drug bottle in the corresponding frame, $\mathbf{O}^D_{n,m}$.

In order to address the bandwidth and computational complexity constraints, the color filtered image is coded and transmitted to the central node. As a result, the huge computational load of morphological operation (*i.e.;* extracting connected piece parameters) and RGB pattern check is moved to the central node. Moreover, the bandwidth required for transmitting encoded color filtered images is significantly decreased, compared to the case where raw/compressed spatial domain data is transmitted. An overview of the proposed resource constrained scheme for color pattern recognition and tracking, distinguishing the parts in sensor and central nodes, is shown in Fig. 4.18(b). Temporal correlation of the consecutive frames is also taken

into account by decreasing the window size in which the color pattern recognition is performed, as the location of the color pattern in the previous frame is determined. As a result, the computational complexity of the whole system and the required sensor-central bandwidth are decreased.

4.4 Experimental Results

In our experiments, we have used the performance of object tracking as one of the evaluation parameters as well as the required sensor-central node bandwidth and the sensor computational complexity. We have implemented the proposed human body tracking, SIFT drug bottle tracking and color-based drug bottle tracking schemes and compared them with different scenarios in terms of the bandwidth and computational complexity. This resource comparison is done while keeping the tracking performance of the proposed method almost constant in different scenarios. The experimental evaluations are done on each of the proposed schemes in sections 4.4.1, 4.4.2 and 4.4.3. Then, these experimental results are analyzed in section 4.4.4.

To evaluate the performance of the proposed human body and drug bottle tracking schemes in terms of the allocated bandwidth and computational power of the sensors, we have used five different multi-view video sequences. Each of these video sequences include three different views, captured from the same scene, at the rate of 30 fps. Each frame of the multi-view video includes 640×480 pixels, each represented by 24 bits in RGB format (8 bits for each of Red, Green and Blue colors). In total, we have captured approximately 60 seconds for each view video sequence, which will provide 1800 frames in each view. A three-view sample frame of one of the used video sequences is shown in Fig. 4.20.

The computational evaluation of the proposed schemes was done by measuring the execution times of sensor nodes in each scenario. Specifically, the execution times are measured for the time the sensor node tasks are performed on a 2.4 GHz Pentium 4 Xeon Central Processing Unit (CPU) with 1 GBytes of memory. It can be stated that the computational load of sensor nodes in each scenario is proportional with the corresponding execution time on this CPU.

(a) (b) (c)

Figure 4.20: A three-view sample frame of test video sequence.

4.4.1 Human Body Tracking

The proposed human body tracking is applied for all the captured video sequences and the required bandwidth for transmitting the foreground map is calculated by averaging over all frames. The computational complexity of the sensor nodes is measured in terms of the running time on a single processor unit, calculated by averaging over all frame of different views.

The proposed scheme is compared with different scenarios in which the human body tracking has almost the same performance. First, the human body tracking is considered to be at the central node which requires transmission of all spatial domain data to the central node. We will examine spatial domain data (8 bit gray frames) transmission, both using H.264/AVC coding and raw data transmission. These two different scenarios are called *center-H264* and *center-raw*, respectively. In a so called *sensor* scenario, the human body tracking is performed on the sensor node, including foreground extraction and articulated model matching and the resulting coordinates of hands and mouth (head center) are transmitted to the central node. Similarly, the computational power of the sensor nodes is measured in terms of the average running time per frame.

The tracking performance is calculated in terms of the Mean Square Error (MSE) of the extracted trajectory from the real one, which is extracted manually. The tracking performance should be the same for all different scenarios in order to compare their allocated resources, *i.e.;* sensor-central node bandwidth and computational power of the sensor nodes. This can be done as the tracking blocks are working independently

Figure 4.21: An example of human body part recognition: In this case, the articulated body model is changes so that the positions of connecting points between the arms and the body match with the physical characteristics of the human body.

and the rate in which the frames are transmitted is chosen so that the decoded frames at the central node have enough quality for providing the same tracking performance. H.264/AVC coding of the video sequence of each camera is done using its open-source implementation, available in [54].

Table 4.1: Experimental Results on Human Body Tracking.

	Tracking Precision (MSE)	Sensor Running Time [sec]	Average Rate per sensor [kbps]
Center-H264	52.3	10.243	1354.9
Center-raw	52.3	~ 0	72,000
Sensor	52.3	718.2701	1.8
Proposed Scheme	52.3	0.4922	116.6

In Table 4.1, the experimental results for the proposed human tracking scheme as well as *Center-H264*, *Center-raw* and *Sensor* are presented. In the *sensor* scenario, a huge amount of computation has to be performed on each sensor node which is completely undesired in practice, although it requires a very low rate communication

bandwidth. In the case of *Central-raw*, the sensor computational complexity is almost zero which results in a huge communication bandwidth among the sensor nodes and the central node. The required bandwidth is reduced 50 times by applying a conventional H.264/AVC coding on the frames. However, the computational complexity of the sensor nodes is high, compared to that of the proposed scheme. Moreover, we could achieve a reasonably lower communication bandwidth than H.264/AVC coding in our scheme. Although it is not a right statement to say the proposed method is definitely the best option among others until the importance of each of the bandwidth and computational resources are not quantified. However, it can be stated that in usual cases in which one of these two factors (*i.e.;* bandwidth or computational complexity) is not much more important that the other, the proposed method is the best option in terms of the required resources. This conclusion can be obtained by comparing the required communication bandwidth and running time of the sensors, simultaneously.

4.4.2 SIFT Drug Bottle Tracking

Similarly, three different scenarios are adopted in this case to be compared with the proposed SIFT drug bottle tracking scheme. In *Center-H264* scenario, the frames are transmitted to the central node after H.264/AVC coding at the sensor node. Then, the SIFT features of the decoded frames are extracted and SIFT object recognition is performed. In this case, the tracking performance is measured in terms of the number of matched SIFT feature points among the input frames and those of the database. As a result, the more matches are found, the better tracking performance would be. Likewise, the sensor node computational complexity is measured in terms of the average running time of the sensor nodes on the same processor unit, used for human body tracking.

The frames of video sequences are transmitted without any coding, in the *Center-raw* scenario, in which the sensor nodes only convert the input RGB frames to the luminance frames. Then, the received frames at the central node will be processed for human body and drug bottle recognition and tracking. In *Center-H264*, the captured gray frames are H.264/AVC coded and transmitted to the central node. Then, the received stream is decoded to reconstruct the gray frames and perform

object recognition and tracking on the reconstructed frames. In *Sensor* scenario, all SIFT object recognition and tracking is done at the sensor nodes and the resulting coordinates of the drug bottle are transmitted to the central node. The proposed method is finally applied on the luminance frames, forcing it to have the same number of matched SIFT features points as the other scenarios.

Table 4.2: Experimental Results on SIFT Drug Bottle Tracking.

	No of Matched SIFT Points	Sensor Running Time [sec]	Average Bandwidth Rate [kbps]
Center-H264	11	10.243	1354.9
Center-raw	11	~ 0	72,000
Sensor	11	1.6016	0.6
Proposed Scheme	10	0.0124	125.7

The resulting experimental statistics for different scenarios are presented in Table 4.2. In cases of *Center-H264*, *Center-raw* and *Sensor* scenarios, the number of matched SIFT points can be easily managed to be the same. As we decrease the size of SIFT descriptor vectors in the proposed scheme, the number of correct matched points is also decreased, in the same situation as other scenarios. Therefore, we changed the number of extracted SIFT features in database templates to get an almost the same number of matched SIFT points, as presented in Table 4.2. It could be understood from the data of Table 4.2 that the *Center-H264* scenario is not a good choice for SIFT object recognition and tracking as requires a lot of bandwidth and computational power resources. The advantage of our scheme compared to the *Sensor* scenario is hard to understand as one is better in terms of the bandwidth and the other in terms of the computational complexity of the sensors. In section 4.4.4, the overall comparison of the proposed methods with the *Sensor* scenario can provide us with a better conclusion.

4.4.3 Color based Drug Bottle Tracking

In section 4.4.2, the drug bottle tracking using the SIFT features was compared with different scenarios in terms of the allocated resources. In this section, we will examine

the proposed color based drug bottle tracking scheme both in terms of the allocated resources and tracking performance. Similar to the SIFT drug bottle tracking, we will compare *Center-H264*, *Center-raw* and *Sensor* scenarios with the proposed scheme. In this case, the tracking performance is measured in terms of the MSE of tracking. Moreover, we will compare the tracking performance of the color based tracking with the proposed SIFT drug bottle tracking in terms of the MSE of tracking to understand how much improvement can be achieved by putting a special color pattern on the drug bottle. Likewise, the real trajectory of the drug bottle is extracted manually and then compared with the extracted trajectories of each different method to calculate the MSE values.

As the color frames are processed in this case, the original color frames are H.264/AVC coded in *Center-H264* scenario and transmitted to the central node. The decoded color frames at the central node are then processed according to the flow chart of Fig. 4.18(a). The same procedure is done in *Center-raw* scenario while no compression is performed at the sensor nodes and raw color frames are transmitted losslessly to the central node. In *Sensor* scenario, all the process of color circle extraction and drug bottle recognition is performed at the sensor node and the resulting coordinates are transmitted to the central node. Both color based and SIFT drug bottle tracking schemes are performed exactly as described in sections 4.3.2 and 4.3.3.

Table 4.3: Experimental Results on Color based Drug Bottle Tracking.

	Tracking Precision (MSE)	Sensor Running Time [sec]	Average Bandwidth Rate [kbps]
Center-H264	12.68	27.657	3483.9
Center-raw	12.68	~ 0	216,000
Sensor	12.68	6.84	0.6
Proposed SIFT Tracking	13.97	0.0201	163.9
Proposed Color Tracking	12.68	2.93	663.5

To be able to have a better comparison among different scenarios, we managed to have almost the same tracking performance for them, as presented in Table 4.3.

Similar to the previous parts, this can be easily done for *Center-H264*, *Center-raw*, *Sensor* and the proposed color based tracking schemes. To change the SIFT tracking performance and get almost the same precision as other scenarios, we changed the number of SIFT feature points, extracted from the input frames.

In Table 4.3, the resulting parameters of our experiments are listed. Similar to the case of SIFT object tracking, the *Center-H264* scenario is not a good choice both in terms of communication bandwidth and the computational complexity of the sensor nodes. As in the *Center-raw* scenario, the required bandwidth is significantly high and it is not a good choice. In terms of the allocated resources, it can be seen that the proposed SIFT tracking method is better than the color based tracking, having sacrificed a little bit of the tracking performance. However, it should be noted that the SIFT tracking in this case requires a huge database of different templates to get an acceptable number of matched SIFT points and reasonable tracking performance.

4.4.4 Overall Analysis of Experimental Results

In the previous subsections of section 4.4, the allocated resources of each part of human body and drug bottle tracking schemes have been evaluated in comparison with different scenarios. In this section, we will compare both human body and drug bottle schemes with *Center-H264*, *Center-raw* and *Sensor* scenarios in terms of the allocated resources. Specifically, the total cost of human body and drug bottle tracking corresponding to the tracking performance, sensor-central node bandwidth and the sensor node computational complexity are calculated in each scenario. Similar to the previous cases, we provide them with almost the same tracking performance and compare them in terms of the allocated resources. The total tracking performance is defined as the sum of mean square errors, corresponding to the human body and drug bottle tracking.

In this section, we consider that the captured frames are processed and transmitted in both **gray** and **color** modes. In the *color mode*, all the RGB information of the pixels are used in extracting the trajectories of the objects of interest. Specifically, the RGB frames are transmitted with and without H.264/AVC coding in *Center-H264* and *Center-raw* scenarios. The received spatial domain RGB data are then processed for human body tracking using the gray frames and color based drug bottle tracking

using the received RGB frames. In the *Sensor* scenario, the RGB frames are used for color based drug bottle tracking while the gray frames are used for human body tracking in parallel. We will not perform SIFT drug bottle tracking in **color mode**. In our scheme, we will extract the foreground map as well as the color filtered frames at the sensor node and transmit them to the center node. As described in section 4.2, the central node is in charge of the remaining steps of trajectory extraction. The overall experimental results of the **color mode** are presented in Table 4.4.

In the **gray mode**, all the captured RGB frames are converted into gray frames at the sensor node. Then, in each of *Center-H264*, *Center-raw*, *Sensor* scenarios, the corresponding procedure (described in sections 4.4.1 and 4.4.2) is adopted. Different scenarios which are implemented in both **gray** and **color** modes of our experiments for our experimental comparisons are listed below:

- *Center-raw:* The raw frames are transmitted to the central nodes without any coding. The central node is in charge of performing all the event detection blocks.

- *Center-H264:* The raw frames are H.264 encoded and transmitted to the central node. The central node will decode the received stream and perform all the event detection blocks.

- *Sensor:* All the event detection blocks are performed at the sensor nodes to find the marginal decisions. The calculated marginal decision is then transmitted to the central node.

- *Proposed Scheme:* The extracted features, used for tracking of human body and drug bottle, are transmitted to the central node. The remaining event detection blocks are performed at the central node by using the received features.

As presented in Table 4.4, *Center-H264* scenario is still the worst choice both in terms of the sensor computational complexity and the sensor-central node bandwidth. *Center-raw* scenario is also not a good choice because it requires a huge communication bandwidth although it does not need any computational power at the sensor nodes. Similarly, the *Sensor* scenario suffers a heavy computational load at the sensor

Table 4.4: Analytical Experimental Results in **Color Mode**.

	Tracking Precision (MSE)	Sensor Running Time [sec]	Average Bandwidth Rate [kbps]
Center-H264	64.98	27.657	3483.9
Center-raw	64.98	~ 0	216,000
Sensor	64.98	725.11	2.4
Proposed Scheme	64.98	3.42	780.1

nodes which results to a very low rate communication bandwidth. The advantage of the proposed scheme over the other scenarios is its low communication and computational resources. However, *Center-raw* or *Sensor* scenarios may be the best option in cases where one of the bandwidth or computational resources is much more important than the other. In such cases, the proposed method may use more computational or bandwidth resources than *Center-raw* or *Sensor* scenarios. In practical cases, the best scenario can be determined by considering the import factors of bandwidth and computational resources, λ_{BW} and λ_{CO}. Specifically, the total cost, $\mathbf{C}_{total}$, associated with each scenario is calculated according to Eq. 1.7. It can be stated that in usual cases where bandwidth and computational resources have approximately the same importance, the proposed scheme is significantly better than other scenarios.

Table 4.5: Analytical Experimental Results in **Gray Mode**.

	Tracking Precision (MSE)	Sensor Running Time [sec]	Average Bandwidth Rate [kbps]
Center-H264	65.92	10.243	1354.9
Center-raw	65.92	~ 0	72,000
Sensor	65.92	719.87	2.4
Proposed Scheme	67.42	0.5046	242.3

In Table 4.5, the resulting parameters of **gray mode** are presented. Obviously, each of the *Center-raw* and *Sensor* scenarios are inefficient in terms of the communication bandwidth or the computational load of the sensors. Moreover, in the *Center-H264* scenario, neither of the sensor-central node bandwidth and the sensor computational power are the lowest among the other schemes. Similar to the **color**

mode, the best option can only be determined by considering the import factors of bandwidth and computational resources. This can be achieved by considering the implementation issues for both cases of the communication and computational power. However, it can be stated that the proposed in-sensor feature extraction scheme for human body and drug bottle tracking is generally the best option in cases where bandwidth and computational resources have almost the same importance.

4.4.5 Color Pattern for Drug Bottle Tracking

As discussed in section 4.2.2, the tracking performance of SIFT drug bottle tracking depends on the number of SIFT feature points which is matched between the input frames and the database. Therefore, providing a large amount of templates in the database to be used for SIFT object recognition will increase the possibility of finding more matched SIFT points for different input frames. However, this may not be feasible in practical cases because of resource limitations in the central node. To address this issue, a special pattern (proposed and discussed in section 4.3.3) can be attached to the drug bottles to make its recognition easier. More specifically, the attached pattern makes the drug bottle tracking more reliable and robust to different changes in the input frames.

The proposed color pattern for recognition of the drug bottles, discussed in section 4.3.3, has made us able not to need a large amount of templates in the database. However, it has some drawbacks as mentioned in the following. One of the drawbacks of the used color pattern arises when a number of different drug bottles are taken by the elderly person and it is essential to distinguish them from each other. Moreover, using the proposed series of red, green and blue circles as the landmarks for the recognition of the drug bottle is sensitive to some situations in which the background scene has also similar colors or circle patterns.

To address these issues, we can use a more complex color pattern which includes more types of landmarks for its recognition in complex scenes. For instance, the RGB color pattern (proposed in section 4.3.3) can be used as the boundaries of a whole color pattern which includes a special checker board pattern. In summary, using different simple patterns to build up a more complex pattern can be useful for increasing the reliability and robustness of drug bottle recognition in different scenes.

Furthermore, a similar color filtering step is capable of extracting the feature points of the input frames, used for drug bottle tracking. As a result, using a combination of different simple patterns for the drug bottle tracking will not require more bandwidth or computational resources than the case where a simple RGB pattern was used.

Chapter 5

Conclusions and Future Work

Medical smart homes are becoming increasingly important in providing a safe living environment for elderly persons. Such homes should have the capability to monitor the vital signs and daily activities of its occupant(s) using efficient, unobtrusive and low-cost technologies. In this thesis, the important task of event detection applied to taking medication, is studied. In contrast to other published works which concentrates on the reliability and detection performance of the system, we focus on reducing the allocated resources for this task for improved real-time applications, but without compromising the system's performance.

The event detection task is done by using a multi-camera system with a central processing unit. Each camera node has a local processor which extracts the useful information and transmits them to the central node for event detection. In such a scenario, it is desired to decrease the allocated resources (sensor-central node bandwidth and computational complexity of the sensor nodes) while the detection performance is not affected significantly. To address this goal, we have come up with two different approaches: first approach is based on the concept of multi-view video coding and the second approach distributes the steps of event detection between the sensor and central nodes efficiently.

In the first approach, we transmit the captured video sequences by using an efficient multi-view video coding scheme to the central node. Then, the central node is in charge of decoding the received stream and making decision about the occurrence of the event. In this approach, we studied a configuration of cameras in which they

are arranged tightly so that the captured frames of adjacent cameras are correlated. In such system, the inter-view similarity of the frames can increase the compression ratio in a joint encoding and decoding scenario. In other words, a bandwidth saving can be achieved if we can transmit the multi-view video by taking advantage of this inter-view similarity. Although the same rate-distortion performance is theoretically expected for separate encoding and joint decoding, it has not yet been practically achieved. We proposed a classification based feedback-assisted scheme which achieves a significantly better performance than separate H.264/AVC coding in cases where inter-sensor communication is not feasible. Our proposed multi-view video coding scheme outperforms by almost 1 dB the separate H.264 encoding (around 17% bit saving) in terms of rate-distortion performance while the computational complexity of the encoders (sensor nodes) is not increased. This can be interpreted as a 17% reduction in the required bandwidth for the same quality of transmitted frames (the same event detection performance).

In the second approach, our problem is addressed by studying the blocks used in the event detection. Specifically, we have analyzed the computational complexity of each step of event detection. Different steps of event detection are then assigned to the sensor or the central nodes in a way such that the computational and bandwidth constraints are relatively satisfied. More specifically, the useful small-size information for event detection task that do not require a high computational power for their calculation, are extracted at the sensor nodes. The experimental results show a trade-off between the allocated bandwidth and computational resources. In other words, the proposed scheme may not be the best option in cases where one of the bandwidth or computational resources is much more important than the other. However, a significant amount of (a reduction of 90% in some cases) bandwidth and computational power savings can be achieved, in usual cases where bandwidth and computational resources have almost the same importance. Therefore, the best option can only be determined by considering the import factors of bandwidth and computational resources.

5.1 Recommendations for Future Work

Improvement of the proposed event detection scenario should be considered. In the first approach where multi-terminal video coding of captured sequences was studied, a low rate feedback channel is required for transmitting the classification results of blocks. Although using this feedback channel is essential for transmission of these classification bits, it may not be feasible in some of the practical cases. Therefore, it can be a very good step to develop a blind classification to be used for multi-terminal video coding of captured video input.

In our work, we are considering the fusion of marginal decisions of each sensor, *i.e.;* $\hat{a}_m(n)$'s, to detect the occurrence of the event and estimate $\hat{a}(n)$. As an improvement to our work, one may try to fuse the data during the tracking of the objects rather than fusing the marginal decisions. More specifically, using a calibrated or un-calibrated camera system, one may use the correspondence between the objects in different camera frames. Such correspondence can increase the tracking performance of the multi-camera system and consequently, the overall performance of event detection is improved. Furthermore, the correlation among different view frames may be useful to slightly reduce the resources used for transmit the information from the sensor nodes to the central node. This can be done by studying the correlation among the extracted features of different views and distributed coding of them.

An improvement in the required bandwidth can be achieved by considering the correlation among the extracted features in different sensor nodes. Specifically, the correlation between different view frames results in a correlation between the extracted foreground map (used for human body tracking). Therefore, using the distributed source coding for transmission of this foreground map can possibly save a significant amount of bandwidth resources in addition to what we achieved in the second approach.

Although using different type of sensors for detection of events in the smart home may increase the implementation cost, it will potentially increase the reliability of the system. In fact, the data captured by a number of vision sensors are fused to detect the occurrence of a single event, in our scenario. Fusing the data captured from a different type of sensor (like RF tags and its sensors) with the vision data can increase the detection performance of the system.

Finally, it should be noted that the event detection process is done after the raw frames are sensed at the sensor nodes. In other words, we need to store (buffer) the spatial domain data at the sensor nodes to extract the useful features from them. Computational resources are inefficiently used because of sensing the spatial domain data rather than the features. Therefore, a better design in terms of computational load of sensor nodes can be obtained by modifying the sensing device to capture the features, used for tracking, instead of sensing the raw spatial domain data.

Appendix A

Mathematical Relation between the number of SW bits and classification threshold, ΔDC_{TH}

Considering the proposed Slepian-Wolf (SW) coding scheme for blocks which are marked to be SW coded, we derive the mathematical relation of the I frame block classification threshold, $\Delta_{DC_{TH}}$, in this appendix. Assume $[A(i,j)]$ and $[B(i,j)]$ where $1 \leq i,j \leq 8$ are the block to be SW coded and its side information, respectively. Therefore, the classification criteria of Eq. 3.5 is as follows:

$$\Delta_{DC} = \frac{1}{64} \sum_{1 \leq i,j \leq 8} [A(i,j) - B(i,j)]^2 \tag{A.1}$$

Representing the DCT coefficients of $[A(i,j)]$ and $[B(i,j)]$ by $[A^{DCT}(i,j)]$ and $[B^{DCT}(i,j)]$, and using Parswal equality, we obtain:

$$\begin{aligned} \Delta_{DC} &= \tfrac{1}{64} \textstyle\sum_{1 \leq i,j \leq 8} [A(i,j) - B(i,j)]^2 \\ &= \tfrac{\alpha}{64} \textstyle\sum_{1 \leq i,j \leq 8} [A^{DCT}(i,j) - B^{DCT}(i,j)]^2 \end{aligned} \tag{A.2}$$

where α is a constant. Neglecting the quantization error associated with the quantization of Eq. 3.15, Δ_{DC} can be approximated as follows:

$$\begin{aligned} \Delta_{DC} &\approx \\ &\frac{\alpha}{64} \sum_{1 \leq i,j \leq 8} [A^{DCT,Q}(i,j) - B^{DCT,Q}(i,j)]^2 \cdot Q^2(i,j) \end{aligned} \tag{A.3}$$

where $A^{DCT,Q}(i,j)$'s and $B^{DCT,Q}(i,j)$'s are the quantized DCT coefficients, calculated according to Eq. 3.15.

For an error free SW encoding/decoding (using the proposed scheme of section 3.1.5), the values to be SW encoded, $A^{DCT,Q}(i,j)$, should fall into a bounded interval around the corresponding side information, $B^{DCT,Q}(i,j)$. Without loss of generality, such constraint can be formulated as in Eq. A.4:

$$|A^{DCT,Q}(i,j) - B^{DCT,Q}(i,j)| \leq 2^{sb-1}, \quad {}_{1 \leq i,j \leq 8} \tag{A.4}$$

Assuming the case in which all the $A^{DCT,Q}(i,j)$'s lay on the boundaries of the constraint of Eq. A.4, the criteria of Eq. 3.5 can be re-written in the following form:

$$\Delta_{DC} \lesseqgtr \Delta_{DC_{TH}} \tag{A.5}$$

where $\Delta_{DC_{TH}}$ is obtained as follows:

$$\begin{aligned} \Delta_{DC_{TH}} &\approx \frac{\alpha}{64} \sum_{1 \leq i,j \leq 8} \{2^{sb-1}.Q(i,j)\}^2 \\ &= \frac{\alpha.2^{sb-1}}{64} \sum_{1 \leq i,j \leq 8} Q^2(i,j) \end{aligned} \tag{A.6}$$

It should be noted that the derived threshold is an approximation.

Bibliography

[1] U. N. Population Division, DESA, "World population ageing 1950-2050: Executive summary," January 2010. [online] Available: http://www.un.org/esa/population/publications/.

[2] R. E. Roush, "Aging populations." [online] Available: http://www.thehormoneshop.com/A/aging.htm.

[3] M. Porter, "How falls can fracture lives," 2009. Times Online Magazine [online] Available: http://www.timesonline.co.uk/tol/life_and_style/health/article6538618.ece.

[4] Dehydration and Water Balance [online]. Available: http://www.merck.com/mmhe/sec12/ch158/ch158b.html.

[5] Helping older people to take prescribed medication in their own home [online]. Available: http://www.scie.org.uk/publications/briefings/files/briefing15.pdf.

[6] University of Rochester Smart Home Research Laboratory [online]. Available: http://www.centerforfuturehealth.org/smart_home/Smart_home.html.

[7] M. Chan, D. Estve, C. Escriba, and E. Campo, "A review of smart homes–present state and future challenges," *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55 – 81, 2008.

[8] Intelligent Assistive Technology and Systems, Toronto Rehabilitation Centre [online]. Available: http://www.ot.utoronto.ca/iatsl/.

[9] University of Sherbrooke DOMUS Laboratory [online]. Available: http://domus.usherbrooke.ca/.

[10] SFU Living Lab [online]. Available: http://www.sfu.ca/livinglab/.

[11] Y. Chen, M. Hannuksela, L. Zhu, A. Hallapuro, M. Gabbouj, and H. Li, "Coding techniques in multiview video coding and joint multiview video model," in *Picture Coding Symposium, 2009. PCS 2009*, pp. 1 –4, 6-8 2009.

[12] Y. Yang, V. Stankovic, W. Zhao, and Z. Xiong, "Multiterminal video coding," in *IEEE International Conference on Image Processing*, vol. 3, pp. III–25–III–28, September 2007.

[13] Y. Yang, V. Stankovic, Z. Xiong, and Z. Wei, "Two-terminal video coding," *IEEE Transactions on Image Processing*, vol. 18, pp. 534–551, March 2009.

[14] MIT House_n Project [online]. Available: http://architecture.mit.edu/house_n/.

[15] MIT PlaceLab [online]. Available: http://architecture.mit.edu/house_n/placelab.html.

[16] Medical Automation Research Center at UVA [online]. Available: http://marc.med.virginia.edu/.

[17] M. Alwan, S. Dalal, S. Kell, and R. Felder, "Derivation of basic human gait characteristics from floor vibrations," in *2003 Summer Bioengineering Conference*, June 2003.

[18] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," in *International Conference on Information and Communication Technologies: from Theory to Applications*, pp. 23–28, April 2006.

[19] A. Arcelus, M. H. Jones, R. Goubran, and F. Knoefel, "Integration of smart home technologies in a health monitoring system for the elderly," in *Advanced Information Networking and Applications Workshops, International Conference on*, vol. 2, (Los Alamitos, CA, USA), pp. 820–825, IEEE Computer Society, 2007.

[20] Technology Assisted Friendly Environment for the Third Age (TAFETA) Smart Technology [online]. Available: http://www.tafeta.ca/.

[21] M. Chan, E. Campo, D. Estve, and J.-Y. Fourniols, "Smart homes – current features and future perspectives," *Maturitas*, vol. 64, no. 2, pp. 90 – 97, 2009.

[22] MPEG Working Documents [online]. Available: http://mpeg.chiariglione.org/working_documents.htm.

[23] Introduction to Multiview Video Coding [online]. Available: http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-mvc/index.htm.

[24] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceeding of IEEE*, vol. 93, pp. 71–83, January 2005.

[25] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, pp. 471–480, July 1973.

[26] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, pp. 1–10, January 1976.

[27] X. Guo, Y. Lu, F. Wu, D. Zhao, and W. Gao, "Wynerziv-based multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp. 713–724, June 2008.

[28] D. Gavrila, J. Giebel, and S. Munder, "Vision-based pedestrian detection: the protector system," in *IEEE Symposium on Intelligent Vehicles*, pp. 13 – 18, 14-17 2004.

[29] H. Zhou and D. Kimber, "Unusual event detection via multi-camera video mining," in *18th International Conference on Pattern Recognition*, vol. 3, pp. 1161 –1166, 0-0 2006.

[30] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Semi-supervised adapted hmms for unusual event detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 611 – 618, June 2005.

[31] X. Li and F. Porikli, "A hidden markov model framework for traffic event detection using video features," in *International Conference on Image Processing*, vol. 5, pp. 2901 – 2904, October 2004.

[32] S. Hongeng and R. Nevatia, "Large-scale event detection using semi-hidden markov models," in *The Ninth IEEE International Conference on Computer Vision*, (Washington, DC, USA), p. 1455, IEEE Computer Society, 2003.

[33] M.-C. Tien, Y.-T. Wang, C.-W. Chou, K.-Y. Hsieh, W.-T. Chu, and J.-L. Wu, "Event detection in tennis matches based on video data mining," in *IEEE International Conference on Multimedia and Expo*, pp. 1477 –1480, June 2008.

[34] S. M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," pp. 133–146, 2006.

[35] G. Mohammadi, F. Dufaux, T. H. Minh, and T. Ebrahimi, "Multi-view video segmentation and tracking for video surveillance," in *SPIE Mobile Multimedia/Image Processing, Security and Applications*, April 2009.

[36] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," *IEEE Workshop on Motion and Video Computing*, pp. 169 – 174, December 2002.

[37] T. Berger, "Multi-terminal source coding," *The Information Theory Approach to Communications*, 1997.

[38] Y. Oohama, "Gaussian multiterminal source coding," *Information Theory, IEEE Transactions on*, vol. 43, pp. 1912–1923, Nov 1997.

[39] Y. Oohama, "Rate-distortion theory for gaussian multiterminal source coding systems with several side informations at the decoder," *Information Theory, IEEE Transactions on*, vol. 51, pp. 2577–2593, July 2005.

[40] S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *Signal Processing Magazine, IEEE*, vol. 19, pp. 51 –60, mar 2002.

[41] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," *IEEE Transactions on Information Theory*, vol. 49, pp. 626 – 643, March 2003.

[42] S. Pradhan and K. Ramchandran, "Generalized coset codes for distributed binning," *Information Theory, IEEE Transactions on*, vol. 51, pp. 3457–3474, Oct. 2005.

[43] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, pp. 80–94, September 2004.

[44] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "On code design for the slepian-wolf problem and lossless multiterminal networks," *IEEE Transactions on Information Theory*, vol. 52, pp. 1495–1507, April 2006.

[45] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On multiterminal source code design," *Information Theory, IEEE Transactions on*, vol. 54, pp. 2278–2302, May 2008.

[46] Joint Multiview Video Model (JMVM) Reference Software [online]. Available: http://ftp3.itu.int/av-arch/jvt-site/2006_10_Hangzhou/JVT-U207.zip.

[47] R. Puri and K. Ramchandran, "Prism: a new robust video coding architecture based on distributed compression principles," in *Allerton Conference on Communication, Control and Computing*, October 2002.

[48] M. Flierl and B. Girod, "Multiview video compression," *IEEE Signal Processing Magazine*, vol. 24, pp. 66–76, November 2007.

[49] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1474 – 1484, November 2007.

[50] P. Ferre, D. Agrafiotis, and D. Bull, "Fusion methods for side information generation in multi-view distributed video coding systems," in *IEEE International Conference on Image Processing*, vol. 6, pp. VI–409–VI–412, October 2007.

[51] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Fusion-based Multiview Distributed Video Coding," in *4th ACM International Workshop on Video Surveillance and Sensor Networks*, Lecture Notes in Computer Science, IEEE, 2006.

[52] K. Sayood, *Introduction to Data Compression.* Morgan Kaufmann Publishers, third ed., 2006.

[53] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *International Journal of Computer Vision*, vol. 70, no. 1, pp. 41–54, 2006.

[54] H.264/AVC Reference Software (JM73) [online]. Available: http://iphome.hhi.de/suehring/tml/download/old_jm/jm73.zip.

[55] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257 –286, Feb 1989.

[56] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[57] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computer Survey*, vol. 38, no. 4, pp. 1–45, 2006.

[58] A. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Object detection, tracking and recognition for multiple smart cameras," *Proceedings of the IEEE*, vol. 96, pp. 1606 –1624, October 2008.

[59] F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, (Los Alamitos, CA, USA), IEEE Computer Society, March 2003.

[60] Y. Dedeoglu, "Moving object detection, tracking and classification for smart video surveillance," Master's thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey, August 2004.

[61] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 246–252, 1999.

[62] R. Y. D. Xu and M. Kemp, "An iterative approach for fitting multiple connected ellipse structure to silhouette," *Pattern Recognition Letters*, vol. In Press, Corrected Proof.

[63] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[64] D. G. Lowe, "Object recognition from local scale-invariant features," in *International Conference on Computer Vision*, pp. 1150–1157, September 1999.