RESOURCE-EFFICIENT PORTABLE VIDEO COMMUNICATION SYSTEM DESIGN FOR WILDLIFE MONITORING AND INTERACTION TRACKING

A Dissertation presented to the Faculty of the Graduate School

University of Missouri - Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

---

by

Xiwen Zhao

Dr. Zhihai He, Dissertation Supervisor

DECEMBER 2011

The undersigned, appointed by the dean of the Graduate School, have examined the dissertation entitled

RESOURCE-EFFICIENT PORTABLE VIDEO COMMUNICATION SYSTEM DESIGN FOR WILDLIFE MONITORING AND INTERACTION TRACKING

presented by Xiwen Zhao,

a candidate for the degree of doctor of philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Dr. Zhihai He

Dr. James Keller

Dr. Dominic Ho

Dr. Wenjun Zeng

ACKNOWLEGMENTS

I deeply thank my supervisor, Dr. Zhihai He, for his support and guidance throughout my five-year journey of PhD study. His amiability, integrity and enthusiasm had a profound impact not only on my academic study but also on my personal life.

I would also like to express my deep gratitude to Dr. James Keller, Dr. Dominic Ho and Dr. Wenjun Zeng to serve as members in my doctoral committee and provide insightful suggestions and supervision for my research. They provide guidance by making great examples of work ethics and passion for perfection.

I would to extend my appreciation to my colleagues, Jay Eggert, York Chung, Wenqing Dai, Zhongna Zhou, Xin Li, Xi Chen and Li Liu, for their help and friendship.

A special thank goes to Dr. Tony Han, Betty Barfield and Shirley Holdmeier for their invaluable help.

Last but not least, to my beloved wife, daughter and son, and my mom, for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

ABSTRACT

In this research, we focus on algorithm development and system design for resource-efficient portable video communication system design and their application in wildlife monitoring and interaction tracking. The capability of seeing what an animal sees in the field is very important for wildlife activity monitoring and research. We design an integrated video and sensor system, called *DeerCam* and mount it on free-ranging animals so as to collect important video and sensor data about their activities in the field. From the video and sensor data collected by DeerCam, wildlife researchers will be able to extract a wealth of sciatic data for studying the behavior patterns of wildlife species and understanding the dynamic of wildlife systems.

In this dissertation, we focus on the following four tightly coupled research issues:

(1) ***Energy minimization***. Video compression is computationally intensive and energy-consuming. However, portable video communication devices for mobile video monitoring, especially those in wildlife monitoring and environmental tracking, are often small in size and light in weight. They have limited energy supply for data processing. One of the central challenging issues in portable video communication system design is to minimize the energy consumption of video compression so as to extend the operational lifetime of devices. In this research, we develop joint power-rate-distortion (P-R-D) methods and algorithms for complexity control and energy minimization of portable video encoders. We demonstrate that, given a video encoder, which has already been fully optimized using existing software and hardware techniques, we can further reduce its energy consumption significantly using P-R-D.

(2) ***Intelligent resource allocation and utility maximization***. The objective of video-based wildlife monitoring is to collect important visual information about animals' activities in the field for behavior modeling and other wildlife research tasks. Therefore, the overall system performance should be measured by the utility of video data collected by the DeerCam system for wildlife research purposes (e.g. behavior modeling). In this research, we develop methods to maximize the utility function under resource constraints.

(3) ***Efficient image encoder***. Because of animal motion, the video samples captured by the animal-mounted DeerCam system often suffer from dramatic motion and content change. In this case, a significant amount of video frames and image regions are encoded with the INTRA mode. Furthermore, the image data often has a significant amount of high-frequency structural components, such as trees and grasses. How to develop an image / video compression scheme to efficiently represent and encode structural components becomes an important problem in our research. To address this issue, we explore various approaches, including local structure prediction to efficiently learn, predict, represent, and encode local image structures, super-spatial structure prediction to find an optimal prediction of structure components within the previously encoded image regions, and inter-structure prediction to find an optimal prediction of structure components within the encoded structure components. Our extensive experimental results demonstrate that the proposed methods are very competitive and even outperform the state-of-the-art image compression methods.

(4) ***Animal interaction detection for event-driven wildlife monitoring***. The inter- and intra-specific interaction of wildlife animals is one of the most interesting activities to wildlife researchers. Video accounts of interactions could aid in disease transmission modeling by revealing the frequency and nature of contacts between animals. Many wildlife diseases, such as

chronic wasting disease (CWD), have been a central challenge to wildlife managers. Therefore, we develop an animal interaction detection method using supervised learning methods. By integrating this detection functionality into our DeerCam, it is able to detect events of animal interactions which will trigger the on-board video encoding system to encode video samples. This will significantly reduce the amount of video data to be encoded and improve the utility of the visual sensing data. It will also provide important reference for sub-sequent wildlife behavior analysis.

CHAPTER 1

INTRODUCTION

1.1 Background

The biological relationship between wildlife and humans has never been more intertwined. Outbreaks of infectious wildlife diseases, such as chronic wasting disease (CWD), threaten wildlife populations, human life, food safety and our national economy [1]. Due to the limitations of current wildlife monitoring technologies, the behavior of free-ranging animals, the dynamics of wildlife systems, and the spread of some wildlife diseases remains largely unknown to us. Lack of scientific knowledge about the behavioral interactions and dynamics of wildlife systems significantly limits our ability to effectively manage wildlife resources and control wildlife diseases. For example, in current practice, one choice to fight against wildlife diseases such as CWD is removal of all individuals within an area of an outbreak, since we have little knowledge about how the disease might propagate within the dynamic wildlife system. Clearly, this approach is ineffective and damages the national economy. Therefore, there is a pressing need to develop a new generation of technologies to monitor wildlife behaviors and interactions and study the dynamics of wildlife systems.

In the past few decades, engineers and wildlife researchers have been developing advanced communication technologies for wildlife monitoring [1, 2]. The state-of-the-art technologies for wildlife monitoring are radio tracking and sensor networks. Radio

tracking uses a radio transmitter or receiver (attached to an animal) to collect its location information [1]. There are three distinct types of radio-tracking technologies that are in use today: *very high frequency (VHF) radio tracking* [3], *satellite tracking* [4], and *Global Positioning System (GPS) tracking* [1, 5]. Recent technological advances in hardware miniaturization of sensors, low-power microprocessor design, and wireless ad hoc networking have enabled the deployment of large-scale wireless sensor networks (WSN's) [6, 7, 9, 10]. The WSN technology provides the research community an enabling platform to simultaneously monitor a large group of free-ranging animals at granular scales. For example, ZebraNet, developed at Princeton University [5], utilizes GPS-based radio tracking and sensor networks to track the movement of a group of zebras and study animal migrations and inter-species interactions. GPS receivers are also used for animal tracking in Telenor R&D's Electronic Shepherd project [8] and UC Davis's Southern California Puma Project [11]. Embedded sensor networks have been developed by the research teams at UC Berkeley [13], UCLA [12], as well as in Australia [14] for habitat and health monitoring, using sensors to collect temperature, humidity, and other biological information on research animals.

However, the aforementioned wildlife monitoring technologies do not offer visual information. Looking back at the history of wildlife research, we find that wildlife researchers have been pursuing a dream, a dream to *see* what the animals *see* in the field without disturbing their natural behaviors, in a cost-effective way. Wildlife researchers have found that for accurate behavior analysis and interaction modeling, it is imperative to obtain some visual information about the animal's activity, its resource selection, as well as the environmental context of the behavior [15]. Otherwise, we are kept "blind"

from the animals and fail to understand important behavioral attributes of wildlife species. For example, with animal's location only, we do not know what the animal is doing, how it is doing, and why it is doing like this. Recent research [18] also shows that by not considering what the animal is doing at these point locations, the researcher may obtain a biased estimate of animal's resource selection and behavior patterns. Wildlife researchers have also observed that pooling across behaviors within different environmental contexts will de-emphasize the importance of some habitats for critical behaviors [18]. This is perhaps the most important technological challenge in wildlife tracking studies.

Another important reason for collecting visual information about animals' activities is interaction modeling for disease tracking and control. Most *epidemic wildlife diseases* spread through direct animal contacts (e.g., via direct contact with saliva) and animals' interactions with the environment and other species. With current technologies, all we have available for analysis is a series of approximate location and movement estimates for a number of animals, and we do NOT know whether animals are directly interacting or not. By collecting and analyzing direct visual information, we can study the animal's close interactions with other animals and their environment, which enables us to understand the dynamics of wildlife systems and to develop quantitative propagation models for wildlife diseases.

In this dissertation, we will present our research effort on resource-efficient portable video communication system design for wildlife activity monitoring and interaction tracking. More specifically, in this research project[1], we will design an integrated video

[1] *This project is funded by National Science Foundation in collaboration with University of Florida.*

encoding and sensor data collection system, called *DeerCam*, and deploy it on animals to collect important video and sensor information about animals' daily activities in the field over an extended period of time. With intelligent information analysis and fusion, we are able to study animals' behavior patterns and answer many challenging wildlife research questions. We present our DeerCam system architecture, discuss the major design challenges, and present our approaches to addressing these challenges.

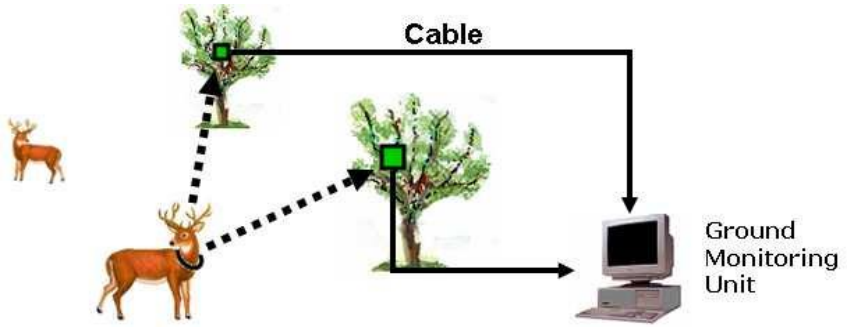


Figure 1.1: Illustration of wildlife activity monitoring using DeerCam.

1.2 System Framework, Design Goals and Challenges

In this section, we describe the proposed system framework for wildlife activity monitoring, explain our design goals, and discuss major design challenges.

1.2.1 System Framework

Figure 1 illustrates our proposed framework for wildlife activity monitoring. We

design an integrated video encoding and sensor data collection system, called *DeerCam*, and mount it on animals. The DeerCam captures video information about animal's daily activities in the field, compresses the video data and stores the bit stream in an on-board storage device. We realize direct wireless data transmission is very difficult because wireless signals attenuate dramatically when the animal moves away, especially in forested terrain. According to our experience, the average transmission distance is only about 20 meters in a wooded area. Therefore, we choose to temporarily store the compressed video data on-board. We also set up several wireless access points near places where the animal visits often, for example, food stations. When the animal returns repeatedly to one of these access points, the compressed video data is uploaded through wireless transmission to a video server in a ground monitoring unit, which connects to the wireless access point via an Ethernet cable. In addition, at the completion of the planned experiment, the DeerCam system can be physically retrieved and the video data can be then directly downloaded to a computer for further analysis.

Besides video data, the DeerCam system is also equipped with a suite of sensors, such as accelerometer, GPS, temperature, and light sensors, on-board to collect various aspects of information regarding animal activity in the field. From the video data, assisted by intelligent computer vision algorithms (e.g. objects detection and classification), we will be able to extract important information regarding animal food selection and activity patterns, fuse it with sensor data, and use the resulting data to study animal's behavior patterns at different times, locations, and environmental contexts and answer important wildlife research questions pertaining to wildlife resource management, interaction tracking, and disease propagation modeling.

We call the proposed system for wildlife activity monitoring *DeerCam* and select deer as the study subject for the following three major reasons. Firstt, recent CWD outbreak and associated national initiatives are related to deer [15]. For example, recent outbreaks of CWD in many U.S. states (e.g., Colorado, South Dakota, Illinois, Nebraska, Kansas, Wisconsin, etc.) have caused significant economical loss and posed a potential threat to human health [25]. Second, deer are the most ubiquitous large mammal in the U.S., thus deer diseases potentially impact everyone in the nation. Third, deer are simply a starting point for this research and eventually the technologies developed in this project will be applied to other species, especially with the advances in hardware miniaturization of sensors [9, 26].

1.2.2 Design Goals and Challenges

There are two basic approaches to wildlife activity monitoring: *passive* and *animal-mounted*. In the passive approach (also called camera traps), the camera and sensor unit is deployed on trees or other fixtures to capture images of animals when they pass by. This passive approach, because of its relative simplicity in system design and deployment, has been widely used in wildlife activity monitoring. For a detailed review of passive camera traps, see [16, 17]. The animal-mounted approach, which mounts the camera and sensor unit directly on live animals, is much more technically challenging. However, it has the capability of seeing what the animal sees in the field and collecting direct and exact information about animal activities and their environmental contexts. Furthermore, because the system moves with the animal, it is able to observe animal activities

anywhere anytime, revealing rare behaviors that we have never been able to see before.

A. Design Goals

In this project, we choose the second animal-mounted approach because of its significant impact in wildlife activity monitoring. Our design goal is *long-lived* and *unobtrusive* wildlife video monitoring. We would like to collect the animals' activity information under various environmental settings for 6 months. The long observation period is typically necessary for wildlife behavior study, especially for behavior pattern analysis across multiple seasons [5, 15]. Unobtrusive observation is essential for studying natural behaviors of animals. To this end, the sensing devices attached to animals should be light-weight such that the impact on animals' health and the disturbance to their natural behaviors are minimized. A generally accepted rule is that the weight of attached devices (including battery) should be less than 5% of the animal's body weight [23]. Smaller percentages are recommended for avifauna due to the increased energetic demands of flight and the sensitivity of avian reproductive success [24]. Another important aspect of unobtrusiveness is that, once the device is deployed, it should not require re-capture of the animal to change batteries or re-configure the system.

B. Design Challenges

The unique requirements in wildlife activity monitoring pose significant challenges in system design. More specifically, we face four major challenges and in the following, we will explain these challenges in more detail.

(1) Energy minimization

As discussed above, the DeerCam system is expected to operate for an extended period of time, for example 3 months, to allow cross-season behavior monitoring, and more importantly, battery change is not allowed once the system is deployed. According to our experience, batteries typically contribute to a majority (about 70-90%) of system weight, as we can see from Figure 1.1. The *unobtrusiveness* requirement indicates that the system weight should be kept as light as possible. This implies that the number of batteries attached to the system should be kept as few as possible and the amount of energy supply is thus very limited. In video-based wildlife activity monitoring, digital video data is voluminous, which has to be efficiently compressed before being stored or transmitted. However, efficient video compression often involves highly sophisticated motion prediction, transform, and coding schemes and is computationally intensive and energy-consuming [19]. Therefore, energy minimization of video encoding and lower-power system design is the most important challenge in DeerCam system design.

(2) Intelligent resource allocation and utility maximization

The objective of video-based wildlife monitoring is to collect important visual information about animals' activities in the field for behavior modeling and other wildlife research tasks. Therefore, the overall system performance should be measured by the utility of video data collected by the DeerCam system for wildlife research purposes (e.g. behavior modeling). Therefore, our design objective is to maximize the utility function under resource constraints. As we will discuss in more detail in Chapter 2, the system resource (bit and energy) consumption is controlled by a set of parameters. How to model the inherent relationship between resource control parameters and the video data utility, and how to optimally configure these control parameters so as to maximize the utility

function under resource constraints are challenging issues in DeerCam system design.

(3) Efficient image compression

Because of animal motion, the video samples captured by the animal-mounted DeerCam system often suffer from dramatic motion and content change, as we can see from Figure 1. 2. In this case, a significant amount of video frames and image regions are encoded with the INTRA mode. Furthermore, the image data often has a significant amount of high-frequency structural components, such as trees and grasses, as we can see from Figure 1. 3. How to develop an image / video compression scheme to efficiently represent and encode structural components becomes an important problem in our research. To address this issue, we explore various approaches, including local structure prediction to efficiently learn, predict, represent, and encode local image structures, super-spatial structure prediction to find an optimal prediction of structure components within the previously encoded image regions, and inter-structure prediction to find an optimal prediction of structure components within the encoded structure components. Our extensive experimental results demonstrate that the proposed methods are very competitive and even outperform the state-of-the-art image compression methods.

(a) frame 162

(b) frame 163

(c) frame 164

Figure 1.2: Three consecutive frames of a video clip that was captured by DeeCam. They suffer from dramatic content change.

(a)

(b)

Figure 1.3: Two frames of video clips that were captured by DeerCam. They contain a significant amount of tree or grass structures. It is important to efficiently represent and encode these high-frequency structure components.

(4) Animal interaction activity detection

The inter- and intra-specific interaction of wildlife animals is one of the most interesting activities to wildlife researchers. Video accounts of interactions could aid in disease transmission modeling by revealing the frequency and nature of contacts between animals. Many wildlife diseases, such as chronic wasting disease (CWD), have been a central challenge to wildlife managers. Therefore, we develop an animal interaction detection method. By integrating the animal interaction detection functionality into our DeerCam, DeerCam could provide crucial information about contact rates necessary to understand potential disease spread.

The animal interaction detection provides very important information for the energy minimization and storage saving of DeerCam as well. DeerCam can record the video only when the animal interaction is detected.

1.3 Thesis Outline

The rest of the thesis is organized as follows. In Chapter 2, we present our various approaches to address the first two challenges: *Energy minimization* and *Intelligent resource allocation and utility maximization* of energy-aware portable video communication system for wildlife activity monitoring. We develop joint power-rate-distortion (P-R-D) algorithms for complexity control and energy minimization. We also develop methods to maximize the utility function under resource constraints. We demonstrate that by incorporating the third dimension of power consumption into conventional R-D analysis, P-R-D analysis gives us one extra dimension of flexibility in resource allocation and energy minimization, and allows us to significantly reduce energy consumption.

In Chapter 3 - Chapter 6, we propose several approaches to address the third challenge: *Efficient image compression*. As we know, one major difficulty in image compression is to efficiently represent and encode high-frequency structure components in images, such as edges, contours, and texture regions. We also know that images are non-stationary source data and it is important to learn local image structures and adjust the image representation and prediction scheme in an adaptive manner. Motivated by this knowledge, in Chapter 3, we propose a scheme so-called *local structure learning and*

prediction to learn local image structures and efficiently predict image data based on this structure information. Our extensive experimental results demonstrate that this scheme outperforms the state-of-the-art lossly image compression schemes such as JPEG2000.

In Chapter 4, we extend the technique of *local structure learning and prediction* approach to lossless image compression. Our extensive experimental results demonstrate that the proposed method outperforms the state-of-the-art lossless image compression schemes such as Content Adaptive Lossless Image Coding (CALIC).

In Chapter 5, we develop an efficient lossless image compression scheme called *super-spatial structure prediction*. This super-spatial prediction is motivated by motion prediction in video coding, attempting to find an optimal prediction of structure components within previously encoded image regions. Our extensive experimental results demonstrate that the proposed scheme is very competitive and even outperforms the state-of-the-art image lossless compression methods.

We observe that in *super-spatial structure prediction*, the optimal prediction references of image structure components to be encoded are usually the previously encoded structure components. We can limit the search range of those prediction references to the previously encoded structure components to reduce the searching time with slightly decreased prediction accuracy. Motivated by this idea, in Chapter 6, we develop an efficient image compression scheme based on inter-structure prediction. This so-called *inter-structure prediction* attempts to find an optimal prediction of structure components within the encoded structure components. We consider only lossless image

compression. Our extensive experimental results demonstrate that the proposed scheme is very competitive.

In Chapter 7, we address the fourth challenge: *Animal interaction activity detection*. We develop an animal interaction detection method using supervised learning methods. By integrating this detection functionality into our DeerCam, it is able to detect events of animal interactions which will trigger the on-board video encoding system to capture video samples. This will significantly reduce the amount of video data to be encoded and improve the utility of the visual sensing data. It will also provide important reference for sub-sequent wildlife behavior analysis.

We summarize this dissertation in Chapter 8.

CHAPTER 2

ENERGY-AWARE PORTABLE VIDEO COMMUNICATION SYSTEM DESIGN FOR WILDLIFE ACTIVITY MONITORING

As discussed in Chapter 1, the unique requirements in wildlife activity monitoring pose significant challenges in system design. In this Chapter, we address two of these challenges: (1) energy minimization, and (2) intelligent resource allocation and sensor data utility maximization.

2.1 DeerCam System

In this section, we introduce our DeerCam system design, explain its central scheme in resource control, and characterize its behavior in resource utilization.

2.1.1 DeerCam System Design

Figure 2.1 shows the camera and sensor unit of our DeerCam system. The size of this unit is about the size of a PDA. It is based on Crossbow Stargate and Mote [27]. It has the following major components: an embedded 400 MHz Intel PXA255 XScale microprocessor, a USB-camera for video capture, a PCMCIA card for 802.11-based wireless data transmission, a Compact Flash (CF) card for storing compressed video bit

streams, and a multi-sensor board with accelerometer, GPS, temperature, and light sensors. The system also includes a drop-off control unit whose functionality will be explained in the following section. The video data captured by the digital camera is compressed by an energy-efficient MPEG-4 video encoder [19] operating on the embedded microprocessor. The compressed bit streams are temporarily stored in the 4.0 G Bytes CF card. We deploy several wireless routers near places (e.g. a food station) that the animal visits often. When the animal comes within the communication range of the router, the wireless transmission module will be activated to upload the compressed video data to a router and release the storage space in the CF card, as illustrated in Figure 2.1.

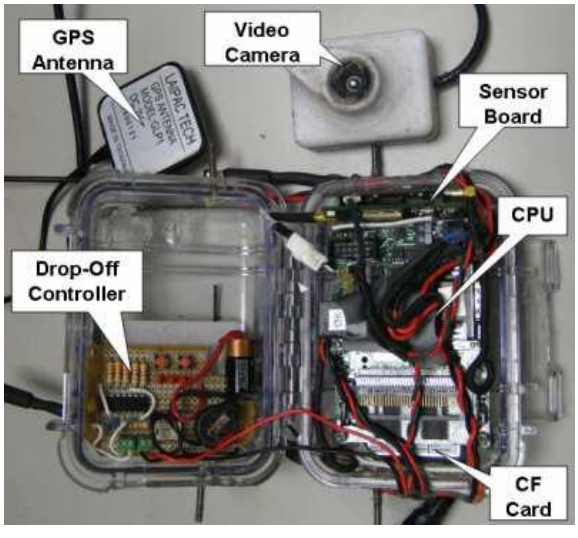


Figure 2.1: Major components in the camera and sensor unit of the DeerCam.

Figure 2.2 (a) shows the camera and sensor unit attached to a collar. There are two sets of batteries, one on each side of the collar. Inside the collar, there is an inner tube which is filled with air and the air pressure is adjusted such that the collar can be comfortably fitted onto the animal's neck, as shown in Figure 2.2 (b). At the top of the collar, there is a drop-off unit which is controlled by the drop-off control unit as shown in

Figure 2.1. The drop-off control unit has a timer which allows us to set a specific time for the drop-off collar to release. The drop-off can be also triggered by other conditions, such as end of battery lifetime. To trigger the drop-off, the control unit generates a high-voltage signal which causes a squib inside the drop-off unit to fire. When the squib fires, gas pressure is generated, moves a bolt inside the unit, and releases the collar. Once dropped off, the collar sends out a radio signal repeatedly so that we can track it in the field and retrieve the system.

Another technical challenge in DeerCam system design is housing the system. A rugged housing is critical for protecting the system from damages, moisture, water, and extreme weather conditions. Wildlife species are often aggressive. They may smash or rub the system against trees. In our DeerCam system design, we have paid special attention to rugged housing design, as shown in Figure 2.2. Another important issue is whether the DeerCam system will affect the animal's health and change its behavior. According to recent studies [15], if the animal-mounted device weighs less than 3% of the animal body weight, the device won't have an negative impact on its health. Our research animals weigh approximately 160 pounds. Our current DeerCam system shown in Figure 2.2 (a) weighs about 4 pounds, which is well below the weight threshold. Our research has not revealed changes in the behavior pattern of the animal.

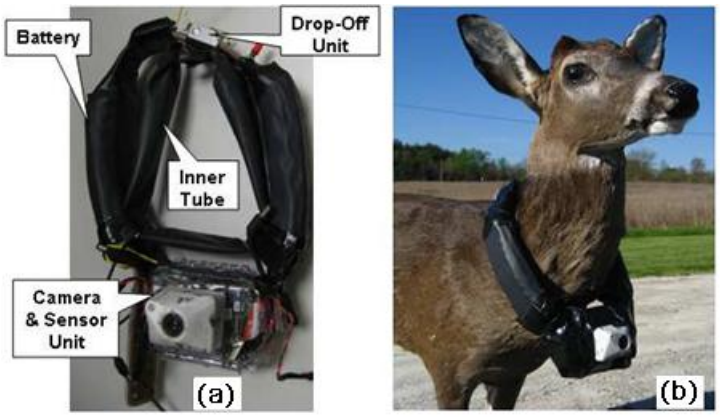


Figure 2.2: (a) Configuration of the DeerCam system; (b) a DeerCam system deployed on a deer.

2.1.2 Duty Cycle-Based Operation

Video encoding is energy-consuming. The limited batter power and the long lifetime requirement prohibit us from continuous video monitoring. In addition, from the wildlife behavior analysis perspective, there is also no need to collect video continuously for 6 months. Therefore, we use a duty cycle-based operation scheme in the DeerCam system, as illustrated in Figure 2.3 (left). Once the video sensor (Stargate) is waken up, it captures and encodes a short video segment (for example, 30 seconds) about the animal's activity, and goes back to the sleep state again. Each encoded video clip is called a *video sample*. The duty cycle can be characterized by three major parameters: average video sample duration δ, average sampling interval Δ, and picture quality D. In video compression, the picture quality is often measured by MSE (mean squared error) distortion.

We observe that configuration of these three duty-cycle parameters should be event-driven and should depend on the current activity state of the animal. In this project, we

use a multi-level sensor-driven duty-cycle design, as illustrated in Figure 2.4 (b). More specifically, the low-power Mote sensor in our DeerCam system operates continuously, collecting acceleration, audio, and GPS data. The multi-level duty cycle has four major levels. At the bottom level, there is a baseline duty cycle which wakes up the video sensor on a regular basis (for example, sun rise and sunset) or based on a preset schedule. On top of this, there are three levels of duty cycles triggered by acceleration, audio, and GPS sensors. From the temporal pattern of acceleration data, we can tell the major activity types, such as bedding / standing, walking, running, and eating, as shown in Figure 2.5 (a). A sudden change in the audio signal, for example, animal screaming, often indicates an interesting event. From the GPS data, we can tell if the animal approaches interesting habitats or areas. From this set of sensor data, we try to estimate what is the animal is doing and determine if this is an interest event. If so, the video sensor will be waked up to capture a video sample about the animal activity and its environment context. Based on the animal activity and event types, we can configuration of those three duty-cycle parameters (δ, Δ, D) accordingly. For example, when the animal is sleeping or resting, we can choose a very low sampling frequency; but when the animal is feeding or interacting with others, we would like to increase the sampling frequency, the sample duration, as well as the picture quality, if needed, to gather more detailed information. In Section 2.2.3-B, we will discuss how to configure these duty-cycle parameters so as to maximize the overall utility function under system resource constraints.

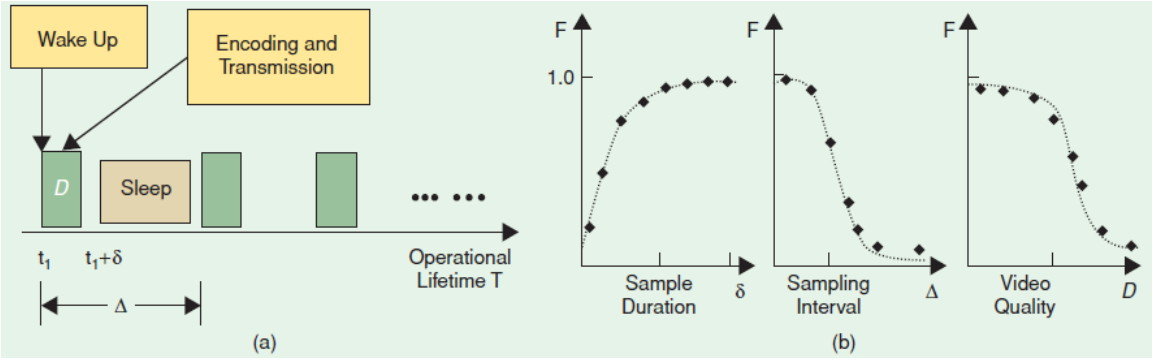


Figure 2.3: Duty cycle of video encoding and wireless transmission.

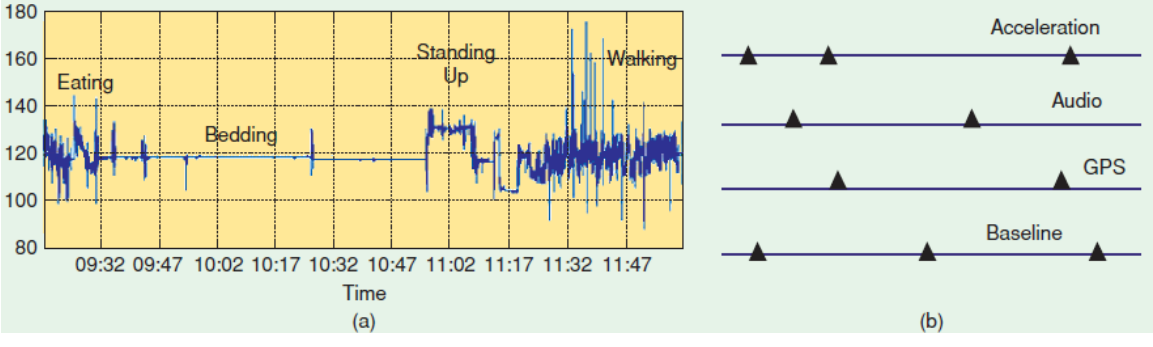


Figure 2.4: (a) An example of acceleration data collected by DeerCam; (b) multi-level duty cycle design.

2.1.3 Energy Characterization of the DeerCam System

To optimize the energy consumption of DeerCam, the first step is to characterize and understand the energy consumption behavior of the system. Specifically, we need to know the energy consumption levels of major system components, such as the embedded CPU, mote sensor board, USB camera, and storage devices, at different system operation modes. To do this, we use the power measurement setup depicted in Figure 2.5 (a) and

follow a power consumption measurement procedure described in [28]. The operating voltage of the Stargate system is 3.8-5.0V [27]. During the video capturing and encoding process, only these required device components, such as camera and CF card, are turned on, while the remaining idle components on the Stargate, such as Ethernet connection, are configured to shut off. The video encoding pipeline has been optimized such that memory access is reduced as much as possible. According to our experience, for this type of wildlife activity monitoring videos with a size of CIF (352×288) at 7-10 frames per second (fps), the average bit rate is about 400 kbps[1]. During wireless video transmission, the average bandwidth is 2-3 Mbps. Therefore, the required data transmission time is about 10-20% of the total video encoding time. Figure 2.5 (b) shows the fractions of energy consumption for major system components. Here, we assume that a low-power USB camera from Logitech with a current draw of about 73mA is used. It can be seen that the video encoder (processor) consumes a significant portion of the total energy and wireless transmission energy is about ¼ of the encoding energy due to delay-tolerant short-range wireless data transmission. Experimental studies in the literature (e.g. [29]) also show similar energy consumption behaviors of portable video devices in other communication settings, for example, live video streaming over wireless LAN.

[1]This is the average encoding bit rate for video samples of different animal activities, including feeding, bedding, walking, running, etc.

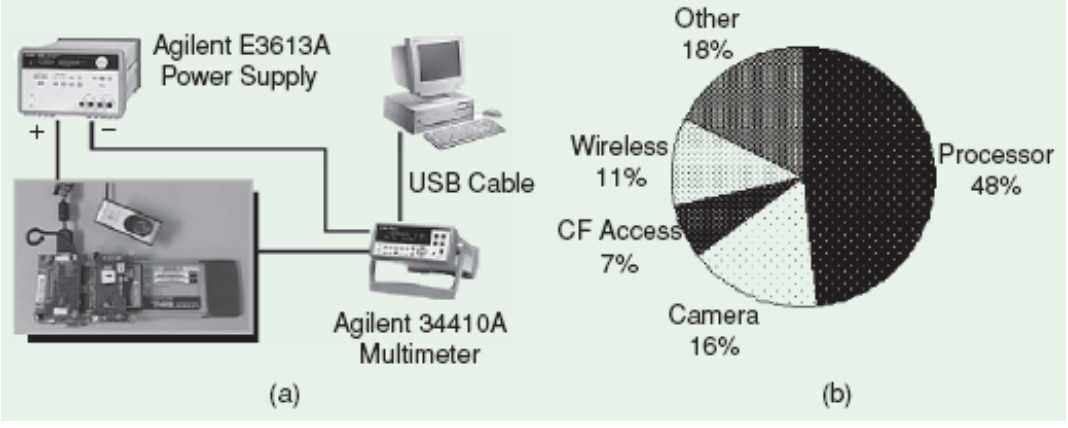


Figure 2.5: Average energy consumption of major components of the Stargate system.

2.2 Power-Rate-Distortion Analysis for Video Encoding Energy Minimization

As discussed in Section 2.1, most of the computational resources of the DeerCam system are used by the video encoder. Therefore, minimizing the video encoding energy plays a critical role in maximizing the operational lifetime of the DeerCam system. In this section, we present a power-rate-distortion (P-R-D) approach to video encoding energy minimization.

To reduce the energy consumption of video encoders, a number of algorithms, software and hardware energy-minimization techniques, including low-complexity encoder design [30-32], low-power embedded video encoding [33, 34], adaptive power control [29, 35, 36], and joint encoder and hardware adaptation [37, 39, 40] have been developed. These algorithms focus on encoder complexity (and power consumption) reduction through heuristic adaptation or control instead of systematic energy optimization. This is because they lack an analytic model to characterize the optimum

trade-off between energy saving and encoding performance [19]. In addition, even with existing energy saving technologies, the operational lifetime of portable video electronics is still very short, which has become one of the biggest impediments to our technology future. Therefore, developing new energy optimization methods has become a urgent task.

Recently, we have been exploring a new approach to video encoding energy minimizing using P-R-D analysis and optimization. The P-R-D analysis tries to answer the following important question: *what is the minimum coding distortion (or maximum video quality) that we can achieve under bit rate and power constraints, and how we can achieve it?* We recognize that, from an information-theoretic perspective, incorporating power consumption into classical R-D analysis for generic source coding is a very difficult problem. Therefore, we choose to attack this problem from a design-analysis-optimization perspective. To successfully attack this problem, the following four major issues need to be carefully addressed: (1) The **design** problem - how to design an energy-scalable video encoder so that we can control its energy consumption? (2) The **modeling** problem - how to model its P-R-D behavior? (3) The **optimization** problem -what is the minimum power consumption for a video encoder to achieve a target R-D performance, or what is maximum R-D performance of a video encoder under a power consumption constraint? (4) The **control** problem - how to accurately control the video encoder to achieve the minimum energy consumption or maximum R-D performance? In the following sections, we explain these four problems in more detail and present our major approaches to addressing these problems.

2.2.1 Energy-Scalable Video Encoder Design and Operational P-R-D Analysis

Our basic approach to designing an energy-scalable video encoder is to introduce a set of complexity control parameters $\{\gamma_1,\gamma_2,\cdots,\gamma_L\}$ to control (scale down) the computational complexity of major encoding operations of the video encoder. With DVS (Dynamic Voltage Scaling)[2], a recently developed power control technology for microprocessors [31], this complexity-scalable video encoder can be translated into an energy-scalable video encoder [19].

More specifically, the operational P-R-D analysis has the following three major steps. In the **first** step, we group major encoding operations into several modules, such as motion prediction, pre-coding (transform and quantization), mode decision, and entropy coding, and then introduce a set of control parameters $\Gamma = [\gamma_1,\gamma_2,\cdots,\gamma_L]$ to control the power consumption of these modules. Therefore, the encoder complexity C is then a function of these control parameters, denoted by $C(\gamma_1,\gamma_2,\cdots,\gamma_L)$. Within the DVS (dynamic voltage scaling) design framework [21], the microprocessor power consumption, denoted by P, is a function of computational complexity C, therefore, also a function of Γ, denoted by

$$P = \Phi(C) = P(\gamma_1,\gamma_2,\cdots,\gamma_L) \tag{2.1}$$

where $\Phi(\cdot)$ is the power consumption model of the microprocessor [21]. For example, according to our measurement, the power consumption model of the Intel PXA255 XScale processor used in DeerCam is shown in Figure 2.6 (solid line). It can be well approximated by the following expression

$$P = \Phi(C) = \beta \times C^{\gamma}, \gamma = 2.5 \tag{2.2}$$

[2]A number of sold processors, including the Intel XScale microprocessor used in many PDAs [38], support this DVS power control feature.

where β is a constant. In the **second** step, we execute the video encoder with different configurations of complexity control parameters and obtain the corresponding R-D data, denoted by $D(R;\gamma_1,\gamma_2,\cdots,\gamma_L)$. Note that this step is computationally intensive and is intended for offline analysis to obtain the P-R-D model only.

In the **third** step, we perform optimum configuration of the power control parameters to maximize the video quality (or minimize the video distortion) under the power constraint. This optimization problem can be mathematically formulated as follows:

$$\min_{\{\gamma_1,\gamma_2,\cdots,\gamma_L\}} D(R;\gamma_1,\gamma_2,\cdots,\gamma_L) \quad s.t. \quad P(\gamma_1,\gamma_2,\cdots,\gamma_L) \le P \tag{2.3}$$

where P is the available power consumption for video encoding. Given the R-D data set $\{D(R;\gamma_1,\gamma_2,\cdots,\gamma_L)\}$, this minimization problem can be easily solved using offline brute-force search. The optimum solution, denoted by $D(R,P)$, describes the P-R-D behavior of the video encoder. The corresponding optimum complexity control parameters are denoted by $\{\gamma_i^*(R,P)\}$, $1 \le i \le L$. Figure 2.7 shows the P-R-D functions $D(R,P)$ for two test video sequences, "Foreman" and "Football", both encoded by our complexity-scalable MPEG-4 encoder at CIF (352 × 288) size and 10 fps. We used two complexity control parameters, the number of SAD (sum of absolute difference) computations and the fraction of skipped macroblocks (MBs). Here, a fast algorithm, called diamond search, is used for motion estimation [22]. It should be noted that in both plots the encoding power is normalized by the maximum encoder power P_{max} where no complexity control is applied.

From those experimental results obtained in operational P-R-D analysis, we can see that the P-R-D functions have an exponential behavior. For the convenience of analysis, we propose to approximate them using the following model

$$D(R,P) = \sigma^2 2^{-\lambda R \cdot g(P)}, 0 \le P \le 1 \tag{2.4}$$

Here, σ^2 represents the variance of encoded picture. If it is a motion predicted video frame, σ2 is the variance of the difference picture after motion compensation. λ is a P-R-D model parameter which characterizes the resource (bits and energy) utilization efficiency of the video encoder. $g(P)$ is the inverse power consumption function $\Phi^{-1}(\cdot)$ after proper normalization such that $g(0) = 0$ and $g(1) = 1$. According to (2.2), we have

$$g(P) = P^{\frac{1}{\gamma}} \tag{2.5}$$

Here, $\gamma = 2.5$ for the Stargate microprocessor. For other microprocessors with DVS capabilities, we typically have $1 \le \gamma \le 3$ [21].

To use this P-R-D model for real-time video encoding, there are two major issues that need to be addressed. Firstt, we need to estimate the P-R-D model parameter λ in (2.4) from real-time video encoding statistics. Second, we need to determine the optimum complexity control parameters $\{\gamma_i^*(R,P)\}$, $1 \le i \le L$ of the video encoder for the input video segments. In our previous work [19], we observe that the P-R-D behavior of a video segment is closely related to its scene activity level, which is characterized by several feature variables, such as variance of motion vectors, Intra macroblock ratio, variance of difference pictures, etc. Based on this observation, we have developed a low-complexity scheme to estimate λ from motion statistics which can directly collected from the video encoder. To determine optimum complexity control parameters for a given

input video sequence, we use a learning and classification approach. More specifically, we set a large set of training video sequences, which have a wide range of scene activity characteristics. Using the operational P-R-D analysis procedure, we obtain the P-R-D model, i.e., the minimum distortion function $D(R,P)$ and the optimum complexity control parameters $\{\gamma_i^*(R,P)\}$ for each video sequence. We then classify the training video sequences according to their scene activity levels into several (say, 5-7) clusters. Since the video sequences with each cluster have similar P-R-D behaviors, we can compute the average of their optimum complexity control parameters for each cluster and store it into a database. During real-time video encoding, we first classify the input video segment according to its scene activity level and determine its cluster. We then use the average optimum complexity control parameters of this cluster to control the input video segment.

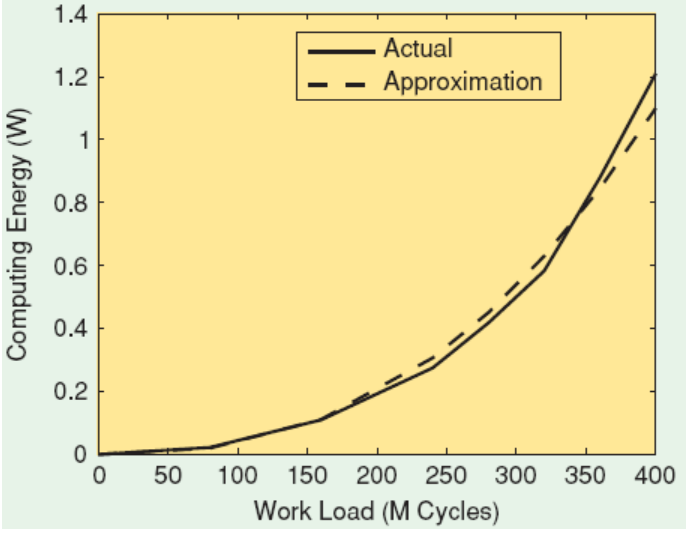


Figure 2.6: Power consumption model with DVS.

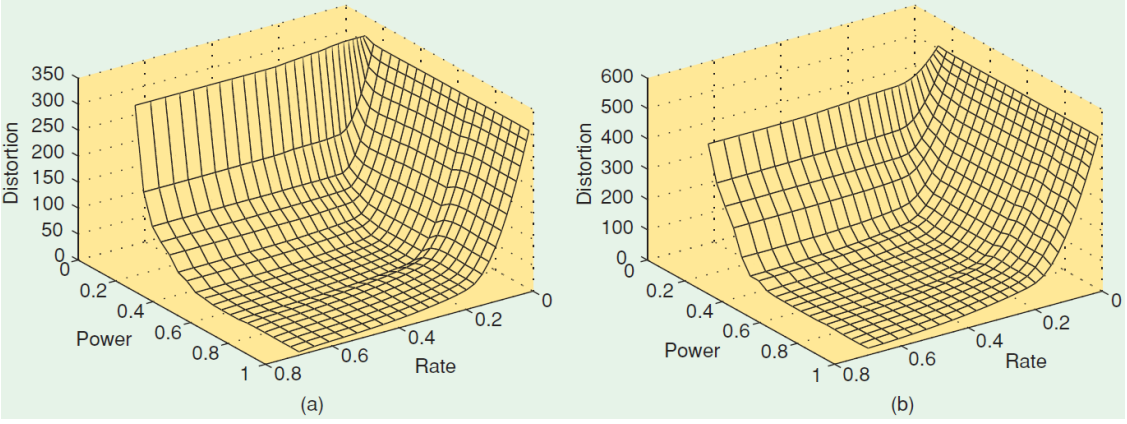


Figure 2.7: Operational P-R-D functions of "Foreman" (top) and "Football" (bottom) CIF videos encoded with MPEG-4 at 10 fps.

2.2.2 P-R-D Based Energy Minimization

Once the P-R-D model of the video encoder and the corresponding encoder complexity control parameters are obtained, we can then use this model to minimize the energy consumption of video encoding. By incorporating the third dimension of power consumption into conventional R-D analysis, the P-R-D analysis gives us one extra dimension of flexibility in resource allocation and opens up a host of new opportunities for energy minimization. In the following, we use two examples, (A) trade-off between computation and communication and (B) trading bits for joules (energy), to demonstrate how the P-R-D model can be used for energy minimization of portable video devices.

A. Trade-off between Computation and Communication

The P-R-D analysis enables us to explore the energy trade-off between computation (video encoding) and communication (wireless transmission) so as to achieve significant energy saving gain. According to the P-R-D model, video encoding power P is a function

of encoding bit rate R and distortion D, denoted by $P(R, D)$. The wireless data transmission power P_t is given by $P_t = c_t \cdot R$, where c_t is the energy cost that is needed for successful transmission of one data bit. It depends on transmission distance and path loss index [44]. This is a simplified model to demonstrate the energy tradeoff between video encoding and wireless transmission. In our case, it is reasonable because the delay is large and the transmission distance is relatively small and we can consider c_t to be the average transmission energy cost. For a given video encoding distortion (or equivalently picture quality) D, if we decrease the encoding power P, the encoder will generate more bits and a higher bit rate R is needed to achieve the target distortion This implies more power is needed for wireless data transmission. As illustrated in Figure 2.8, this leads to an energy tradeoff between video encoding and wireless data transmission. This suggests that, in many practical video encoding scenarios where the system has access to sufficient storage (or buffer) space or transmission bandwidth, the per-bit energy cost in wireless data transmission is relatively low, while the video encoder operates under severe energy constraints, we can lower the encoding power to an optimum level O, as illustrated in Figure 2.8 with a triangle, to minimize the overall power consumption. Our initial study shows that by exploring this type of energy trade-off between computation and communication, we are able to reduce the overall energy consumption by 30-60%, depending on the input video characteristics and wireless transmission condition. In practice, we need to consider fading channel and time-varying characteristics of input videos. For detailed treatment of this issue, see [41].

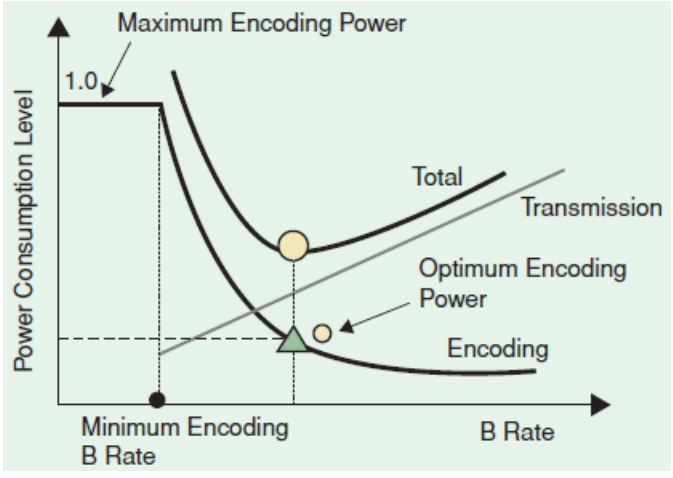


Figure 2.8: Energy tradeoff between video encoding and wireless data transmission.

B. Trading Bits for Joules

Our second observation is that, within the P-R-D analysis framework, the encoding power P is coupled with the encoding bit rate R. More specifically, for a given distortion D, the encoding power is a function of encoding bit rate, denoted by $P = P(R, D)$. This implies that we can minimize the total power consumption of video encoding by performing optimum bit allocation, just like in optimum bit allocation in conventional R-D analysis and optimization. To further illustrate this idea, let us consider the following example. Suppose we want encode a long video sequence on the DeerCam or a PDA with a bit budget R_T (say 2G bytes) at coding distortion D_0 (say 34 dB). We partition the video sequence into N segments $\{V_n | 1 \le n \le N\}$. Let R_n and P_n be the encoding bits and power consumption used by V_n. According to the P-R-D model, we have $P_n = P_n(D_0, R_n)$. Then, the energy minimization problem can be formulated as

$$\min_{\{R_n\}} P = P_n(D_0, R_n) \qquad s.t. \quad \sum_{n=1}^{N} R_n \le R_T \tag{2.6}$$

Our major result can be summarized as follows: if the input video data is non-stationary with time-varying scene characteristics, the P-R-D model enables us to explore the input source diversity, trade bits for joules (energy) between different video segments as outlined in (2.6), and achieve significant encoding energy saving by performing optimum bit allocation. In practice, a video scene under surveillance or monitoring often experiences a series of events with time-varying scene activity patterns. Therefore, the video sequence to be encoded by the portable device is often highly non-stationary. For example, in our DeerCam system, the animal has a wide variety of daily activities, such as feeding, walking, and bedding, which cause significant content changes in the input video. This observation also holds in personal video recording and many remote video surveillance applications. In [42], we demonstrate that, using this type of P-R-D based energy minimization, we are able to save the video encoding energy by up to 50%, depending on the non-stationary scene characteristics of the input videos. For a detailed treatment of this issue, see [42].

2.2.3 Additional Energy Minimization Schemes for DeerCam

Besides those two energy minimization schemes discussed in the previous section, we can also explore additional approaches for energy minimization based on the P-R-D analysis framework. In the following, we present two basic ideas to be further explored in our next step of research: (A) cross-layer energy minimization and (B) utility-driven resource allocation and energy minimization.

A. Cross-Layer Energy Minimization

Cross-layer energy minimization is able to achieve significant energy saving by

exploring joint design of hardware scheduling, encoder P-R-D control, and wireless transmission [37, 39, 40, 43]. The P-R-D model links the encoder resource parameters at the application layer with other resource parameters at the physical and link layers, which enables us to develop cross-layer energy minimization schemes. In cross-layer energy minimization for portable video communication systems, we need to consider the mobility of portable devices, the time-varying characteristics of wireless fading channels, the queuing behaviors of the processing (encoding) and transmission buffers, the energy allocation between video encoding and wireless transmission, and their joint impact to the end-to-end video quality, as well as to the overall energy saving performance.

B. Utility-Driven Resource Allocation and Energy Minimization (make sure all headings are labeled correctly as in other journal papers or thesis.)

The second idea that can be explored for energy minimization in practical system design is utility-driven resource allocation and energy minimization. In practical applications, such as video surveillance, remote monitoring, and environmental tracking applications, the task of video encoding and streaming over portable devices is to collect important visual information about the targets, events, and their environmental contexts. Therefore, the performance of the portable video communication system should be measured by the amount of useful information that has been collected by the portable device, i.e., the utility of the video data. For example, the performance of the DeerCam system should be measured by the amount of visual information that is useful for wildlife behavior analysis. In the literature [6], network capacity and operation lifetime are two major performance metrics that are extensively used for energy minimization. However, within the context of video surveillance and monitoring, these two performance metrics are not sufficient. The amount of useful visual information, or the utility of video data,

cannot be simply measured by its data size (in bits) or time duration (operational lifetime). A portable device may spend all of its energy on processing and transmitting a lot of bits which however contain little useful information for practical purposes.

Therefore, it is critical to perform utility-driven resource allocation and energy minimization such that the portable device spends its energy and other resources intelligently. To this end, we need to address the following two major issues. First, we need to establish a utility function which describes the inherent relationship between the resource control parameters and the overall utility. Second, based on this utility function, we can optimally configure the resource control parameters to maximize the utility function under the resource constraint. As a variation of this optimization procedure, we can also minimize the energy consumption while still satisfying the utility requirement. In the following, we use the DeerCam as an example to discuss how we could address these two issues.

As discussed in Section 2.1.2, the DeerCam operates on a duty cycle, which is controlled by three major parameters, (δ, Δ, D). We can see that these three parameters have an direct impact on resource consumption. For example, if we increase the sample duration δ, reduce the sampling interval Δ, or decrease the video distortion D (or equivalently increase the video quality), the video encoder will use more energy and generate more bits. Therefore, we also refer to them as resource control parameters. The overall utility of the collected video samples is a function of these three resource control parameters, denoted by $\mathbf{U}(\delta, \Delta, D)$.

To set up a utility function $\mathbf{U}(\delta, \Delta, D)$ for wildlife monitoring, we propose to use a procedure similar to perceptual video quality evaluation. More specifically, we capture a

set of original video samples using animal-mounted video cameras, which cover a wide range of habitats and plant species. We then each original video sample with different configuration of resource control parameters (δ, Δ, D). We organize a panel of wildlife experts and develop a video evaluation protocol to present each processed video sample to the expert panel and ask them to score from 0 to 10 the utility of each processed video sample for wildlife behavior analysis purposes. Based on scores provided by the expert panel, using statistical modeling, we can estimate the expression of $\mathbf{U}(\delta, \Delta, D)$ and establish a utility function. It should be noted that this utility function is fully application-specific and knowledge-driven.

Meanwhile, we analyze the relationship between energy consumption and these resource control parameters. We call this as the energy consumption function, denoted by $\mathbf{E}(\delta, \Delta, D)$. Now the utility-driven energy minimization problem becomes: optimally configuring the resource control parameters to minimize the energy consumption while satisfying the utility requirement, which can be mathematically formulated as:

$$\min_{(\delta,\Delta,D)} \mathbf{E}(\delta, \Delta, D) \qquad s.t. \quad \mathbf{U}(\delta, \Delta, D) \geq \mathbf{U_0} \tag{2.7}$$

where $\mathbf{U_0}$ is the minimum utility requirement. We expect that this utility-driven energy minimization approach will significantly reduce energy consumption.

2.3 Conclusion

The capability of seeing what animal sees in the field is very important for wildlife monitoring and research. In this paper, we have introduced the DeerCam system, an integrated video and sensor system for animal-mounted wildlife monitoring. From the

video and sensor data collected by DeerCam, wildlife researchers will be able to extract a wealth of scientific data for studying the behavior patterns of wildlife species and understanding the dynamic of wildlife systems. We have presented the system architecture of DeerCam, explained our system design goals and discussed major design issues. One of the central challenges in DeerCam system design is energy minimization. We have presented a new approach for energy minimization of portable video devices: P-R-D analysis and optimization. Results demonstrate that by incorporating the third dimension of power consumption into conventional R-D analysis, P-R-D analysis gives us one extra dimension of flexibility in resource allocation and energy minimization and allows us to significantly reduce energy consumption.

CHAPTER 3

LOCAL STRUCTURE LEARNING AND PREDICTION FOR EFFICIEENT LOSSY IMAGE COMPRESSION

The key in efficient image compression is to explore source correlation so as to find a compact representation of image data. A natural image can be often separated into two types of image regions: structure and non-structure regions. Non-structure regions, such as smooth image areas, can be efficiently represented with conventional spatial transforms, such as KLT (Karhunen Lòeve transform), DCT (discrete cosine transform) and DWT (discrete wavelet transform) [45, 47]. However, structure regions, which consist of high-frequency structure components and curvilinear features in images, such as edges, contours, and texture regions, cannot be efficiently represented by these linear spatial transforms [46]. They are often hard to compress and consume a majority of the total encoding bit rate.

Recently, researchers have developed various methods and algorithms to efficiently explore image correlation within these structure components and incorporate them into existing image/video compression systems. For example, several modified image transforms, such as curvelet [48], ridgelet [46], and contourlet [49], attempt to incorporate geometric image features into conventional wavelet transforms. Because of their over-complete representation, designing an efficient image compression system based on these transforms remains a challenging and open problem. Another important

type of techniques, called directional or orientation-adaptive wavelets, have been developed in the literature [50-53]. By adapting the direction of wavelet filtering, these techniques are able to efficiently explore local image correlation along geometrical features and significantly improve the image compression efficiency. While these methods have been focused on efficient wavelet transform design, in this work, we focus on *spatial image prediction* and *adaptive image representation* and study their impact on wavelet-based image compression.

Spatial image prediction has been a key component in efficient image and video compression. A number of highly efficient image prediction schemes have been used in the state-of-the-art image / video compression systems. For example, an adaptive image prediction scheme is proposed in LOCO-I (LOw COmplexity LOssless COmpression for Image) image compression [54]. Each pixel is first predicted using a fixed predictor, called MED (Median Edge Detector). The prediction residuals are then modeled by a TSGD (Two-Sided-Geometric Distribution) [54]. A gradient-based scheme is proposed in CALIC (Context-based Adaptive Lossless Image Coding) [55] for adaptive image prediction. The predicted value is then further adjusted using bias cancellation and error feedback. While LOCO-I and CALIC are using pixel-level spatial prediction, H.264 Intra coding employs a block-based spatial prediction scheme [56], which supports different block sizes and prediction directions. We observe that, in pixel-level prediction, it is often hard to explicitly explore local geometric image features as in H.264 Intra coding because of the large overhead for encoding geometric information [56]. On the other hand, block-based image prediction schemes suffer from performance degradation in prediction when the pixels to be predicted are further away from prediction reference. In this work, we

propose to address these issues by designing an efficient vector-based spatial image prediction scheme.

Another important motivation of this work is adaptive image representation. As we know, in transform coding of images using DCT or DWT, we use a fixed set of basis functions to approximate or represent the target image data [45, 47]. It has been observed that images are non-stationary source data. A natural image often contains a lot of local image features, such as edge, textures, and patterns, which evolve from regions to regions. Therefore, for efficient image compression, it is important to learn local image structures and adjust the image representation and prediction scheme in an adaptive manner.

In this work, we propose to learn local image structures and predict image data based on this structure information. Our basic idea is to determine the best set of basis functions from previous reconstructed image data in the neighborhood and use this basis function set to efficiently represent and predict the image data to be encoded. We prove that this basis function set can be obtained with local singular value decomposition. We find that this local structure learning and adaptive image prediction procedure is very efficient for image regions with significant structure components. However, because of its relatively large overhead and high computational complexity, it is less efficient for other image regions, such as smooth areas. To address this issue, we propose a simple yet efficient scheme to segment an image into two types of regions: structure regions and non-structure regions. In the proposed image compression system, structure regions are encoded with structure prediction, while non-structure regions are encoded with CALIC.

Our experimental results demonstrate that the proposed method is very competitive and even outperforms the state-of-the-art lossless image compression methods.

The rest of this chapter is organized as follows. The local structural learning and prediction scheme is presented in Section 3.1. Section 3.2 explains our image content separation algorithm which can efficiently decompose an image into three classes of regions: structure regions, non-structure regions and transition regions. Smooth-painting of transition and non-structure regions is also explained. In Section 3.3, we will first study how to design an efficient image encoder based local structure learning and prediction. Experimental results are presented in Section 3.4. Section 3.5 concludes the chapter and discusses our future work.

3.1 Local Structure Learning for Efficient Spatial Image Prediction

In this section, we explain the central idea of local structure learning and prediction and discuss how it can be used for efficient spatial prediction of image data.

3.1.1 Local Structure Learning

We observe that a natural image generally contains a significant amount of structural components which evolve slowly from regions to regions. For example, Figure 3.3.1(a) shows the image *Barbara* and Figure 3.1(b) shows four patches (32×32 blocks) extracted from different locations of the image. It can be seen that, at different locations, the local image data exhibits different structural characteristics. Therefore, it is important

to learn these local image structures, extract a local set of basis functions for efficient representation / approximation of each local image patch, and accurately predict its neighboring pixels. In lossless image compression, we can use the original image pixels as prediction reference. However, in lossy image compression, we need to use those previously encoded or reconstructed image pixels for prediction reference, as illustrated in Figure 1(c). For convenience, we call these reconstructed image pixels as *reference image data*.

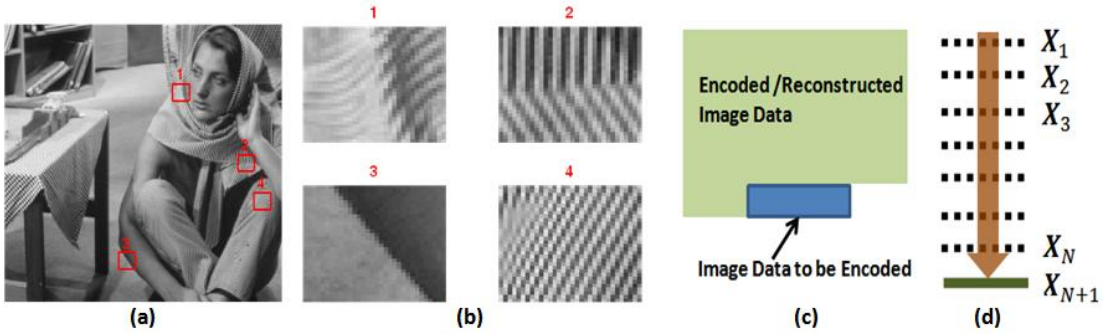


Figure 3.1: (a) The *Barbara* image; (b) four image blocks extracted from *Barbara*; (c) predictive image coding; (d) illustration of local structure learning and prediction.

We observe that the size of the prediction unit is an important parameter in spatial image prediction. For example, in pixel-level and block-based prediction, the prediction unit sizes are one pixel and one block, respectively. As discussed in Section 3.1, when the unit size is too small, the amount of prediction and coding overhead will be too large if we want to explore local geometric image features. However, if we use a larger prediction unit, for example, a block, the overall prediction efficiency will decrease. In

this work, we attempt to find a good trade-off between these two and propose to perform spatial image prediction on a vector basis: predict a vector of pixels in the next image row or column. For example, in our experiments, we set the vector size to be 1×4 in vertical prediction or 4×1 in horizontal prediction. In Section 3.2, we will explain this in more detail.

Now, the local image structure learning and prediction problem can be formulated as follows: given a set of reference vectors (of pixels), $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3 \cdots \boldsymbol{X}_N\}$, which have already been encoded and reconstructed, as shown in Figure 3.1(d), how to predict vector $\boldsymbol{X}_{N+1}$ in the next row or column such that the prediction error:

$$\boldsymbol{E} = \|\boldsymbol{X}_{N+1} - \boldsymbol{f}(\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3 \cdots \boldsymbol{X}_N)\|_2 \tag{3.1}$$

is minimized. Here, $\boldsymbol{f}(\cdot)$ is a predictor function and we use the mean squared error (or L_2-norm) to measure the prediction performance. We denote the vector size by L. To determine the predictor function, we attempt to learn the local image structure of $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3 \cdots \boldsymbol{X}_N\}$. More specifically, we find a small set of orthonormal basis functions $\{\varphi_1, \varphi_2, \cdots \varphi_K\}$, where $K \ll N$, to approximate $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3 \cdots \boldsymbol{X}_N\}$. Let

$$\widehat{\boldsymbol{X}}_n = \sum_{k=1}^{K} c_{nk}\varphi_k, \;\; c_{nk} = (\boldsymbol{X}_n, \varphi_k) \tag{3.2}$$

be the prediction of $\boldsymbol{X}_n$, $1 \le n \le N$. We need to find the best set of basis functions $\{\varphi_1, \varphi_2, \cdots \varphi_K\}$ such that the overall prediction or approximation error is minimized, i.e.,

$$\min_{\{\varphi_1, \varphi_2, \cdots \varphi_K\}} \boldsymbol{E} = \sum_{n=1}^{N} \left\| \boldsymbol{X}_n - \sum_{k=1}^{K} (\boldsymbol{X}_n, \varphi_k) \cdot \varphi_k \right\|_2 \tag{3.3}$$

In Appendix A, we prove that the optimum solution $\varphi_k$is the *k*-th eigenvector in the singular value decomposition of matrix $\boldsymbol{X}$. More specifically, we consider $\boldsymbol{X}_n$ as a column vector and $\boldsymbol{X}$ a $L \times N$ matrix, $\boldsymbol{X} = [\boldsymbol{X}_1\, \boldsymbol{X}_2\, \boldsymbol{X}_3\, \cdots \boldsymbol{X}_N]$. Suppose that its singular value decomposition is

$$\boldsymbol{X} = \boldsymbol{U}_{L\times L}\, \boldsymbol{\Lambda}_{L\times N}\, \boldsymbol{V}_{N\times N}.$$

Then, the optimum solution $\varphi_k$is the *k*-th column vector of matrix $\boldsymbol{U}_{L\times L}$**.** Here, $\boldsymbol{\Lambda}_{L\times N}$ is a diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{L-1} \geq \sigma_L$. The corresponding minimum approximation error is given by

$$\boldsymbol{E}^* = \sum_{i=K+1}^{L} \sigma_i^2. \tag{3.4}$$

It can be seen that if we increase *K*, the number of basis functions for signal representation, the total approximation error will decrease.

3.1.2 Local Structure Prediction

The optimum basis function set $\{\varphi_1, \varphi_2, \cdots \varphi_K\}$ captures the major structural characteristics of the reference image data $\boldsymbol{X}$. We believe that the image data to be predicted, i.e., vector $\boldsymbol{X}_{N+1}$ shares similar structural characteristics since $\boldsymbol{X}_{N+1}$ is the immediate neighbor of image patch $\boldsymbol{X}$. Once the optimum basis function set $\{\varphi_1, \varphi_2, \cdots \varphi_K\}$ is obtained, we decompose $\boldsymbol{X}_n$ onto these basis functions and compute the decomposition coefficients $c_{nk} = (\boldsymbol{X}_n\, , \varphi_k)$. To predict $\boldsymbol{X}_{N+1}$ from $\boldsymbol{X}$, we extrapolate their decomposition coefficients. More specifically, let

$$\hat{c}_{N+1,k} = \boldsymbol{\mathcal{L}}(c_{1k}, c_{2k}, \cdots c_{Nk})$$

be the one-step extrapolation of $\{c_{1k}, c_{2k}, \cdots c_{Nk}\}$. In this work, for simplicity, we use linear extrapolation. Then,

$$\widehat{\boldsymbol{X}}_{N+1} = \sum_{k=1}^{K} \hat{c}_{N+1,k} \varphi_k$$

forms the prediction of $\boldsymbol{X}_{N+1}$.

3.1.3 Local Structure Prediction in Images

To successfully apply the structure prediction proposed in Section 3.2 to image compression, the following two issues need to be carefully addressed: (1) selecting the vector direction and (2) selecting reference vectors.

(1) *Selecting Vector Direction*

The proposed structural prediction scheme operates on a vector basis. There are two basic choices in selecting the next prediction vector: *row* (*horizontal*) and *column* (*vertical*). The selection of vector direction depends on the correlation direction of the local image data. If the image data has strong correlation with its vertical neighbors on the top, we need to choose row vectors for prediction, as illustrated in Figure 3.2(a). Similarly, if the image data has strong correlation with its horizontal neighbors on its left, we need to use column vectors for prediction, as illustrated in Figure 3.2(b). Since different image regions have different correlation structures, the decision on vector direction needs to be adaptive. However, the adaptive decision cannot be made on a

vector basis. Otherwise, the scenario illustrated in Figure 3.2(c) will arise, which make it very difficult for us to select reference image data for prediction and to manage the image encoding process. To address this issue, we propose to decide the vector direction on a block basis, as illustrated in Figures 2(a) and (b). All vectors within one block use the same vector direction, either horizontal or vertical. Once the first vector (row or column) of the block is predicted, encoded, and reconstructed, it will become a part of the reference image data during the prediction of second vector. This procedure is repeated until all vectors (rows or columns) of block are predicted and encoded.

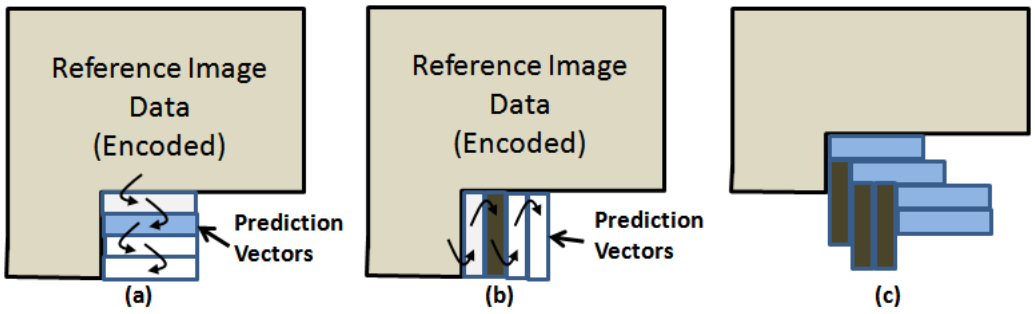


Figure 3.2: Row prediction and column prediction.

In this work, we observe that the optimum vector prediction direction is highly correlated with the direction of local gradients. To see this, in Figure 3.3, we use a brute-force approach to find the optimum vector prediction direction, predicting the image block using both row and column vectors and determine which vector direction has a smaller prediction residual. We also compute the local gradient of this image block and let $0 \leq \theta \leq 2\pi$ be the direction angle. Let $[x, y]$ with $x^2 + y^2 = 1$ be the coordinate of

this angle. Define a classification feature $C_f = |y| - |x|$. We can see that if $C_f > 0$, the block has near vertical local gradients; otherwise, it has near horizontal local gradients.

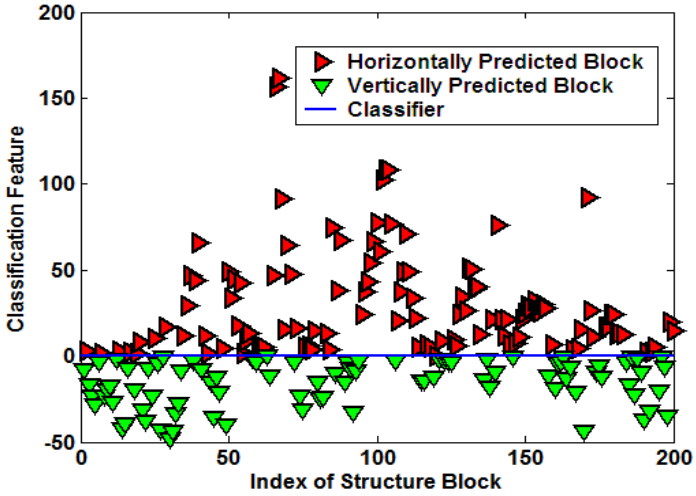


Figure 3.3: Correlation between vector direction and local gradient direction.

Figure 3.3 shows the value of C_f for all 200 structure blocks in image *Lena*. Red and yellow triangle represents blocks whose optimum predictions are horizontal and vertical, respectively. It can be seen that the optimum prediction vector and the classification feature C_f are highly correlated. This implies that, in actual image prediction, we can use the local gradient direction to estimate the optimum prediction direction.

(2) *Selecting Reference Vectors*

The reference vectors are selected from the reference image data. In doing so, we need to consider the local image correlation structure so as to maximize the overall prediction efficiency. Similar to H.264 Intra coding, we choose 8 spatial prediction

directions, with four directions, C-0, C-1, C-2, and C-3 for column prediction and another four, R-0, R-1, R-2, and R-3 for row predictions, as illustrated in Figure 3.4(a). Figures 4(b) and (c) are two examples showing how the reference vectors are selected for directions C-2 and R-1. For each direction, we compute the approximation error as defined in (4). The direction with the minimum approximation error will be used for actual prediction. The direction information will be encoded as overhead information and sent to the decoder. In Section 3.3, we will explain the image encoder design in more detail.

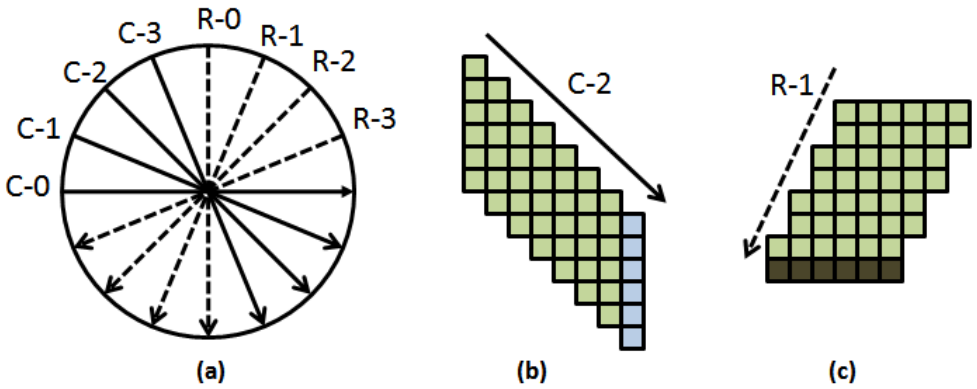


Figure 3.4: (a) 8 directions for row and column predictions; (b) reference image data selection for column prediction at direction C-2 and (c) for row prediction at direction R-1.

To predict the next row or column vector from the reference vectors, we use the structure learning method described in Section 3.1 to determine the basic vectors. For example, Figure 3.5(b) shows two basis vectors for those reference vectors in Figure 3.5(a). Figure 3.5(c) shows the corresponding decomposition coefficients. As explained

in Section 3.1, to predict the next row or column vector, we extrapolate the decomposition coefficients of those reference vectors. Figure 3.5(a) show the prediction result (the dash-triangle line) in comparison with the actual vector (the solid-square line). It can be seen that the prediction result is very close to the actual one. Certainly, this is just one example to demonstrate the process of structural prediction. In the following sections, we will evaluate the proposed structure prediction systematically.

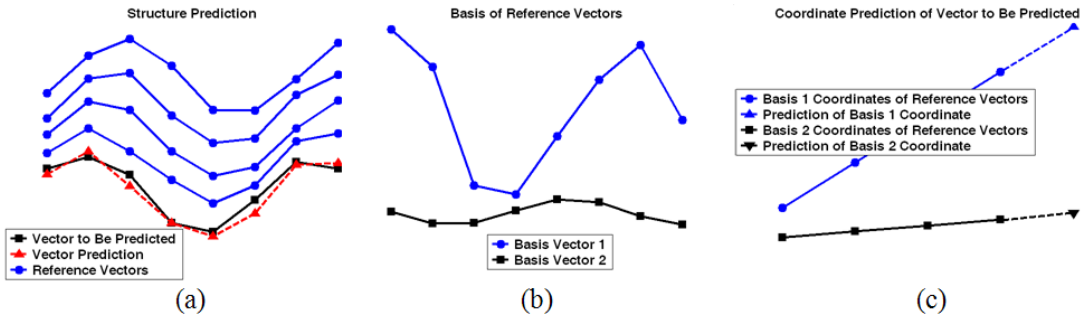


Figure 3.5: Structure prediction: (a) four reference vectors used for prediction of the next row; (b) two basis vectors obtained from structure learning; (c) extrapolation of decomposition coefficients for structure prediction.

3.2 Image Content Separation

The local structure learning method developed in Section 3.1 is able to capture local image structures for efficient spatial image prediction. From the experimental results in Section 3.4, we will see that this method works very efficiently for image regions with a significant amount of structural components. However, its efficiency will degrade in non-

structure or smooth image regions because it needs to encode and transmit overhead information about the prediction directions. In addition, its computational complexity is relatively high. Therefore, we propose a content separation scheme to separate the image data into three regions: *structure, non-structure, and transition regions*.

To explain the proposed content separation scheme, let us start with a 1-D example. Figure 3.6(a) shows a 1-D signal (e.g. one image row). It has one high-frequency component, which could be an edge in the image. Our first step in image content separation is to identify those image regions that cannot be efficiently represented or predicted by conventional spatial transform (e.g. DWT)[1]. In this work, we encode these structure regions with local structure prediction, as explained in Section 3.1. We then fill in these structure regions such that the resulting signal is "*maximally smooth*", as illustrated in Figure 3.6(b). For convenience, we refer to this operation as *smooth-painting*. Here, "maximally smooth" means the signal has a minimum amount of high-frequency components and the corresponding coding bit rate is minimized. In general, lower-frequency and smoother signals often require small coding bit rates [45].

[1] A spatial transform can be also considered as data prediction. For example, in DWT, a pixel is predicted by a weighted summation of its neighboring pixels during wavelet filtering.

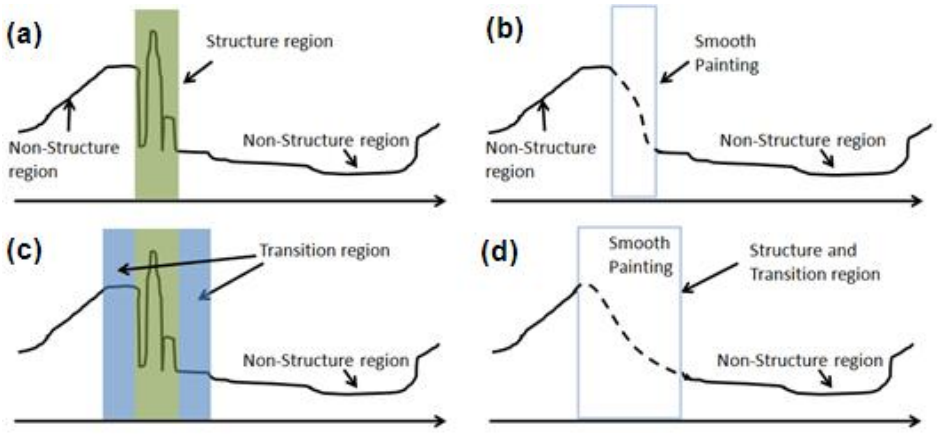


Figure 3.6: Content separation into structure, non-structure, and transition regions.

We observe that structure image regions are often very small in size. For example, an edge only occupies a long narrow image region. In addition, the contrast between pixel values at both sides of the edge is often very high, especially at strong edges. This presents a challenge in smooth painting. Even after smooth painting, the signal in the structure region is often still a high-frequency one, as shown in Figure 3.6(b). To address this issue, we propose to create a transition region, as illustrated in Figure 3.6(c). Since the size of the region has been increased, the signal after smooth-painting will be much smoother, as illustrated in Figure 3.6(d). There are three major issues that need to be carefully addressed: (1) how to identify the structure regions? (2) How to smooth-paint the structure and transition regions such that the image is maximally smooth? (3) How to determine the size of the transition region?

3.2.1 Identifying Structure Regions

In this work, we make the decision of structure prediction on a block basis. More specifically, we partition the image into blocks (e.g. 4×4 blocks). We then classify these blocks into two categories: structure and non-structure blocks. Structure blocks will be encoded with structure prediction. As discussed in Section 3.3, we will encode the non-structure blocks (regions) using wavelet-based JPEG2000 image compression after smooth painting. As discussed in the above, the DWT can be also considered as a spatial prediction scheme: a pixel is being predicted by a weighted summation of its neighboring pixels with the wavelet filter coefficients as weights. Our basic idea is that, if a block can be efficiently predicted by DWT, or equivalently, the high-pass filter outputs are very small, we classify it into the non-structure region. Otherwise, we classify it into structure regions.

Figure 3.7 shows the detailed procedure of our classification scheme. After DWT and subband decomposition, we set the high-frequency subbands to zero. After inverse DWT, we compare the reconstructed image again the original one on a block basis. If the block difference is large than a threshold, this block is considered as a structure block, otherwise, a non-structure block. For simplicity, we use the sum of absolute difference (SAD) to measure the block difference. The threshold is chosen based on the percentage of structure blocks that we want to encode with structure prediction. For example, if we like to choose 10% of blocks as structure blocks, we simply choose those top 10% block with the highest SAD values. Note that the a binary classification map with "1" for structure blocks and "0" for non-structure blocks needs to be encoded and sent to the decoder, as shown in Figure 3.8(b). To reduce the coding bit rate for this block classification map, we need to make the classification more coherent by removing those

isolated 1's and 0's. To end this, we apply morphological filtering operations to the initial classification result, as shown in Figure 3.7. Figure 3.8(c) shows the morphological filtering output.

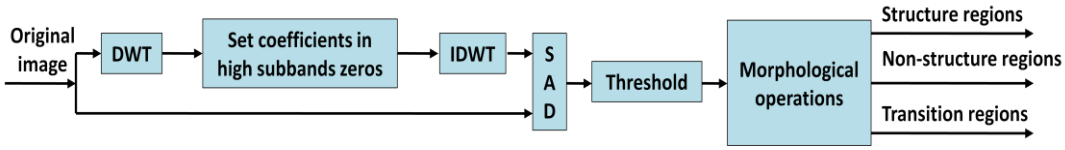


Figure 3.7: Classification of structure and non-structure image regions.

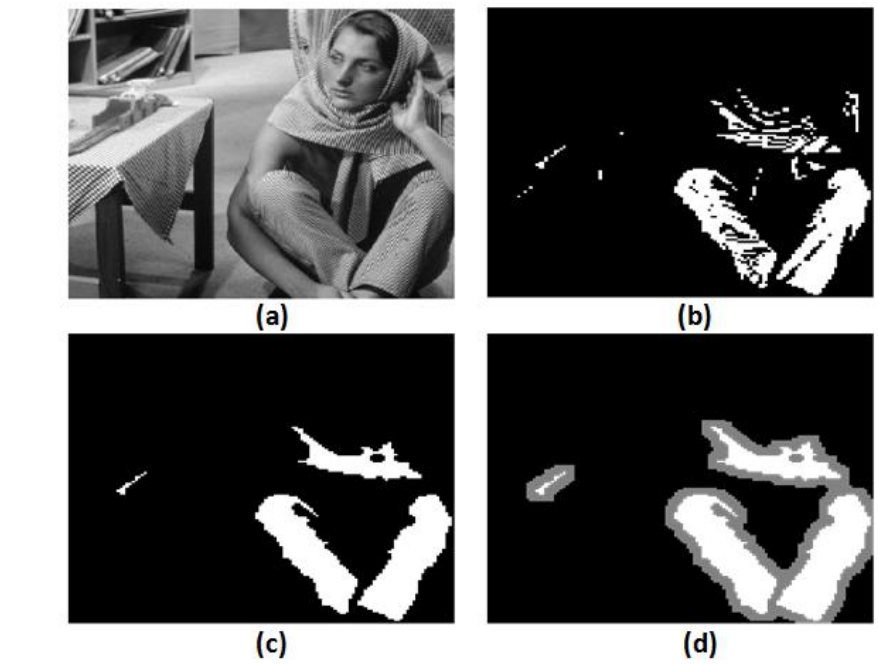
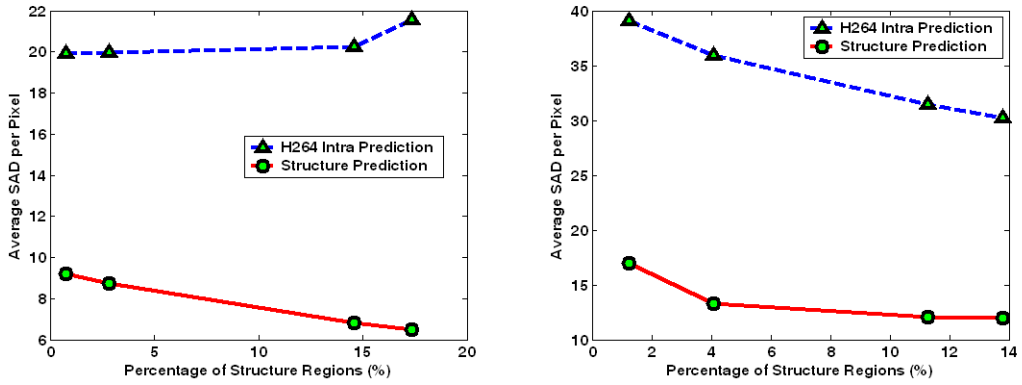


Figure 3.8: An example of image content separation: (a) original *Barbara* image; (b) initial classification result; (c) after morphological filtering; (d) augmented with transition regions.

In Figure 3.9, we compare the proposed structure prediction scheme with H.264 Intra prediction, one of the state-of-the-art schemes for image prediction. The horizontal axis shows the percentage of image blocks that are encoded as structure blocks. The vertical axis shows the SAD (sum of absolute difference) of blocks after structure prediction and H.264 Intra prediction. In both prediction schemes, we use the original pixels as the prediction reference. It can be seen that for both images *Lena* and *Barbara*, the block SAD in structure prediction is about 2-3 times smaller than that of H.264 Intra prediction.



(a) (b)

Figure 3.9: Average SAD (sum of absolute difference) of structure blocks using structure prediction and H.264 intra prediction for images (a) Lena and (b) Barbara.

3.2.2 Smooth-Painting of Structure and Transition Regions

Figure 3.6(d) illustrates the basic idea of smooth-painting: filling the structure and transition regions so as to generate a smooth image, denoted by $\boldsymbol{I}_s$. The objective of smooth-painting is to minimize the coding bit rate of image $\boldsymbol{I}_s$. As discussed in Section 3.3, image $\boldsymbol{I}_s$ will be encoded with wavelet image encoders, such as JPEG2000. As we know, during wavelet transform and subband decomposition, the image is decomposed into a series of high-frequency subbands plus one low-frequency subband. Here, a subband is called high-frequency if its subband data is the output of the high-pass filter. Therefore, to minimize the coding bit rate, we need to minimize the energy of these high-frequency subbands during smooth-painting. Let $\{p_1, p_2, \cdots p_M\}$ be values of pixels to be filled into the structure and transition regions smooth painting. The corresponding smooth image is denoted by $\boldsymbol{I}_s(p_1, p_2, \cdots p_M)$. Let $\mathbb{E}_H[\, \boldsymbol{I}_s(p_1, p_2, \cdots p_M)]$ be the energy of the high-frequency subbands. Now, the problem of smooth painting can be formulated as follows:

$$\min_{\{p_1, p_2, \cdots p_M\}} \mathbb{E}_H[\, \boldsymbol{I}_s(p_1, p_2, \cdots p_M)]. \tag{3.6}$$

In general, it will be very difficult to find an optimum solution for this problem. In this work, we propose to reduce this 2-D image smooth-painting problem into a 1-D problem and develop a sub-optimal solution.

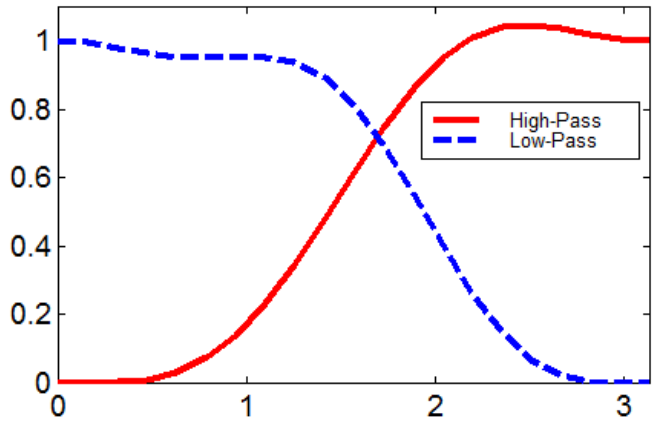


Figure 3.10: Magnitude spectrums of the low-pass and high-pass filters of Debauches (9, 7) wavelet.

Let $\boldsymbol{x} = [x_1, x_2, \cdots x_N]$ be one row or column of image data, and $[x_{P+1}, x_{P+2}, \cdots x_{P+M}]$ be the segment of pixels to be smooth-painted, as illustrated in Figure 3.6(d). The objective of smooth-painting is to minimize the high-frequency subband energy of $\boldsymbol{x}$. Let $H_1(z)$ be the high-pass filter of the wavelet filter bank with discrete Fourier transform $H_1(e^{jw})$. The high-pass filtering output is given by $Y[e^{jw}] = X[e^{jw}]H_1[e^{jw}]$. According to Parseval's theorem, its energy is given by

$$\sum_{i=1}^{N} y_i^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |Y[e^{jw}]|^2 \, dw = \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_1[e^{jw}]|^2 \cdot |X[e^{jw}]|^2 dw. \tag{3.7}$$

Now, the smooth-painting problem becomes:

$$\min_{[x_{P+1}, x_{P+2}, \cdots x_{P+M}]} \frac{1}{2\pi}\int_{-\pi}^{\pi} |H_1[e^{jw}]|^2 \cdot |X[e^{jw}]|^2 dw \tag{3.8}$$

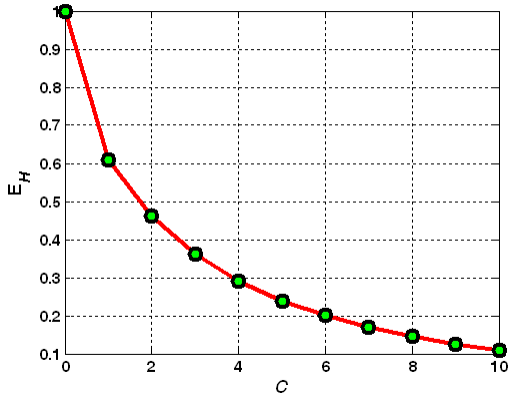
We recognize that a direct method to find the optimum solution for this problem will have very high computational complexity. In this work, we propose to use an indirect method to find a sub-optimum solution. In this following, we take the Debauches (9, 7) wavelet, which is used in our experiments in Section 3.4, as an example to explain our method. We consider the objective function in (8) as a weighted summation of $|X[e^{jw}]|^2$ with $|H_1[e^{jw}]|^2$ as weights. In Figure 3.10, we plot the $|H_1[e^{jw}]|^2$, the magnitude spectrum of the high-pass filter $H_1(z)$, as well as the spectrum of the low-pass filter $H_0(z)$. To minimize this weighted summation, we need to decrease the value of $|X[e^{jw}]|^2$ within the right-half of the spectrum since the weight $|H_1[e^{jw}]|^2$ is relatively large. In the left half, $|X[e^{jw}]|^2$ can be large since the weight $|H_1[e^{jw}]|^2$ is small. Note that the spectrum of the low-pass filter has this property. Therefore, we propose to use the low-pass filter $H_0(z)$ to shape the spectrum of $|X[e^{jw}]|^2$ through filtering. Based on the above observation, we propose the following simple yet efficient algorithm for smooth-painting:

Step 1. ***Initialization***. Fill in the missing pixels $[x_{P+1}, x_{P+2}, \cdots x_{P+M}]$ with linear interpolation.

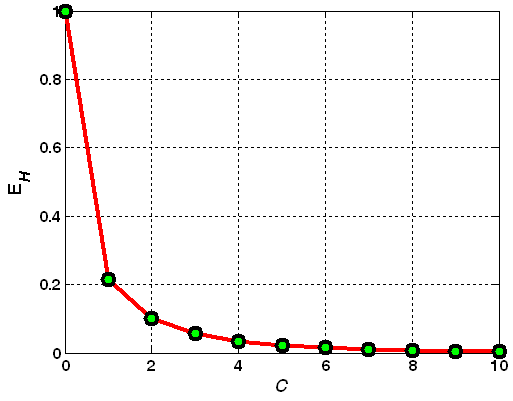
Step 2. ***Low-pass filtering***. Apply the low-pass filter to $\boldsymbol{x} = [x_1, x_2, \cdots x_N]$. To reduce computational complexity, we can apply low-pass filtering only within the neighborhood of these missing pixels. Denote the filter output by $\boldsymbol{x'}$.

Step 3. ***Restore original pixels.*** Copy the pixels $[x_1, x_2, \cdots x_P]$ and $[x_{P+M+1}, \cdots x_N]$ from $\boldsymbol{x}$ into $\boldsymbol{x}'$.

Step 4. Repeat Steps 2 and 3 for C times. In this work, we set C to be an empirical number of 5.



(a)



(b)

Figure 3.11: The energy of high-frequency subbands as a function of iterations of smooth-painting on (a) the 420th row of image *Lena* and (b) the 450th row of image *Barbara*.

Figure 3.11 shows that the energy of high-frequency subbands decreases very quickly with the number of iterations in smooth-painting on two 1-D test sequences: (a) the 420th row of image *Lena* and (b) the 450th row of image *Barbara*. It can be seen that after about 5 iterations, the high-frequency subband energy is much smaller than the original one. To extend this procedure to 2-D images, we simply apply the 1-D smooth-painting both horizontally and vertically. We can see that the basic operation in smooth-painting is interpolating the structure and transition regions using the low-pass filter. Therefore, the length of the low-pass filter, denoted by Δ_F, plays a critical role in determining the size of transition region. Let Δ_S be the length (either horizontal or vertical) of the structure region and Δ_T the length of the transition region. We use the following formula to determine Δ_T:

$$\Delta_T = \begin{cases} 0 & if\ \Delta_S > \beta\Delta_F, \\ \max\{B, \dfrac{\beta\Delta_F - \Delta_S}{2}\} & otherwise. \end{cases} \tag{3.9}$$

Here, B is the block size; β is a scaling coefficient larger than or equal to 1.0. In this work, we set it to be an empirical value, 1.5.

3.3 Image Compression Based on Structure Prediction

Based on the structure prediction and image content separation proposed in previous sections, we develop an image encoding and decoding system. As illustrated in Figure 3.12, the proposed image encoding system has the following major steps:

Step 1. ***Image content separation***. Using the method described in Section 3.2, we separate the image into structure, non-structure, and transition regions. We apply smooth-painting to the non-structure image region to create a smooth image. The binary classification map is encoded with arithmetic coding.

Step 2. ***Encode the non-structure image***. The non-structure image is encoded with JPEG2000 and reconstructed at the encoder side.

Step 3. ***Encode transition region (blocks).*** Note the transition region is used to provide sufficient space for smooth-painting of the non-structure region. This implies that it has to be encoded separately from the non-structure region. We observe that the transition region is often as smooth as the non-structure region. Since the non-structure image has already been reconstructed, we can use it to predict the transition region (or blocks). In this work, a simple DC prediction is used. The prediction residual is encoded with 4×4 integer DCT, uniform quantization, and arithmetic coding. The transition region is also reconstructed at the encoder.

Step 4. ***Encode the structure region (blocks)***. The structure blocks are encoded in a raster scan order. For each block, using the structure learning and prediction method described in Section 3.1, we compute the best prediction direction and prediction residual. The prediction residual is encoded with 4×4 integer DCT, uniform quantization, and arithmetic coding. The prediction direction is also encoded with arithmetic coding and sent to the decoder.

Compared to conventional wavelet-based image encoders, such as JPEG2000, the proposed image encoder introduces the following major computations: SVD for structure learning and prediction, smooth-painting with local low-pass filtering, 4×4 integer DCT, uniform quantization, and arithmetic coding of structure and transition blocks. SVD is a complicated matrix computation. However, its fast algorithm is available. In addition, we only apply structure learning to the structure region, a relatively small portion of the image. This will help us reduce the overall computation complexity. As discussed in Section 3.5, in our future work, various ideas can be explored to reduce the encoding complexity. Table 3.1 summarizes the average computational complexity of major components in percentage of the overall encoder complexity.

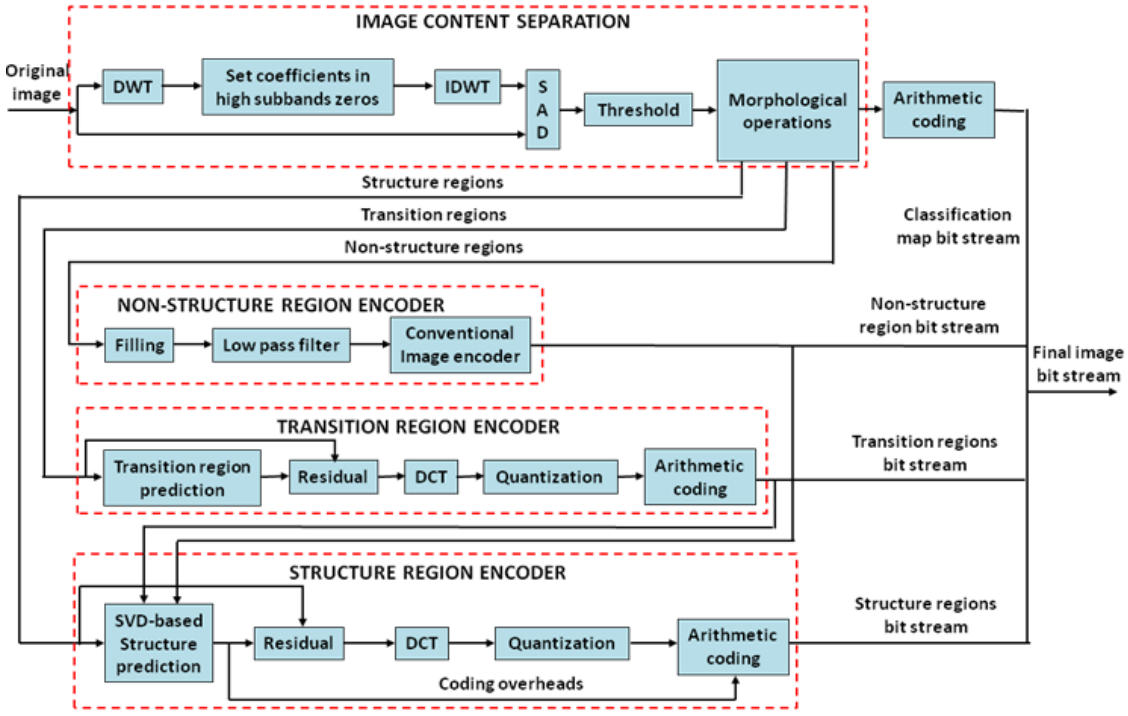


Figure 3.12: Image encoding based on structure prediction.

Table 3.1: Computational complexity of major encoding modules.

Encoder Module	Computational Complexity (%)
Structure learning	45
Smooth-painting	5
Encoding of the structure region	15
Encoding of the transition region	5
Encoding of the non-structure image	30

3.4 Experimental Results

We have implemented the proposed image compression algorithm and compared its performance with JPEG2000 image compression [57], the state-of-the-art image compression scheme. In the following experiments, we use a block size of 4×4. The Debauches (9, 7) wavelet is used in JPEG2000 image compression. Figure 3.13 shows the test images, *Lena*, *Barbara*, and *Zone-Plate* [51], all in 512×512 grayscale format. Figures 14-16 show the rate-PSNR (peak signal-to-noise ratio) curves of the proposed algorithm in comparison with JPEG2000. It can be seen that, for images *Lena* and *Barbara*, the proposed algorithm improves the picture quality by about 0.5 dB. For the Zone-Plate image, the quality improvement goes up to 2 dB.

The output bit stream of the proposed image encoder consists of bits for the following major syntax components: the image classification map, structure regions (blocks), transition, and non-structure regions. Table 3.2 shows the percentages of bits of these major syntax components at different bit rates for images *Lena.* The results for *Barbara*

are shown in Table 3.3. In these experiments, the top 10% of blocks are classified as structure blocks. It can be seen that at lower bit rates structure blocks use more bits.

In Tables 4 and 5, we evaluate the impact of percentage of structure blocks on the overall image compression performance. If no structure blocks are used, the proposed image encoder becomes the standard JPEG2000 image encoder. We encode images *Lena* and *Barbara* with different percentages of structure blocks at various bit rates and the results are shown in Tables 4 and 5. It can be seen that, at higher bit rates, we need to use a higher percentage of structure blocks to achieve the optimum performance. In general, we observe that 10-15% of structure blocks achieve the near optimum performance.

Figure 3.13: Test images: *Lena*, *Barbara*, and *Zone-Plate*.

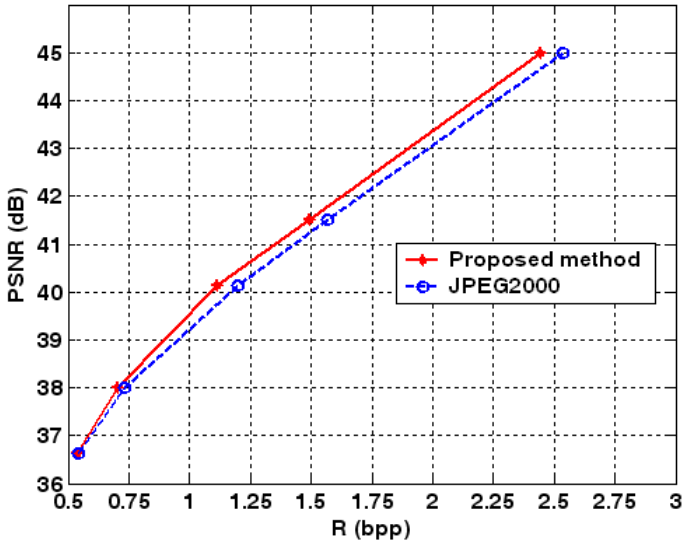


Figure 3.14: Compression performance evaluation on image *Lena*.

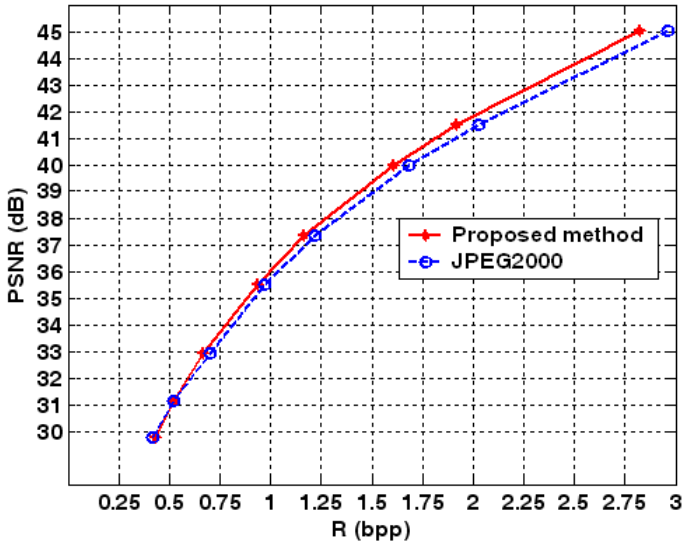


Figure 3.15: Compression performance evaluation on image Barbara.

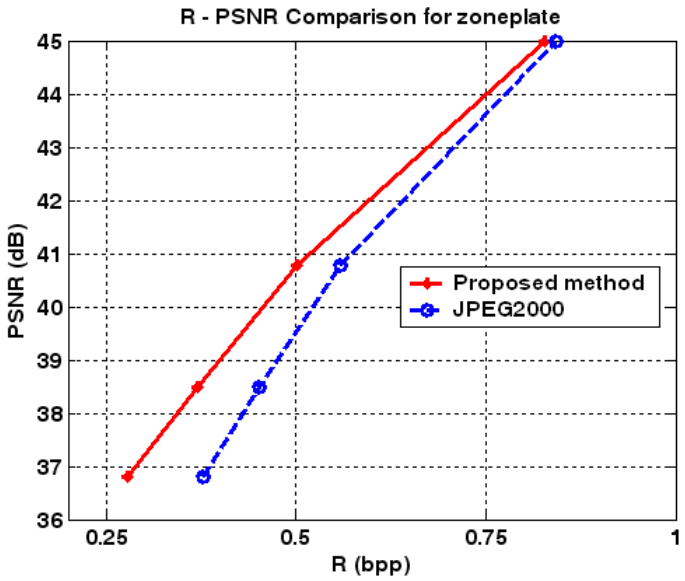


Figure 3.16: Compression performance evaluation on image Zone-Plate.

Table 3.2: Percentages of bits of major syntax components for image *Lena*

Total Bit Rate (bpp)	Percentage of Bits (%)			
	Image Classification Map	Structure Region	Transition Regions	Non-structure Regions
0.26	4.0	50.0	16.1	29.9
0.38	7	43.2	18.1	36.0
0.54	1.9	38.0	19.0	41.1
0.70	1.5	33.4	18.8	46.4
1.11	0.9	25.1	16.4	57.6
1.49	0.7	20.5	14.2	64.6

Table 3.3: Percentages of bits of major syntax components for image *Barbara*

Total Bit Rate	Percentage of Bits (%)			
	Image	Structure	Transition	Non-

(bpp)	Classification Map	Regions	Regions	structure Regions
0.39	2	51.1	19.8	26.9
0.59	1.4	47	21.4	34.5
0.82	1.0	36.8	21.0	41.2
1.02	0.8	33.4	20.6	45.2
1.40	0.6	28.5	19.2	51.7
1.69	0.5	25.7	18.0	55.9

Table 3.4: Image PSNR at different percentages of structure blocks for image *Barbara*

Bit Rate (bpp)	JPEG2000 (θ=0%)	Image PSNR (dB) at Different Percentages of Structure Blocks		
		θ =3%	θ =9%	θ =18%
0.25	33.1614	**33.0105**	34149	30.1599
0.50	36.2433	36.2921	**36.1552**	35.3507
0.75	38.0824	38.1652	**38.2422**	38.0615
1.00	39.2736	39.3277	39.5477	**39.6707**
1.25	40.3432	40.4249	40.6409	**40.8808**

Table 3.5: Image PSNR at different percentages of structure blocks for image *Lena*

Bit Rate (bpp)	JPEG2000 (θ=0%)	Image PSNR (dB) at Different Percentages of Structure Blocks			
		θ=4%	θ =11%	θ =16%	θ =22%
0.50	30.8315	**31.0528**	30.7973	30.1767	29.7799
0.75	33.4392	33.7074	**33.7085**	33.3659	37764
1.00	35.7463	35.9087	**36.008**	35.8056	35.4944
1.25	37.5203	37.6815	**37.844**	37.7367	37.593
1.5	39.0103	39.1203	**39.3501**	39.3221	39.1364
1.75	40.3464	40.4224	40.6706	**40.6852**	40.6797

3.5 Conclusion

In this chapter, we have developed an image compression algorithm based on structure learning and prediction. When learning local image structures, we attempt to find a small number of basis vectors whose linear combinations are able to closely approximate local image patches. By extrapolating these linear combination coefficients, we can efficiently predict neighboring pixels of the local image patch. To design an efficient image encoder based structure prediction, we have introduced the ideas of separation of an image into structure, transition, and structure regions and smooth-painting. Our experimental results demonstrate that the proposed algorithm outperforms JPEG2000 image compression.

CHAPTER 4

LOCAL STRUCTURE LEARNING AND PREDICTION FOR EFFICIENT LOSSLESS IMAGE COMPRESSION

In Chapter 3, we proposed local structure learning and prediction for efficient lossy image compression. As an extension, in this chapter, we apply this technique for lossless image compression with some necessary modifications. Our extensive experimental results demonstrate that the proposed method outperforms the state-of-the-art lossless image compression schemes such as LOCO-I and CALIC.

The rest of the paper is organized as follows. The local structure learning and prediction scheme is presented in Section 4.1. Section 4.2 explains our image content separation algorithm which can efficiently decompose an image into two classes of regions: structure regions and non-structure regions. In Section 4.3, we will study how to design an efficient lossless image encoder based on local structure learning and prediction. Experimental results are presented in Section 4.4. Section 4.5 concludes the paper and discusses our future work.

4.1 Local Structure Prediction for Lossless Image Compression

In Section 2.1, we explained the central idea of local structure learning and prediction and discussed how it can be used for efficient spatial prediction of image data. In this section, we investigate its prediction performance for lossless image compression.

The gradient-adjusted prediction (GAP) [58] and H.264 Intra prediction [56] are two efficient image prediction methods. The former is used in CALIC. In Figure 4.1, we compare the prediction performance of our structure prediction method with these two prediction techniques for the *Barbara* image. Figure 4.1(a) shows the original *Barbara* image. Figure 6(b)-(d) show the absolute value of the residual image obtained by the structure prediction, the GAP, and H.264 intra prediction. It can be seen that the structure prediction archives better prediction than GAP and H.264 intra prediction, especially at the image areas with significant structure components, e.g., the areas of the table cloth, hood and pants. Quantitatively, the average absolute prediction error of the structure prediction for the *Barbara* image is 10.9/pixel while that of GAP and H.264 intra prediction are 20.3/pixel and 33.8/pixel, respectively. In Section 4.4, we will present more results to demonstrate the performance of structure prediction.

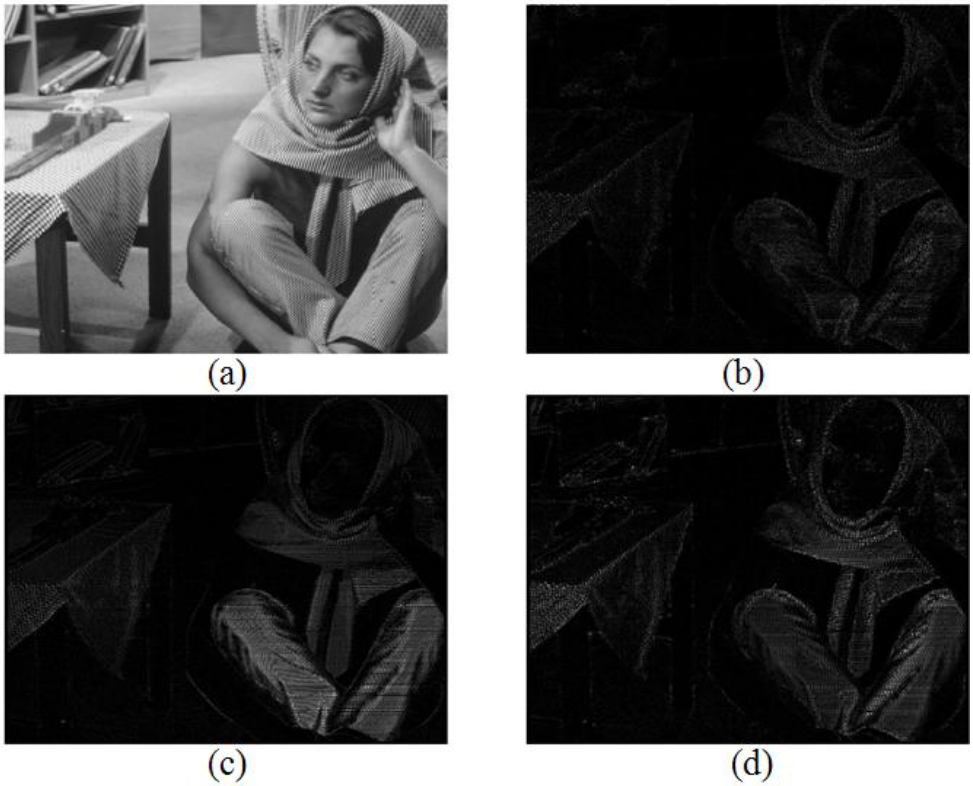


Figure 4.1: Prediction performance comparison: (a) the original *Barbara* image; (b) the absolute residual image of structure prediction; (c) the absolute residual image of GAP prediction; (d) the absolute residual image of H.264 intra prediction.

4.2 Image Content Classification

As discussed in Chapter 3 and further exemplified in Section 4.1, the local structure learning and prediction method is able to capture local image structures for efficient spatial image prediction. It works very efficiently for image regions with a significant amount of structure components. However, its efficiency will degrade in non-structure or smooth image regions because it needs to encode and transmit overhead information about the prediction. In addition, its computational complexity is relatively high.

Therefore, we propose a content separation scheme to separate the image data into two regions: *structure* and *non-structure regions*.

In this work, we make the decision of structure prediction on a block basis. More specifically, we partition the image into blocks (e.g. 4×4 blocks). We then classify these blocks into two categories: structure and non-structure blocks. Structure blocks are encoded with structure prediction. Non-structure blocks are encoded with conventional lossless image compression methods, such as CALIC [58]. CALIC is a spatial prediction based scheme, in which GAP (Gradient Adjusted Prediction) is used for adaptive image prediction. Our basic idea is that, if a block can be predicted more efficiently by GAP than by structure prediction, we classify it into the non-structure regions. Otherwise, we classify it into structure regions. For the completeness of this paper, we include the original definition of GAP [58]. Let $I[i,j]$ be the current pixel to be predicted. As illustrated in Figure 4.2, we define

$$\begin{gathered} I_n = I[i,j-1], I_w = [i-1,j], I_{ne} = I[i+1,j-1], I_{nw} = I[i-1,j-1], \\ I_{nn} = I[i,j-2],\ I_{ww} = I[i-2,j], I_{nne} = I[i+1,j-2]. \end{gathered} \tag{4.1}$$

The gradients along the horizontal and vertical directions at pixel $I[i,j]$ are estimated by

$$\begin{aligned} d_h &= |I_w - I_{ww}| + |I_n - I_{nw}| + |I_n - I_{ne}|, \\ d_v &= |I_w - I_{nw}| + |I_n - I_{nn}| + |I_{ne} - I_{nne}|. \end{aligned} \tag{4.2}$$

Then, the GAP scheme predicts the value of $I[i,j]$ as follows

IF $(d_v - d_h > 80)$ {sharp horizontal edge} $\hat{I}[i,j] = I_w$;
ELSE IF $(d_v - d_h < -80)$ {sharp vertical edge} $\hat{I}[i,j] = I_n$;
ELSE {
$$\hat{I}[i,j] = \frac{I_w + I_n}{2} + \frac{I_{ne} - I_{nw}}{4};$$
IF $(d_v - d_h > 32)$ {horizontal edge} $\hat{I}[i,j] = \frac{\hat{I}[i,j]+I_w}{2}$;
ELSE IF $(d_v - d_h > 8)$ {weak horizontal edge} $\hat{I}[i,j] = \frac{3\hat{I}[i,j]+I_w}{4}$;
ELSE IF $(d_v - d_h < -32)$ {vertical edge} $\hat{I}[i,j] = \frac{\hat{I}[i,j]+I_n}{2}$;
ELSE IF $(d_v - d_h < -8)$ {weak vertical edge} $\hat{I}[i,j] = (3\hat{I}[i,j] + I_n)/4$.
}

where $\hat{I}[i,j]$ is the predicted value of $I[i,j]$. For each block, we compute SAD between its original values and predicted ones.

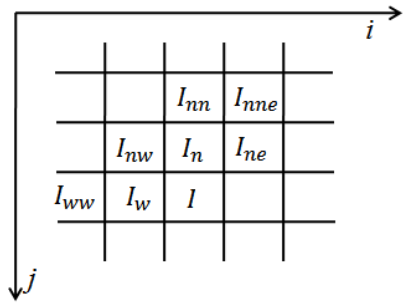


Figure 4.2: The GAP prediction scheme.

In this work, we study two classification methods. In the **first** method, as illustrated in Figure 4.3, we independently apply GAP and structure prediction to the original image. We then compare the two predicted images against the original image on a block basis.

For simplicity, we use the sum of absolute difference (SAD) to measure the block difference:

$$SAD = \sum_{i=1}^{L}\sum_{j=1}^{L} |I[i,j] - \hat{I}[i,j]|. \tag{4.3}$$

where $I[i,j]$ is the pixel of the original block, and $\hat{I}[i,j]$ that of the prediction. We then compare the block SAD values of GAP and structure prediction. If the GAP SAD is larger than the structure prediction SAD, this block is considered as a structure block. Otherwise, it is a non-structure block. In the **second** classification method, we just apply GAP to the original image and choose the fraction (e.g. 20%) of blocks with the highest prediction error as structure block and the rest as non-structural blocks. We can see that the second method has much lower computational complexity than the first method. However, its classification performance will be sub-optimal. Figure 9(b) and (c) show the classification results by these two methods with white pixels representing the structure blocks.

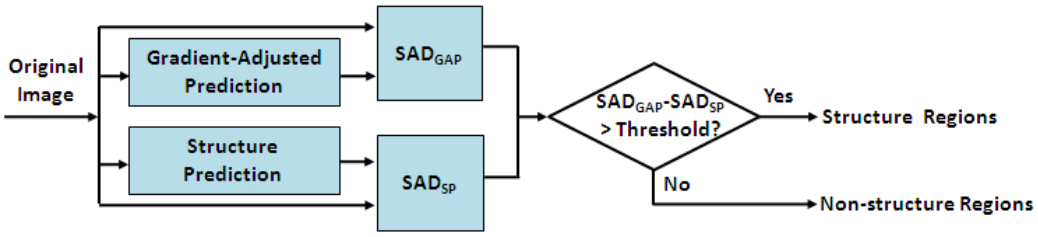


Figure 4.3: Classification of structure and non-structure regions.

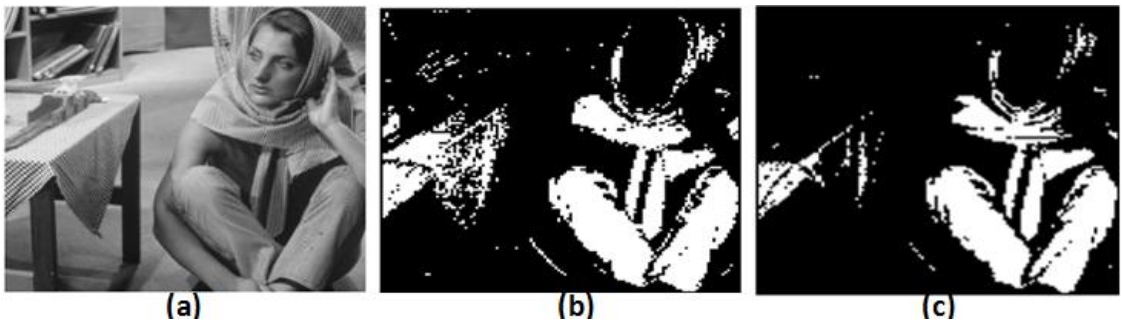


Figure 4.4: (a) The *Barbara* image; (b) classification result by the first method; (c) classification result by the second method.

4.3 Image Coding Based on Structure Prediction

Based on the structure prediction and image content separation proposed in previous sections, we develop a lossless image encoding system. The proposed image encoding system has the following major steps:

Step 1. ***Image content separation***. Using the method described in Section 4.2, we separate the image into structure regions (Figure 4.5(c)) and non-structure regions (Figure 4.5(b)). We apply filling operation to structure regions to create an image in which the values of pixels in structure regions are replaced with 128 (Figure 4.5(d)). The binary classification map is encoded with arithmetic coding and sent to the decoder.

Step 2. ***Encode the non-structure regions***. The non-structure regions are encoded with CALIC in raster scan order. With the help of the binary classification map, CALIC identifies and encodes the pixels of non-structure regions in the filled non-structure image

(Figure 4.5(d)). Note that even though the filled regions are skipped for encoding, they are necessary to provide contexts for their neighboring pixels to be encoded with CALIC.

Step 3. ***Encode the structure regions (blocks)***. The structure blocks are encoded in a raster scan order. For each block, using the structure learning and prediction method described in Section 4.1, we compute the best prediction parameters and prediction residual. The prediction parameters include the prediction direction and the number of reference vectors. The prediction residual and prediction parameters are encoded with arithmetic coding and sent to the decoder.

Compared to conventional lossless image encoders, such as CALIC, the proposed image encoder introduces the following major computations: image content classification, structure prediction in structure block coding, and arithmetic coding of prediction residuals and other side information. Structure prediction in structure block coding is computationally expensive due to SVD computation for structure learning and prediction. However, its fast algorithm is available. In addition, we only apply structure prediction coding to the structure blocks, a usually small portion of the image. This will help us reduce the overall computation complexity.

Figure 4.5: (a) The original *Barbara* image; (b) non-structure regions; (c) structure regions; (d) filled non-structure image.

4.4 Experimental Results

In this section, we implement the proposed image compression algorithm and compare its performance with three other state-of-the-art lossless image encoders, JPEG2000 [59], JPEG-LS/LOCO-I [60] and CALIC [61]. In the following experiments, we use a block size of 4×4.

The output bit stream of the proposed image encoder consists of bits for the following major syntax components: image classification map, non-structure regions and structure regions, which include prediction residual of structure regions and prediction overheads.

Figure 4.6 shows the test images used in our lossless compression. We restrict our attention to luminance component of images. First, we compare the prediction performance of the structure prediction with H.264 Intra prediction [56] and GAP [58] at structure areas (in terms of absolute prediction residual per pixel). The results are shown in Table 4.1. It can be seen that, our structure prediction achieves much more accurate predictions than both H.264 Intra prediction and GAP for all test images.

Table 4.2 shows the performance of the proposed encoder compared to three other state-of-the-art lossless image encoders, JPEG2000 [59], JPEG-LS/LOCO-I [60] and CALIC [61]. It can be seen that the proposed encoders outperforms all those three encoders and saves the overall bit rate by up to 0.261 bpp. We also observe that the proposed image encoder is very efficient for images with a significant amount of structure components, such as *Baboon*, but relatively less efficient for smooth images, such as *Lena*. Table 4.3 shows the percentage of bits used for the major syntax components mentioned above. I n Table 4.4, we compare those two classification methods presented in Section 4.2. We can see that, the second method, although has a much lower computational complexity, only slightly increases the encoding bit rate by less than 1%, when compared to the first classification method.

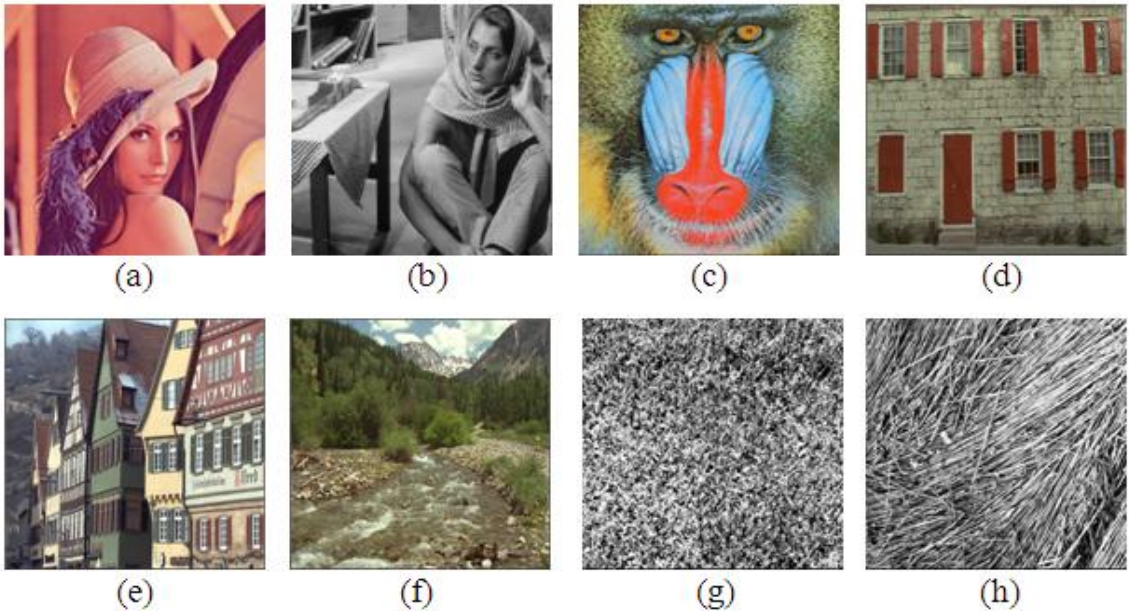


Figure 4.6: Test images from USC and Kodak image databases.

Table 4.1: Prediction errors of GAP, H.264 Intra prediction and structure prediction for structure regions (in terms of absolute prediction residual per pixel).

Image	GAP	H.264 Intra Prediction	Structure Prediction
Lena	20.6	14.5	**7.3**
Barbara	33.8	20.3	**10.9**
Baboon	14.0	13.7	**8.7**
kodim01	17.0	15.8	**9.5**
kodim08	30.3	28.1	**15.2**
kodim13	14.7	14.4	**9.3**
USC1.2.01	34.8	38.9	**22.5**
USC1.2.03	33.6	31.9	**19.8**

Table 4.2: Performance comparison with JPEG2000, JPEG-LS/LOCO-I and CALIC (bit rate in bpp). Note that the classification threshold in the proposed encoder is

manually chosen and may not be the best in terms of overall coding efficiency. It also may be different for different images.

Image	JPEG2000	JPEG-LS/LOLO-I	CALIC	Proposed Encoder	Bitrate Saving
Lena	4.347	4.245	4.097	**4.084**	**-0.013**
Barbara	4.819	4.863	4.586	**4.522**	**-0.064**
Baboon	6.145	6.041	5.898	**5.722**	**-0.176**
kodim01	5.475	5.268	5.091	**5.034**	**-0.057**
kodim08	5.562	5.285	5.049	**5.036**	**-0.013**
kodim13	6.144	5.962	5.818	**5.662**	**-0.156**
USC1.2.01	7.399	7.177	6.973	**6.78**	**-0.193**
USC1.2.03	7.414	7.071	6.873	**6.612**	**-0.261**

Table 4.3: Percentage of bits used for image classification map (ICM), non-structure regions (NSR) and structure regions (SR), which include prediction residual of structure regions and prediction overheads.

Image	Percentage of Bits		
	ICM	NSR	SR
Lena	0.07%	99.21%	0.72%
Barbara	0.24%	86.59%	13.17%
Baboon	0.36%	8.05%	91.59%
kodim01	0.61%	80.24%	19.16%
kodim08	0.23%	95.03%	4.74%
kodim13	0.39%	7.64%	91.98%
USC1.2.01	0.76%	45.82%	53.43%
USC1.2.03	0.73%	34.36%	64.91%

Table 4.4: Performance comparison between old and new classifications in terms of overall coding efficiency.

Image	Classification		Bite Rate Difference	%
	Method One	Method Two		

Lena	4.084	4.085	**+0.001**	**+0.02%**
Barbara	4.522	4.529	**+0.007**	**+0.16%**
Baboon	5.722	5.728	**+0.006**	**+0.11%**
kodim01	5.034	5.059	**+0.025**	**+0.5%**
kodim08	5.036	5.046	**+0.01**	**+0.2%**
kodim13	5.662	5.655	**-0.007**	**-0.12%**
USC1.2.01	6.78	6.822	**+0.042**	**+0.62%**
USC1.2.03	6.612	6.67	**+0.068**	**+1%**

4.5 Conclusion

In this work, we have developed an efficient lossless image compression algorithm based on structure prediction. We classified an image into structure regions and non-structure regions. The structure regions are encoded with structure prediction while the non-structure regions are encoded with existing image compression schemes, such as CALIC. Our extensive experimental results demonstrated that the proposed scheme is very efficient in lossless image compression, especially for images with significant structure components.

In our future work, we will explore various ideas to further reduce the computational complexity of the encoding algorithm, especially the structure prediction. We will also study how the structure prediction could be integrated with lifting-based wavelet transform for adaptive image prediction and transform.

CHAPTER 5

SUPER-SPATIAL STRUCTURE PREDICTION FOR EFFICIENT LOSSLESS IMAGE COMPRESSION

As previously mentioned, the key challenge in image compression is to efficiently represent and encode high-frequency image structure components, such as edges, patterns, and textures. In this work, we develop an efficient lossless image compression scheme called *super-spatial structure prediction*. This super-spatial prediction is motivated by motion prediction in video coding, attempting to find an optimal prediction of structure components within previously encoded image regions. We find that this super-spatial prediction is very efficient for image regions with significant structure components. Our extensive experimental results demonstrate that the proposed scheme is very competitive and even outperforms the state-of-the-art image lossless compression methods.

Spatial image prediction has been a key component in efficient lossless image compression [58, 62]. Existing lossless image compression schemes attempt to predict image data using their spatial neighborhood. We observe that this will limit the image compression efficiency. A natural image often contains a large number of structure components, such as edges, contours, and textures. These structure components may repeat themselves at various locations and scales. Therefore, there is a need to develop a more efficient image prediction scheme to exploit this type of image correlation.

The idea of improving image prediction and coding efficiency by relaxing the neighborhood constraint can be traced back to sequential data compression [63, 64] and vector quantization for image compression [65]. In sequential data compression, a substring of text is represented by a displacement / length reference to a substring previously seen in the text. Storer extended the sequential data compression to lossless image compression [66]. However, the algorithm is not competitive with CALIC in terms of coding efficiency. During vector quantization (VQ) for lossless image compression, the input image is processed as vectors of image pixels. The encoder takes in a vector and finds the best match from its stored codebook. The address of the best match, the residual between the original vector and its best match are then transmitted to the decoder. The decoder uses the address to access an identical codebook, and obtains the reconstructed vector. Recently, researchers have extended the VQ method to visual pattern image coding (VPIC) [67] and visual pattern vector quantization (VPVQ) [68]. The encoding performance of VQ-based methods largely depends on the codebook design. To our best knowledge, these methods still suffer from lower coding efficiency, when compared with the state-of-the-art image coding schemes.

In the intra prediction scheme proposed by Nokia [69], there are 10 possible prediction methods: DC prediction, directional extrapolations and block matching. DC and directional prediction methods are very similar with those of H.264 intra prediction [56]. The block matching tries to find the best match of the current block by searching within a certain range of its neighboring blocks. As mentioned earlier, this neighborhood constraint will limit the image compression efficiency since image structure components may repeat themselves at various locations.

In the fractal image compression [70], the self-similarity between different parts of an image is used for image compression based on contractive mapping fixed point theorem. However, the fractal image compression focuses on contractive transform design, which makes it usually not suitable for lossless image compression. Moreover, it is extremely computationally expensive due to the search of optimum transformations. Even with high complexity, most fractal-based schemes are not competitive with the current state of the art [71].

The rest of the paper is organized as follows. The super-spatial prediction for image compression is presented in Section 5.1. Section 5.2 explains our image content classification algorithm. In Section 5.3, we will study how to design an efficient lossless image encoder based on super-spatial prediction. Experimental results are presented in Section 5.4. Section 5.5 concludes this chapter.

5.1 Super-spatial Structure Prediction for Efficient Image Compression

In this section, we explain the basic idea of super-spatial prediction and how it can be used for efficient image compression.

We observe that a real world scene often consists of various physical objects, such as buildings, trees, grassland, etc. Each physical object is constructed from a large number of structure components based upon some pre-determined object characteristics. These structure components may repeat themselves at various locations and scales. For

example, Figure 5.1 (a) shows the image *Barbara* and Figure 5.1 (b) shows four patches (32×32 blocks) extracted from different locations of the image. It can be seen that they share very similar structure characteristics. Therefore, it is important to exploit this type of data similarity and redundancy for efficient image coding.

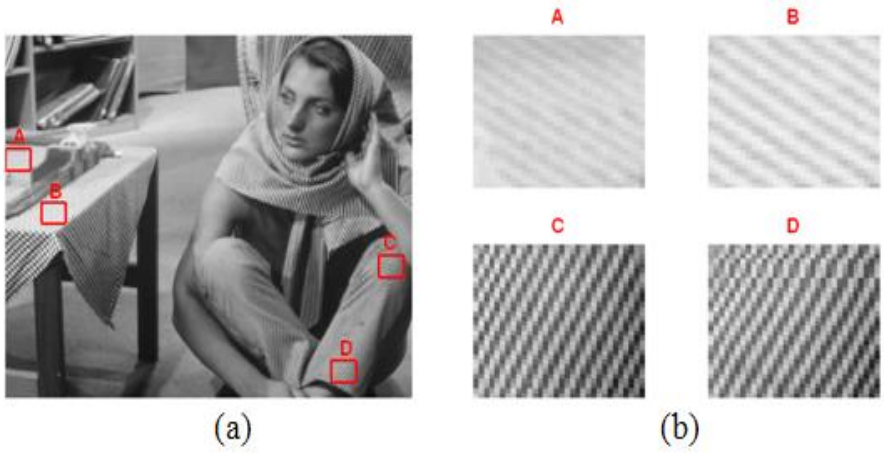


Figure 5.1: (a) The *Barbara* image; (b) four image blocks extracted from *Barbara*.

The proposed super-spatial prediction borrows the idea of motion prediction from video coding, as illustrated in Figure 5.2. In motion prediction, we search an area in the reference frame to find the best match of the current block, based on some distortion metric. The chosen reference block becomes the predictor of the current block. The prediction residual and the motion vector are then encoded and sent to the decoder. In super-spatial prediction, we search within the previously encoded image region to find the prediction of an image block. In order to find the optimal prediction, at this moment, we apply brute-force search. The reference block that results in the minimum block

difference is selected as the optimal prediction. For simplicity, we use the sum of absolute difference (SAD) to measure the block difference. Besides this direct block difference, we can also introduce additional H.264-like prediction modes, such as horizontal, vertical, and diagonal prediction [56], as illustrated in Figure 5.3(a). Here, block B is the current block to be encoded and block A is the reference block from the previous reconstructed image region. The best prediction mode will be encoded and sent to the decoder.

As in video coding, we need to encode the position information of the best matching reference block. To this end, we simply encode the horizontal and vertical offsets, (dx, dy), between the coordinates of the current block and the reference block using context-adaptive arithmetic encoder.

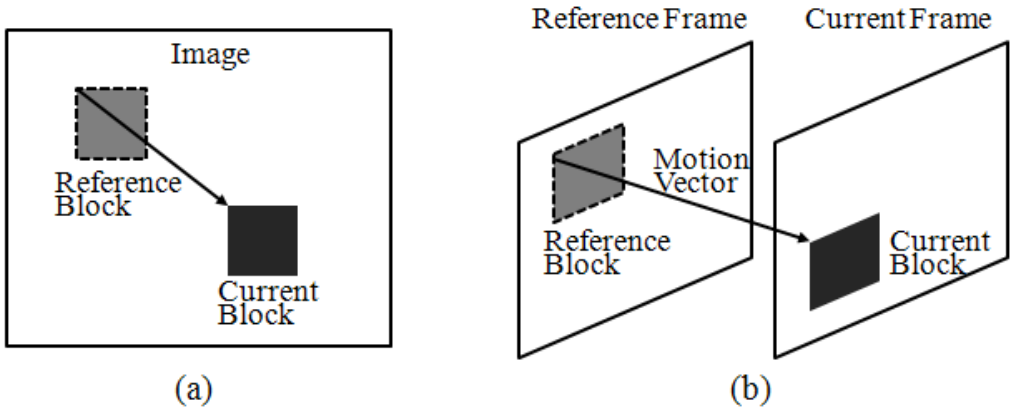
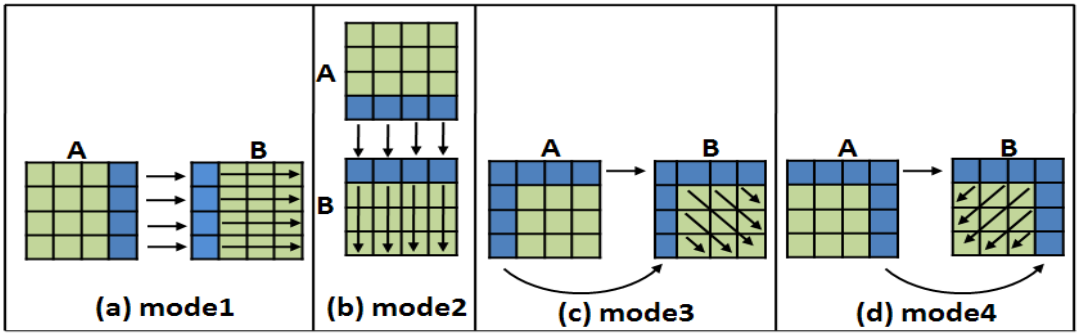


Figure 5.2: (a) Super-spatial prediction; (b) motion prediction in video coding.



(a)

(b)

Figure 5.3: (a) addition prediction modes; (b) prediction reference map for the *Barbara* image.

We observe that the size of the prediction unit is an important parameter in the super-spatial prediction. When the unit size is small, the amount of prediction and coding overhead will become very large. However, if we use a larger prediction unit, the overall prediction efficiency will decrease. In this work, we attempt to find a good trade-off

between these two and propose to perform spatial image prediction on block basis. For example, in our experiments, we set the block size to be 4×4.

We can see that, when compared to VQ-based image encoders, the proposed super-spatial prediction scheme has the flexibility to incorporate multiple H.264-style prediction modes. When compared to other neighborhood-based prediction methods, such as GAP and H.264 Intra prediction, it allows the block to find the best match from the whole image which will significantly reduces the prediction residual.

5.2 Image Block Classification

From the experimental results in Section 5.4, we will see that super-spatial prediction works very efficiently for image regions with a significant amount of structure components. However, due to its large overhead and high computational complexity, its efficiency will degrade in non-structure or smooth image regions. Therefore, we propose a block-based image classification scheme. More specifically, we partition the image into blocks (e.g. 4×4 blocks). We then classify these blocks into two categories: structure and non-structure blocks. Structure blocks are encoded with super-spatial prediction. Non-structure blocks are encoded with conventional lossless image compression methods, such as CALIC. CALIC is a spatial prediction based scheme, in which a gradient-adjusted prediction (GAP) is used for adaptive image prediction [58]. We propose to explore two classification methods. In the first method (denoted by Method_A), we compare the GAP prediction against the super-spatial prediction. Our basic idea is that, if a block can be predicted more efficiently by GAP than super-spatial prediction, we

classify it into the non-structure regions. For simplicity, we use SAD to measure the prediction performance. Otherwise, we classify it into structure regions. In the second method (denoted by Method_B), we simply perform GAP prediction on the original image and compute the prediction error for each block. If the prediction error is larger than a given threshold, then it is considered as a structure block. Otherwise, it is classified as a non-structure block. We can see that Method_A has a much higher computational complexity than Method_B since it needs to perform super-spatial prediction for each block. Figure 5.4 shows a classification result for the Barbara image using Method_A. The white pixels in Figure 5.4(b) indicates a structure block.

Figure 5.4: (a) The *Barbara* image; (b) classification result using Method_A.

5.3 Image Coding Based on Super-spatial Structure prediction

Based on the super-spatial prediction and image content separation proposed in previous sections, we develop a lossless image coding system. The proposed image encoding system has the following major steps. First, using the method described in Section 5.2, we classify the image into non-structure regions and structure regions as shown in Figure 5.5(b) and (c), respectively. The classification map is encoded with CABAC. Second, based on the classification map, our encoder switches the prediction between the GAP prediction and super-spatial prediction. The prediction residual is then encoded the CALIC scheme.

We observe that the super-spatial prediction has relatively high computational complexity because it needs to find the best match of the current block from previous reconstructed image regions. Using the Method_B prediction, we can classify the image into structure and non-structure regions and then apply the super-spatial prediction just within the structure regions since its prediction gain in the non-structure smooth regions will be very limited. This will significantly reduce overall computational complexity.

Figure 5.5: (a) The original *Barbara* image; (b) non-structure regions; (c) structure regions.

5.4 Experimental Results

We have implemented the proposed super-spatial prediction scheme in CALIC [58], a very efficient lossless image compression scheme which outperforms other state-of-the-art coding methods, such as JPEG-LS and LOCO-I. In this work, we choose a block size of 4×4. The output bit stream of the proposed encoder consists of bits for the following major syntax components: image classification map, bits for non-structure regions, bits for prediction residual of structure blocks, addresses of reference blocks, and prediction mode.

We first evaluate the prediction efficiency of the proposed super-prediction scheme. We use Method_B to classify the image block and choose the top 30% of blocks as structure blocks. We then apply three schemes to predict the image data within the structure regions: GAP from CALIC, H.264 Intra prediction which is a very efficient spatial prediction scheme proposed in H.264 video coding, and the proposed super-spatial

prediction. We measure the SAD of the prediction residual. The results are summarized in Table 5.1. We can see that on average the H.264 Intra prediction is more efficient than GAP prediction. (However, H.264 Intra prediction needs to encode overhead information, such as prediction modes [56].) Our super-spatial scheme is able to significantly reduce the prediction error. Compared with H.264 Intra prediction, it is able to reduce the SAD by up to 79% within these structure regions.

In Table 5.2, we compare the coding bit rate of the proposed lossless image coding method based on super-spatial prediction with CALIC [58], one of the best encoders for lossless image compression. We can see that the proposed scheme outperforms CALIC and save the bit rate by up to 13%, especially for images with significant high-frequency components. Table 5.3 shows the percentages of bits used by three major syntax components: structure regions (SR), non-structure regions (NSR), and overhead information (including classification map, reference block address, and prediction mode). In Table 5.4, we evaluate two image classification methods, Method_A and Method_B, as discussed in Section 5.2. We can see that, the Method_B, although has low computational complexity, its performance loss is very small, when compared to Method_A. In Table 5.5, we evaluate two super-spatial prediction methods: (1) prediction of the current block from previous reconstructed image regions and (2) prediction from previous reconstructed *structure* image regions. We can see that limiting the prediction reference to structure regions only increases the overall coding bit rate by less than 1%. This implies most structure blocks can find its best match in the structure regions.

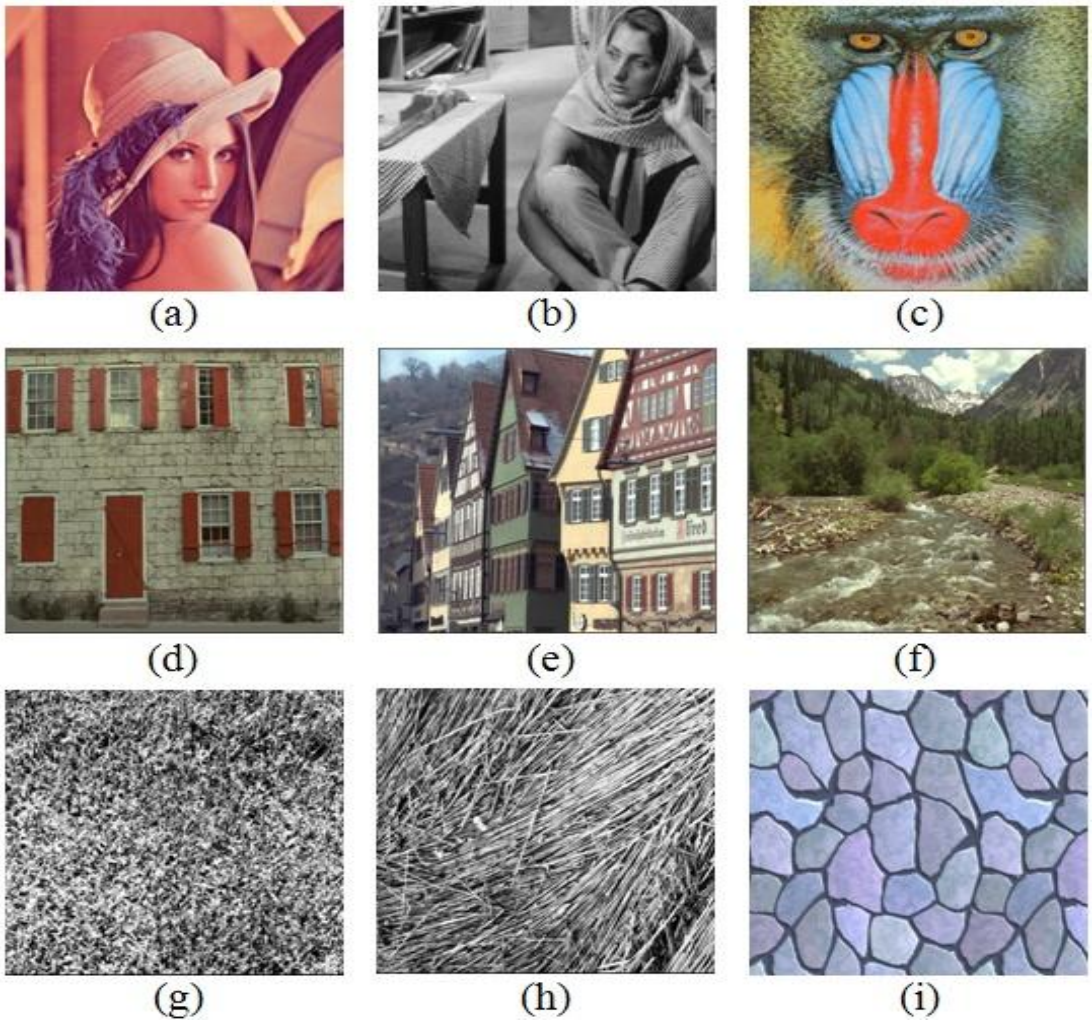


Figure 5.6: 9 test images from USC and Kodak image databases.

Table 5.1: Prediction performance comparison on the structure regions.

Test Image	GAP	H.264 Intra Prediction	Structure Prediction	Saving over H.264 Intra Prediction
Lena	20.2	15.9	7.0	56%
Barbara	22.7	17.8	6.6	63%
Baboon	22.0	20.5	11.4	44%
kodim01	16.0	16.0	8.1	49%
kodim08	18.3	21.8	8.4	61%
kodim13	22.2	20.8	11.2	46%
USC1.2.01	31.2	37.8	16.4	57%
USC1.2.03	31.4	31.6	13.4	57%
mosaic	15.7	12.1	2.5	79%

Table 5.2: Compression performance comparison with CALIC.

Test Images	CALIC Bit Rate (bpp)	This Work Bit Rate (bpp)	Bit Rate Saving
Lena	4.097	4.086	-0.011
Barbara	4.58	4.471	-0.109
Baboon	5.898	5.71	-0.188
kodim01	5.091	4.998	-0.093
kodim08	5.049	5.008	-0.041
kodim13	5.818	5.638	-0.18
USC1.2.01	6.973	6.603	-0.37
USC1.2.03	6.873	6.382	-0.491
Floor	3.855	3.4	-0.455

Table 5.3: Percentages of bits of major syntax components.

Test Image	Percentage of Bits		
	NSR	SR	Overhead
Lena	99%	0.8%	0.2%
Barbara	69%	25%	6%
Baboon	55%	36%	9%
kodim01	68%	25%	7%
kodim08	73%	21%	6%
kodim13	51%	39%	10%
USC1.2.01	18%	68%	14%
USC1.2.03	14%	71%	15%
Floor	86%	10%	4%

Table 5.4: Performance comparison between Method_A and Method_B image classification methods.

Image	Method_A	Method_B	Percentage of Bit Increase
Lena	4.086	4.086	0%
Barbara	4.471	4.463	-0.18%
Baboon	5.71	5.706	-0.07%
kodim01	4.998	5.011	0.26%
kodim08	5.008	5.033	0.50%
kodim13	5.638	5.671	0.59%
USC1.2.01	6.603	6.612	0.14%
USC1.2.03	6.382	6.387	0.08%
Floor	3.4	3.546	4.29%

Table 5.5: Impact of search range of super-spatial prediction

Image	Search in Encoded Image Domain	Search in Encoded Structure Regions	Percentage of Bit Increase
Lena	4.086	4.086	-0%
Barbara	4.471	4.506	+0.78%
Baboon	5.71	5.736	+0.46%
kodim01	4.998	5.041	+0.86%
kodim08	5.008	5.083	+1.50%
kodim13	5.638	5.687	+0.87%
USC1.2.01	6.603	6.633	+0.45%
USC1.2.03	6.382	6.415	+0.52%
Floor	3.4	3.666	+7.82%

5.5 Conclusion

In this work, we have developed a simple yet efficient image prediction scheme, called super-spatial prediction. It is motivated by motion prediction in video coding, attempting to find an optimal prediction of structure components within previously encoded image regions. When compared to VQ-based image encoders, it has the

flexibility to incorporate multiple H.264-style prediction modes. When compared to other neighborhood-based prediction methods, such as GAP and H.264 Intra prediction, it allows the block to find the best match from the whole image which significantly reduces the prediction residual by up to 79%. We classified an image into structure regions and non-structure regions. The structure regions are encoded with super-spatial prediction while the non-structure regions are encoded with existing image compression schemes, such as CALIC. Our extensive experimental results demonstrated that the proposed scheme is very efficient in lossless image compression, especially for images with significant structure components.

In our future work, we shall develop fast and efficient algorithms to further reduce the complexity of super-spatial prediction. We also notice that when the encoder switches between structure and non-structure regions, the prediction context is broken and it will degrade the overall coding performance. In our future work, we shall investigate more efficient schemes for context switching.

CHAPTER 6

INTER-STRUCTURE PREDICTION FOR EFFICIENT LOSSLESS IMAGE COMPRESSION

As mentioned in previous chapters, a real world scene consists of various physical objects, such as buildings, trees, grassland, etc. Each physical object is constructed from a large number of structural components based upon some pre-determined object characteristics. These structural components may repeat themselves at various locations of the image. For efficient representation and coding of images, it is critical to develop efficient methods to capture this type of similarity or redundancy at the structural component level.

In this work, we develop an efficient image compression scheme based on inter-structure prediction. This so-called inter-structure prediction attempts to find an optimal prediction of structure components within the encoded structure components. We consider only lossless image compression. Our extensive experimental results demonstrate that the proposed scheme is very competitive and even outperforms the state-of-the-art image compression methods such as LOCO-I and CALIC.

The rest of the paper is organized as follows. We first give an overview of the proposed lossless image compression algorithm in Section 6.1. Section 6.2 describes our image content classification algorithm. In Section 6.3, we explain the optimum reordering, prediction, and efficient encoding of structural components. The conditional

indexing of structural components is presented in Section 6.4. The experimental results are given in Section 6.5 and Section 6.6 concludes the chapter.

6.1 Algorithm Overview

A real world scene consists of various physical objects, such as buildings, trees, grassland, etc. Each physical object is constructed from a large number of structural components based upon some pre-determined object characteristics. These structural components may repeat themselves at various locations of the image. For efficient representation and coding of images, it is critical to develop efficient methods to capture this type of similarity or redundancy at the structural component level.

Figure 6.1 shows the overview of the proposed image compression scheme based on inter-structure prediction. First, we classify the input data (e.g., the Barbara image in Figure 6.2(a)) into two categories: structural components (Figure 6.2(c)) and non-structural image areas (Figure 6.2(b)). For low complexity, we use block-based classification. We apply image smoothing (low-pass filtering) to the non-structural image areas to create a smoothed image (Figure 6.2(d)), which is compressed by conventional lossless image compression systems, such as CALIC [58].

Let $\{S_m | 1 \leq m \leq M\}$ be the set of structural blocks that have been extracted from the image. We will develop a minimum spanning tree method for optimum re-ordering and prediction of these structural blocks. In this inter-structure prediction scheme, a structural block can be predicted by any block from the set, which will significantly improve the prediction and coding efficiency.

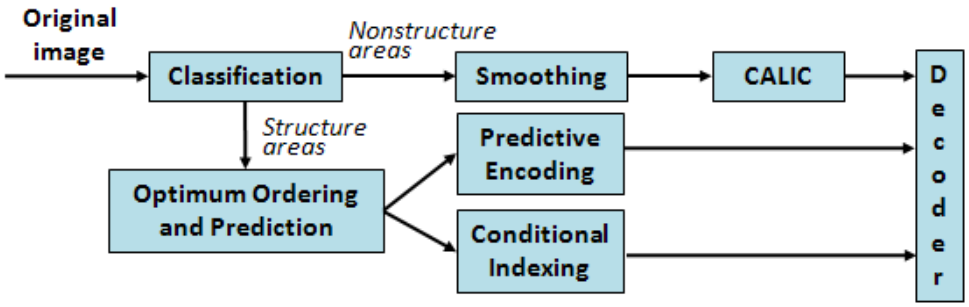


Figure 6.1: Overview of the proposed image compression scheme.

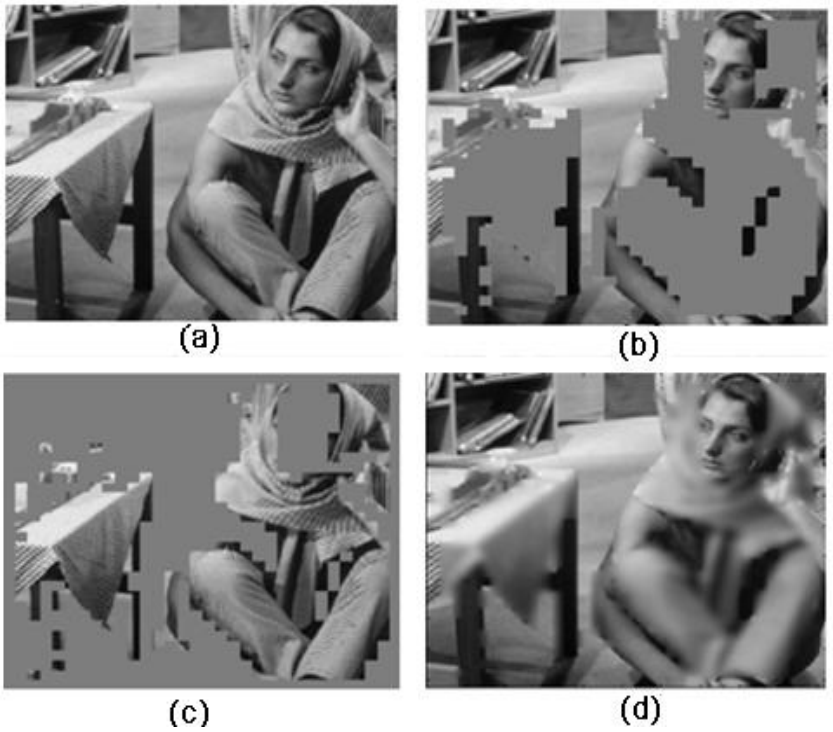


Figure 6.2: (a) The original Barbara image; (b) non-structure image areas; (c) structural components; (d) the smoothed non-structure image.

6.2 Classification of Structural Components

We design a scheme for classifying and extracting structural components for lossless image compression.

We make the decision of structure prediction on a block basis. More specifically, we partition the image into blocks (e.g., 4×4 blocks). We then classify these blocks into two categories: structure and non-structure blocks. Structure blocks will be encoded with super-spatial prediction. As discussed in Section 6.1, we will encode the non-structure blocks (regions) using conventional image compression technique such as CALIC after smoothness. Figure 6.3 shows the procedure of our classification scheme.

After CALIC prediction, which will be described in a little detail later on, we compare the predicted image against the original one on a block basis. If the block difference is larger than a threshold, this block is considered as a structure block, otherwise, a non-structure block. For simplicity, we use SAD (sum of absolute difference) to measure the block difference. The threshold is chosen based on the percentage of structure blocks that we want to encode with super-spatial prediction. For example, if we choose 40% of blocks as structure blocks, we simply choose the top 40% blocks with the highest SAD values. Note that a binary classification map with "1" for structure blocks and "0" for non-structure blocks needs to be encoded and sent to the decoder.

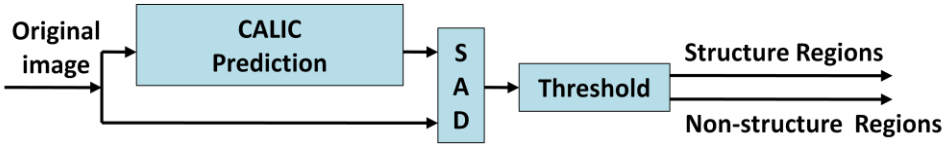


Figure 6.3: Classification of structure and non-structure blocks.

6.3 Optimum Prediction of Structural Components

In this section, we explain the optimum reordering, prediction, and efficient encoding of structural components.

6.3.1 Optimum Reordering of Structural Blocks

Let $\{S_m | 1 \leq m \leq M\}$ be the set of structural blocks that have been extracted from the image. We recognize that a structural block S_m may not be best predicted by its previous neighbor S_{m-1}. This implies that we can re-organize these structural blocks to maximize the overall prediction efficiency. We also allow one block to be used as prediction reference for multiple blocks. This becomes a minimum spanning tree problem. Figure 6.5 shows one example of such prediction. Each node represents a structural block. Nodes 3 and 5 are predicted from Node 1. Nodes 4 and C are predicted from Node 3. To this end, we use the energy (e.g. SAD) of the prediction residual as the distance measure. We use the Prim's algorithm to construct the minimum spanning tree. The tree structure and node information are losslessly encoded and sent to the decoder.

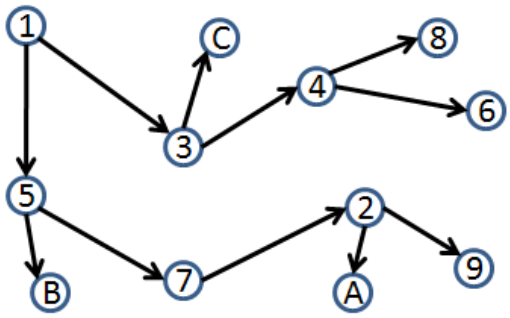


Figure 6.4: Minimum spanning tree for optimum prediction of structural blocks.

If we restrict one block can be used as reference for only one other block (i.e., chain prediction instead of tree-based prediction), this becomes a Hamilton path problem: finding an optimum path to traverse all structure blocks such that the overall prediction residual energy is minimized. This is a NP-hard problem.

6.3.2 Optimum Prediction of Structural Blocks

In order to minimize the prediction residual energy of a structural block, five prediction modes are proposed: mode0, 1, …, 4, which are described below. As shown in Figure 6.6, the reference block and the current block to be predicted are denoted as A and B respectively.

Mode 0: In this mode, A is directly used as the prediction of B.

Mode 1: This mode captures the horizontal structure of B. In this mode, the rightmost column of A is first used as the prediction of the leftmost column of B. The leftmost column of B is then reconstructed and used as the prediction of the rest columns of B.

Mode 2: This mode captures the vertical structure of B. In this mode, the bottom row of A is first used as the prediction of the top row of B. The top row of B is then reconstructed and used as the prediction of the rest rows of B.

Mode 3: This mode captures the diagonal-down-right structure of B. In this mode, the top row and leftmost column of A are first used as the predictions of the top row and the leftmost column of B respectively. The top row and the leftmost column of B are then

reconstructed and used as the predictions of the rest pixels of B in the diagonal-down-right direction.

Mode 4: This mode captures the diagonal-down-left structure of B. In this mode, the top row and rightmost column of A are first used as the predictions of the top row and the rightmost column of B respectively. The top row and the rightmost column of B are then reconstructed and used as the predictions of the rest pixels of B in the diagonal-down-right direction.

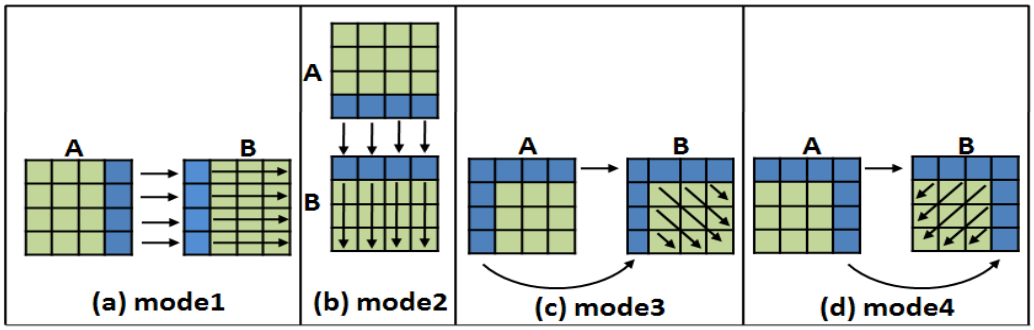


Figure 6.5: Four of five prediction modes of structural blocks.

The mode that results in the minimal SAD between a structural block and its prediction is selected as the best prediction mode. The residuals of the structural blocks are encoded. The information of best prediction modes is encoded as well as overhead information to help the reconstruction.

6.4 Conditional Indexing of Structural Components

The reordering indices of structural blocks need to be encoded as overhead information. The simplest way to do so is directly encoding the reordering indices using an entropy coding technique, such as Huffman coding or arithmetic coding. However, in order to encode the reordering indices more efficiently, we propose a so-called conditional indexing scheme. The basic idea behind this scheme is that the spatial correlation of a structural block and its encoded reconstructed neighbors is used to reduce the space of the reordering indices. We illustrate the conditional indexing scheme using an example as follows.

Assume there are five structural blocks. They are denoted as {A, B, C, D, E} in original raster scan order. They are reordered as {D, A, E, B, C} after optimum reordering, which implies that the reordering indices are {3, 0, 4, 1, 2}. As shown in Figure 6.7, in the conditional indexing scheme, the new reordering index of A is first generated as follows: The neighboring row above A and the column left to A are used as the predictions of the top row and the leftmost column of A respectively. The SAD between the top row and the leftmost column of A and their predictions are then calculated. The similar procedure is repeated to B, C, D and E to obtain the SADs between the neighboring row and column of A and the top row and the leftmost column of those blocks. Once the five SADs are obtained, they are sorted in increasing order. The position of A's SAD in the sorted SADs is finally determined and used as A's new reordering index. The similar procedure of generating A's new reordering index is repeated to B, C, D and E consecutively and the new reordering indices of {A, B, C, D, E} can be finally generated, for example, {1, 2, 1, 0, 2}. Compared to the old reordering indices, {3, 0, 4, 1, 2}, whose space size is 5 and entropy 2.32, the new reordering indices

has smaller space size, 3, and entropy 1.52, which implies that fewer bits are needed to encode the new reordering indices. At the decoder side, a similar procedure is first conducted to obtain the sorted SADs for each structural block location in the image. The corresponding block is then located based on its new reordering index.

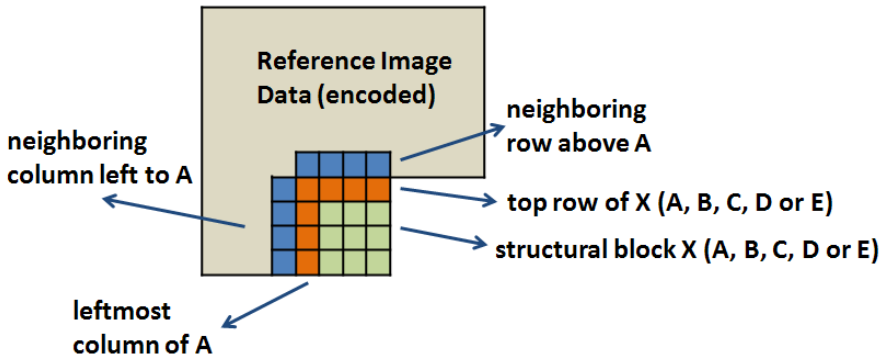


Figure 6.6: Generation of the conditional index of structural block A.

6.5 Experimental Results

We report results on lossless image compression. We compare the proposed scheme with CALIC [58], a very efficient lossless image compression scheme which outperforms other state-of-the-art coding methods, such as JPEG-LS/LOCO-I [62].

Figure 6.8 shows the test images used in lossless compression. Table 1 shows the performance comparison (in terms of bit rates in bits per pixel, bpp). It also shows the percentage of bits used for the smooth image area, structural components, and other overhead, such as index bits. It can be seen that the proposed scheme outperforms CALIC and save the overall bit rate by up to 0.3 bpp.

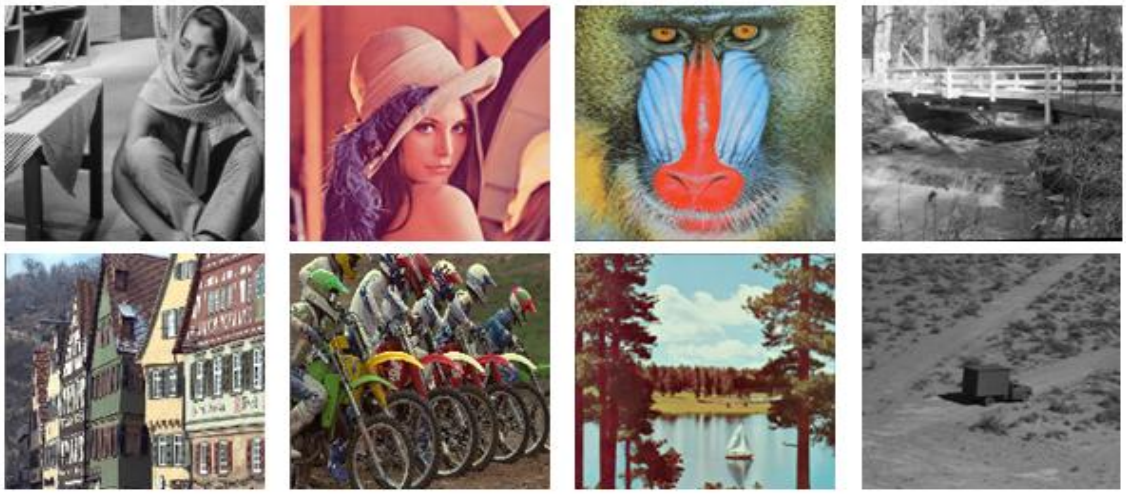


Figure 6.7: Test images from USC and Kodak image databases.

Table 6.1: Performance comparison with CALIC (bit rate in bpp) and percentage of bits used for the smooth image area, structural components, and other overhead, such as index bits.

Image	CALIC	This Work	Percentage of Bits		
			Smooth	SC	Others
Barbara	4.58	4.42	53%	42%	5%
Lena	4.11	3.96	55%	40%	5%
Baboon	7.36	7.01	62%	35%	3%
Bridge	4.46	4.21	61%	33%	6%
KodIM08	5.04	4.90	57%	39%	4%
KodIM05	4.85	4.78	50%	44%	6%
Lake	4.88	4.81	56%	40%	4%
Truck	4.46	4.24	61%	35%	4%

6.6 Conclusion

In this work, we have developed an efficient lossless image compression scheme based on inter-structure prediction. We classified an image into structural components

and non-structure image areas. The non-structure image areas, after smoothing, are encoded with existing image compression schemes, such as CALIC. Based on a minimum spanning tree, we developed an optimum prediction scheme for structural components. Our extensive experimental results demonstrated that the proposed scheme is very efficient in lossless image compression.

One major drawback of the proposed method is its computational complexity. The major complexity lies in the construction of the minimum spanning tree for optimum prediction. In our future work, we will investigate some sub-optimum algorithms to provide a good trade-off between complexity and compression performance.

The success of the super-spatial prediction in lossless image compression implies a potential application in lossy image compression. As our future work, we will apply our super-spatial prediction to lossy image compression with some necessary modifications.

CHAPTER 7

WILDLIFE ANIMAL INTERACTION DETECTION

7.1 Introduction

Video accounts of interactions between wildlife animals could aid in disease transmission modeling by revealing the frequency and nature of interactions. Many wildlife diseases, such as chronic wasting disease (CWD), have been a central challenge to wildlife managers and DeerCam could provide crucial information about contact rates necessary to understand potential disease spread.

However, the interactions between wildlife animals, such as deer, are with low frequency and the search of those interactions from large amounts of videos is very human-labor-expensive. For example, we collected a total of 96.4 hours of video data from our past four DeerCam deployments and observed only 11 interactions between white-tailed deer comprising a total of 22.8 minutes. The interaction frequency is 0.39% (22.8 minutes / 96.4hours) [72]. It will be an unbearable load for wildlife researchers to search the interactions between wildlife animals in thousands of hours of video data, which is not uncommon. Therefore, it will be very valuable to develop a tool that can automatically search a large amount of video data and output the video segments with interactions between wildlife animals. This is our initial motivation of developing a wildlife animal interaction detection algorithm.

The wildlife animal interaction detection can have its application in the energy consumption minimization and data storage saving in DeerCam as well. For example, some wildlife researchers may be only interested in interactions between wildlife animals rather than other activities. In this case, the animal interaction detection unit can be integrated in DeerCam to detect the animal interactions between the host animal, whom DeerCam was mounted on, and other animal individuals. The digital camera of DeerCam only encodes the video only when the animal interactions are detected. Otherwise, the camera does not encode the video. By this way, the energy of DeerCam can be reserved to encode the videos of animal interactions and the sensor data rather than other data that is not interesting to the wildlife researchers. The data storage room in DeerCam is limited and it can be saved to store only those interesting videos and sensor data as well.

7.2 Approach Overview

7.2.1 Problem Simplification

Figure 7.1: Three snapshots of a deer interaction in a video captured by DeerCam.

From December 2006 to March 2008, we had four DeerCam deployments. We collected nearly 100 hours of video data. We observed 11 interactions between host deer and other white-tailed deer. Three snapshots of such an interaction are shown in Figure 7.1. We can see from these snapshots that the host animal is usually invisible in the video frames and if animals are detected we can be certain that there must be an interaction in

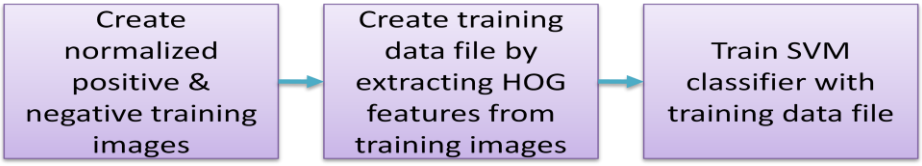
those frames. Therefore, the problem of animal interaction detection is simplified to a problem of animal detection.

The sliding window detector based on HOG (Histogram of Oriented Gradients) feature descriptor and SVM (Support Vector Machine) classifier provides the state-of-the-art performance for human detection [73, 74]. Naturally, the question can be raised: Can we borrow this human detection algorithm for our animal detection?

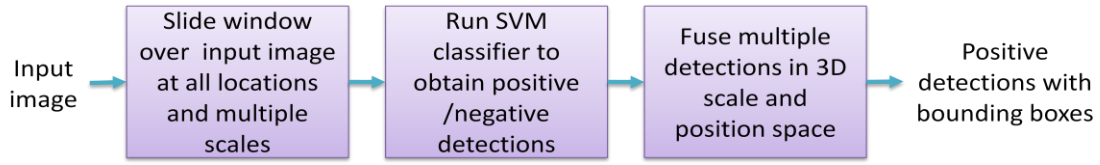
It will be a challenging task to detect an animal's body in images due to the large variety of poses. However, it will be much easier to detect an animal's face. For example, in Figure 7.1, the deer has different poses in those three frames. In frame 3, it is even partially visible, which makes the detection of its existence more difficult. However, it will be a much easier task to detect its face in those frames. We can be certain that there must be an interaction between two deer if deer faces are detected in those frames.

Based on above discussion, we simplified the task of animal interaction detection to that of animal detection then further to that of animal face detection. Since the host animals of our current DeerCam system are deer and based on our observations, they usually have interactions only with deer. Therefore, we currently apply our animal face detection to deer only. We can easily extend our deer face detection to other animal spices as well in future.

7.2.2 Overview of Approach



(a)



(b)

Figure 7.2: Training and detection phases of our deer face detection (a) training phase, (b) detection phase.

As mentioned in the previous section, we adopt the methodology of the sliding window detector based on HOG feature and SVM classifier to our deer face detection. The framework is shown in Figure 7.2.

In the training phase, the first step is to create the training data. The positive samples are collected by manually cropping deer faces from images containing deer and resized to fixed resolution. The negative samples are randomly cropped from the same set of images at the regions not containing deer faces. The HOG features are then extracted from these positive and negative sample images and stored in a training data file. The training data are finally fed into the SVM classifier to train the classifier.

During the detection phase, the input image is scanned by a window of particular size at all locations and multiple scales. For each location and scale, the HOG feature is extracted for that windowed region and the SVM classifier is run to produce deer face or non-deer face decision. The image regions containing deer faces and their neighboring

regions usually produce multiple firings. It is necessary to fuse these overlapping firings into a single final detection. We adopt multi-scale object localization technique to do so.

7.3 HOG Feature Descriptor

7.3.1 HOG Feature

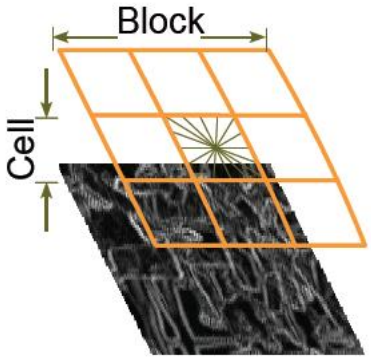


Figure 7.3: HOG feature descriptor [73].

Histogram of Oriented Gradient descriptor provides a dense overlapping representation of image regions by calculating image gradient orientations in histogram. As shown in Figure 7.3, an image region is divided into smaller rectangle ones, called "cell". For each cell, gradient is calculated for every pixel and its magnitude is used to vote to one of a number of predetermined orientation bins. The orientation histograms of multiple adjacent cells, called "block", are accumulated and normalized over the block to introduce better resistance to illumination variation.

7.3.2 HOG Feature Extraction Procedure

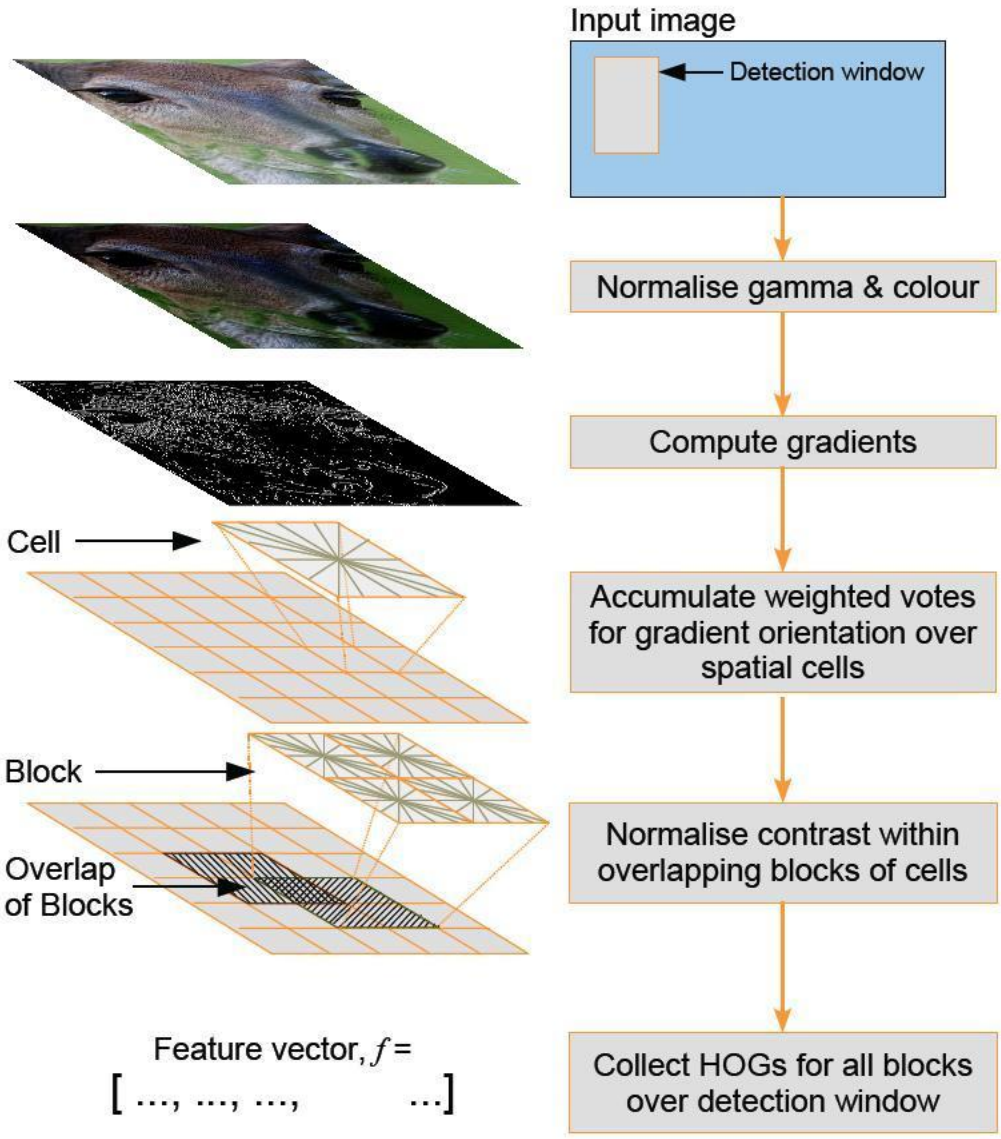


Figure 7.4: The procedure of HOG feature extraction [73].

HOG feature is computed for every detection window of an input image. The procedure is shown in Figure 7.4. The first step applies gamma and color normalization to increase robustness against illumination effects. The second step calculates gradient for every pixel in each cell and votes to the orientation histogram. The third step normalizes

the orientation histograms of the cells within the overlapping block. The final step collects HOGs for all blocks over the detection window to generate a feature vector.

7.4 SVM Classifier

The SVM classifier is currently one of the predominant classifiers due to its good performance and efficiency. A SVM classifier constructs a hyperplane in a multi-dimensional space. The hyperplane of a good SVM classifier can produce an optimal separation that has the largest distance between the training data points of positive and negative classes. We use a dense version of SVMLight for our application that minimizes the memory consumption of large feature descriptors.

7.5 Multi-Scale Object Localization

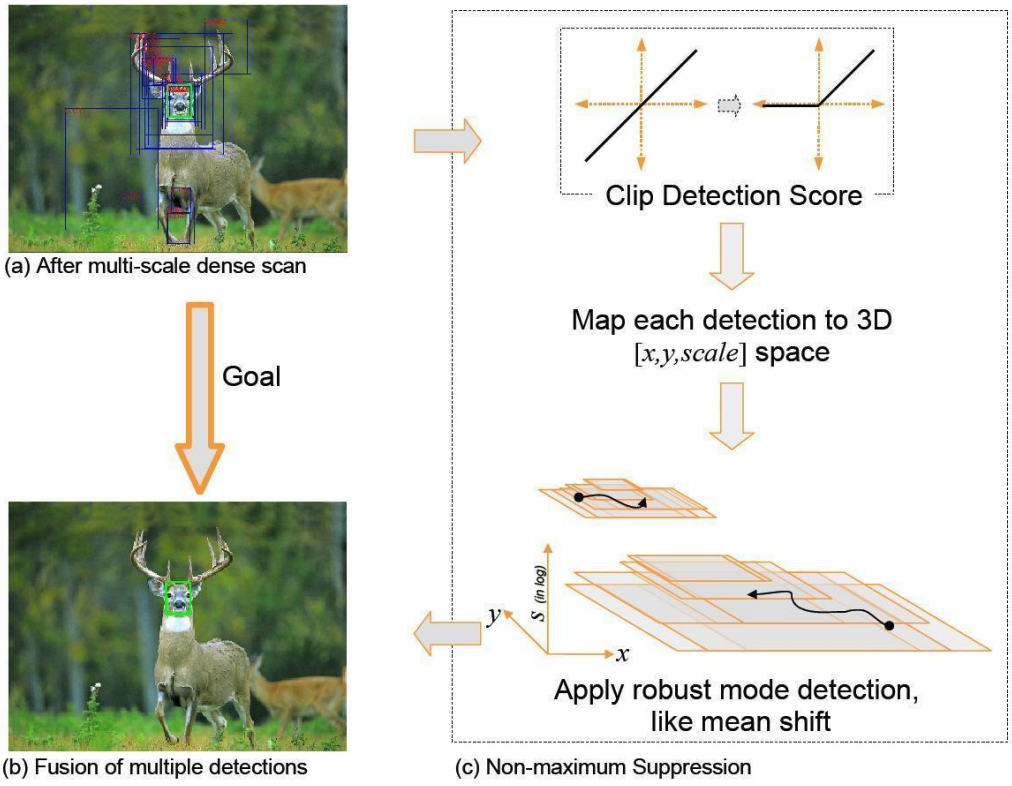


Figure 7.5: Non-maximum suppression for fusion of multiple overlapping detections [73].

Non-maximum suppression in [73] is adopted to fuse overlapping detections. The overview of this algorithm is provided in Figure 7.5. Readers are advised to refer [73] for a detailed description of this algorithm. Figure 7.5(a) is a typical result after scanning the SVM classifier over the input image at all locations and multiple scales. There are multiple overlapping detections at the region that contains the positive object. Figure 7.5(b) shows the result of fusing the overlapping detections into a final single detection using non-maximum suppression. The procedure of non-maximum suppression is demonstrated in Figure 7.5(c). The first stage converts the scores of the SVM classifier into non-negative values via clipping function. The second stage maps all detections to a 3-D location and scale space. The final stage applies mean-shift algorithm to cluster the overlapping detections together.

7.6 Experimental Results

7.6.1 Training Data Sets

We collected 104 deer images with different resolutions from internet using Google search tool, along with their right-left reflections. Five of those images are shown in Figure 7.6 below for example. From those images, we manually label/crop deer faces and resize them to 72×96 as positive training samples, as shown in Figure 7.7 for example.

Larger size of sample images produces better detection performance but higher computational complexity. We choose size of 72×96 to make a good tradeoff between performance and complexity. The negative samples have the same size and are randomly cropped from the same set of images at the regions that do not contain deer faces. Some of those negative samples are shown in Figure 7.8 for example.

Figure 7.6: Examples of deer images from which deer face images are cropped then normalized as positive training samples.

Figure 7.7: Examples of positive training samples.

Figure 7.8: Examples of negative training samples that are cropped from the deer images at fixed resolution.

7.6.2 Parameter Setting

In our deer face detector, we use the following parameters: RGB color space; no gamma correction; $[-1\ 0\ 1]$ gradient filter for both vertical and horizontal directions; 9 orientation bins in $0 - 180^0$ for gradient histogram; 3×3 cell block of 4×4 pixel cells; block stride of 6×6 pixels; sliding detection window of 72×96 pixels with 4 pixels of margin on all four sides; sliding stride of 8×8 pixels.

7.6.3 Experimental Results

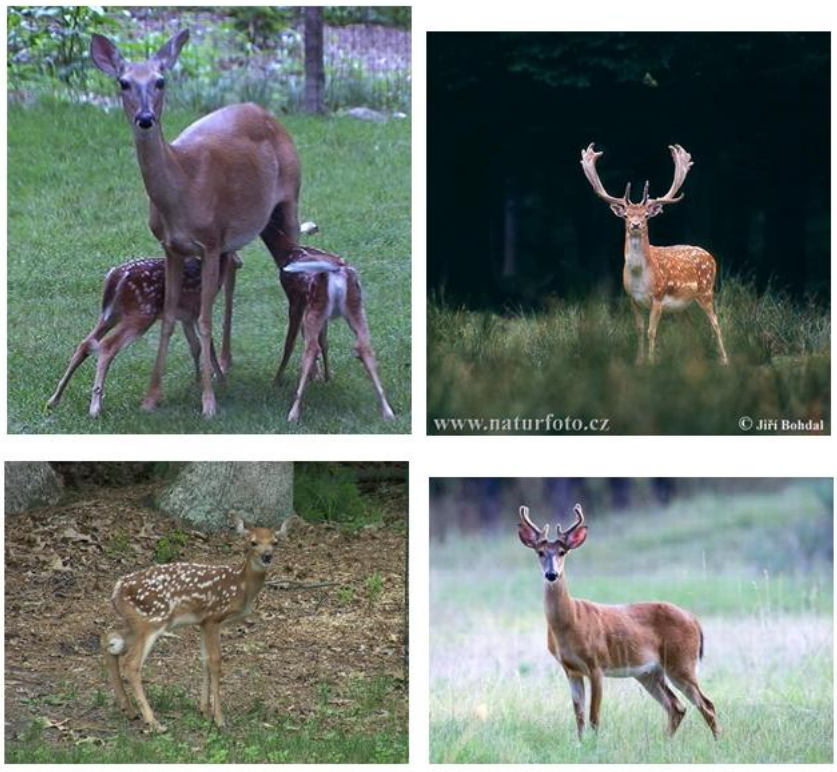
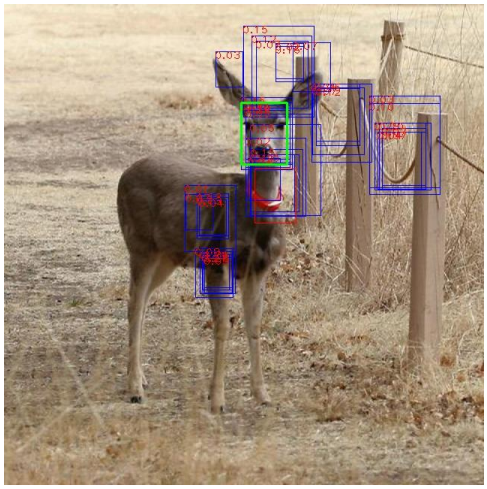


Figure 7.9: Examples of normalized test images with width of 640 pixels.

We test our animal face detector on 26 deer images that we collected from internet using Google search tool. These images originally have different resolutions but are normalized to images whose widths are 640 pixels while their original width-height ratios are unchanged. Some of the normalized images are shown in Figure 7.9 for example. The deer faces in 23 images are correctly detected and some examples are shown in Figure 7.10. The deer faces in 3 images are not correctly detected and two examples are shown in Figure 7.11. The detection rate of our deer face detector for this test image set is 88.5% (23/26). Figure 7.12 is the ROC curve of our deer face detector.



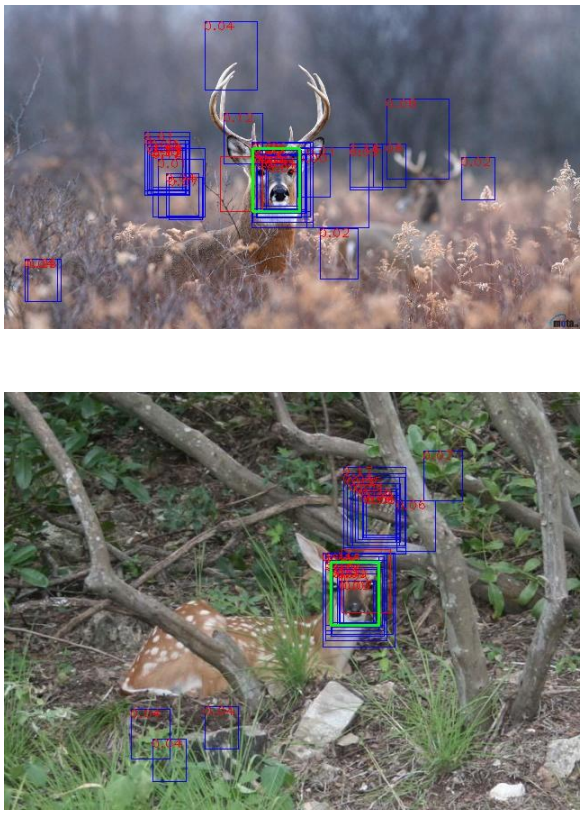
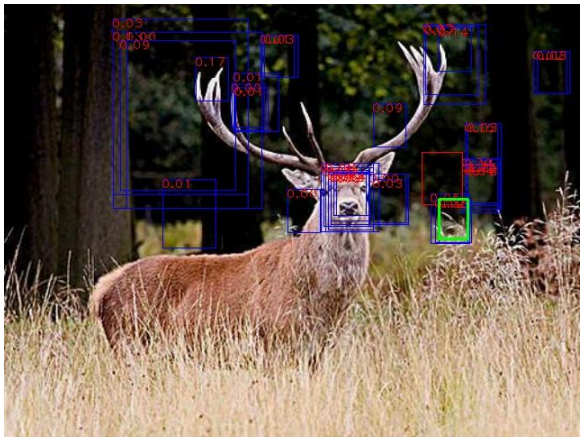


Figure 7.10: Examples of correct detection.



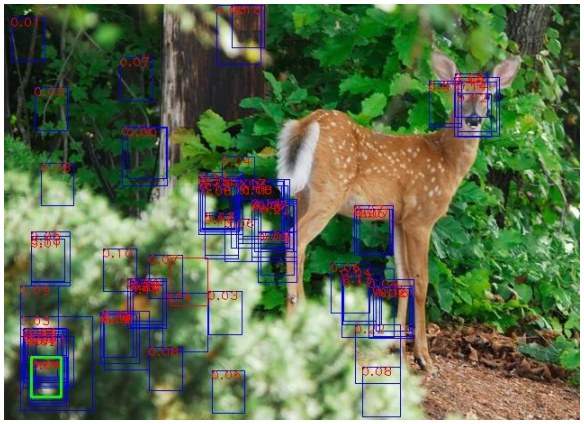


Figure 7.11: Examples of incorrect detections.

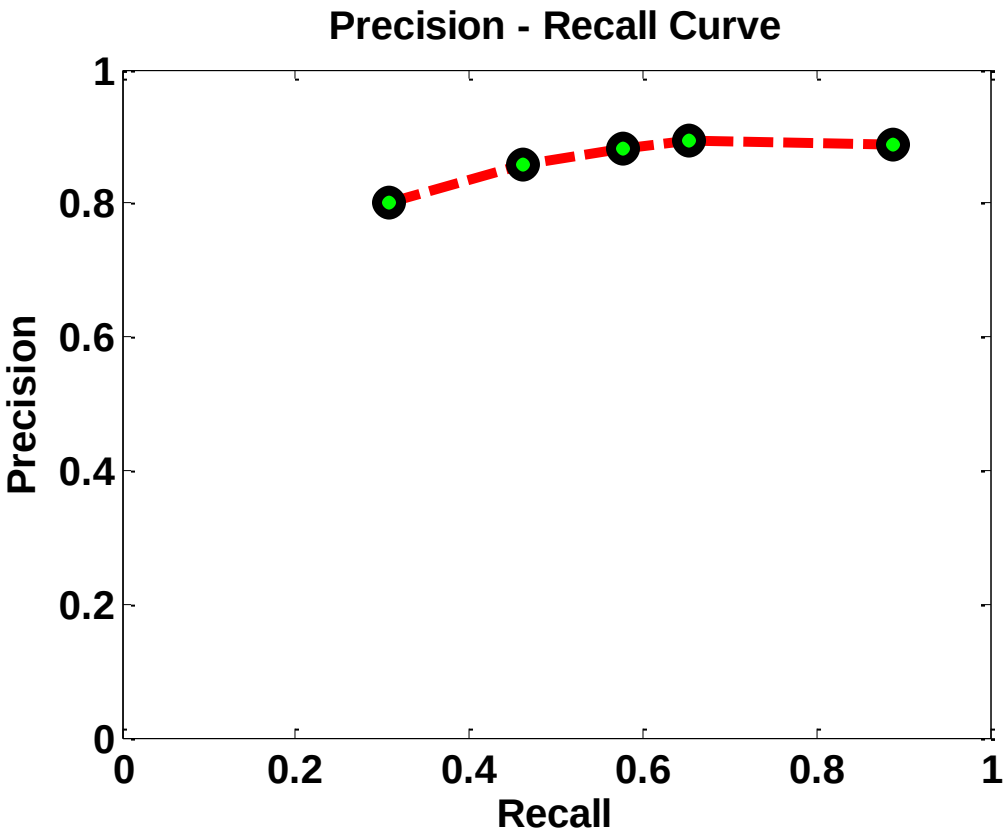


Figure 7.12: ROC curve of our deer face detector.

7.7 Conclusion

In this work, we developed an animal interaction detection method using supervised learning methods. By integrating this detection functionality into our DeerCam, it is able to detect events of animal interactions which will trigger the on-board video encoder to encode and save interesting videos. This will significantly reduce the amount of video data to be encoded and improve the utility of the visual sensing data. It will also provide important reference for sub-sequent wildlife behavior analysis.

CHAPTER 8

CONCLUDING REMARKS AND FUTURE WORKS

In this chapter, we first summarize our works presented in Chapter 2 – 7, then introduce our future works.

8.1 Concluding Remarks

In this research, we focus on algorithm development and system design for resource-efficient portable video communication system design and their application in wildlife monitoring and interaction tracking. The capability of seeing what an animal sees in the field is very important for wildlife activity monitoring and research. We design an integrated video and sensor system, called DeerCam and mount it on free-ranging animals so as to collect important video and sensor data about their activities in the field. From the video and sensor data collected by DeerCam, wildlife researchers will be able to extract a wealth of sciatic data for studying the behavior patterns of wildlife species and understanding the dynamic of wildlife systems.

In this dissertation, we focus on the four tightly coupled research issues:

(1) Energy minimization.

(2) Intelligent resource allocation and utility maximization.

(3) Efficient image encoder.

(4) Animal interaction detection for event-driven wildlife monitoring.

In Chapter 2, we presented our various approaches to address the first two challenges: *Energy minimization* and *Intelligent resource allocation and utility maximization* of energy-aware portable video communication system for wildlife activity monitoring. We developed joint power-rate-distortion (P-R-D) algorithms for complexity control and energy minimization. We also developed methods to maximize the utility function under resource constraints. We demonstrated that by incorporating the third dimension of power consumption into conventional R-D analysis, P-R-D analysis gives us one extra dimension of flexibility in resource allocation and energy minimization, and allows us to significantly reduce energy consumption.

In Chapter 3 - Chapter 6, we proposed several approaches to address the third challenge: *Efficient image compression*. More specifically, in Chapter 3, we developed an image compression algorithm based on structure learning and prediction. When learning local image structures, we attempt to find a small number of basis vectors whose linear combinations are able to closely approximate local image patches. By extrapolating these linear combination coefficients, we can efficiently predict neighboring pixels of the local image patch. To design an efficient image encoder based structure prediction, we introduced the ideas of separation of an image into structure, transition, and structure regions and smooth-painting. Our experimental results demonstrate that the proposed algorithm outperforms JPEG2000 image compression.

In Chapter 4, we presented an efficient lossless image compression algorithm based on structure learning and prediction. We classified an image into structure regions and non-structure regions. The structure regions are encoded with structure prediction while the non-structure regions are encoded with existing image compression schemes, such as

CALIC. Our extensive experimental results demonstrated that the proposed scheme is very efficient in lossless image compression, especially for images with significant structure components.

In Chapter 5, we designed a simple yet efficient image prediction scheme, called super-spatial prediction. It is motivated by motion prediction in video coding, attempting to find an optimal prediction of structure components within previously encoded image regions. When compared to VQ-based image encoders, it has the flexibility to incorporate multiple H.264-style prediction modes. When compared to other neighborhood-based prediction methods, such as GAP and H.264 Intra prediction, it allows the block to find the best match from the whole image which significantly reduces the prediction residual by up to 79%. We classified an image into structure regions and non-structure regions. The structure regions are encoded with super-spatial prediction while the non-structure regions are encoded with existing image compression schemes, such as CALIC. Our extensive experimental results demonstrated that the proposed scheme is very efficient in lossless image compression.

In Chapter 6, we proposed an efficient lossless image compression scheme based on inter-structure prediction. We classified an image into structural components and non-structure image areas. The non-structure image areas, after smoothing, are encoded with existing image compression schemes, such as CALIC. Based on a minimum spanning tree, we developed an optimum prediction scheme for structural components. Our extensive experimental results demonstrated that the proposed scheme is very efficient in lossless image compression.

In Chapter 7, we addressed the fourth challenge: *Animal interaction activity detection*. We developed an animal interaction detection method using supervised learning methods. By integrating this detection functionality into our DeerCam, it is able to detect events of animal interactions which will trigger the on-board video encoding system to capture video samples. This will significantly reduce the amount of video data to be encoded and improve the utility of the visual sensing data. It will also provide important reference for sub-sequent wildlife behavior analysis.

8.2 Future Works

In our future works, we will explore various ideas to further reduce the computational complexity of the encoding algorithms proposed in Chapter 3 - 6, especially the structure learning and prediction. We will also study how the structure learning and prediction could be integrated with lifting-based wavelet transform for adaptive image prediction and transform.

In our future works, we shall develop fast and efficient algorithms to further reduce the complexity of super-spatial prediction (Chapter 5). We also notice that when the encoder switches between structure and non-structure regions, the prediction context is broken and it will degrade the overall coding performance. Therefore, we shall investigate more efficient schemes for context switching.

One major drawback of the lossless image compression algorithm based on inter-structure prediction (Chapter 6) is its computational complexity. The major complexity lies in the construction of the minimum spanning tree for optimum prediction. In our

future work, we will investigate some sub-optimum algorithms to provide a good trade-off between complexity and compression performance.

The success of the lossless image compressions presented in Chapter 4 – 6 implies their potential applications in lossy image compression. As our future work, we will apply them to lossy image compression with some necessary modifications.

We should also further improve our animal interaction detection method in aspects of complexity and detection performance.

APPENDIX A

In this appendix, we prove that the optimum basis functions which minimize the following approximation error

$$\min_{\{\varphi_1, \varphi_2, \cdots \varphi_K\}} \boldsymbol{E} = \sum_{n=1}^{N} \left\| \boldsymbol{X}_n - \sum_{k=1}^{K} (\boldsymbol{X}_n , \varphi_k) \cdot \varphi_k \right\|_2 \tag{A.1}$$

are the first K singular vectors of $\boldsymbol{X}$. Note that

$$\left\| \boldsymbol{X}_n - \sum_{k=1}^{K} (\boldsymbol{X}_n , \varphi_k) \cdot \varphi_k \right\|_2^2 = \boldsymbol{X}_n^t \boldsymbol{X}_n - \sum_{k=1}^{K} c_{nk}^2, \tag{A.2}$$

where $c_{nk} = (\boldsymbol{X}_n , \varphi_k)$. Note that

$$\sum_{n=1}^{N} \sum_{k=1}^{K} c_{nk}^2 = tr(\boldsymbol{X^t \Phi \Phi^t X}) = tr(\boldsymbol{\Phi^t X X^t \Phi}). \tag{A.3}$$

Here, $\boldsymbol{\Phi} = [\varphi_1, \varphi_2, \cdots \varphi_K]$ and we use the result $tr(AB) = tr(BA)$. According to (A.2) and (A.3), the problem in (A.1) becomes

$$\max_{\{\varphi_1, \varphi_2, \cdots \varphi_K\}} tr(\boldsymbol{\Phi^t X X^t \Phi}). \tag{A.4}$$

Suppose $\boldsymbol{X}$ has SVD decomposition $\boldsymbol{X} = \boldsymbol{U}_{L\times L}\, \boldsymbol{\Lambda}_{L\times N}\, \boldsymbol{V}_{N\times N}$. Then, we have $\boldsymbol{XX^t} = \boldsymbol{UDU^t}$ with $\boldsymbol{D} = diag[\lambda_1, \lambda_2, \cdots \lambda_L]$ where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$. Write $\boldsymbol{Z} = \boldsymbol{U^t}\, \boldsymbol{\Phi}$ which is an $L \times K$ matrix with orthonormal columns. According to the property of matrix trace, $tr(ABC) = tr(CAB)$, we have

$$tr(\boldsymbol{\Phi^t X X^t \Phi}) = \quad tr(\boldsymbol{Z^t D Z}) = tr(\boldsymbol{Z Z^t D}) = tr(\boldsymbol{PD}) = \sum_{i=1}^{L} P_{ii}\lambda_i, \tag{A.5}$$

where $\boldsymbol{P} = [P_{ii}] = \boldsymbol{Z Z^t}$. Note that $P_{ii} \geq 0$ and $P_{ii} \leq 1$ because $\boldsymbol{Z}$ is a projection matrix. We also have

$$\sum_{i=1}^{L} P_{ii} = tr(\boldsymbol{P}) = tr(\boldsymbol{Z Z^t}) = tr(\boldsymbol{Z^t Z}) = k, \tag{A.6}$$

because matrix $\boldsymbol{Z}$ is an $L \times K$ matrix with orthonormal columns. Therefore, the optimization problem in (A.4) becomes

$$\max_{\{P_{ii}\}} \sum_{i=1}^{L} P_{ii}\lambda_i, \;\; s.t. \;\; \sum_{i=1}^{L} P_{ii} = k, 1 \geq P_{ii} \geq 0. \tag{A.7}$$

Note that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$. The optimum solution of (A.7) is given by $P_{11} = P_{22} = \cdots P_{KK} = 1$ and $P_{ii} = 0$ for $L \geq i > K$. This implies that φ_i is the i-th column of matrix $\boldsymbol{U}$.

BIBLIOGRAPHY

[1] L.D. Mech and S.M. Barber, A critique of wildlife radio-tracking and its use in national parks: a report to the U.S. National Park Service, U.S. Geological Survey, Northern Prairie Wildlife Research Center, Jamestown, N.D. 78 pages, 2002.

[2] J.J. Millspaugh and J.M. Marzluff (editors). Radio Tracking and Animal Populations, Academic Press, San Diego, California, USA, 2001, 467 pages.

[3] L.D. Mech, A Handbook Of Animal Radio-tracking, Univ. of Minn. Press, p. 108, 1983.

[4] NASA Satellite Tracking of Threatened Species. http://sdcd.gsfc.nasa.gov/ISTO/satellite_tracking/, 2002.

[5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, ―Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with Zebranet,'' Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS-X), San Jose, CA, Oct. 5–9, 2002.

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, ―Wireless sensor networks: a survey,'' Computer Networks, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[7] H. Gharavi and K. Ban, ―Vision-based ad-hoc sensor networks for tactical operations," World Wireless Congress, 3G Wireless 2002, San Francisco, May 2002.

[8] B. Thorstensen et al. –Electronic Shepherd: A Low-Cost, Low-Bandwidth, Wireless Network System," in Proc. Second Intl. Conference on Mobile Systems, Applications and Services, June 2004.

[9] J.M. Kahn, R.H. Katz, and K.S.J. Pister, –Mobile Networking for Smart Dust," ACM/IEEE Intl. Conf. on Mobile Computing and Networking (Mobi-Com 99), Seattle, WA, Aug. 17–19, 1999.

[10] G. Pottic and W. Kaiser, –Wireless integrated network sensors," Communications of the ACM, vol. 43, no. 5, pp. 51–58, May 2000.

[11] UC Davis Wildlife Health Center, Southern California Puma Project, http://www.vetmed.ucdavis.edu/whc/scp/, 2004.

[12] R. Szewczyk, E. Osterwell, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, –Habitat monitoring with sensor networks," Communications of ACM, vol. 47, no. 6, p. 3440, 2004.

[13] R. Szewczyk, A. Mainwaring, J. Polastre, and D. Culler, –An Analysis of a Large Scale Habitat Monitoring Application," in Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (Sen-Sys), Nov. 3–5, 2004.

[14] K. Mayer, K. Taylor, and K. Ellis, –Cattle Health Monitoring Using Wireless Sensor Networks," The 2nd IASTED International Conference on Communication and Computer Networks, Cambridge Massachusetts, Nov. 8–10, 2004.

[15] J. Beringer, J.J. Millspaugh, J. Sartwell, and R. Woeck, –Real-time video recording of food selection by captive white-tailed deer," Wildlife Society Bulletin, vol. 32, no. 3, 2004.

[16] T.L. Cutler and D.E. Swann, –Using remote photography in wildlife ecology: a review," Wildlife Society Bulletin, no. 27, pp. 571–581, 1999.

[17] D.I. King, R.M. DeGraaf, P.J. Champlin, and T.B. Champlin, –A new method for wireless video monitoring of bird nests," Wildlife Society Bulletin, no. 29, pp. 349–353, 2001.

[18] A.B. Cooper and J.J. Millspaugh, Accounting for variation in resource availability and animal behavior in resource selection studies. Pages 243–274 in J. J. Millspaugh and J. M. Marzluff, editors. Radio Tracking and Animal Populations. Academic Press, San Diego, California, USA, 2001.

[19] Z. He, Y. Liang, L. Chen, I. Ahmad, and D. Wu, –Power-rate-distortion analysis for wireless video communication under energy constraint," IEEE Transactions on Circuits and System for Video Technology, May, 2005.

[20] T. Wiegand, –Text of Committee Draft of Joint Video Specification (ITU-T Rec. H.264—ISO/IEC 14496-10 AVC)," Document JVTC167, 3rd JVT Meeting, Fairfax, Virginia, USA, May 6–10, 2002.

[21] T. Burd and R. Broderson, –Processor Design for Portable Systems," Journal of VLSI Signal Processing, vol. 13, no. 2, pp. 203–222, Aug. 1996.

[22] B. Zeng, R. Li, and M.L. Liou, "Optimization of fast block motion estimation algorithms," IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 6, pp. 833–844, Dec. 1997.

[23] H.D.J.N. Aldridge and R.M. Bringham, "Load carrying and maneuverability in an insectivorous bat: A test of the 5% rule of radio-telemetry," Journal of Mammalogy, vol. 69, pp. 379–382.

[24] J.C. Withey, et al, "Effects of tagging and location error in wildlife radiotelemetry studies," Radio Tracking and Animal Populations (Millspaugh, J.J. and Marzluff, J.M., eds), pp. 43–75, Academic Press, 2001.

[25] J.R. Fischer, L.H. Creekmore, R.L. Marchinton, S.J. Riley, S.M. Schmitt, and E.S. Williams, "External review of chronic wasting disease management in Wisconsin,'' Oct. 10, 2003.

[26] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Mobile networking for smart dust," in ACM MOBICOM, Seattle, WA, Aug. 1999.

[27] http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Stargate_Datasheet.pdf

[28] C.B. Margi, V. Petkov, K. Obraczka, and R. Manduchi, "Characterizing energy consumption in a visual sensor network testbed," Proceedings of 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM), Mar. 2006, pp. 8–15.

[29] X. Lu, Y. Wang, and E. Erkip, "Power efficient H.263 video transmission over wireless channels," Proceedings of 2002 International Conference on Image Processing, Rochester, New York, Sept. 2002.

[30] K. Hyungjoon, N. Kamaci, and Y. Altunbasak, "Low-complexity rate-distortion optimal macroblock mode selection and motion estimation for MPEG-like video coders," IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 7, pp. 823–834, July 2005.

[31] I.M. Pao and M.T. Sun, "Statistical Computation of Discrete Cosine Transform in Video Encoders," Journal of Visual Communication and Image Representation, vol. 9, no. 2, pp. 163–170, June 1998.

[32] D.S. Turaga, M. van der Schaar, and B. Pesquet-Popescu, "Complexity scalable motion compensated wavelet video encoding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 8, pp. 982–993, Aug. 2005.

[33] J. Chen and K.J.R. Liu, "Low-power architectures for compressed domain video coding co-processor," IEEE Transactions on Multimedia, vol. 2, no. 2, pp. 111–128, June 2000.

[34] J. Villasenor, C. Jones, and B. Schoner, "Video Communications using Rapidly Reconfigurable Hardware," IEEE Transactions on Circuits and Systems for Video Technology, vol. 5, pp. 565–567, Dec. 1995.

[35] P. Agrawal, J.-C. Chen, S. Kishore, P. Ramanathan, and K. Sivalingam, "Battery power sensitive video processing in wireless networks," Proceedings IEEE PIMRC'98, Boston, Sept. 1998.

[36] W.P. Burleson, P. Jain, and S. Venkatraman, "Dynamically Parameterized Architecture for Power-Aware Video Coding: Motion Estimation and DCT," Proceedings of the Second USF International Workshop on Digital and Computational Video, 2001.

[37] V. Akella, M. van der Schaar, and W.-F. Kao, "Proactive Energy Optimization Algorithms for Wavelet-Based Video Codecs on Power-Aware Processors," IEEE International Conference on Multimedia and Expo, pp. 566–569, July 6–8, 2005.

[38] Intel Inc, "Intel XScale Technology," http://www.intel.com/design/intelxscale.

[39] D.G. Sachs, W. Yuan, C.J. Hughes, A.F. Harris, S.V. Adve, D.L. Jones, R.H. Kravets, and K. Nahrstedt, "GRACE: A Cross-Layer Adaptation Framework for Saving Energy," Sidebar in IEEE Computer, special issue on Power-Aware Computing, Dec. 2003, pp. 50–51.

[40] D.G. Sachs, S. Adve, and D.L. Jones, "Cross-layer Adaptive Video Coding to Reduce Energy on General-Purpose Processors," Proceedings of the International Conference on Image Processing, 2003 (ICIP '03), Barcelona, Spain, Sept. 2003.

[41] Z. He and D. Wu, "Resource allocation and performance limit analysis of wireless video sensors," IEEE Transactions on Circuits and System for Video Technology, vol. 16, no. 5, pp. 590–599, May 2006.

[42] W. Cheng, X. Chen, and Z. He, "Doubling of the operational lifetime of portable video communication devices using power-rate-distortion analysis and control," Proceedings of International Conference on Image Processing, Atlanta, GA, Oct. 2006.

[43] V. Akella, M. van der Schaar, and W.-F. Kao, "Proactive Energy Optimization Algorithms for Wavelet-Based Video Codecs on Power-Aware Processors," IEEE International Conference on Multimedia and Expo, July 6–8, 2005, pp. 566–569.

[44] T.S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, New Jersey, 1996.

[45] M. Vetterli, and J. Kovacevic, "Wavelets and subband coding," *Prentice Hall Englewood Cliffs*, NJ 1995.

[46] M. N. Do, and M. Vetterli, "The finite ridgelet transform for image representation," *IEEE Trans. Image Processing*, vol. 12, no. 1, pp. 16-8, Jan. 2003.

[47] D. S. Taubman, and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwar Academic Publishers, 2002.

[48] J.L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.,* vol. 11, no. 6, pp. 670-84, Jun. 2002.

[49] M. N. Do, and M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, IEEE Transactions Image on Processing, vol. 14, no. 12, pp. 2091-2106, Dec. 2005.

[50] W. Ding, F. Wu, and S. Li, "Lifting-based wavelet transform with directionally spatial prediction," in *Proc. Picture Coding Symposium 2004*, San Francisco, CA, USA, Dec. 2004.

[51] D. Wang, L. Zhang, A. Vincent, and F. Speranza, –Curved wavelet transform for image coding," *IEEE Transactions on Image Processing*, vol. 15, No. 8, pp. 2413-2421, Aug. 2006.

[52] D. Taubman, and A. Zakhor, –Orientation adaptive subband coding of images," *IEEE trans. on Image Processing,* vol. 3, no 4, 421-437, July 1994.

[53] P. Vandergheynst, and J.F. Gobbers, –Directional dyadic wavelet transforms: Design and algorithms," *IEEE Trans. Image Process*, vol. 11, no. 4, pp. 363-72, Apr. 2002.

[54] M. J. Weinberger, G. Seroussi, and G. Sapiro, –The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS", *IEEE Transactions on Image Processing*, Vol. 9, No. 8, August 2000, pp.1309-1324.

[55] X. Wu, and N. Memon, –Context-Based, Adaptive, Lossless Image Coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp.437-444, April. 1997.

[56] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, –Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, June, 2003.

[57] Jasper JPEG2000 encoder, http://www.ece.uvic.ca/ mdadams/jasper/.

[58] X. Wu, and N. Memon, –Context-Based, Adaptive, Lossless Image Coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp.437-444, April. 1997.

[59] http://jpeg2000.epfl.ch/download/jj2000_4.1-src.zip.

[60] http://www.hpl.hp.com/loco/jlsimV100.zip.

[61] http://www.ece.mcmaster.ca/~xwu/calicexe/calic8e.exe.

[62] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS", *IEEE Trans. Image Process.*, vol. 9, no. 8, pp.1309-1324, Aug. 2000.

[63] J. Ziv, and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. 23, no. 3, pp. 337-343, May 1977.

[64] J. Ziv, and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Trans. Inform. Theory*, vol. 24, pp. 530-536, Sep. 1978.

[65] V. Sitaram, C. Huang, and P. Israelsen, "Efficient codebooks for vector quantization image compression with an adaptive tree search algorithm," *IEEE Trans. Commun.*, vol. 42, no. 11, Nov. 1994.

[66] J. Storer, "Lossless image compression using generalized LZ1-type methods," in *Proc. IEEE Data Compression Conference (DCC*'96), pp. 290-299.

[67] D. Chen, and A. Bovik, "Visual pattern image coding," *IEEE Trans. Commun.*, vol. 38, no. 12, Dec. 1990.

[68] F. Wu and X. Sun, "Image compression by visual pattern vector quantization (VPVQ)," in *Proc. IEEE Data Compression Conference (DCC*'2008), pp. 123-131.

[69] http://wftp3.itu.int/av-arch/video-site/0005_Osa/a15j19.doc.

[70] *Y. Fischer, "Fractal image compression," SIGGRAPH'92 course notes.*

[71] B. Wohlberg, and G. de Jager, "A review of the fractal image coding literature," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1716-1729, Dec. 1999.

[72] R. J. Moll, J. J. Millspaugh, J. Beringer, J. Sartwell, Z. He, J. A. Eggert, and X. Zhao, "A terrestrial animal-borne video system for large mammals," *Computers and Electronics in Agriculture*, vol. 22, no. 6, pp. 133-139, May. 2009.

[73] N. Dalal, PhD thesis: "Finding people in images and videos," Institute National Polytechnique de Grenoble, 2006.

[74] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," *IEEE International Conference on Computer Vision (ICCV 2009)*, Kyoto, 2009.

PUBLICATIONS

Journal Papers:

[1] **Xiwen Zhao**, and Zhihai He, "Lossless image compression using super-spatial structure prediction," *IEEE Signal Processing Letters*, vol.7, no.4, pp. 383-386, Apr. 2010.

[2] **Xiwen Zhao**, and Zhihai He, "Local structure learning and prediction for efficient lossless image compression," submitted to Elsevier *Journal of Visual Communication and Image Representation*.

[3] Zhihai He, Jay Eggert, Wenye Cheng, **Xiwen Zhao**, Joshua J. Millspaugh, Remington J. Moll, Jeff Beringer, and Joel Sartwell, "Energy-aware portable video communication system design for wildlife activity monitoring," *IEEE Circuits and Systems Magazine,* vol.8, no.2, pp. 25-37, 2008.

[4] Remington J. Moll, Joshua J. Millspaugh, Jeff Beringer, Joel Sartwell, Zhihai He, Jay A. Eggert, and **Xiwen Zhao**, "A terrestrial animal-borne video system for large mammals," *Computers and Electronics in Agriculture***,** vol. 22, no. 6, pp. 133-139, May. 2009.

Conference Papers:

[1] **Xiwen Zhao**, "A 3D microwave imaging method for earth observation," *Proc. CISP*, Oct. 2010.

[2] **Xiwen Zhao**, "LMMSE equalization and DFE in 2-D ISI channels," *Proc. CISP,* Oct. 2011.

[3] **Xiwen Zhao**, and Zhihai He, "Local structure learning and prediction for efficient lossless image compression," *Proc. ICASS*, Mar. 2010.

[4] **Xiwen Zhao**, and Zhihai He, "Lossless image compression using super-spatial prediction of structural components," *Picture Coding Symposium (PCS' 2009*), May. 2009.

[5] **Xiwen Zhao**, and Zhihai He, "Local structure learning and prediction for efficient image compression," *Proc. SPIE*, vol. 7257, 725713 (2009).

[6] Yongfei Zhang, Shiyin Qin, **Xiwen Zhao**, and Zhihai He, "Content-aware packet scheduling for multi-session video streaming over wireless mesh networks," *Proc. SPIE*, vol. 7257, 72570Y (2009).

VITA

Xiwen Zhao received the B.E. degree in thermal engineering from Tsinghua University, Beijing, China, in 1998, the M.E. degree in electrical engineering from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2002, the M.S. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, in 2006. He is now pursuing the Ph.D. degree in electrical and computer engineering at the University of Missouri, Columbia. His research interests include image processing and compression, video processing and compression, computer vision.