2007

Selected topics in video coding and computer vision

Congxia Dai
West Virginia University

Selected Topics in Video Coding and Computer Vision

Congxia Dai

Dissertation submitted to the
College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical Engineering

Xin Li, Ph.D., Chair
Oscar Divorra Escoda, Ph.D.
Donald A. Adjeroh, Ph.D.
Natalia A. Schmid, Ph.D.
Arun A. Ross, Ph.D.

Lane Department of Computer Science and Electrical Engineering

Morgantown,West Virginia
2007

Keywords: Video Coding, Intra Prediction, Infrared Imagery, Pedestrian Detection and Tracking, Multi-view Video Alignment

ABSTRACT

Selected Topics in Video Coding and Computer Vision

Congxia Dai

Video applications ranging from multimedia communication to computer vision have been extensively studied in the past decades. However, the emergence of new applications continues to raise questions that are only partially answered by existing techniques. This thesis studies three selected topics related to video: intra prediction in block-based video coding, pedestrian detection and tracking in infrared imagery, and multi-view video alignment.

In the state-of-art video coding standard H.264/AVC, intra prediction is defined on the hierarchical quad-tree based block partitioning structure which fails to exploit the geometric constraint of edges. We propose a geometry-adaptive block partitioning structure and a new intra prediction algorithm named geometry-adaptive intra prediction (GAIP). A new texture prediction algorithm named geometry-adaptive intra displacement prediction (GAIDP) is also developed by extending the original intra displacement prediction (IDP) algorithm with the geometry-adaptive block partitions. Simulations on various test sequences demonstrate that intra coding performance of H.264/AVC can be significantly improved by incorporating the proposed geometry adaptive algorithms.

In recent years, due to the decreasing cost of thermal sensors, pedestrian detection and tracking in infrared imagery has become a topic of interest for night vision and all weather surveillance applications. We propose a novel approach for detecting and tracking pedestrians in infrared imagery based on a layered representation of infrared images. Pedestrians are detected from the foreground layer by a Principle Component Analysis (PCA) based scheme using the appearance cue. To facilitate the task of pedestrian tracking, we formulate the problem of shot segmentation and present a graph matching-based tracking algorithm. Simulations with both OSU Infrared Image Database and WVU Infrared Video Database are reported to demonstrate the accuracy and robustness of our algorithms.

Multi-view video alignment is a process to facilitate the fusion of non-synchronized multi-view video sequences for various applications including automatic video based surveillance and video metrology. In this thesis, we propose an accurate multi-view video alignment algorithm that iteratively aligns two sequences in space and time. To achieve an accurate sub-frame temporal alignment, we generalize the existing phase-correlation algorithm to 3-D case. We also present a novel method to obtain the ground-truth of the temporal alignment by using supplementary audio signals sampled at a much higher rate. The accuracy of our algorithm is verified by simulations using real-world sequences.

Acknowledgments

I want to gratefully and sincerely thank my advisor Prof. Xin Li with whom I have been fortunate to work during the last four years at West Virginia University. Without his advice, patience and encouragement this work could not have been done. I especially appreciate Prof. Xin Li for spending his valuable time discussing with me, guiding and training me into a professional researcher.

I would like to thank all the faculty members in the Lane Department of Computer Science and Electrical Engineering for offering an outstanding education and research environment. Especially, I would like to thank Prof. Donald Adjeroh, Prof. Arun Ross, Prof. Natalia Schmid, and Dr. Oscar Divorra for reading this dissertation and for providing precious suggestions towards its completion.

My special thanks go to Dr. Peng Yin, and Dr. Oscar Divorra for offering me the valuable internship opportunity at Thomson and leading me to the field of advanced video coding. The internship gave me industry experience and yielded new ideas which have become a part of this thesis. It is a wonderful learning experience to work with both of you, and thank you for all that you have done for me.

I would like to thank Yunfei Zheng for sharing his deep thoughts and inspiring discussions with me on various research problems. I am also grateful to all my friends at WVU: Ting Liu, Fei Nan, Jinyu Zuo... for their friendship and warm-hearted help.

Finally, I would like to thank my parents and my wife for their constant love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the past two decades, the rapid growth of imaging, computing and communication technologies has stimulated the prominent emergence of many video applications. Modern video based applications heavily rely on two categories of techniques: video processing and computer vision. Video processing mainly deals with the problems of employing mathematical tools to achieve enhanced visual quality or compact representation of video signals. Examples of video processing include video restoration, supper resolution, and compression. The goal of computer vision is to endow computers with the ability to understand imagery. Applications such as object detection, tracking and recognition, and video content analysis have received increasingly more attention in recent years. In this thesis, we choose to study three specific topics from both categories, and present novel approaches to attack these problems. The selected topics are intra prediction in block-based video coding, pedestrian detection and tracking for infrared imagery, and multi-view video alignment.

1.1 Intra Prediction for Video Coding

Video coding is a technique used to achieve compact representations of video data by removing their redundancies. Because of their large size, video signals have to be compressed for transmission and storage purposes. Therefore since 1980's, video coding has been a research topic of great interest in both academia and industry, and a wide variety of video coding algorithms [1–7] have been proposed. Among the existing video coding strategies, block-based hybrid video coding is the most widely used due to its applicability to a broad range of video content and adaptability for hardware implementations. Therefore all the existing video coding standards (e.g. MPEG-1, MPEG-2, H.263, MPEG-4 and H.264/AVC, etc.) belong to this category. Block-based video coding systems have two coding modes: inter coding and intra coding. Inter coding models the current video frame by referencing to other video frames. Intra coding essentially treats a video frame like a still image. Since the temporal redundancy is higher than the spatial redundancy, inter coding usually contributes more to the overall compression efficiency than intra coding in block-based video coding systems. Therefore, during the development of block-based video coding, significant research activities have been focused on the optimization of inter coding algorithms.

However, intra coding is also important for at least the following reasons: intra refreshment, error resilience, and special video play modes (fast forward and reverse). The intra coding procedure in current popular block-based video coding systems is usually comprised of intra prediction, transformation, quantization, and entropy coding. Where intra prediction functions as a modeling procedure of the current block of data with previously coded information of the same video frame, and prediction errors (residues) are transformed, quantized, and encoded by certain entropy coding techniques. Therefore, a properly designed intra prediction algorithm reduces the redundancy of prediction residues leading to the improvement of the intra coding efficiency. Unlike inter prediction, early block-based video coding systems (e.g. H.261, MPEG-1) do not have an explicit intra prediction procedure for intra coding. The intra coding scheme of MPEG-1 for example is conceptually similar to JPEG the image coding standard. Indeed, the concept of intra prediction was first introduced in H.263+ where the prediction algorithm was built in the domain of DCT (Discrete Cosine Transform) transform coefficients [8]. In H.264/AVC the state-of-art video coding standard, intra prediction scheme is designed based on a hierarchical qual-tree block partitioning structure. According to the comparative study conducted in [9], the intra coding scheme of H.264/AVC outperforms the still image coding standard JPEG and achieves comparable performance to JPEG2000.

Despite the superior performance, the intra coding scheme of H.264/AVC is not optimized. One possible improvement originates from the limitation of using quad-tree block partitioning structure to represent natural images. Since natural images are often approximated by 2D signals with piecewise smooth characteristics, studies in approximation theory have shown that quad-tree block partitioning structures are suboptimal for such models in terms of rate and distortion (R-D) [10, 11]. This limitation is mainly due to the fact that quad-tree based structures fail to exploit the geometric constraint of edges [12].

Based on this observation, we propose a geometry-adaptive block partitioning structure to exploit geometric constraint along image edges for enhancing the intra coding performance of block-based video coding systems. Upon the the proposed geometric block partitions, a set of prediction/modeling schemes have been designed to generate pixel predictions for the partitioned regions. This intra prediction algorithm is referred to as geometry-adaptive intra prediction (GAIP). Simulation results show that when compared to standard intra prediction schemes of H.264/AVC, the proposed GAIP algorithm provides more efficient representation of piecewise smooth image regions, and therefore leads to enhanced intra coding performance.

Modeling texture information of images is a very challenging task for the intra video coding purpose. Local prediction schemes often give poor performance on texture patterns. Therefore, in [13–16], algorithms have been proposed to model image texture patterns via nonlocal information. In this thesis, we also study the problem of modeling texture information for intra video coding by extending the intra displacement prediction (IDP) algorithm presented in [13] with our geometry-adaptive block partitioning structure. The resulting algorithm is named as geometry-adaptive intra displacement prediction (GAIDP). Experimental results show that the GAIDP algorithm is efficient to model texture patterns that reappear within the same video frame. More importantly, we demonstrate that when the GAIP and GAIDP algorithms are jointly enabled in an H.264/AVC video coding system, the intra coding performance can be significantly improved by approximately 6.9% average bit savings or equivalently 0.45dB average PSNR gain on twelve standard test video sequences.

1.2 Pedestrian Detection and Tracking in Infrared Imagery

The first computer vision problem we tackle is pedestrian detection and tracking for infrared imagery. Pedestrian detection and tracking in visible spectrum has been extensively studied over past decades, and various algorithms [17–30] have been developed to meet the challenges arising from the variety of body pose, clothing, illuminating condition, and occlusion. Successful pedestrian detection and tracking algorithms in visible spectrum have been applied to many important applications from video surveillance to intelligent vehicles. However under certain circumstances such as night conditions or bad weathers, sensing in visible spectrum becomes infeasible, which calls for the imaging modalities beyond visible spectrum. In recent years, the cost of thermal sensors has dramatically reduced and infrared (IR) sensors with high dynamic range and sensitivity have become widely deployed in night-vision and all-weather surveillance applications. Stimulated by the decreasing cost of IR sensors, a flurry of research works [31–35] on pedestrian detection and tracking for infrared imagery have emerged.

In the visible spectrum, the human vision system (HVS) is often used as the benchmark for the robustness and accuracy of machine vision systems. However, as we enter the infrared spectrum, there are several critical questions to be considered: Are the techniques developed in visible spectrum directly applicable to IR imagery? How can we efficiently represent IR images for object detection and tracking purposes? Is simulating HVS still the right approach, if objects are invisible to human eyes but still detectable by thermal sensors? All these questions suggest that for object tracking and classification beyond the visible spectrum (OTCBVS), physical principles of various imaging modalities become more relevant than psychophysics. Therefore, we argue that it is the mathematical modeling of sensor data instead of HVS that plays the fundamental role in OTCBVS.

In this thesis, we present a new approach towards pedestrian detection and tracking for infrared imagery using the appearance cue. In the proposed technique, a layered representation is first introduced and a generalized expectation-maximization (EM) algorithm is developed to separate infrared images into background (still) and foreground (moving) layers regardless of camera panning. To accurately locate individual pedestrians from the foreground layer, the appearance cue is employed via a modified Principle Component Analysis (PCA) algorithm. PCA templates with varying sizes are sequentially applied to detect pedestrians at multiple scales to accommodate different camera distances. To facilitate the task of pedestrian tracking, we formulate the problem of shot segmentation and present a graph matching-based tracking algorithm that jointly exploits the appearance and distance information. Experimental results with both the OSU Infrared Image Database and the WVU Infrared Video Database are reported to demonstrate the accuracy and robustness of our algorithm.

1.3 Multi-view Video Alignment

The second computer vision problem that we want to study in this thesis is multi-view video sequence alignment. Many video applications often require video sequences of high spatial and temporal resolutions. Examples include automatic video-based surveillance [36, 37], video metrology for athletic events [38], video-based modeling and rendering of 3-D scenes [39], and tele-immersion [40]. However any single video camera has limited spatial and temporal resolutions. Therefore a flexible and cost efficient way to achieve higher spatial and temporal resolutions is to fuse video sequences shot by multiple low cost cameras. Video alignment is a technique that manipulates multiple video sequences to facilitate this fusion purpose. In recent years, various algorithms for aligning video sequences have been proposed [41–45].

In this thesis, we propose an accurate technique for temporally aligning two video sequences of the same scene captured by non-synchronized cameras. An iterative procedure is proposed to successively align the sequences in space and time; The existing two-dimensional phase-correlation method [46] is generalized into three dimensions to achieve sub-frame accuracy. The ground-truth[1] of sub-frame temporal alignment is obtained by using supplementary audio signals sampled at a much higher rate. The accuracy

[1] To assess the accuracy of our algorithm, we recorded together with each video sequence a piece of audio signal. Since they have a much higher sampling rate, the alignment of audio signals provides much more accurate estimates of temporal displacements, and therefore can be used as the ground-truth for the video alignment.

of our technique is demonstrated by experimental results using real-world sequences.

1.4 Organization and Contributions

In this thesis, we study three selected topics from video coding and computer vision. For each topic we shall start with our motivation on why we choose to study it by presenting related background information. Then theoretical analysis of the selected problem from our own point of view will be presented. Based on the analysis, we shall describe in detail our approaches to attack these problems. Extensive simulations will be reported and discussed to verify the effectiveness of the proposed approaches. Finally, we shall conclude each topic by providing some perspectives on its future research directions.

Chapter 2 of this thesis studies the problem of intra prediction in block-based video coding. We shall start with theoretical analysis of the limitations of the intra prediction schemes of the state-of-art video coding standard H.264/AVC. Noticing that the quad-tree based block partitions fail to exploit the redundancies along image edges, we present a novel geometry-adaptive block partitioning structure. Based on the geometric block partitions, a new intra prediction scheme named geometry-adaptive intra prediction (GAIP) is proposed. To explore the problem of modeling texture information for intra video coding, we extend the existing intra displacement prediction (IDP) algorithm with our geometry-adaptive block partitions. The new texture prediction algorithm is referred to as geometry-adaptive intra displacement prediction (GAIDP). simulations on both the GAIP and GAIDP algorithms are reported and discussed to verify their effectiveness.

In Chapter 3 we propose a novel approach for pedestrian detection and tracking in infrared imagery by exploiting the appearance cue. In our algorithms, a layered representation is first introduced to separate infrared images into background (still) and foreground (moving) layers. Pedestrian detection is accomplished in the foreground layer by using the appearance cue. The tracking task is formulated as a graph matching problem where the correspondent assignment between individual pedestrians of successive video frames is measured by exploiting both the geometric approximity and the appearance similarity. To address the problem of temporal discontinuities caused by scene changes and non-uniform temporal sampling rates, a shot-segmentation algorithm is developed to cut a sequence into several temporally correlated shots. The tracking task is performed within each segmented shot. In order to better understand as well as to demonstrate the effectiveness of the proposed algorithms, simulations with both the OSU thermal image database and WVU infrared video database are presented and discussed at the end of this chapter.

Chapter 4 deals with the problem of multi-view video alignment. we present an accurate video alignment algorithm using phase correlation. Noticing that the spatial and temporal alignments of two non-synchronized video sequences of the same dynamic scene are intertwined with each other, we propose an iterative procedure to successively align the sequences in space and time. To achieve sub-frame accuracy of the temporal alignment, we generalize the existing 2-D phase-correlation algorithm to 3-D case. Another contribution of our work is that we invent a simple but accurate method to obtain the ground-truth of temporal alignment by utilizing supplementary audio signals sampled at a much higher rate.

In Chapter 5, we present concluding remarks for this thesis and provide possible directions of future research.

Chapter 2

Geometry-Adaptive Intra Prediction for Video Coding

2.1 Introduction

Due to their huge size, transmitting and storing raw video data can easily swamp any practical communication and storage resources. For example, a video sequence of CIF resolution with 8-bit precision per pixel sampled at 30 frames per second requires a data rate of 36.5Mbit/s. Consequently, digital video must be compressed in order to make a better use of available transmission and storage resources.

Generally speaking, video coding is a process that removes the spatial and temporal redundancies of video signals to reduce the size of their digital representations. Based on the fidelities of the compressed video sequences to their original formats, video coding schemes can be divided into two classes: lossless video coding and lossy video coding. In lossless video coding [47], the primary target is to pursue without any distortion the most compact descriptions of video sources which are given by their entropies according to Shannon's theory [48]. Comparing to the lossless case, lossy video coding often achieves much higher compression by allowing distortions with acceptable perceptual quality loss. Therefore, the ultimate goal of lossy video coding is to find the optimal tradeoff between the bit rate and the perceptual quality. In this chapter we shall focus on lossy video coding.

Due to the rapid growth of multimedia communication and processing in the past two decades, video coding has experienced a fast and steady progress. Since 1980's, two international standardization organizations ISO and ITU-T have released recommendations for universal video coding standards [49]. In 1991, ISO released the first draft of MPEG-1 [50] for audiovisual storage on CD-ROM, and MPEG-2 [51] (ITU-T H.262) was released in 1994 for HDTV applications. ITU-T released their first video coding standard H.261 [52] for ISDN networks in 1993. After that the ITU-T H.263 [53] video coding standard was released in 1996 for low bit rate communications over PSTN networks followed by its extensions: H.263+ [8] and H.263++ released in 1998 and 1999 respectively. In 2003, JVT (Joint Video Team) released the final draft of the newest video coding standard H.264/AVC [54] which is also known as MPEG-4 [55] part 10. It is well accepted that H.264/AVC represents the current state-of-art video coding technique for the following reasons: First, H.264/AVC provides services to a broad area of multimedia communication applications ranging from low bit rate, low complexity video streaming over wireless and mobile networks to high bit rate, high quality HDTV broadcasting over cable networks. Second, H.264/AVC has been reported to achieve 50% average bit rate savings over H.263 or MPEG2 for equivalent perceptual quality [56].

In addition to the standardization activities, a variety of video coding techniques have been proposed from different perspectives of modeling video contents. Segmentation based video coding [1, 2] splits video frames into arbitrary shaped regions based on rate and distortion criterion, and the information (region shape, motion, and texture) required for representing each segmented region is encoded separately. Model based video coding [3–5] employs a set of predefined models (3D object model, illumination model, and camera model) to describe the 3D scene. At the encoding stage, the models are manipulated and deformed to adapt to the video content. Then the analyzed model parameters and matching errors are encoded and transmitted. The decoder reconstructs the video content by synthesizing the 3D scene using the received information. Sub-band video coding [6, 7] decomposes video signals into a number of frequency bands using a set of filter banks in both the spatial and the temporal domain. The transform coefficients in each sub-band are quantized and encoded. Compression is obtained by carefully designed adaptive bit allocations in different frequency bands. Because of its inherently embedded nature, sub-band video coding owns a scalable coding structure which is desirable to many video transmission applications with variable bandwidth constraints.

Block-based hybrid video coding is the most widely used video coding technique, whose popularity mainly comes from its generality, computational simplicity, and adaptability of hardware implementations. Therefore, all of the aforementioned mainstream video coding standards belong to this category. In block-based hybrid video coding, each video frame is partitioned into macroblocks first. During the encoding process, macroblocks are predicted, transformed, quantized, and encoded by some entropy coding technique. Most block-based video coders have two basic coding modes: intra and inter modes. In intra coding mode, a macroblock is predicted using the spatial information of the current video frame. While in the inter mode, a macroblock is predicted by exploiting the temporal redundancies among successive video frames. Normally in a block-based hybrid video coding system, the inter coding mode achieves higher compression efficiency than the intra coding mode with comparable visual quality. This is due to the fact that in most video signals temporal redundancy is much higher than the spatial one. Therefore, during the development of block-based hybrid video coding, researchers have devoted more attentions to optimizing the performance of the inter coding.

However, intra coding is also important for at least the following reasons: First, since an inter coded video frame is predicted from the previously coded frames, an sequence of successively inter coded frames suffers quality degradation due to the accumulation of quantization errors. Therefore, an intra coded frame is usually required to refresh the frame quality after a certain number of inter coded video frames. Second, many multimedia services, such as mobile TV and mobile teleconferencing applications, require transmitting compressed video data over networks of varying bandwidth and error rate characteristics. Transmission errors appearing in the coded bitstreams may lead to serious problems at the decoding process. In these transmission applications, properly increasing the portion of intra coded data in coded bitstreams is a way to relieve error propagation problems and enhance the error robustness of compressed video data [49].

Aiming at enhancing the coding efficiency of the intra mode in block-based video coding, in this chapter, we present a novel geometry-adaptive intra prediction scheme in which wedgelet like discontinuities are used to defined separate coding regions where different statistical/waveform modeling tools can be used. To verify the proposed intra prediction scheme, we have implemented it under the framework of the state-of-art video coding standard H.264/AVC. Extensive experimental results show that when incorporated with the proposed intra prediction scheme, the efficiency of H.264/AVC intra coding can be significantly improved. The rest of this chapter is organized as follows: Section 2.2 reviews existing intra prediction schemes for block based video coding. In section 2.3 we discuss in detail our geometry-adaptive block partitioning scheme for intra prediction. In section 2.4 we extend the existing intra displacement prediction schemes by applying the proposed geometry-adaptive block partitioning structure to address the problem of modeling texture regions in intra coding. Finally, concluding remarks of this chapter are presented in section 2.5.

2.2 Overview of Existing Intra Prediction Schemes

Intra prediction is an effective procedure to reduce spatial redundancy of video data in intra video coding. In current block-based video coding systems, the intra predicted sample block is subtracted from the original block and the resulting residual block is transformed, quantized, and coded by certain entropy coding technique. The first intra prediction scheme was introduced as the Advanced Intra Coding Mode in H.263+ video coding standard [8], where intra prediction is performed in the block DCT (Discrete Cosine Transform) domain with three prediction modes: DC, Vertical DC and AC, and Horizontal DC and AC. In the DC mode, only the DC coefficient of the current 8x8 block is predicted from its above and left neighbors. In the Vertical/Horrizontal DC and AC mode, the DC and first row/column of AC coefficients of the current 8x8 block are vertically/horrizontally predicted from those of the block to the above/left.

In H.264/AVC video coding standard [54], intra coding is based on a hierarchical quadtree-partition of the luminance component of an intra 16x16 macroblock into four 8x8 blocks (INTRA8x8) or sixteen 4x4 blocks (INTRA4x4). For each block size, a set of intra prediction schemes using decoded causal neighbor information, have been carefully designed as follows: There are four prediction modes (Vertical, Horizontal, DC,

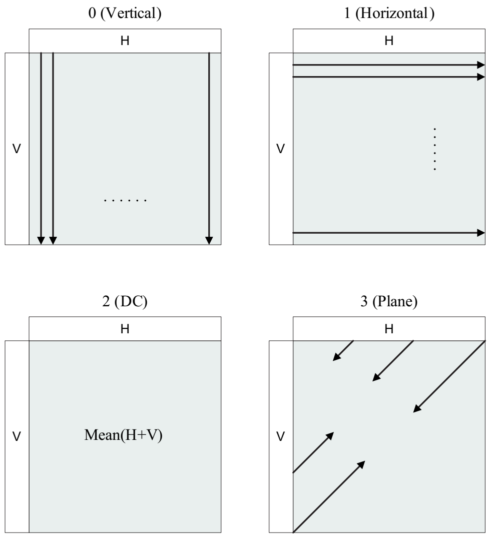


Figure 2.1: INTRA16x16 prediction modes in H.264/AVC.

and Plane) defined for INTRA16x16 macroblocks, and nine prediction modes (eight directional predictions, and DC) defined for INTRA8x8 and INTRA4x4 blocks. The prediction modes for INTRA8x8 luminance blocks and INTRA4x4 blocks are just similar. Besides, the two 8x8 Chrominance blocks have four prediction modes which are similar to those defined for INTRA16x16 luminance blocks except that the orders of the Vertical and DC modes are switched. The prediction modes of INTRA16x16 and INTRA4x4 luminance blocks are illustrated in Fig. 2.1 and Fig. 2.2 respectively. At the encoding stage, a rate and distortion (R-D) optimization procedure selects the best macroblock partition and associated prediction modes among all the possible choices.

This intra prediction scheme together with enhanced prediction residue coding procedures have significantly improved the intra coding efficiency of H.264/AVC over previous video coding standards such as H.263+ and H.263++ [56]. Moreover, in [9] performance comparisons between intra coding of H.26L (the prototype of H.264/AVC) and existing still image coding standards (JPEG and JPEG2000) were conducted. Based on the experimental results, the author concluded that over a wide range of targeted bit rates, H.26L significantly outperforms JPEG in both objective (PSNR) and subjective measures. Although on average, JPEG2000 performs slightly better than H.26L, for small

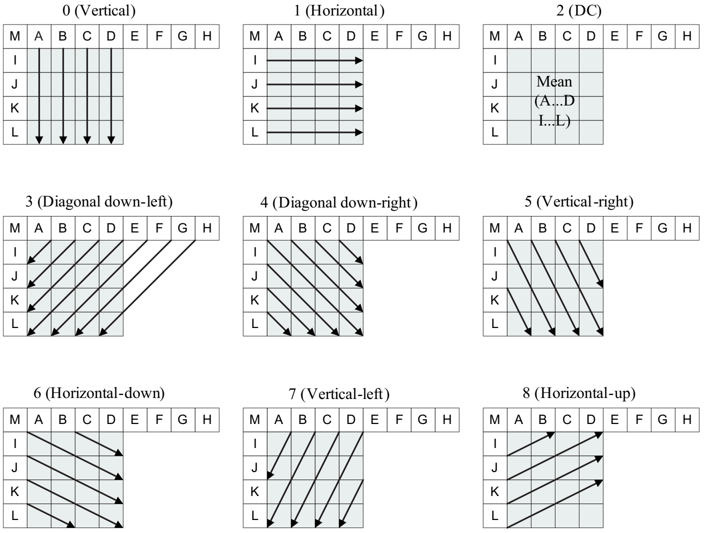


Figure 2.2: INTRA4x4 prediction modes in H.264/AVC.

sized images at low bit rates, H.26L can achieve higher coding gain than JPEG2000 with less perceptual artifacts. According to [9], the superior performance of H.264/AVC intra coding indeed attributes to the quadtree-based block partition structure and the directional prediction schemes shown in Fig. 2.2. First of all, the quad-tree partitioning structure adapts automatically to the non-stationary nature of natural images: bigger blocks are often used to represent smooth regions, while smaller blocks tend to aggregate around edges or textured regions. Second, directional prediction schemes exploit some geometric redundancy by extrapolating previously decoded neighboring pixels [56].

Despite the aforementioned advantages, H.264/AVC intra prediction strategy can still be further improved. First, the quadtree-based block partition structure is not R-D efficient to represent piecewise smooth images. For instance, if we consider a piecewise smooth image model illustrated in Fig. 2.3 where two different smooth regions, with different smoothness properties, are separated by an edge, small blocks will tend to accumulate around the boundary due to the difficulty of predicting both regions with a single model. In near-edge areas, tree-based partition leads to separately code different blocks with similar data with unnecessary overhead. In effect, H.264/AVC standard does not take into account large scale geometry in images for efficient intra coding.

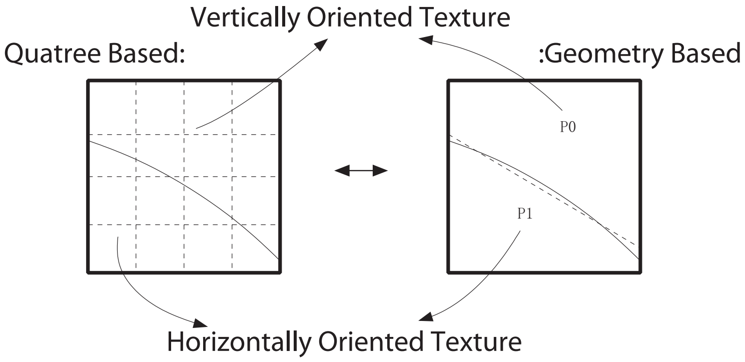


Figure 2.3: Quadtree (Left) vs Geometry (Right) partitioning of 2D piecewise smooth signals.

Second, as shown in Fig. 2.1 and Fig. 2.2, the intra prediction modes of H.264/AVC rely on the neighboring decoded pixels (predictors). Sometimes, pixels located farther from these predictors can not be well modeled, leading to higher prediction errors and losses in coding efficiency. This case is shown in Fig. 2.4 where a region of the Foreman image predicted by H.264 is displayed. Obviously H.264 intra prediction scheme performs poorly around the edges in Fig. 2.4. As we can imagine, these inaccurate predictions will generate high prediction errors leading to the loss of coding efficiency. Based on these observations, in section 2.3 we present in detail a novel way of representing intra video data, called geometry-adaptive intra prediction (GAIP). Specifically, we will demonstrate that theoretically geometry-adaptive block partition is R-D more efficient than the quadtree based block partition for representing signals with piecewise smooth characteristics. In addition to the better partitioning structure, geometry-adaptive block partition also provides the flexibility of representing each partitioned region with different modeling tools and this flexibility helps to mitigate some of the limitations of intra prediction schemes used in H.264/AVC therefore reducing the prediction error around edge regions as shown in Fig.2.4.

Natural images usually contain plentiful texture information which is highly non-stationary and abundant of high frequency components. However H.264/AVC intra prediction schemes (as illustrated in Fig. 2.1 and Fig. 2.2) are not able to model texture

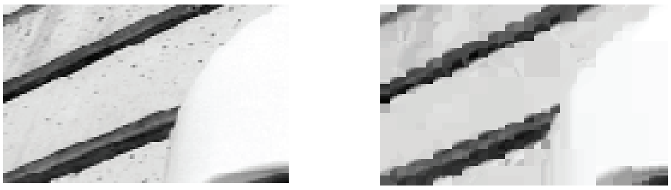


Figure 2.4: H.264 intra prediction result: Left (original image), Right (predicted by H.264)

information well. One possible explanation is that complex texture patterns are difficult to be predicted by simple low frequency operations (extrapolation along certain directions) using limited local information. In recent years, several texture prediction techniques [13–16] have been proposed to facilitate the modeling of texture information of intra video data by exploiting the available non-local information within the same video frame. In section 2.4, we study the problem of modeling intra video data via texture prediction. To do this, we extend the intra displacement prediction (IDP) algorithm proposed in [13] by incorporating our geometry-adaptive block partitioning structure proposed in section 2.3. We show that this geometry-adaptive intra displacement prediction (GAIDP) when jointly applied with the GAIP algorithm described in section 2.3 can significantly improve the intra coding performance of H.264/AVC.

2.3 Geometry-Adaptive Intra Prediction

In this section, we present in detail the concept of the geometry-adaptive intra prediction. We shall start with the motivations for introducing the geometry-adaptive intra prediction. Then we discuss the geometry-adaptive intra prediction by defining geometry-based block partitions together with the predicting/modeling tools for the partitioned regions. In section 2.3.4, we show how the geometry-adaptive intra prediction is incorporated in the standard H.264/AVC intra coding scheme. At the end of this section, simulation results will be presented to demonstrate the effectiveness of the geometry-adaptive intra prediction.

2.3.1 Motivations

Natural images are often approximated by two-dimensional signals with piecewise smooth characteristics, especially for low bit rate representations. Considering such a signal model, it has been shown that quadtree-based structures are R-D suboptimal for coding purposes [10, 11]. This is because tree partitioning, with homogeneous approximation models within leaves, is often unable to exploit the redundancy along region boundaries. As shown in Fig. 2.3, one may intuitively consider that a good compression approach should exploit region shapes and code, with no further splitting, each one of the disjoint regions with homogeneous characteristics: *P0* and *P1*. This is what has been actually proved in terms of R-D for the case of piecewise-smooth images in [10], where the authors show that distortion reduces with the rate, in a way close to the optimal, when wedge-like partitions are used within tree leaves to code piecewise-smooth images, having smooth contours. To help readers better understand this motivation, here we briefly describe the theoretical analysis presented in [10].

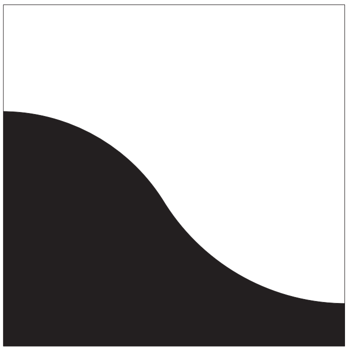


Figure 2.5: A simple piecewise smooth image model.

Consider a simple piecewise smooth image model (in Fig. 2.5) $f(x, y)$ defined as follows:

$$f(x,y) = \begin{cases} 1, & if\ b(x) \leq y \\ 0, & otherwise \end{cases} \tag{2.1}$$

Where $(x, y) \in [0, 1] \times [0, 1]$, and $b(x) \in \mathcal{C}^p$ is the smooth partitioning curve which is p-times continuously differentiable and has a finite length. An optimal way (Oracle based method) to code this image is to spend all available bit rate R to coding the partitioning curve $b(x)$. Assume that $\hat{b}(x)$ and $\hat{f}(x,y)$ are the coded versions of $b(x)$ and $f(x,y)$ respectively. Then the resulting distortion function $D_{opt}(R,f)$ should be written as:

$$\begin{aligned} D_{opt}(R,f) &= \int_{(x,y)\in[0,1]^2}(f(x,y)-\hat{f}(x,y))^2dxdy \\ &\leq \int_{x\in[0,1]}|b(x)-\hat{b}(x)|dx \\ &\leq (\int_{x\in[0,1]}(b(x)-\hat{b}(x))^2dx)^{1/2} \\ &= (D(R,b))^{1/2} \end{aligned}$$

Since it is proven in [57] that coding a 1D curve $b(x) \in \mathcal{C}^p$ using a proper wavelet basis can result in a distortion $D(R,b) \sim R^{-2p}$, the oracle based method for coding the image $f(x,y)$ will have a R-D performance as:

$$D_{opt}(R,f) \sim R^{-p} \tag{2.2}$$

When a quad-tree based wavelet[1] coder is used for this image model, at level j, wavelet basis functions have a support size of 2^{-j}. Denote n_j the number of dyadic squares at level j intersecting with the curve $b(x)$. Then $n_j \sim 2^j$ as illustrated in Fig. 2.6. Therefore there are $O(2^j)$ significant wavelet coefficients at level j. If the wavelet decomposition is performed up to level J, the total number of nonzero wavelet coefficients to be coded is

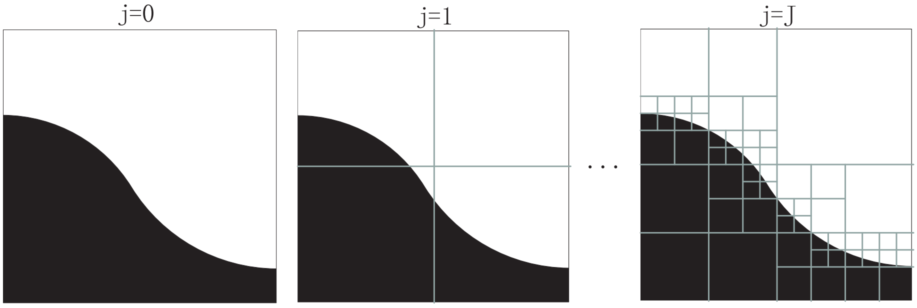


Figure 2.6: Quad-tree based wavelet decomposition of a piecewise smooth image.

[1] Please note that any specific tools for representing the quad-tree partitioned blocks is not essential here. Wavelet is used to keep the analysis mathematically tractable.

$N_J \sim \sum_{j=0}^{J} 2^j \sim 2^J$. Assuming wavelet coefficients decay like $C_{j,k} \sim 2^{-j}$ at level j, then a quantization step size of $\Delta \sim 2^{-J}$ is fine enough to code all nonzero coefficients. In other words, J bits are required to represent each nonzero coefficient. Therefore the total number of bits required to code the nonzero wavelet coefficients up to level J is:

$$R \sim N_J J \sim J 2^J. \tag{2.3}$$

The total distortion is the sum of the quantization error and wavelet series truncation error:

$$\begin{aligned} D_{tree}(R, f) &\sim N_J \Delta^2 + \sum_{j=J+1}^{\infty} n_j 2^{-j} \\ &\sim 2^{-J} \end{aligned} \tag{2.4}$$

Combining Eq.(2.3) and Eq.(2.4), we derive the R-D performance of the quadtree-based wavelet coder:

$$D_{tree}(R, f) \sim \frac{\log R}{R} \tag{2.5}$$

Now let us discuss the geometry-adaptive coding algorithm proposed in [10]. In this algorithm, the image is also decomposed into dyadic squares using quad-tree based partitions. The difference is that each dyadic square (edge node) that intersects with the edge curve $b(x)$ is represented by a Wedgelet mode which consists of two smooth regions separated by a straight line. In this way, the smooth curve $b(x)$ is actually approximated by the concatenation of line segments as shown in Fig.2.7. Assume that when this coding scheme is applied to the image in Fig. 2.5, the quad-tree based partitioning is performed up to level J, then the smallest dyadic square will be of size 2^{-J}, and $2J$ bits will be used to code the two vertices of the wedgelet within an edge node. Also notice that as illustrated in Fig. 2.7 we need 2 bits to indicate the type of each node within the tree, which can be one of the following: black, white, edge, or intermediate (an edge node that is further split). Denote N_a and N_e the number of total nodes and edge nodes respectively. Then the total number of bits required for coding this image is:

$$R = 2N_a + 2JN_e$$

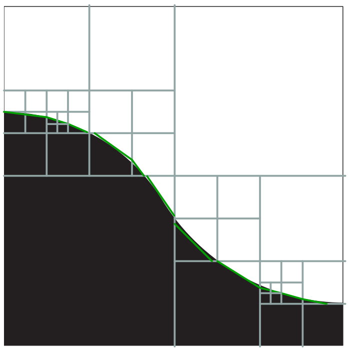


Figure 2.7: Wedgelet representation of a piecewise smooth image. Wedgelets are denoted by green line segments.

Since N_a is bounded by $N_a \leq 2^J$ (in Fig.2.6), and $N_a \geq N_e$, the total number of bits can be written as:

$$R \sim J2^J \tag{2.6}$$

Denote $\hat{b}(x)$the line segment representation of the curve $b(x)$. Since the size of the smallest dyadic square is 2^{-J}, and the quantization step size of the vertices of each line segment is $\Delta = 2^{-J}$ (J bits for each vertex), the maximum distance between $b(x)$ and $\hat{b}(x)$ is bounded by:

$$\max_{x \in [0,1]} |b(x) - \hat{b}(x)| \leq C\Delta 2^{-J} = C2^{-2J}$$

Therefore the total distortion can be written as:

$$\begin{aligned} D_{tree+wedgelet}(R,f) &= \int_{(x,y)\in[0,1]^2} (f(x,y) - \hat{f}(x,y))^2 dxdy \\ &\leq \int_{x\in[0,1]} \max|b(x) - \hat{b}(x)|dx \\ &\leq C2^{-2J} \end{aligned} \tag{2.7}$$

Combining Eq.(2.6) and Eq.(2.7), the R-D performance of this Wedgelet coder can be shown as:

$$D_{tree+wedgelet}(R,f) \sim \frac{\log R}{R^2} \tag{2.8}$$

Comparing Eq.(2.2), Eq.(2.5), and Eq.(2.8), we can conclude that quad-tree based coding algorithm is R-D less optimal than the method incorporated with Wedgelet representations for coding piecewise smooth signals as defined in Eq.(2.1). If the edge curve $b(x) \in \mathcal{C}^2$, the wedgelet coding algorithm approaches the oracle based method in terms of R-D.

In addition to the theoretical improvements, when modeling piecewise smooth images, geometry-adaptive block partitioning allows to adaptively select different models for each partition depending on the signal while considering the geometric structure of object boundaries. This may help to mitigate some of the limitations of intra prediction schemes used in H.264/AVC, e.g. reducing the prediction error around edge regions as shown in Fig.2.4. Unlike quad-tree based partitions where a uniform predicting scheme is used for each block, the geometry based structure automatically provides the flexibility to represent a block with two different models either from the neighboring predictors or by the statistics from within the partitioned regions, at the same time when object boundaries can be described.

2.3.2 Definition of Geometry-Based Block Partitions

In this section, based on previous discussions, we present the definition of geometry-based block partitions to extend the intra coding scheme of H.264/AVC. Ideally the geometric single-edge representation of a block can be modeled by an arbitrary parametric curve $f(x, y, \vec{p})$, where $\vec{p}$ represents the model parameters. In our current work, a first order polynomial model is adopted to generate the splitting Wedgelets for quad-tree partitioned blocks within a video frame. Although our work focuses on intra coding, we notice that the concept of Wedgelet-based block partitions has been proposed elsewhere [58, 59] for the inter coding purpose.

Given a block of finite size L, a partitioning line segment within this block can be implicitly defined by its level-set [60] parametric model:

$$f(x, y, \vec{p}) = x \cos\theta + y \sin\theta - \rho, \quad (x, y) \in [-L/2, L/2]^2 \tag{2.9}$$

where $\vec{p} = [\rho, \theta]$ are model parameters, and ρ and θ represent respectively the partitioning radius and angle as illustrated in Fig. 2.8. Based on this parametric line mode, geometric

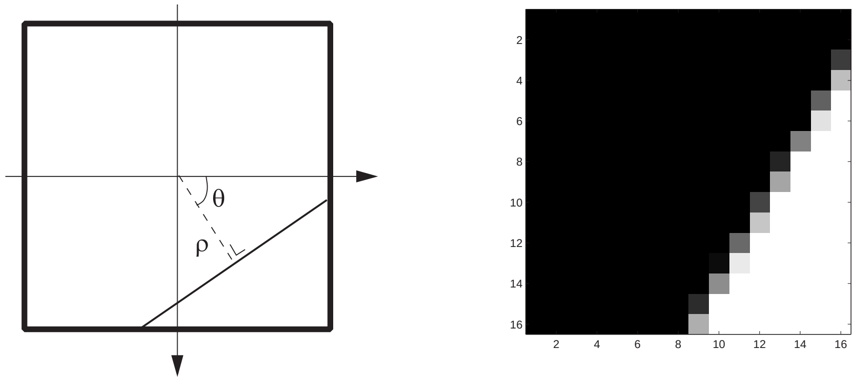


Figure 2.8: Left: Line partition of a block based on geometric parameters θ and ρ. Right: Example of wedge-like partition with $\theta = \pi/6$ and $\rho = 4$. White color indicates one of the partitions, black marks the complementary partition. Gray intermediate values show "partial-surface" pixels.

partitions of the block are defined such that each pixel $(x, y) \in [-L/2, L/2]^2$ ideally can be classified as:

$$Partition(x, y) = \begin{cases} if\ \ f(x, y) > 0 & Partition\ 0 \\ if\ \ f(x, y) = 0 & Line\ Boundary \\ if\ \ f(x, y) < 0 & Partition\ 1 \end{cases} .$$

However due to the discrete nature of digital images, the partitioning line may cross some pixels, and those pixels can not be classified to either partition. Hence, when building the partition masks, they may be labeled as "partial surface" pixels (in Fig. 2.8), with a label different from 1 and 0. "Partial surface" pixels can be labeled with some value in between. This way of labeling also reflects how much "Partial surface" pixels are weighted as if they are fully classified into each region (e.g. a value of 1 would be completely, 0.5 would be half-half, 0 nothing). Therefore the labeling of pixels is formally defined as:

$$Label(x, y) = \begin{cases} if & f(x, y) \geq 0.5 & then\ 1 \\ if & 0.5 > f(x, y) > -0.5 & then\ f(x, y) + 0.5 \\ if & f(x, y) \leq -0.5 & then\ 0 \end{cases} . \quad (2.10)$$

As shown in Fig. 2.8, $Label(x, y) = 1$ indicates that pixel is included within one partition, and $Label(x, y) = 0$ indicates that it is within the complementary partition. The rest values indicate, for that particular pixel, it is partially classified.

For coding purposes, line parameters (ρ and θ) can not be represented in their continuous form. Hence a dictionary of possible partitions is *a priori* defined as follows:

$$\rho : \rho \in [0, \frac{\sqrt{2}L}{2}),\ \rho \in \{0, \Delta\rho, 2 \cdot \Delta\rho, \dots\},$$

and

$$\theta : \begin{cases} \theta \in [0, \pi) & if\ \rho = 0 \\ \theta \in [0, 2\pi) & otherwise \end{cases},\ \theta \in \{0, \Delta\theta, 2 \cdot \Delta\theta, \dots\}.$$

In here, $\Delta\rho$ and $\Delta\theta$ are the selected sampling steps for the radius and angle data respectively. Depending on the target bit rate, these can be modified in order to maximize R-D coding efficiency.

2.3.3 Predicting/Modeling Schemes for Partitioned Regions

Given the geometric block partitions defined in the previous section, we need to design proper prediction methods for the partitioned regions. In this section we present two predicting/modeling schemes (in Fig. 2.9) that are used to generate sample predictions for the partitioned regions by exploiting either the neighboring decoded information or the statics inside the partitioned regions.

Linear Directional Prediction

Considering a partitioned block of size $N \times N$ together with $3N + 1$ predictors (decoded neighboring pixels), the directional predicting scheme is defined such that every pixel $p(x, y)$ inside a partitioned region is predicted along the predicting direction φ from the predictors as shown in Fig. 2.9(a,b,c), where $\varphi \in [0, \pi),\ \varphi \in \{0, \Delta\varphi, 2 \cdot \Delta\varphi, \dots\}$. Depending on the number of intersecting points of the line passing through (x, y) with orientation φ (e.g. the dashed lines in Fig. 2.9) and the coordinate axes, the pixel value $p(x, y)$ may be predicted in one of following three cases: 1) When there are two intersecting points $(x', 0)$, and $(0, y')$ (in Fig. 2.9(a)), the pixel values of these intersecting

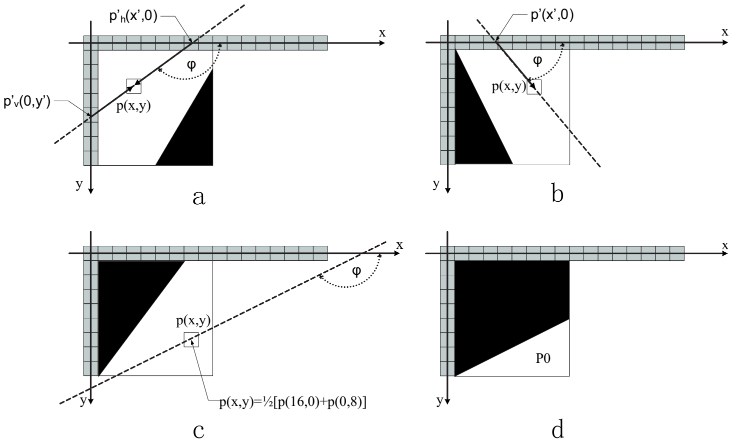


Figure 2.9: The linear directional predicting(a,b,c) and DC modeling(d) schemes for partitioned regions (gray squares representing the neighboring predictors).

points: $p'_h(x', 0)$ and $p'_v(0, y')$ are interpolated from their nearest predictors. Then the pixel value $p(x, y)$ is linearly interpolated from $p'_h(x', 0)$ and $p'_v(0, y')$ along the direction of φ; 2) When there is only one intersecting point, for example, the point $(x', 0)$ in Fig. 2.9(b), $p(x, y)$ is predicted by simply copying (extrapolated) the pixel value $p'(x', 0)$ which is interpolated from its nearest predictors; 3) If there is no such intersecting point within the range of the predictors, the prediction of pixel $p(x, y)$ is set to be the average of the two ending predictors (as shown in Fig. 2.9(c)). Note that each directional intra prediction mode defined in H.264/AVC (in Fig. 2.2) generates sample predictions by only copying the values of the predictors on one side of the block. While in our scheme, the pixel value can be predicted from the predictors on both sides of the block. Therefore our scheme may produce more accurate predictions at the price of moderate computational complexity.

Linear directional prediction scheme requires to signal the decoder values of the predicting directions φ. Given a predefined quantization step size $\Delta\varphi$, a straightforward way is to code the index of φ with a fixed-length code by assuming φ a uniform distribution. However, recalling our piece-wise smooth assumption, it is reasonable to infer that instead of being uniformly distributed, φ is more likely to be along the partitioning edge than crossing it. Based on this observation, a better coding strategy is to differentially code φ with respect to the partitioning edge orientation ω as illustrated in Fig. 2.10,

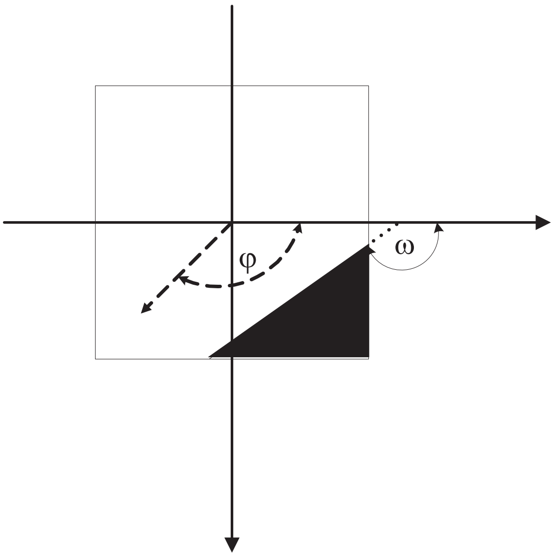


Figure 2.10: Predicting direction φ is differentially coded with respect to the partitioning edge orientation ω.

where ω is derived from the partitioning line parameter θ by $\omega = \theta \pm \frac{\pi}{2}$. Denote $\delta = \varphi - \omega$ the difference between the predicting direction and the partitioning edge orientation, and let the quantization step sizes $\Delta\varphi$ and $\Delta\theta$ to be multiples of each other. To guarantee a fully reversible quantization of δ, the index of δ is defined as:

$$Index(\delta) = \begin{cases} \frac{\varphi - \omega}{\Delta\varphi} & if\ \Delta\varphi \leq \Delta\theta \\ \frac{\varphi - ROUND(\frac{\omega}{\Delta\varphi}) \times \Delta\varphi}{\Delta\varphi} & otherwise \end{cases} . \quad (2.11)$$

The derived index of δ is then encoded using signed variable length codes.

DC Modeling

Sometimes, the linear directional predicting scheme based on neighboring predictors is not able to provide accurate predictions for partitioned regions, due to the limited accessibility of the neighboring pixels. For example, in Fig. 2.9(d), pixels inside region $P0$ might not be accurately predicted from neighboring predictors, because they are separated by the partitioning edge. In this case, we can model the region as a smooth polynomial of certain order. In our current implementation, for simplicity purpose, we choose zero order polynomial or the DC value to model that region. Therefore any pixel $p(x, y) \in P0$ is estimated by:

$$\hat{p}(x, y) = DC(P0) = \sum_{p(x,y) \in P0} p(x, y)$$

In spite of its simple form, the DC modeling scheme is very useful to reduce the prediction errors shown in Fig. 2.4 which is unavoidable in any directional prediction schemes using decoded neighboring predictors.

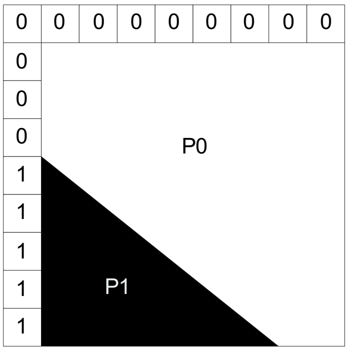


Figure 2.11: Predictive coding of the DC value of the partitioned regions

Since in the DC modeling scheme, the DC value of a partitioned region is estimated from the pixels to be encoded, when the DC modeling scheme (DC mode) is selected to represent a region, we have to signal the decoder explicitly the estimated DC value. To fully exploit the decoded information, we predict the DC value of a partitioned region by the mean of the decoded neighboring pixels that are adjacent to that region (in Fig. 2.11). Then the difference between the DC value and its prediction is encoded using signed variable length codes. For example, denote $S_0 = \{p : label(p) = 0\}$ the set of all predictors that is adjacent to region $P0$ in Fig. 2.11, the predicted DC value of region $P0$ is computed as: $\hat{DC}(P0) = \frac{\sum_{p \in S_0} p}{|S_0|}$.

2.3.4 Intra Macroblock Encoding Procedure

To implement our geometry-adaptive intra prediction scheme, we introduce two geometric modes named INTRA16x16GEO and INTRA8x8GEO into the standard H.264/AVC mode table. The mode INTRA16x16GEO is defined for the geometric partition of macroblocks and inserted after the standard mode INTRA4x4. The INTRA8x8GEO mode is defined for the 8x8 blocks inside a macroblock, therefore a 1-bit flag is defined for each 8x8 block to distinguish it from the standard INTRA8x8 mode. Note that for the mode INTRA16x16GEO, sometimes, both of two partitioned regions of a macroblock may select the same predicting direction φ, and the existence of a geometric partition is redundant. To resolve this pathological case, a 1-bit flag is used for each macroblock to indicate whether it is partitioned.

Given a block of size 16x16 or 8x8, there are a set of possible geometric partitions $\mathcal{P} = \{P(1), P(2), \ldots, P(N)\}$ and a set of predefined predicting and modeling schemes $\Omega = \{\varphi_1, \varphi_2, \ldots, \varphi_M, DC\}$ associated with the partitioned regions. The optimal geometric partition and associated predicting/modeling schemes for representing the block is selected such that an R-D cost function is minimized:

$$(P_{best}, \omega_{1best}, \omega_{2best}) = \min_{\forall P \in \mathcal{P} \ and \ \forall \omega_1, \omega_2 \in \Omega} (D_p(P, \omega_1, \omega_2) + \lambda_p R_p(P, \omega_1, \omega_2))$$

Where$D_p(P, \omega_1, \omega_2)$ is the distortion measure between the original block $(I_o(x, y))$ and its predicted version $(\hat{I_o}(x, y))$ generated by using the partition P and corresponding prediction schemes $\omega_1, and\, \omega_2$ for each region. Examples of the distortion measure include SSE (sum of square error) and SAD (sum of absolute difference), etc. $R_p(P, \omega_1, \omega_2)$ is the number of bits required to represent the geometric partition P together with the prediction schemes $\omega_1, and\, \omega_2$. λ_p is a Lagrangian multiplier which is a function of the system quantization parameter (QP).

At the encoding stage, for every intra macroblock, the encoder selects among all possible coding modes: INTRA4x4, INTRA16x16, INTRA8x8, INTRA16x16GEO, and INTRA8x8GEO, the optimal one that results in the lowest R-D cost: $J = D_m + \lambda_m R_m$. Where D_m is a distortion measure (SSE or SAD) between the original macroblock $(I_o(x, y))$, and its decoded version $(I_d(x, y))$, λ_m is a Lagrangian multiplier, and $R_m =$

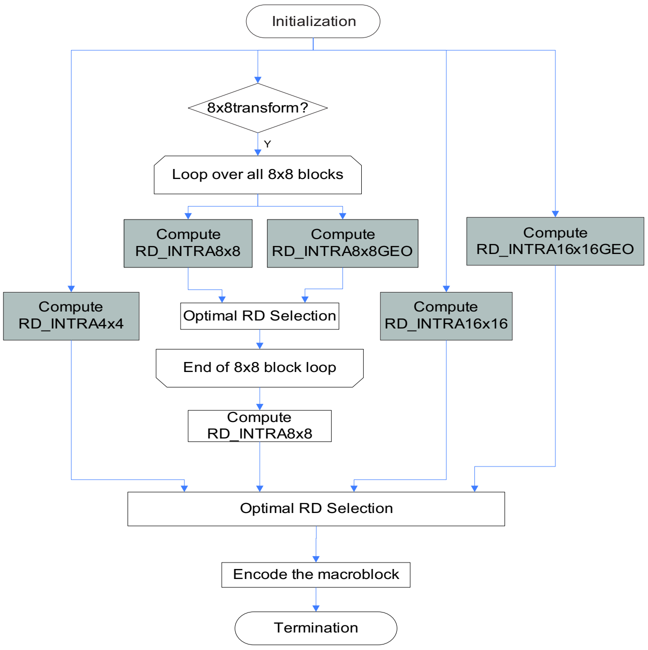


Figure 2.12: Intra macroblock encoding procedure

$R_{mode} + R_{pred} + R_{residue}$ is the total number of bits to encode the macroblock, where R_{mode} is the number of bits to indicate the coding mode (e.g. INTRA4x4), R_{pred} is the number of bits to represent the predicting scheme (e.g. $P, \omega_1, and\ \omega_2$), and $R_{residue}$ is the number of bits required for the quantized transform coefficients of the residual signal. Fig. 2.12 illustrates the intra macroblock encoding procedure of H.264/AVC when the proposed geometry-adaptive intra prediction modes are enabled.

Here, we want to point out that the Lagrangian multipliers λ_p and λ_m for the optimal geometric parameter selection and mode decision procedures are functions of the quantization parameter (QP). In this thesis we adopt the mathematical expressions of these functions provided in [61], which have been empirically determined from the statistics of extensive coding experiments.

2.3.5 Simulation Results and Discussions

In this section, we present experimental results of our geometry-adaptive intra prediction algorithm. The experiments are carefully designed to help us better understand the performance of the proposed algorithm, which is dependent on both the characteristics of the video data and the choice of coding parameters.

We select several popular test video sequences with different spatial resolutions in our experiments as presented in Table 2.1. In all experiments these sequences are coded exclusively as I (intra) frames. To evaluate the coding results, the Peak-Signal-to-Noise-Ratio (PSNR) is used as the objective distortion measure. Given a video frame with spatial resolution $W \times H$, PSNR is defined by:

Resolutions	Sequences
352x288 (CIF)	Foreman, Paris
176x144 (QCIF)	Car Phone
320x240 (SIF)	Duck dodgers
480x480	Tiger

Table 2.1: Test video sequences.

$$PSNR(I, \hat{I}) = 10 \log_{10} \frac{255^2}{\frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} (I(x, y) - \hat{I}(x, y))^2}$$

Where I and $\hat{I}$ denote the original and the decoded video frames. The bit rate is calculated from the total number of bits used for encoding the video frame.

To make a fair coding performance comparison between our geometry extended H.264/AVC intra coding scheme and the standard version, during the experiments the encoder settings are set exactly the same for both cases. The mainly used settings are: VLC coding, Enabling the 8x8 block partition and transform (FRext) for intra coding, and the standard deblocking filter is on. Throughout the experiments we use the popular Bjontegaard's average coding gain to measure the performance difference between the two intra coding schemes, which is calculated according to the convention proposed in [62]. The H.264/AVC reference software JSVM 6.0 [61] has been used as the compliant codec to conduct all experiments.

In our first experiment, we employ a fixed set of basic geometric parameters $\Delta\rho$, $\Delta\theta$, and $\Delta\varphi$ for our geometry-adaptive intra prediction scheme. The values of the geometric parameters are empirically determined and given as: $\Delta\rho = 1$, $\Delta\theta = \pi/16$, and $\Delta\varphi = \pi/32$. Based on these basic geometric parameters, in our present settings the actual parameters for modes INTRA16x16GEO and INTRA8x8GEO are derived as follows: $\Delta\rho_{16\times16} = \Delta\rho_{8\times8} = \Delta\rho$, $\Delta\theta_{16\times16} = 1/2\Delta\rho_{8\times8} = \Delta\theta$, and $\Delta\varphi_{16\times16} = 1/2\Delta\varphi_{8\times8} = \Delta\varphi$. Table 2.2 summarizes the coding performance comparison between the geometry extended H.264/AVC intra coding scheme and the standard version at various quantization parameters (QP) for all the test video sequences listed in Table 2.1. It can be observed from

Sequences	number of frames	QP	H.264		H.264+GEO		PSNR diff(dB)	Bit saving(%)	ave. Bit saving	ave. PSNR gain
			YPSNR(dB)	Bit rate(kbps)	YPSNR(dB)	Bit rate(kbps)				
Foreman	30	28	37.67	1967.75	37.68	1868.15	0.01	5.06%	-11.19%	0.60
		32	35.17	1249.87	35.24	1151.42	0.07	7.88%		
		36	32.88	800.06	33.07	719.01	0.19	10.13%		
		40	30.69	520.49	31.04	465.82	0.35	10.50%		
Cartoon	60	28	38.46	1568.72	38.45	1526.17	-0.02	2.71%	-5.23%	0.42
		32	35.69	1114.68	35.71	1072.35	0.02	3.80%		
		36	32.89	780.32	33.02	743.06	0.13	4.78%		
		40	30.10	547.75	30.32	519.72	0.22	5.12%		
Paris	30	28	36.72	4182.66	36.72	4140.79	-0.01	1.00%	-2.03%	0.16
		32	33.47	2941.46	33.45	2893.58	-0.02	1.63%		
		36	30.50	1999.65	30.51	1950.46	0.01	2.46%		
		40	27.67	1327.99	27.74	1291.81	0.07	2.72%		
Tiger	30	28	40.19	4509.31	40.24	4276.97	0.04	5.15%	-8.82%	0.62
		32	37.20	3082.80	37.32	2884.26	0.12	6.44%		
		36	34.41	2031.44	34.66	1899.42	0.26	6.50%		
		40	31.74	1353.94	32.13	1284.95	0.40	5.10%		
Car phone	60	28	38.27	508.67	38.24	487.82	-0.02	4.10%	-7.61%	0.54
		32	35.35	351.98	35.36	328.80	0.01	6.59%		
		36	32.61	239.72	32.70	220.29	0.09	8.10%		
		40	29.84	158.88	30.07	148.92	0.23	6.27%		

Table 2.2: Performance comparison between the geometry extended H.264/AVC intra coding (H.264+GEO) and the standard H.264/AVC (H.264).

Table 2.2 that the geometry extended intra coding scheme outperforms H.264/AVC for all test video sequences at various QP values, and up to 11% average bit rate savings have been achived for the Foreman sequence. Besides, the results presented in Table 2.2 suggest that our geometry-adaptive intra prediction scheme provides more coding gain for higher QP values. This is mainly due to the fact that encoded images lose more texture details by coarser quantization operations, and therefore appear more "piecewise-smooth".

To better illustrate how the geometry-adaptive intra prediction contributes to the enhanced coding performance, using the Foreman sequence as an example, we compare the prediction results generated by the two intra coding schemes in Fig. 2.13. By simple visual inspection, prediction improvements of geometry extended H.264/AVC can be easily observed. Indeed, edges regularity and regions smoothness are significantly enhanced in Fig. 2.13(a) with respect to Fig. 2.13(b). In terms of coding efficiency, this translates into the reduction of prediction residual energy, few number of quantized non-zero transform coefficients. This demonstrates the edge preserving characteristics and better modeling capabilities of the proposed geometric intra prediction scheme. Fig. 2.13(d) shows in detail which blocks are coded using the geometry-adaptive modes, along with the encoded partition wedges. It can be observed that geometric partitions tend to aggregate around edge regions where H.264/AVC standard intra prediction scheme

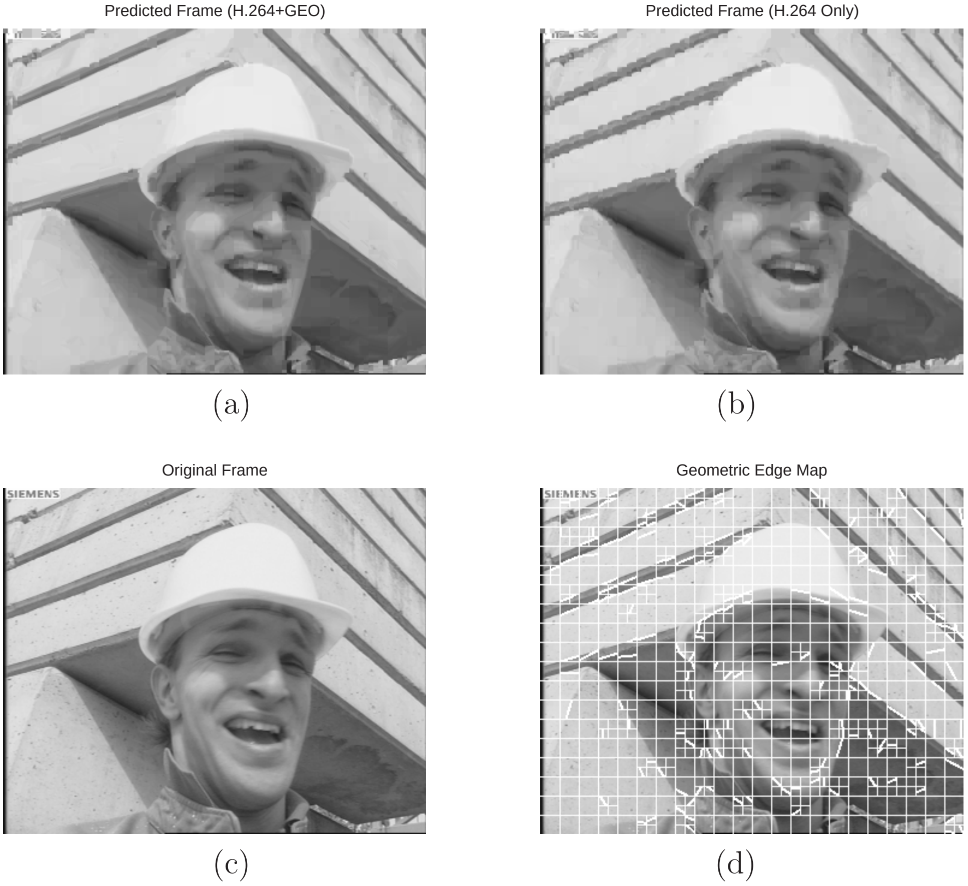
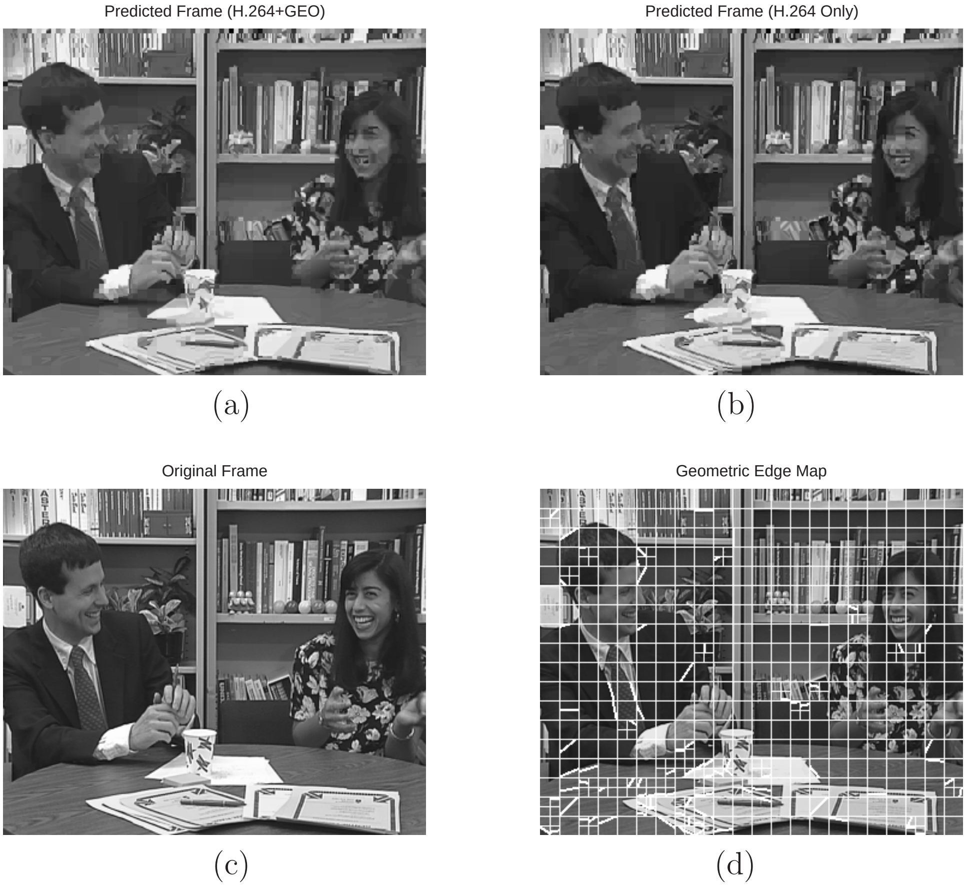


Figure 2.13: Intra predicted pictures for the 15th frame of Foreman(CIF) at QP=28. (a): Intra prediction by geometry extended H.264/AVC (H.264+GEO). (b): Intra prediction by H.264/AVC (H.264). (c): The original picture. (d): The geometric block partitioning map for H.264+GEO.

produces poorer predictions. Also, some wedge partitions are used in non-edge areas in order to better model luminance changes and local gradients.

Among all the test video sequences, in Table 2.1, we find that Paris achieves only a marginal amount of average coding gain when coded with the geometry extended intra coding scheme. This is due to the fact that Foreman contains large amount of piecewise-smooth regions, while Paris on the contrary is mainly composed of abundant complex structures and textures. To justify this, we present in Fig. 2.14 the prediction results of a frame of Paris generated by the two intra coding algorithms. Comparing Fig. 2.14(a) to Fig. 2.14(b), we find that for Paris our geometry-adaptive intra prediction does not



(a) (b)

(c) (d)

Figure 2.14: Intra predicted pictures for the 30th frame of Paris(CIF) at QP=28. (a): Intra prediction by geometry extended H.264/AVC (H.264+GEO). (b): Intra prediction by H.264/AVC (H.264). (c): The original picture. (d): The geometric block partitioning map for H.264+GEO.

provide significantly better predictions over the standard prediction scheme, and the limited improvements appear to aggregate around regions with strong edges (e.g. the edge of the table and the man's hands). This observation is justified by Fig. 2.14(c). Comparing Fig. 2.14(c) to Fig. 2.13(c), we find that much less blocks are predicted by the geometric modes for Paris than Foreman. In fact, according to our experiments, for Foreman, there are more than 30% blocks predicted by the geometric modes, while the percentage drops to about 17% for Paris.

We also study the effect of the geometric parameters on the R-D performance of the geometry extended intra coding scheme. Specifically, we compare three parameter settings: $Par1 = \{\Delta\rho = 0.5, \Delta\theta = \pi/32, \Delta\varphi = \pi/32\}$, $Par2 = \{\Delta\rho = 1, \Delta\theta = \pi/16, \Delta\varphi = \pi/16\}$, and $Par3 = \{\Delta\rho = 2, \Delta\theta = \pi/8, \Delta\varphi = \pi/8\}$. As we can see from Fig. 2.15, the geometry extended intra coding scheme performs better with finer geometry parameters. The improvement comes from two aspects: First, the finer resolutions of the geometric partitioning parameters θ,and ρ provide better approximations of the arbitrarily orientated edges; Second, the finer resolution of the directional prediction angle φ supports

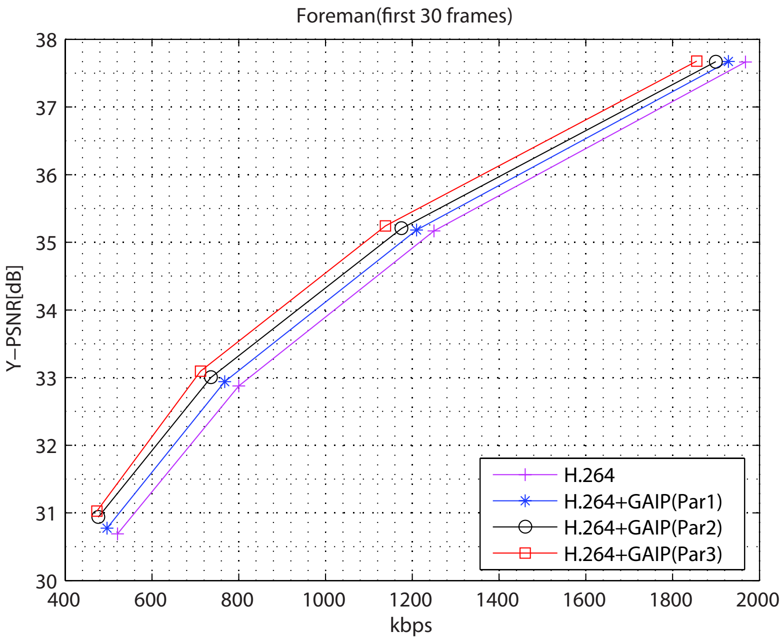


Figure 2.15: Rate-Distortion curves using different sets of geometric parameters for Foreman: $Par1 = \{\Delta\rho = 0.5, \Delta\theta = \pi/32, \Delta\varphi = \pi/32\}$, $Par2 = \{\Delta\rho = 1, \Delta\theta = \pi/16, \Delta\varphi = \pi/16\}$, and $Par3 = \{\Delta\rho = 2, \Delta\theta = \pi/8, \Delta\varphi = \pi/8\}$.

more accurate predictions of directional structures which are not necessarily aligned with the partitioning edge orientation θ inside image blocks. In fact, besides Foreman, we have observed the similar relationship between the parameter resolution and the coding performance for other test video sequences. However, the price of employing finer geometric parameters is the increased computational complexity of the encoder. Besides, for our current implementation, we have found that no significant coding gain can be achieved for geometric parameters finer than those in $Par1$.

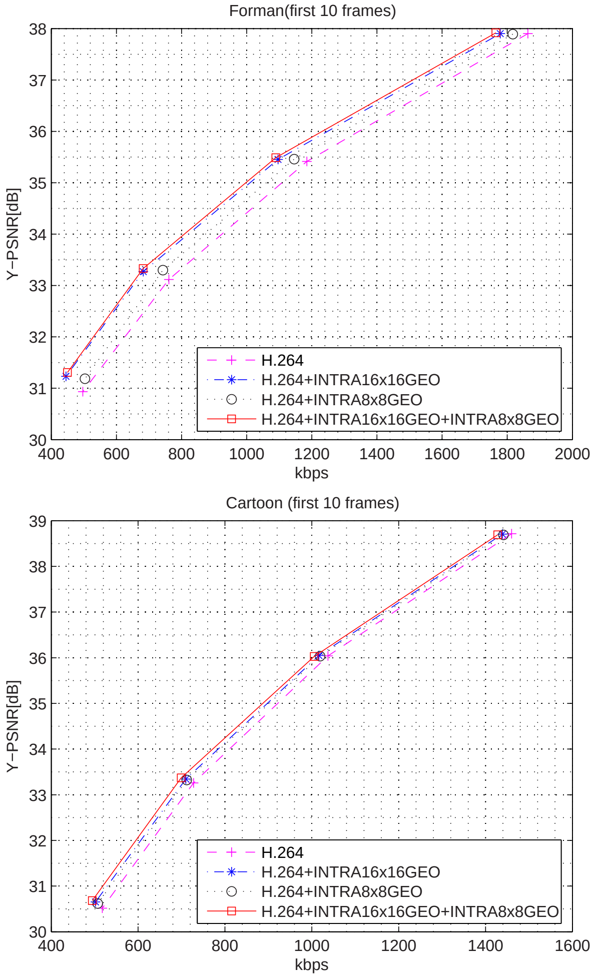


Figure 2.16: Performance comparison between INTRA16x16GEO and INTRA8x8GEO using Foreman and Cartoon.

As it has been mentioned in the previous section, our geometry-adaptive intra prediction is implemented by introducing two geometric modes: INTRA16x16GEO and

INTRA8x8GEO. At the end of this section, we compare the contributions of these two geometric modes to the overall coding performance of the geometry extend H.264 intra coding scheme. Fig. 2.16 shows the experimental results of enabling each of the modes individually and jointly for Foreman and Cartoon. We can see from Fig. 2.16, for Foreman, mode INTRA16x16GEO performs much better than mode INTRA8x8GEO. This is due to the fact that many of the strong edges in Foreman are straight lines, and the Wedgelet partition defined on larger blocks are more efficient to represent them in terms of R-D. However, for the Cartoon case, we observe that the two modes performs almost equally, and we believe this is due to that Cartoon contains both linear strong edges and non-linearly shaped curves. The latter ones obviously are better approximated by smaller Wedgelet partitions.

2.4 Geometry-Adaptive Intra Displacement Prediction

In the previous section, we have discussed the geometry-adaptive intra prediction (GAIP) for enhancing the intra coding efficiency of H.264/AVC. Simulation results show that the GAIP algorithm performs well as expected for images with piecewise-smooth characteristics. However, we also observed that limited coding gain is achieved for video sequences containing significant amount of texture information (e.g. Paris). In deed, as we can see in Fig. 2.14(a), and Fig. 2.14(b) neither the standard scheme nor the GAIP algorithm is able to generate accurate predictions around texture regions. Since textures represent the non-stationary and high frequency components of image signals, previously discussed intra prediction schemes which apply low frequency operations (e.g. interpolation and extrapolation) on local decoded information are not suitable to model complex texture patterns of natural images. Therefore in this section we shall discuss the problem of modeling texture information for intra video coding. More specifically we study the intra displacement prediction (IDP) scheme proposed in [13] in which the texture information of a block is modeled from global decoded information (all the decoded pixels) of the current frame and extend the original IDP scheme with our geometry-adaptive block partitioning structure. The rest of this section is organized as follows: In section 2.4.1, we briefly review the existing texture prediction schemes for intra coding; In section 2.4.2 we discuss in detail the geometry-adaptive intra displacement prediction algorithm (GAIDP). Finally the simulation results will be presented in section 2.4.3. Experimental results show that when the previously presented GAIP algorithm and the GAIDP scheme are jointly applied significant average coding gain can be achieved for the tested video sequences at low bit rate regime.

2.4.1 Review of Existing Texture Prediction Algorithms

Modeling the texture information of natural images is a challenging and keenly studied research topic in computer vision and image processing. In the context of image/video intra coding, a desired texture modeling scheme should generate the sample prediction that best matches the original signal in terms of rate and distortion rather than the visual quality. Aiming at enhancing the performance of video intra coding, in recent years several texture prediction techniques have been proposed [13–16]. Although they are different approaches, a common motivation of these texture prediction algorithms is to search within the decoded region of an intra coded picture the best candidate of the texture pattern to be coded. In the following paragraphs three typical texture prediction algorithms will be briefly discussed.

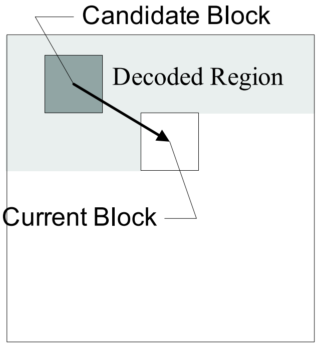


Figure 2.17: Intra displacement prediction (IDP).

***Intra Displacement Prediction (IDP)*:** As proposed in [13], the basic idea of the IDP algorithm is to apply the block motion compensated prediction procedure for inter coding in H.264/AVC to intra coding. In the IDP algorithm, the current block is predicted by referencing to the decoded region within the same frame via a displacement vector as shown in Fig. 2.17. At the encoding stage, a displacement vector estimation procedure searches within the decoded region the best prediction of the current block in terms of rate and distortion. Similar to the motion compensated inter prediction this algorithm requires explicit transmitting the estimated displacement vectors as overheads.

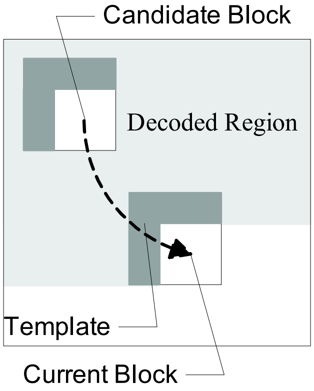


Figure 2.18: Template matching (TM).

***Template Matching (TM)*:** Template Matching (TM) is originally a mathematical tool for texture synthesis problems in computer vision [63,64], and was introduced in [65] and [14] for inter prediction and intra prediction in video coding respectively. In [14] the template matching algorithm searches the previously coded region for the candidate sample prediction block whose neighborhood (template) best matches that of the block to be encoded. Since the search region and the neighborhood of the current block are known at both the encoder and the decoder, no additional side information (e.g. motion vectors) is needed, and identical prediction can be achieved on both sides [16]. The basic procedure of the Template Matching technique is shown in Fig. 2.18.

***Extended Texture Prediction (ETP)*:** Since both the IDP and the TM techniques have their own pros and cons, a new texture prediction method named Extended Texture Prediction (ETP) was proposed in [16] recently to fuse them together. Specifically, the proposed ETP algorithm combines those two schemes together with a simplified version of H.264 intra prediction (only Horizontal, Vertical and DC modes) in a way such that under the quad-tree partitioning structure of a macroblock, each block or sub-block may

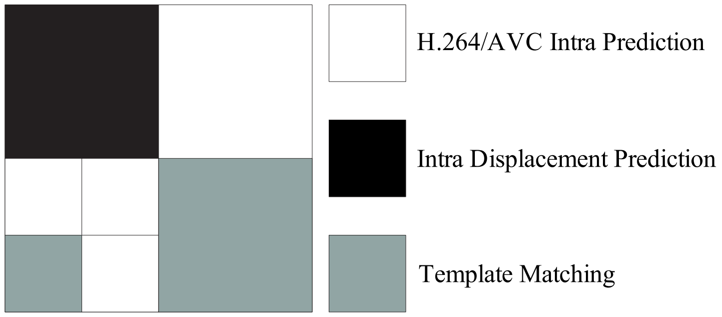


Figure 2.19: Extended Texture Prediction (ETP).

be predicted using either of these schemes based on the RD criterion. For example Fig. 2.19 depicts a possible combination of the prediction modes used by the ETP algorithm when predicting an intra macroblock.

As we can see, the discussed texture prediction algorithms are conceptually similar to the block motion compensated prediction scheme for inter coding. The main difference is that the former ones use the decoded region of the current frame as their reference picture, while the reference picture of the latter one has to be a previously decoded frame. Besides, we also notice that recently the geometry-adaptive block partitioning structure has been proposed to improve the motion compensated inter prediction in video coding [58, 59]. Based on these observations, in this section we propose to extend the IDP algorithm in [13] with our geometry-adaptive block partition structure presented in section 2.3.2. The reason we choose to extend the IDP algorithm mainly comes from the consideration for the decoder complexity: Compared to the other two texture prediction schemes, IDP imposes much less computational complexity at the decoder by explicitly indicating candidate blocks via displacement vectors. This property is crucial to real time video applications.

2.4.2 Geometry-Adaptive Intra Displacement Prediction (GA-IDP)

In this section we shall present the design of the geometry-adaptive intra displacement prediction (GAIDP) algorithm. To implement the GAIDP algorithm we introduce a new mode: INTRA_SEARCH into the H.264/AVC standard mode table. More specifically, the INTRA_SEARCH mode is inserted after the INTRA16x16GEO mode introduced for the GAIP algorithm and before the standard INTRA16x16 modes.

***Macroblock Partitions for GAIDP*:** The original IDP algorithm adopted the same quad-tree macroblock partition structure for the block motion compensated inter prediction in H.264/AVC. In this section, we extend this qual-tree based macroblock partition by introducing the geometry-adaptive block partitioning structure to the 16x16 and 8x8 blocks. Fig. 2.20 shows the macroblock and sub-macroblock (8x8) partitions defined for the GAIDP algorithm. As we can see from Fig. 2.20, the 16x8 and 8x16 macroblock partitions are special cases of the 16x16GEO partitioning structure. Therefore we remove these two special cases from the 16x16GEO partitioning structure to avoid the redundant representation (the similar strategy is also adopted to the 8x8GEO case). The sets of macroblock and sub-macroblock prediction modes associated with the partition scheme in Fig. 2.20 are specified in Table 2.3. Notice that the prediction mode of MODE_SKIP_ISMB in Table 2.3 is just the intra equivalent to the skip mode defined for the motion compensated inter prediction in H.264/AVC [54]. When this mode is used, no information for the macroblock is coded, and the decoder reconstructs the macroblock by only using the information derived from its casual neighbors. When the geometric prediction modes (MODE_16x16GEO_ISMB and BLK_8x8GEO) in Table 2.3 are used to code a macroblock or 8x8 block, the corresponding geometric parameters: θ and ρ have

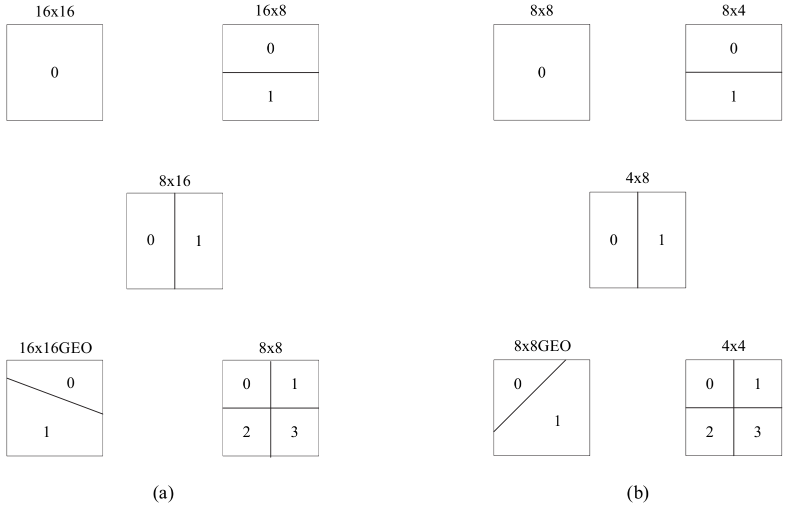
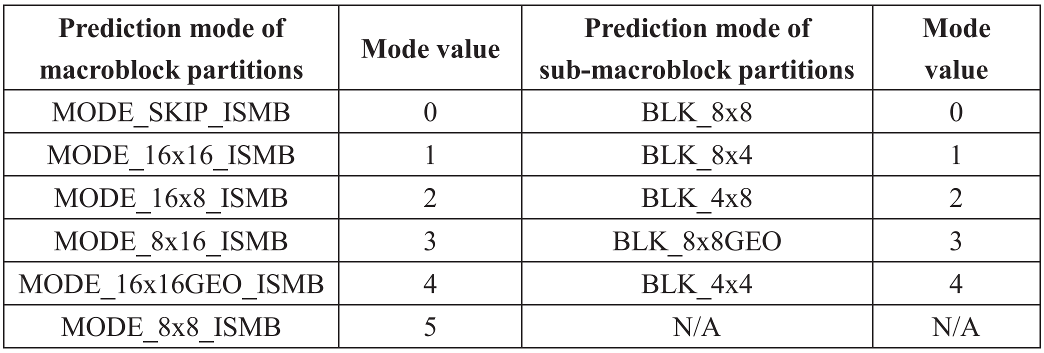


Figure 2.20: The macroblock Partition (a), and sub-macroblock partition (b) structures designed for GAIDP.



Prediction mode of macroblock partitions	Mode value	Prediction mode of sub-macroblock partitions	Mode value
MODE_SKIP_ISMB	0	BLK_8x8	0
MODE_16x16_ISMB	1	BLK_8x4	1
MODE_16x8_ISMB	2	BLK_4x8	2
MODE_8x16_ISMB	3	BLK_8x8GEO	3
MODE_16x16GEO_ISMB	4	BLK_4x4	4
MODE_8x8_ISMB	5	N/A	N/A

Table 2.3: Prediction modes for the macroblock and sub-macroblock partitions of the GAIDP algorithm.

to be sent to the decoder as overheads.

Displacement Vectors (DV): In the GAIDP algorithm, a displacement vector (DV) is used to indicate within the same frame the relative position of the current block to its sample prediction. In our current implementation, displacement vectors with the 1/4 pixel resolution are supported. Since pixel samples at fractional pixel positions are not present in the original video data, estimates of these fractional pixel values have to be generated. To do this, we borrow the sub-pixel interpolation technique specified for the inter prediction purpose in [54] from H.264/AVC. This sub-pixels interpolation procedure can be summarized as follows: the pixel values at 1/2 pixel positions are interpolated from neighboring samples at integer pixel positions using a 6-tap FIR filter, then the pixels at 1/4 pixel positions are linearly interpolated from the nearest two samples at the integer and the 1/2 pixel position. The detailed discussion of this interpolation technique is presented in [66]. During the encoding process, given one of the macroblock partition modes specified in Table 2.3, the best DV for a block or a region of a block is estimated based on the following rate and distortion criterion:

$$DV_{best} = \min_{DV}(\parallel I - \hat{I}(DV) \parallel + \lambda_{motion} R(DV)).$$

Where $\parallel I - \hat{I}(DV) \parallel$ is the distortion (e.g. SSE) between the original region/block and its prediction, $R(DV)$ is the number of bits required to code the DV, and λ_{motion} is the Lagrangian multiplier. Similar to the coding of motion vectors (MV) for the inter prediction case, in our implementation the estimated DV is differentially coded with respect to its predicted value which can be computed from its causal neighbors using a median filter if one or more neighboring macroblocks are coded as the INTRA_SEARCH mode. The standard motion vector prediction procedure is defined in [54] for quad-tree based block partitions. For the geometric block partition case, the task of motion vector prediction becomes more involved and the prediction scheme specified in [67] has to consider the number of block corners present in a partitioned region.

***Adaptive Reference Picture Smoothing*:** The intra reference picture used by the GAIDP algorithm contains visually apparent blocky artifacts (as shown in Fig. 2.21) which are introduced mainly by the quantization of the transformed residues in each 4×4 block. The blocky artifacts have negative impact on the quality of the reference picture, which essentially leads to the efficiency loss of the GAIDP algorithm.

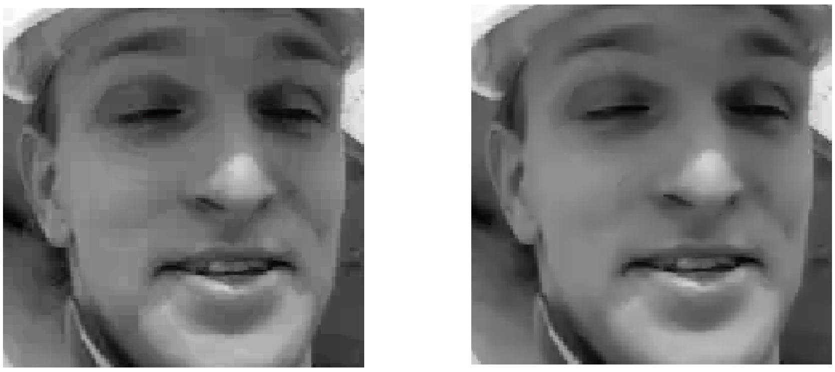


Figure 2.21: Comparison of two intra reference pictures (Left: without smoothing; Right: with adaptive smoothing).

Therefore, in the current the GAIDP algorithm, we have implemented an adaptive smoothing filter to enhance the quality of the intra reference picture. The main challenge for designing this adaptive smoothing filter is to suppress the blocky artifacts as well as to preserve true image edges. Indeed, we have noticed that the in-loop filter [68] of H.264/AVC serves this purpose well. Our adaptive smoothing filter is in fact directly derived from the in-loop filter in [68]. Since the in-loop filtering procedure is very sophisticated and its design is out of the scope of this thesis, we refer interested readers to [68] for detailed discussions. Here, we clarify that our adaptive smoothing filter is almost identical to the in-loop filter in [68], except that we modified the boundary strength[2]

[2] Boundary strength is a value which together with some other parameters indicate if a 4×4 block

(BS) calculation for intra macroblocks coded by the INTRA_SEARCH mode as if they are *inter* coded.

Fig. 2.21 shows two intra reference pictures for the Foreman (CIF) sequence at QP=28. It is easy to observe that the reference picture with adaptive smoothing contains much less blocky artifacts than the one without being smoothed. In fact, more than 1% bit savings can be obtained when our adaptive intra reference smoothing filter is enabled for this sequence.

***Intra Macroblock Encoding and Decoding Procedures*:** Since the essence of the GAIDP algorithm is to predict a macroblock by find its best match in the decoded region of the current frame, an intra reference picture buffer is needed at both the encoder and the decoder to store the decoded region of the current frame. For a video codec with the GAIDP algorithm enabled, both the encoder and the decoder shall update the intra reference picture buffer after each intra macroblock is encoded or decoded no matter this macroblock is coded by the INTRA_SEARCH mode or not. Fig. 2.22 illustrates the intra macroblock encoding procedure of a video codec incorporated with the GAIDP algorithm. As shown in Fig. 2.22, the GAIDP algorithm competes with other intra prediction schemes (e.g. INTRA4x4, INTRA16x16GEO, etc.) and will be

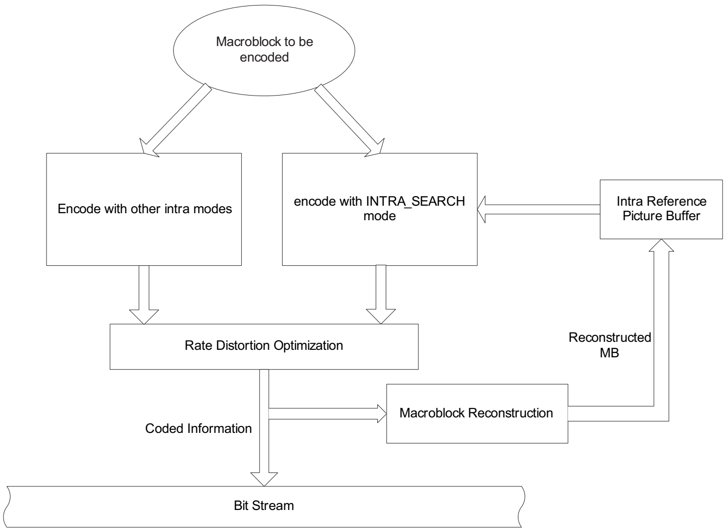

Macroblock to be encoded
Encode with other intra modes
encode with INTRA_SEARCH mode
Intra Reference Picture Buffer
Rate Distortion Optimization
Reconstructed MB
Macroblock Reconstruction
Coded Information
Bit Stream

Figure 2.22: Intra macroblock encoding procedure of a video codec incorporated with the GAIDP algorithm.

boundary shall be filtered or not and the strength of the smoothing filter in [68] selected to predict the current intra macroblock if it leads to the lowest R-D cost. After the macroblock is encoded with the best prediction scheme, it will be reconstructed, and the reconstructed macroblock is sent to the intra reference picture buffer which will then be used to encode the future intra macroblocks. The corresponding intra macroblock decoding procedure is depicted in Fig. 2.23.

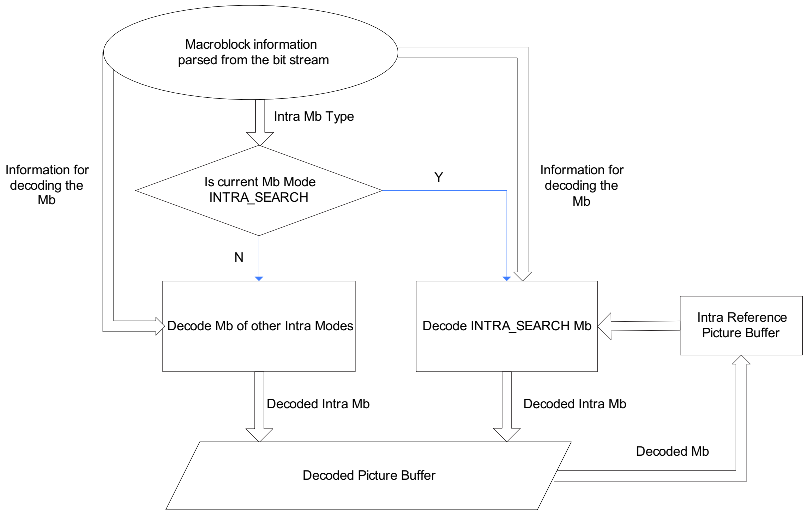
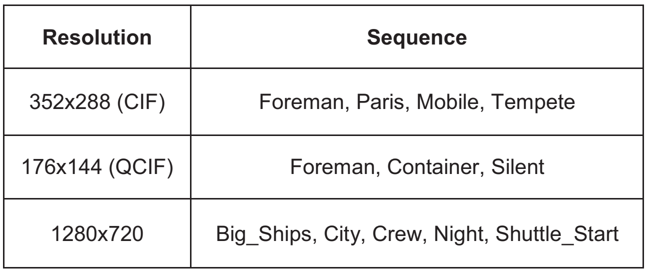

Macroblock information parsed from the bit stream
Intra Mb Type
Information for decoding the Mb
Is current Mb Mode INTRA_SEARCH
Y
Information for decoding the Mb
N
Decode Mb of other Intra Modes
Decode INTRA_SEARCH Mb
Intra Reference Picture Buffer
Decoded Intra Mb
Decoded Intra Mb
Decoded Mb
Decoded Picture Buffer

Figure 2.23: Intra macroblock decoding procedure of a video codec incorporated with the GAIDP algorithm.

2.4.3 Simulation Results and Discussions

To test the coding performance of the GAIDP algorithm, several simulations have been conducted. In this section, we report and discuss the experimental results. In order to have an accurate and thorough evaluation of the algorithm, our simulations are conducted on twelve video sequences which have been recommended as the standard video coding test sequences in a recent VCEG (Video Coding Expert Group) meeting [69]. Table 2.4 contains the test video sequences used for our simulations. The geometric parameters of the experiments are identical to those specified in section 2.3.5. The H.264/AVC baseline profile settings are employed to conduct simulations. Besides, the adaptive smoothing filter discussed in previous section is enabled throughout the experiments.

In this section, we compare the coding performances of the following three cases:H.264 +GAIDP, H.264+GAIP and H.264+GAIDP+GAIP to the original H.264/AVC intra



Resolution	Sequence
352x288 (CIF)	Foreman, Paris, Mobile, Tempete
176x144 (QCIF)	Foreman, Container, Silent
1280x720	Big_Ships, City, Crew, Night, Shuttle_Start

Table 2.4: Test video sequences.

coding. R-D curves of the three coding schemes are shown in Fig. 2.24 and Fig. 2.25. In fact from Fig. 2.24 and Fig. 2.25, several interesting observations can be obtained, which may help us better understand the properties of the GAIDP and GAIP algorithms presented in this chapter.

First, compared to H.264, the intra coding scheme that enables the GAIDP algorithm (H.264+GAIDP) achieves better coding performance on Foreman (CIF and QCIF), City, Big_Ships, Crew, and Mobile sequences. This might be due to the fact that these sequences contain fairly large amount of texture patterns that repeatedly appear within the same video frame (e.g. windows of buildings in City and edges of the wall in Foreman). Besides, for those aforementioned sequences the behavior of the R-D curves of the H.264+GAIDP scheme suggests that GAIDP performs better at high bit rates than low bit rates. This phenomenon probably can be explained as follows: At low bit rates, higher QP values introduce more distortion into the coded macroblocks which are used to build the intra reference picture for the GAIDP algorithm. Therefore the quality of sample predictions generated by GAIDP are often worse at lower bit rates than higher bit rates, which leads to the loss of coding efficiency of the GAIDP algorithm.

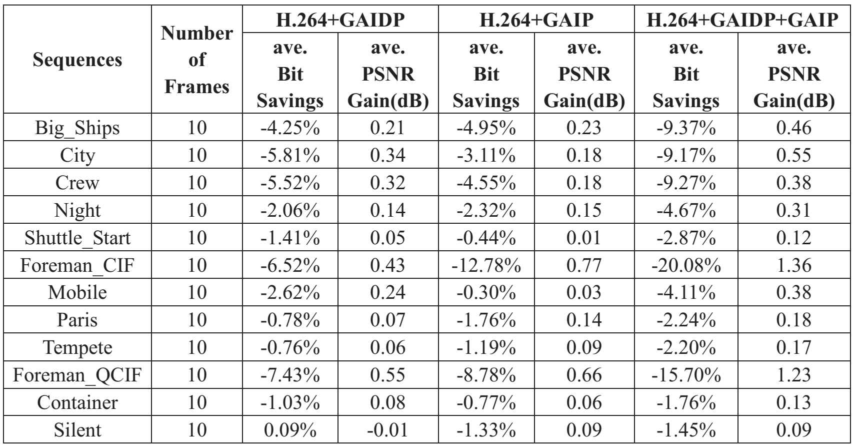
Second, we found that when the GAIP and GAIDP algorithms are both enabled (H.264+GAIDP+GAIP), a better intra coding performance is achieved compared to the other two cases for all test sequences. This observation suggests that the GAIP and GAIDP algorithms are complementary to each other in a way that the GAIP algorithm is good at modeling the piecewise smooth image regions and the GAIDP algorithm helps to handle certain complex texture patterns. The conjecture is also partially justified by the R-D curves in Fig. 2.24 and Fig. 2.25. As we can see for most of the test sequences, at higher bit rates the H.264+GAIDP+GAIP scheme performs similarly to the

H.264+GAIDP scheme, while at lower bit rates the former scheme obviously outperforms the latter one. The reason is that at higher bit rates, texture information is well preserved, while at lower bit rates, certain amount of texture information will be removed due to the quantization operation and images appear smoother.

Third,currently there still exist some of video sequences that can not be well modeled by any of the intra prediction schemes we have discussed in this chapter. For example the Tempete, Paris, and Silent sequences only achieves marginal coding gains even when both GAIP and GAIDP are enabled. In fact, these sequences contain large amount of complex texture patterns that unlike the texture patterns of the City sequence do not simply reappear at different spatial locations within the same video frame. Apparently the current GAIDP algorithm is unable to handle these cases well. In addition to the aforementioned sequences, We notice that the Shuttle_Start and Container sequences also receive very moderate coding gains in our simulations, and these two sequences do not contain lots of texture information. In fact, Since these two sequences contain abundant smooth regions, the original H.264 intra prediction has done a very good job, therefore the potential space for further improvements should be very limited. In Table 2.5, we have listed the average coding gains of the two schemes: H.264+GAIDP, and H.264+GAIDP+GAIP over the original H.264 intra coding scheme. As we can see when our geometry-adaptive intra prediction schemes presented in this chapter are enabled, different degrees of improvement of H.264/AVC intra coding performance have been achieved for all test video sequences. Especially, the Foreman_CIF sequence obtains 20% of bit savings the highest coding gain among all.

2.5 Concluding Remarks

In this chapter, we have studied the topic of intra prediction in block based hybrid video coding systems. Noticing that the traditional quad-tree based block partitioning structure is suboptimal in the sense of rate and distortion to represent video signals in which regions with different statistical characteristics are separated by smooth discontinuities, we proposed a geometry-adaptive block partitioning structure within the quad-tree partitioned blocks (tree leaves). We show in detail the theoretical analysis presented in [10]



Sequences	Number of Frames	H.264+GAIDP		H.264+GAIP		H.264+GAIDP+GAIP	
		ave. Bit Savings	ave. PSNR Gain(dB)	ave. Bit Savings	ave. PSNR Gain(dB)	ave. Bit Savings	ave. PSNR Gain(dB)
Big_Ships	10	-4.25%	0.21	-4.95%	0.23	-9.37%	0.46
City	10	-5.81%	0.34	-3.11%	0.18	-9.17%	0.55
Crew	10	-5.52%	0.32	-4.55%	0.18	-9.27%	0.38
Night	10	-2.06%	0.14	-2.32%	0.15	-4.67%	0.31
Shuttle_Start	10	-1.41%	0.05	-0.44%	0.01	-2.87%	0.12
Foreman_CIF	10	-6.52%	0.43	-12.78%	0.77	-20.08%	1.36
Mobile	10	-2.62%	0.24	-0.30%	0.03	-4.11%	0.38
Paris	10	-0.78%	0.07	-1.76%	0.14	-2.24%	0.18
Tempete	10	-0.76%	0.06	-1.19%	0.09	-2.20%	0.17
Foreman_QCIF	10	-7.43%	0.55	-8.78%	0.66	-15.70%	1.23
Container	10	-1.03%	0.08	-0.77%	0.06	-1.76%	0.13
Silent	10	0.09%	-0.01	-1.33%	0.09	-1.45%	0.09

Table 2.5: The average gains of H.264+GAIDP, H.264+GAIP and H.264+GAIDP+GAIP over the original H.264/AVC intra coding scheme.

that when the geometric block partitions are incorporated into the quad-tree based block partitioning structure a better R-D performance can be obtained for piecewise-smooth images. In our current realization, the smooth discontinuities (partitioning curves) are featured by arbitrarily orientated line segments (Wedgelets) which are parameterized by two geometric parameters: the orientation θ, and the distance ρ.

Aiming at improving the intra prediction performance of H.264/AVC the state-of-art video coding standard, we applied the geometry-adaptive block partitioning structure on the 16x16 and 8x8 blocks of the H.264/AVC intra coding scheme, and designed our own schemes to model each geometrically region using either the neighboring decoded predictors (directional prediction) or the statistics inside (DC modeling). We refer this intra prediction scheme to as geometry-adaptive intra prediction (GAIP). Simulation results show that when GAIP is enabled, the intra prediction performance of H.264/AVC is enhanced for sequences containing piecewise-smooth image regions, which leads to impressive coding gains over the original H.264/AVC intra coding scheme.

Modeling texture information to facilitate image or video coding is a challenging task. In this chapter, we discussed several existing texture prediction techniques proposed for intra video coding. Among all the discussed texture prediction techniques, we chose the IDP algorithm and extended it with our geometry-adaptive block partitioning structure. Simulation results show that the GAIDP algorithm is able to improve the intra coding performance of H.264/AVC for certain video sequences containing repeated texture patterns. More importantly, we found that when GAIP and GAIDP are both enabled, the intra coding performance of H.264/AVC is enhanced for all the twelve test video sequences. Among them, Foreman achieves maximum average coding gain of 20% bit savings which is equivalent to more than 1dB PSNR improvement.

Although in this chapter, we have demonstrated the power of geometry-adaptive block partitioning structure for modeling the intra video data, our current design is still quite simple and may be further improved. For example, currently we only use first order curves to model the smooth discontinuities, which is apparently not sophisticated enough to describe edges of natural images. Therefore, in the future higher order of curves should be considered to represent the partitioning edge of image regions. Besides, as we have discussed in section 2.4.3, the modeling capability of the GAIDP algorithm is limited in the case where texure patterns appear repeatedly within a video frame. For more complex texture patterns the GAIDP algorithm could not provide better predictions than H.264/AVC. In fact, within the block coding framework of hybrid video coders, task of modeling texture information is even more difficult, and to our best knowledge there is no such a texture modeling scheme that is able to significantly enhance the intra coding performance of H.264/AVC for video sequences that contain various texture patterns. Therefore in our own opinion, instead of trying to build a single algorithm that is able to handle complex texture patterns, a more promising way might be to model complex texture patterns by fusing together different predictions provided by several prediction schemes. Each of those prediction schemes is not necessarily very sophisticated and might be only able to handle certain types of texture patterns. In our current implementation, the optimal partitioning edge of a block is computed via exhaustive search. The exhaustive searching scheme is highly computation demanding especially when fine edge parameter resolutions are used in real time applications. Therefore fast searching algorithms shall be explored in the future. Indeed, one possible way of designing the fast algorithm is to exploit the geometric information of neighboring decoded blocks. For example, if we know the block on top of the current one has a vertically orientated edge, then it is more probable that there is an edge passing through the current block with a similar orientation.

Besides the aforementioned possible improvements, there are some other thoughts regarding to the presented geometry-adaptive intra prediction algorithms, which are worth of exploration in the future. First, in this thesis we have shown simulations of our algorithms on video sequences with ideal quality. In many real applications, video data may have non-ideal quality (e.g. contaminated by noise and blur), in such situations, the robustness of the proposed algorithms shall be tested; Second, Similar to the motion compensated inter prediction [70], the performance of the proposed GAIDP algorithm can be affected by illumination variations. Therefore in the future, we shall explore possible ways to accommodate the negative impact of illumination variations.

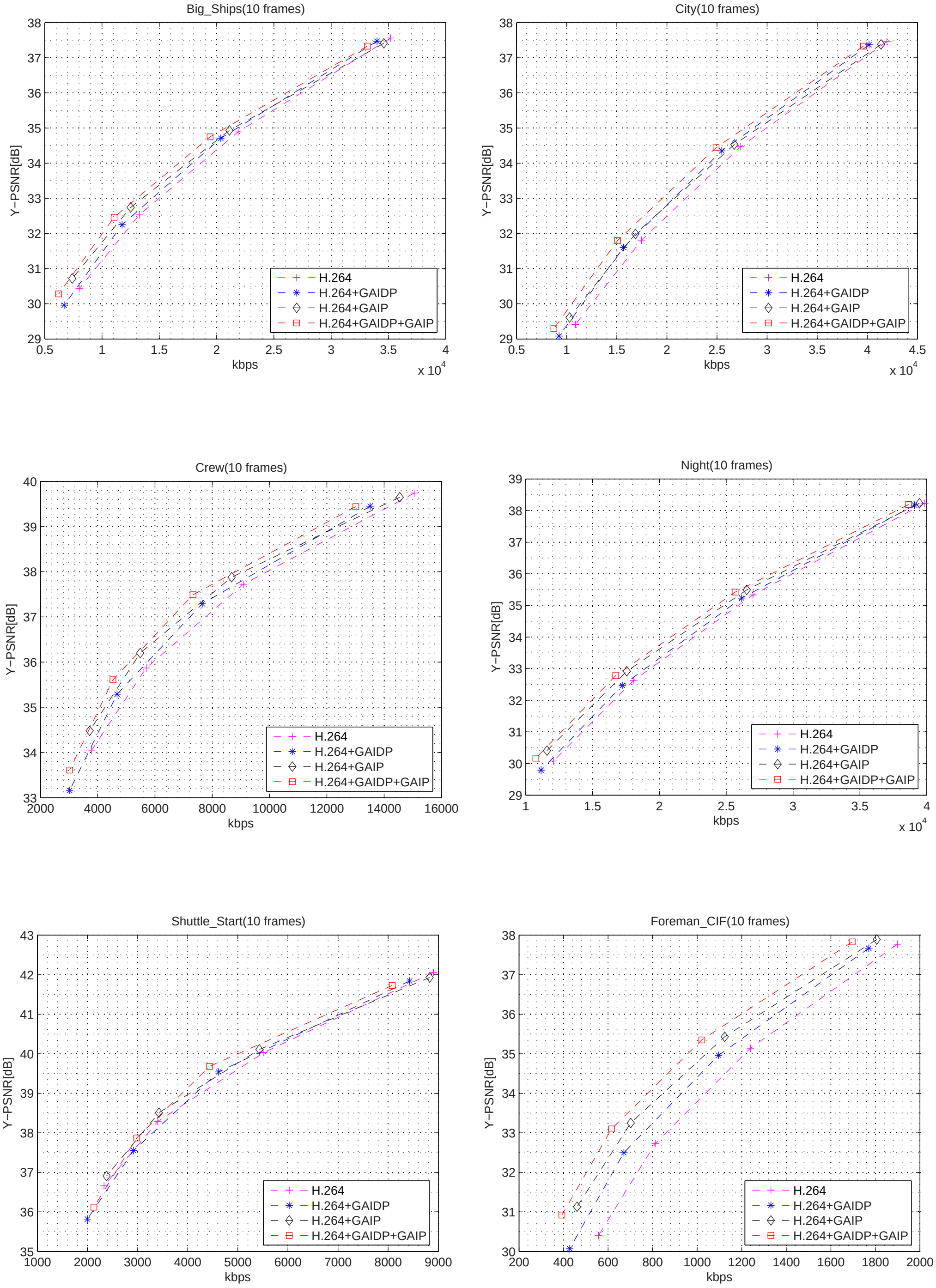


Figure 2.24: Simulation results of three intra coding schemes: H.264, H.264+GAIDP, H.264+GAIP and H.264+GAIDP+GAIP at QP=[28,32,36,40].

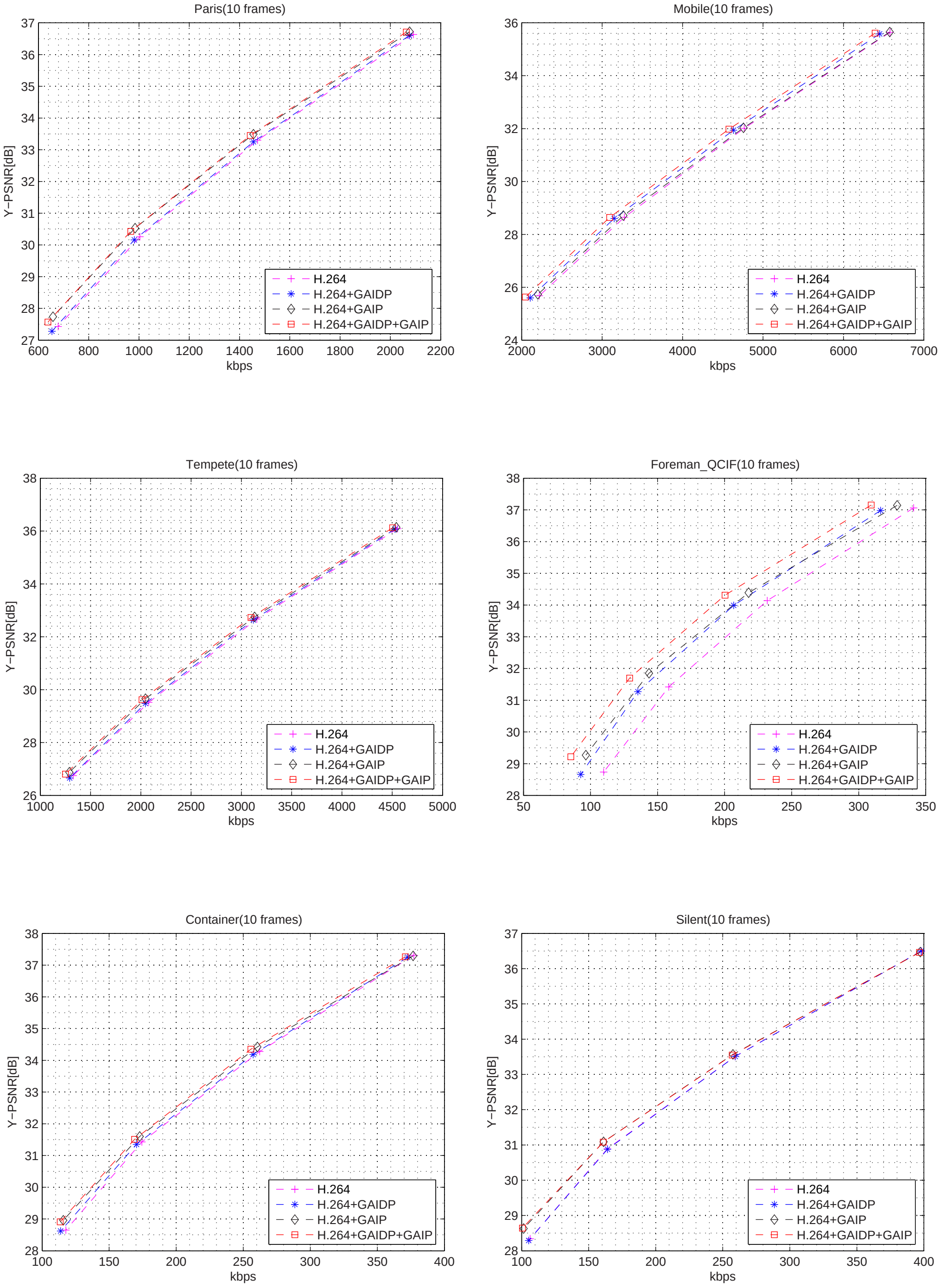


Figure 2.25: Simulation results of three intra coding schemes: H.264, H.264+GAIDP, H.264+GAIP and H.264+GAIDP+GAIP at QP=[28,32,36,40].

Chapter 3

Pedestrian Detection and Tracking in Infrared Imagery

3.1 Introduction

3.1.1 Overview

Pedestrian detection and tracking have been extensively studied in computer vision over the past decade. To meet the challenges arising from large variability of body pose, clothing and environmental factors (e.g. [17,18]), various algorithms have been developed. For example, wavelet based appearance representations with support vector machine (SVM) classifier were proposed in [19,20]. Silhouette and shape-based detection techniques have been adopted in [21–24]. In [25–27], human body, pose and motion are respectively, modeled and utilized for pedestrian detection; Periodicity and self-similarity of human motion analysis is proposed to detect pedestrians in [28]. Feature vectors involving both appearance and motion information are passed to an Adaboost classifier for pedestrian recognition in [18]. Principle component analysis (PCA) and time-delay neural networks are jointly used for object recognition and tracking in [29]. A stereo-based disparity segmentation and neural network-based pedestrian recognition algorithm appears in [30].

Effective pedestrian detection and tracking algorithms in visible spectrum have found many important applications from video surveillance to intelligent vehicles. However, under certain circumstances (e.g., in nights or bad weathers), sensing in visible spectrum becomes infeasible or severely impaired, which calls for the imaging modalities beyond visible spectrum. In particular, the cost of thermal sensors has reduced dramatically in the past decades and we start to witness that infrared (IR) sensors with high dynamic range and sensitivity become more widely deployed in the applications such as night-vision and all-weather surveillance.

Driven by the decreasing cost of IR sensors, there have been a flurry of works on pedestrian detection and tracking in IR imagery recently. In [31], probabilistic templates are used to capture the variations in human shape for pedestrian detection. In [32], support vector machine and Kalman filtering are adopted for detection and tracking, respectively. In [33], the P-tile method is developed to detect human head first, and then human torso and legs are included by local search. In [34], a particle swarm optimization algorithm is proposed for human detection in IR imagery. In [35], a two-stage template-based method with an Adaboosted classifier was presented for pedestrian detection.

In this chapter we present a pedestrian detection and tracking scheme via layered representation. In the proposed algorithm, infrared images are first separated into two layers: a background layer (still layer), and a foreground layer (moving layer). Pedestrians are then detected in the foreground layer using the appearance cue. To facilitate the task of pedestrian tracking, we formulate the problem of shot segmentation and present a graph matching-based tracking method.

3.1.2 Contributions

The contributions of this work are summarized into the following three aspects.

Layered representation. Layered representations have been widely used for object tracking in visible imagery [71–73]. For IR imagery, layered representation is also attractive because it facilitates the statistical modeling of senor data even when motion cue is not directly useable. We propose to decompose an IR image into two layers: background (still objects) and foreground (moving objects). A light version of generalized expectation maximization (GEM) algorithm [71] is developed to dynamically mosaic the background by registering IR images based on the global motion model. When compared with the GEM algorithm for visible imagery [71], ours runs much faster since it only involves appearance and a global motion model.

Pedestrian detection using the appearance cue. We propose to detect pedestrians from the foreground layer by exploiting the appearance information. To obtain accurate localization of individual pedestrians, a multi-scale principle component analysis (PCA) technique [74] is developed to accommodate pedestrians with various sizes in the foreground layer. Compared to the benchmark work presented in [35] using the OSU infrared image database, our appearance based approach seems to achieve improved true-positive performance in the situation of crowed pedestrians and false-positive performance for low-SNR IR imagery.

Shot segmentation and tracking. We address the problem of shot segmentation to facilitate tracking of pedestrians in a long sequence (i.e., with scene changes) or a collection of IR imagery with unknown temporally sampling information (e.g., snapshots taken at random timing). The sequence is first segmented into shots (temporally correlated frames) based on Hausdorff distances; then within each shot, pedestrian tracking is formulated as a matching problem on weighted bipartite graphs. Each pedestrian corresponds to a node and every potential matching between two nodes in adjacent frames corresponds to a weighted edge whose weight reflects the tradeoff between appearance similarity and geometric proximity. When compared with the existing Kalman-filtering based tracking [32], ours does not require any assumption about the characteristics of motion trajectory.

The rest of this chapter is organized as follows: Section 3.2 describes a generalized EM algorithm for dynamic background mosaicing and discusses the detection of polarity switch. Section 3.3 presents an appearance based pedestrian detection technique, which accommodates various pedestrian sizes for pedestrian detection. Section 3.4 covers shot segmentation and matching-based pedestrian tracking. Experimental results are reported and discussed in Section 3.5. We also discuss some challenging situations involving polarity switch of IR imagery in Section 3.5.4. Concluding remarks and future work are included in Section 3.6.

3.2 Two-Layer Representation

3.2.1 Modeling of IR Imagery

Layered representation of image sequences was first introduced by [75] in which video is decomposed into layers with different motions. Since then, numerous studies have followed [73]. For instance, decomposing video into layers was formulated as a maximum-likelihood (ML) estimation of multiple motion models in [76]. In [71], a generalized expectation maximization (GEM) algorithm was developed to learn a mixture of sprites (layers) from a video sequence. Most recently, [73] shows how occlusion and rigidity can be exploited to enable a computationally simple algorithm to jointly estimate the unknown background and rigid shape of the moving object directly from the image intensity values. Layered representations are attractive because they support a variety of high-level vision tasks including recognition, tracking and retrieval.

There are two main issues in layered representation: the number of layers and the layer decomposition. For pedestrian detection and tracking in IR imagery, we suggest that two layers consisting of background and foreground (similar to figure-ground model in [73]) will be sufficient. Background and foreground layers include still and moving objects in the scene, respectively. However, unlike [73] emphasizing the joint utilization of rigidity and occlusion, we argue that simple cues such as shape and appearance are more appropriate for surveillance applications where camera distance is in the middle-to-far range. Such observation also allows us to derive a both conceptually and computationally simple layer decomposition algorithm for IR imagery. Our algorithm can be viewed as a light version of GEM algorithm introduced in [71]: we only consider a simple global translational model to compensate the camera motion and we directly obtain the binary mask without computationally demanding non-linear optimization procedure. Such computational reduction is often critical to IR-related applications (e.g., intelligent vehicles) where real-time implementation is highly desirable.

The following notations are adopted in this chapter. A sequence of IR images are denoted by $I_k(m,n)$ where $k = 1, 2, ..., K$ and $(m,n) \in \Omega = [1,H] \times [1,W]$ are temporal and spatial variables, respectively. Each sequence is assumed to be taken at a medium-to-large camera distance and within a short period of time such that environmental factors such as precipitation and temperature remain unchanged.

Similar to [71] and [73], our two-layer model is described by:

$$I_k = (1 - M_k)B_k + M_k F_k + N_k. \tag{3.1}$$

where B_k, F_k stand for background, foreground layers respectively, M_k represents the foreground mask, and N_k models sensor noise. Note that background layer B_k can be described by a sprite (we shall elaborate on this next) and foreground F_k includes both pedestrian and non-pedestrian moving objects. When compared with the model used in [71], ours does not involve complicated motion characterization, which might not be feasible for IR imagery. We also note that there exists non-parametric techniques for background and foreground modeling of visible imagery in the literature (e.g. [77]) but their computational cost is prohibitive.

Another important difference between IR imagery and visible imagery is the amount of sensor noise. The strength of thermal sensor noise is strong enough to be highly visible. For example, for the IR cameras considered in our experiments, we have found that the additive noise term N_k approximately observes the Gaussian distribution with zero mean and variance of $\sigma_w \in [40, 60]$. Such heavy noise poses a challenge to both background extraction and pedestrian detection. In background extraction, we will suppress noise components by adaptive temporal filtering; in pedestrian detection, we will empirically choose the number of principle components to minimize the noise interference.

3.2.2 Background Extraction

Due to simplification of our two-layer model, the primary task in layer decomposition is to resolve the uncertainty about the binary mask M_k. In the absence of camera motion (i.e. $B_k = B$), EM algorithm can be used to extract the still background. For instance, mask layer M_k and back ground layer $B_k = B$ can be iteratively refined (refer to [78, page 310]). With camera motion [79], we are facing a more general background mosaicking problem [80] where each B_k can be viewed as a subset of the mosaicked image B or sprite [71]. Under the assumption of camera panning motion, we have developed a generalized EM algorithm as shown in Fig. 3.1 where an additional global camera motion compensation step is adopted to handle the sprite generation. At the initialization, phase correlation method [81] is used to register K IR images and produce an initial estimation of $B_k,\ k = 1, 2, ..., K$. At each each iteration, we update M_k by thresholding $|I_k - B_k|$, refine the alignment results by excluding the foreground pixels and then update B_k by adaptively averaging the K registered images.

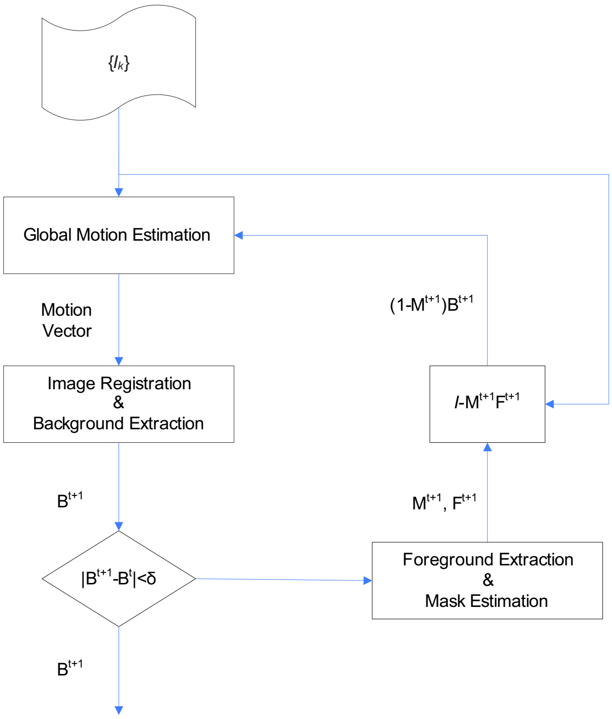


Figure 3.1: Flow chart for background extraction.

The stopping criterion is set to be $||B^{t+1} - B^t||^2 < \delta$, where δ is a small positive number (e.g. 0.01). We have found that such algorithm converges rapidly (typical three iterations) [82]. To improve the robustness, we use morphological filtering to process the mask layer to eliminate small objects (connected components) and fill in the holes of moving objects. After background extraction, the set $\Omega_{mov} = \{(m, n)|M_k(m, n) = 1\}$ consists of connected components $R_1, ..., R_{E_k}$ where E_k is the total number of moving objects.

Motionless pedestrians. One tricky issue that often arises from our short acquisition time assumption is that pedestrians could remain still with respect to the background throughout the whole sequence. In such case of "motionless pedestrians", miss detection is likely to occur unless we exhaustively search the background layer, which defeats the merit of layer decomposition (although motion cue is not directly used in our layer decomposition, thresholding operator in our GEM algorithm can be viewed as motion detector). To overcome such difficulty, we suggest that the problem roots in that motion is a concept relative to time and propose the following engineering solution—i.e., IR sensor can be programmed to take shots either frequently in a day (e.g., every other hour) or at particular chosen timing (e.g., 6AM when it is unlikely to catch pedestrians). We believe that those strategies are also applicable to pedestrian detection in the visible spectrum.

Polarity switch. Another interesting phenomenon with IR imaging is the so called "polarity switch". When it occurs, hot and cold ranges of thermal sensor get reversed: For instance, pedestrians that normally give rise to bright pixels could become dark pixels as shown in Fig. 3.7. This phenomenon definitely poses a challenge to the pedestrian detection task, because the prior knowledge that pedestrians appear brighter than non-pedestrian objects no longer holds in polarity reversed IR images. However, to the best of our knowledge, the mechanism of polarity switch has not been well documented in the literature. It is roughly known that various environmental factors such as outdoor temperature and specular surfaces could trigger the switch of polarity [83]. The test data available for experimental studies are also limited at this point.

To meet the challenge of the polarity switch phenomenon, in this work we have developed a heuristic strategy to detect polarity switched IR sequences by using the outcome of the background extraction procedure. Here, we briefly describe the polarity switch detection strategy as follows: Assuming that all pedestrians would undergo the same switch, we propose to compute the average of $e_k = (F_k - B_k)M_k$ over the set Ω_{mov} and then the polarity of e_{avg} could be used as the indicator of polarity switch (refer to Fig. 3.13). Note that in some situations polarity switch could occur with a single pedestrian (i.e., become a spatially localized event), whose detection is beyond the scope of this work (refer to Fig. 3.14 for more details). Once the polarity switched IR sequences have been identified, we are able to deal with the pedestrian detection task for both normal and polarity switched IR images using the appearance based technique described in the next section.

3.3 Static Pedestrian Detection Using Appearance

Pedestrian detection and tracking deal with the questions of “where are the pedestrians” and “where does this pedestrian go”, respectively. These two classes of questions are closely related but have clearly different objectives. Although detection is often conceived as a preliminary step before tracking, we note that temporal constraint imposed by most tracking algorithms can help the detection task as well. Therefore, we use static pedestrian detection to denote the class of techniques that do not involve any temporal cues.

Spatial cues generated by layer decomposition include both the shape and appearance. The shape cue has been exploited for the pedestrian detection purpose in several previous works (e.g. [31, 35],). For example, in [35], shape-based adaptive filters trained by an adaboosting procedure are applied to locate individual pedestrians. This shape-based approach is very effective as reported for the OSU thermal pedestrian database which covers a wide variety of challenging scenarios including rainy weathers, polarity switch and occluded pedestrians.

Unlike shape information, appearance information is rarely exploited in literature for pedestrian detection tasks in infrared imagery. we conjecture it might be due to the fact that some appearance cues such as color and texture that have been employed for visible spectrum [19] are absent in infrared imagery. Besides, the polarity switch phenomenon poses an obstacle for direct exploiting the appearance cue. However, we argue that appearance information can be very useful for pedestrian detection in infrared imagery if it is properly manipulated. More importantly, as we shall demonstrate in section 3.5.2, our appearance based algorithm seems to be more sensitive compared to the shape based approach in [35].

Among all appearance based techniques, in this work we choose to use a principle component analysis (PCA) based approach to represent pedestrian’s appearance variations in infrared imagery. PCA has been widely used for object recognition and detection in visible spectrum. One famous example is the “eigenface” approach for face detection and recognition in [74]. Despite of its huge impact, the original "eigenface" approach has its own drawbacks: As indicated in [74] this approach is sensitive to sensor noise, illumination variation, and interference of non-face objects. Therefore, recent research works have incorporated advanced classification techniques to enhance its robustness. For example, in [84], a support vector machine (SVM) trained in the "eigenface" space is applied for face detection. For infrared imagery, we notice that the direct application of the original PCA based approach proposed in [74] shall face the similar difficulties as in visible spectrum (Although "illumination variation" does not exist for thermal sensors, we found that the thermal signatures of moving objects and backgrounds vary significantly due to weather conditions and environmental temperatures, which poses the similar effect as "illumination variation"). Besides, the original PCA approach obviously is not applicable to polarity-switched infrared images.

Therefore, to exploit the appearance cue of pedestrians in infrared imagery, we have developed a modified PCA approach based on the layered representation presented in the previous section. To meet the aformentioned challenges for pedestrian detection in infrared imagery, we have carefully revised the steps of the original "eigenface" approach. Our modified PCA approach is comprised of four components: normalization, training, projection, and local aggregation. We shall introduce them one by one in the following paragraphs.

Normalization. Based on the background extraction results, we have estimates of the background layers $\{B_k\}$, the foreground layers $\{F_k\}$, and the masks $\{M_k\}$. The pedestrian appearance information presents in the foreground layers $\{F_k\}$. Due to the significant thermal signature variations of pedestrians (the "polarity switch" phenomenon can be viewed as the extreme case of thermal variations), we found that directly applying the PCA algorithm on the foreground layers $\{F_k\}$ does not guarantee a satisfying detection performance. However, careful investigations on infrared imagery suggest that relative thermal signatures of pedestrians to their local background $\{B_k M_k\}$ vary less significantly. This observation motivates us to normalize the foreground layers.

We first calculate the difference image $e_k = (F_k - B_k)M_k$ for the kth infrared image.

Then e_k is normalized by its maximum absolute value, i.e:

$$\bar{e}_k = \frac{e_k}{e_{k,max}} \tag{3.2}$$

Where $e_{k,max} = \max_{(m,n)} |e_k(m,n)|$. To accommodate polarity-switched sequences detected by the heuristic method presented in Section 3.2, we simply take $\bar{e}_k = |\frac{e_k}{e_{k,max}}|$. Note that both training and detection are performed with the normalized difference image $\bar{e}_k$. This normalization operation is important to accommodate a variety of environmental conditions across the databases we have tested as well as polarity switch. We also note that similar techniques have been used by other object detection algorithms (e.g. [18]).

Training. To serve the multi-scale pedestrian detection purpose, we have manually clipped training pedestrian templates for each of the three training template sets. In our previous work [82], we have chosen a single pedestrian template set with a fixed window size of 30 × 20. This template size performs well for OSU thermal pedestrian database due to the fact that the pedestrian sizes in this database are almost constant. However, such fixed template size becomes less effective as camera distance varies (i.e., lack of scale invariance in [82]). Motivated by previous works [35,85], we propose to use a sequence of template sets with varying sizes. In our current implementation, we have chosen three sets of pedestrian training templates: $T1$, $T2$, and $T3$ sized by 110×40, 60×24, and 30×20 pixels, respectively. Some samples of the training pedestrian templates are illustrated in Fig. 3.2. We follow the same procedure of eigenvector decomposition describe in [74] to derive a set of orthonormal principle eigenvectors $\vec{V_i} = \{v(1), v(2), ..., v(N_i)\},\ i = 1,2,3$ and a corresponding mean vector m_i for each training template set. The optimal value of N_i is determined by the power of noise as well as the subspace structure of signal. For the OSU and WVU thermal pedestrian databases, we have found that a small number of principle eigenvectors, for example: $N_1 = N_2 = 16$, and $N_3 = 12$ give good signal-noise separation results.

Projection. We follow the same eigenvector projection procedure as presented in [74], to obtain an cost map $p(m,n)$ from which the likelihood of the presence of a pedestrian at each pixel position can be derived. We summarize this projection procedure as follows: 1) Given a normalized foreground image $\bar{e}(m,n)$, we extract a block

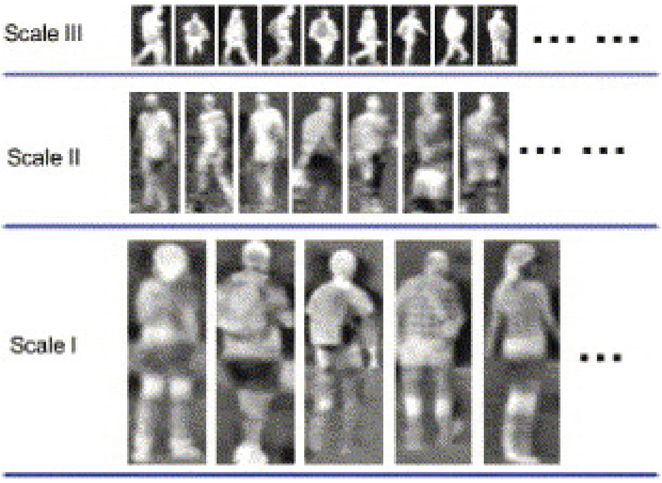


Figure 3.2: Template sets in different scales.

$\bar{e}_b(m,n)$ centering at each pixel location (m,n), and compute $s = \bar{e}_b - m_i$, where m_i is the mean vector of the ith eigenvector set. 2) Project s onto the ith eigenvector set $\vec{V}_i = \{v(1), v(2), ..., v(N_i)\}$ to produce the correspondent set of projection coefficients $\lambda_1, \lambda_2, ..., \lambda_{N_i}$. 3) Compute $\hat{s} = \sum_{n=1}^{N_i} \lambda_n v(n)$. 4) Compute $p(m,n) = MAD(s,\hat{s})$, where $MAD(s,\hat{s})$ is the mean absolute difference between s and $\hat{s}$.

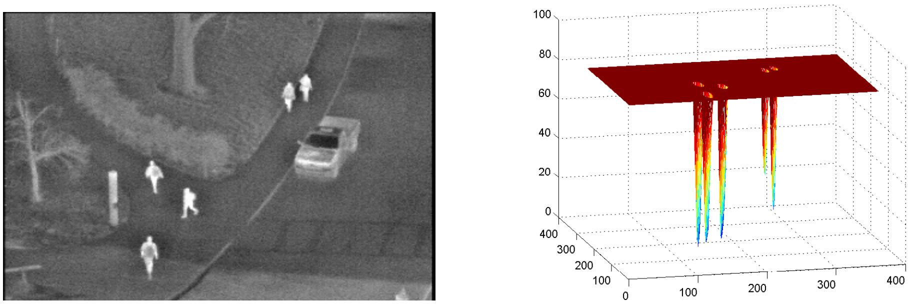


Figure 3.3: Derived cost map after the projection. (Left: the original IR image; Right: the correspondent cost map after the thresholding.)

The above projection procedure actually indicates that when the eigenvector set aligns with a pedestrian, it will produce a prominent local minimum at the center of the pedestrian in the cost map $p(m,n)$. To reduce the noise interference, we propose to exclude weak local minima from the original cost map $p(m,n)$ with a threshold T_i. the value of T_i

is designed to be associated with the thermal signatures of the backgrounds and usually higher for darker backgrounds due to bad weathers, and lower for brighter backgrounds. Fig. 3.3 shows an IR image containing five pedestrians and its correspondent cost map after the thresholding.

For IR images containing pedestrians of variable sizes, we propose to apply the three sets of principle eigenvectors $\vec{V_i} = \{v(1), v(2), ..., v(N_i)\},\ i = 1, 2, 3$ sequentially in a way such that pedestrians of the largest size are detected first, and the correspondent regions of $\bar{e}(m, n)$ are masked. Then pedestrians of smaller sizes are detected in the unmarked regions of $\bar{e}(m, n)$.

Location aggregation. Once the cost map $p(m, n)$ is computed. We first identify all prominent local minima in $p(m, n)$ whose values are below a pre-selected threshold T_i as the candidates. Then we locally aggregate the multiple candidates into one if they are too close to each other. Specifically, if the overlapped area of two candidates is more than 30% of the window area, we aggregate them into one; otherwise they are treated as two adjacent yet different pedestrians. Here T_i is a threshold for the ith eigenvector set, and $T_i \neq T_j$ for $i \neq j$. Note that by using location aggregation, we actually make a balanced trade off between false alarm and miss detection rates for the pedestrian detection task.

Here we highlight the three features of our PCA-based pedestrian detection technique. First, the layered representation effectively reduces the interference of non-pedestrian objects[1]. Second, to compensate thermal signature variations of pedestrians and especially to handle the "polarity switch" problem, we propose a heuristic method in which the extracted foreground F_k layer (pedestrian appearance information) is normalized relative to the local background $B_k M_k$, and the principle component analysis is applied to the normalized foreground layer. Third, on the contrary to our intuitions, we notice that employing a smaller set of principle components can serve the purpose of noise removal and contributes to the accuracy of pedestrian detection. Indeed a similar statement can be found in [84], where the authors claim that compared to face recognition, the face

[1] Layered representation indeed can not reduce the interference of non-pedestrian moving objects like vehicles. This may cause false detections in our algorithm as demonstrated in section 3.5.2. However this situation is extremely rare in both the OSU and the WVU databases, therefore its solution is out of the scope of this work. In section 3.6 we shall discuss the possible directions towards solving this problem.

detection task requires less principle components for a better performance.

3.4 Pedestrian Tracking

Tracking can be viewed as the dynamic extension of static pedestrian detection where motion-related temporal constraint is exploited to establish the correspondence of moving objects across multiple frames. However, such motion-related constraint can only be exploited for video frames whose sample rate is sufficiently high (otherwise they are no different from still images). For an ordered yet non-uniformly sampled collection of IR imagery, tracking is not always possible (e.g., when there is scene change). Therefore, we propose to do shot segmentation before tracking—a shot is defined as a collection of consecutive frames whose adjacent time interval is sufficiently small (e.g., a fraction of second). Tracking will be done within each shot instead for the whole sequence.

3.4.1 Shot segmentation

Based on the above definition, it is reasonable to assume that frames within the same shot look more alike than those outside. In visible imagery, histogram-based techniques are often suitable for shot segmentation [86]. However, histogram becomes less effective for IR imagery because pixels in the still background would dominate those in the moving foreground. Instead, we have developed a fast Hausdorff-distance [87] based shot segmentation algorithm.

Recall the collection of objects in foreground layer is labeled $R_1, R_2, ..., R_{E_k}$. For an object R_k, we collect the endings and intersections along its skeleton and form a feature point set C_k. To measure the distance between two feature point sets C_k and C_{k+1}, the Hausdorff distance has been widely used in the literature of compute vision [88]. We adopt the following definition of Hausdorff distance:

$$H(C_k, C_{k+1}) = \frac{h(C_k, C_{k+1}) + h(C_{k+1} + C_k)}{|C_k| + |C_{k+1}|} \tag{3.3}$$

where $h(X, Y) = \max_{\{\forall\ a \in X\}} \{ \min_{\{\forall\ b \in Y | a \in X\}} \{d(a, b)\}\}$, $d(a, b)$ denotes the Euclidean distance between two points $a \in X$ and $b \in Y$, and $|C_k|$ is the cardinality of set C_k. Note that the above definition has enforced the symmetry, i.e., $H(C_k, C_{k+1}) = H(C_{k+1}, C_k)$.

One practical constraint that has not been considered in the definition of Hausdorff distance is that images have finite size. Therefore, if some pedestrian happens to enter or leave the field of view, Hausdorff distance between two frames could be large even if they are temporally close. To overcome such difficulty, we opt to exclude the pedestrians around image boundary in the calculation of Hausdorff distance. Two frames are grouped together if and only if their Hausdorff distance is below a pre-selected threshold.

3.4.2 Graph theoretic tracking

Within the same shot, pedestrian tracking in IR imagery is often more difficult than that in visible imagery [22,89]. Unlike visible imagery containing color and texture cues, shape is arguably the only cue that can be exploited by tracking in IR imagery. When the camera distance is large, shape discrepancy between two different persons but with similar weight and height is small. The silhouette of a person is constantly varying due to the walking motion. Moreover, when two pedestrians walk closely or pass by each other, the overlapped shape of pedestrians experiences severe deformation, which makes tracking even more difficult.

To overcome the above difficulties, we propose the following strategies that exploit both appearance and location information adaptively at the same time. First, it is a reasonable to assume that a persons appearance does not change suddenly in two consecutive frames of the same shot. To measure the similarity between two pedestrians at different scales, we propose to normalize them to the same scale first (note that the scale information for each pedestrian is available from the detection stage). Then we project the pedestrians onto the eigenvectors $\vec{V_i} = \{v(1), v(2), ..., v(N_i)\}$ in scale i and define their similarity to be the L_2 distance calculated in the space spanned by the eigenvectors. Such distance measuring the similarity in terms of appearance between two pedestrians is denoted by d_{sa}.

Second, we propose to adaptively exploit the cue of photometric similarity and geometric proximity in the spatiotemporal domain. If pedestrians in the scene are far away from each other, geometric proximity is often sufficient for establishing the correspondence (e.g., nearest neighbor rule). In difficult scenarios where multiple pedestrians walk closely or pass by each other, both similarity and proximity will be useful to tracking.

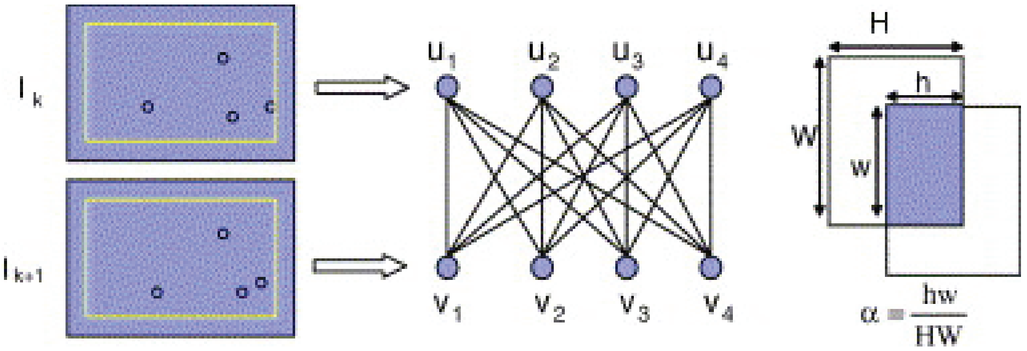


Figure 3.4: The graph theoretic tracking scheme.

As shown in Fig. 3.4, we have implemented the above ideas for two-frame tracking under a graph matching framework [90]. Let G be a weighted graph, in which $2Q$ nodes denote the detected pedestrians: $U = \{u_1, ..., u_Q\}$ from I_k and $V = \{v_1, ..., v_Q\}$ from I_{k+1} (Note that the equal number of pedestrians within each shot is guaranteed by the shot segmentation procedure). For any $u \in U$ and $v \in V$, there is an edge between them whose weight is:

$$w(u, v) = \alpha d_{sa} + (1 - \alpha) d_{eu} \tag{3.4}$$

where d_{sa} has been defined earlier, d_{eu} is the Euclidean distance between the centroid of u and v, and the weighting coefficient α is the overlapping ratio of u and v (i.e., the ratio of overlapped area to pedestrian window size). Note that when $\alpha > 0$, d_{sa} is often much larger than d_{eu} and easily dominates the weight assignment.

With the above-defined weighted graph G, two-frame pedestrian tracking can be formulated as a bipartite matching problem with set U and V. Denote $a_i(U, V), i = 1, 2, ..., Q!$ a one-to-one mapping between set U and set V, and $W(U, V|a_i)$ the total weights of the graph G given the matching operation a_i, the best mach between U and V is determined by: $a_{best} = \min_{\{a_i\}} W(U, V|a_i)$. For example, assuming $Q = 4$, and $a_1 = \{(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4)\}$, then the total weights of the current graph G

formed by the assignment a_1 is $W(U,V|a_1) = \sum_{i=1}^{4} w(u_i, v_i)$. Therefore, given Q pedestrians in each frame, theoretically there are $Q!$ possible ways of constructing the graph G with $Q!$ correspondent total weights $W(U,V|a_i)$, and we are indeed pursing the one matching operation a_i that leads to the lowest total weight.

When Q is large, the solution space of $a_i(U,V)$ shall be very huge, and the exhaustive search is not applicable. However, we found that for the OSU and WVU thermal pedestrian database, pedestrians seldom stay closely in large groups. Therefore, in our implementation, isolated pedestrians are matched first using the geometric cue, then pedestrians standing very close to each other (often 2-3 pedestrians) are matched via exhaustive search. Besides, we also notice there are fast algorithms to solve bipartite graph matching problems such as the graph-cut algorithm in [91].

3.5 Simulation Results and Discussions

In this section, we report our experimental results for OTCBVS benchmark-OSU thermal pedestrian database [35] (acquired by Raytheon 300D thermal sensor and available at http://www.cse.ohio-state.edu/otcbvs-bench/), and WVU IR video database (acquired by Raytheon PalmIR thermal sensor and available at http://www.csee.wvu.edu/ xinl/research/OTCBVS.html). The spatial resolutions of the IR images in the OSU and WVU databases are 360×240, and 320×200 respectively. There are 10 test sequences in OSU thermal database. Each sequence contains 18–73 frames that are taken within one minute but not temporally uniformly sampled (they are the subset of 30 Hz video coming out of IR camera). This database reasonably covers a variety of environmental conditions such as rainy, cloudy and sunny days. In OSU database, the camera is kept still all the time, and the cameraCpedestrian distance is far. Since video sequences in WVU database contain camera panning motion and are acquired at a closer camera distance, we choose two of them to demonstrate the performance of our dynamic background mosaicing and multi-scale pedestrian detection.

3.5.1 Dynamic background mosaicing

We first demonstrate the performance of the proposed dynamic background mosaicing algorithm. In the demonstrated video sequence, there are two pedestrians walking in the same direction as the camera panning motion. Fig. 3.5 shows the background mosaicing results for three disjointed frames in a sequence from WVU database. It can be observed that generalized EM algorithm effectively separate moving pedestrians from the background regardless of camera panning motion. Without any optimization, our MATLAB-based implementation takes 3–5 s to process 30 frames, which is faster than the reported speed in [71] To illustrate the problem with motionless pedestrians, we take sequence #8 in OSU thermal database as an example. If we only use its 24 frames in background extraction, the two pedestrians will be assigned to the background B, which seriously affects the detection performance as observed in [35]. Since we do not have any reference image taken at the same day, we opt to add another 24 frames of sequence #10 with similar thermal characteristics. It can be observed from Fig. 3.6 that incorporating more frames into background extraction alleviates the problem, though some ghost shadow of two pedestrians remains (it has been experimentally confirmed that the ghost shadow does not affect the detection).

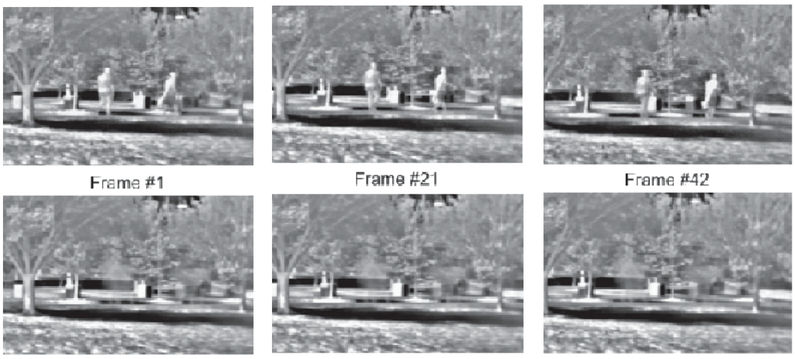


Figure 3.5: Background layer extraction result in the presence of camera panning. (Top: original IR images; Bottom: extracted background layers)

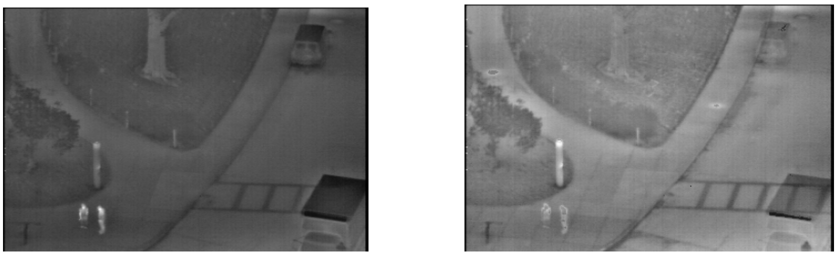


Figure 3.6: Extracted background using 24 frames in #8 only (left) and 48 frames in #8, #10 together (right).

3.5.2 Pedestrian detection

The size of training templates for PCA-based localization is empirically determined for the given database. Table 3.1 shows the detection result for all ten sequences in the OSU database. We have adopted the terminology in [35] to facilitate the comparison (note that Sensitivity=$\#TP/\#People$, PPV=1-$\#FP/\#People$). When compared with [35] our approach noticeably works better on sensitivity performance.

Database	Squence No.	# Frames	# People	#TP		#FP		Sensitivity		PPV	
				[35]	Ours	[35]	Ours	[35]	Ous	[35]	Ours
OSU database	1	31	91	88	91	0	0	0.97	1.00	1.00	1.00
	2	28	100	94	99	0	0	0.94	0.99	1.00	1.00
	3	23	101	101	100	1	2	1.00	0.99	0.99	0.98
	4	18	109	107	109	1	2	0.98	1.00	0.99	0.98
	5	23	101	90	101	0	0	0.89	1.00	1.00	1.00
	6	18	97	93	97	0	0	0.96	1.00	1.00	1.00
	7	22	94	92	94	0	0	0.98	1.00	1.00	1.00
	8	24	99	75	99	1	1	0.76	1.00	0.99	0.99
	9	73	95	95	95	0	0	1.00	1.00	1.00	1.00
	10	24	97	95	94	3	3	0.98	0.97	0.97	0.97
	1-10	284	984	930	979	6	8	0.95	0.99	0.99	0.99
WVU database	1	45	135		135		0		1		1
	2	60	215		215		0		1		1
	3	60	180		159		3		0.88		0.98

Table 3.1: Detection results for OSU thermal pedestrian database (TP - True Positive, FP - False Positive; PPV - positive predictive value (see texts for its definition).

In Fig. 3.7, we show our pedestrian detection results for some challenging cases in the OSU database, including the "polarity switch", groups of pedestrians, and rainy weather. Fig. 3.8 shows some examples of miss detection, false alarm, and arguably correct detection (not included in the ground-truth table due to occlusion). As we can see in the bottom right image of Fig. 3.8, our pedestrian detection algorithm has falsely

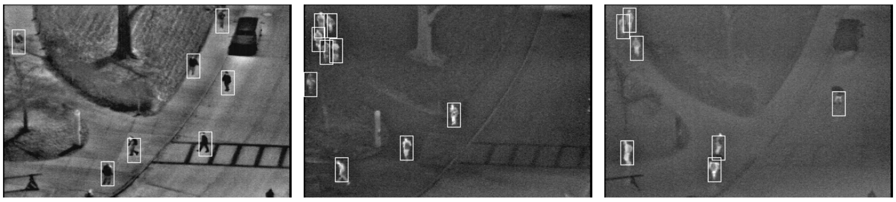


Figure 3.7: Pedestrian detection results for challenging cases in the OSU database. (Left: “polarity switch”; Middle: group of pedestrians; Right: rainy weather)

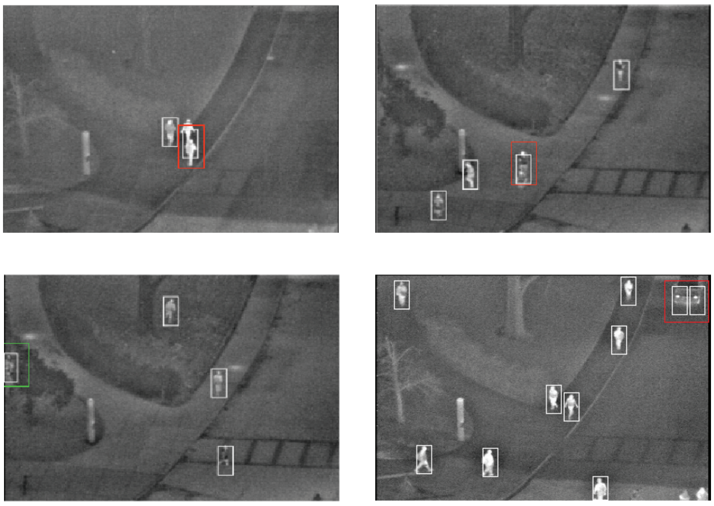


Figure 3.8: Examples of miss detection, false alarm (red box) and arguably correct results (green box).

detected the car entering the scene as two individual pedestrians. This type of false alarm is mainly due to the fact that the linear transformation of PCA based methods is not sufficient to capture the shape variation information of various types of objects. Indeed, in [92] we have proposed to exploit shape information to distinguish pedestrian and non-pedestrian objects. However, because of the extremely limited training examples in the OSU and WVU database, the shape-based classifier has not been integrated in our existing pedestrian detection algorithm.

To demonstrate the performance of our pedestrian detection algorithm across varying scales, we have conducted experiments with WVU infrared video database. As we can see from Table 3.1, we have obtained perfect detection without any errors for the first two video sequences; for the more challenging third sequence with polarity switch, the sensitivity and PPV values are 0.88 and 0.98 respectively. Fig. 3.9 shows the detection results for some frames containing large pedestrian size variations.

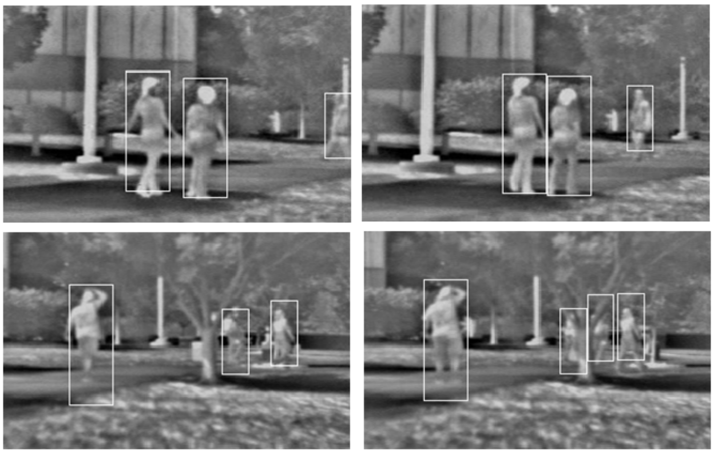


Figure 3.9: Multi-scale detection result with camera panning

3.5.3 Pedestrian tracking

In Fig. 3.10,we also show the shot segmentation result for sequence #9 in OSU database. The total 73 frames are segmented into four separate shots. The starting frames of four shots are shown in Fig. 3.11. In tracking experiments, the following heuristics is used in finding the optimal matching: if a pedestrian is isolated, the best matching can be found by direct local search (i.e. $\alpha = 0$); only for overlapped pedestrians ($\alpha > 0$), we need to exhaustively try out different assignments. Fig. 3.12 shows the tracking result for frames No.13-18 of sequence #5 in OSU database. Each pedestrian is marked by a different color. It can be observed that tracking is successful despite slight overlapping of pedestrians. Note that since our tracking is based on detection results it will not work if some pedestrian is missed at the detection stage.

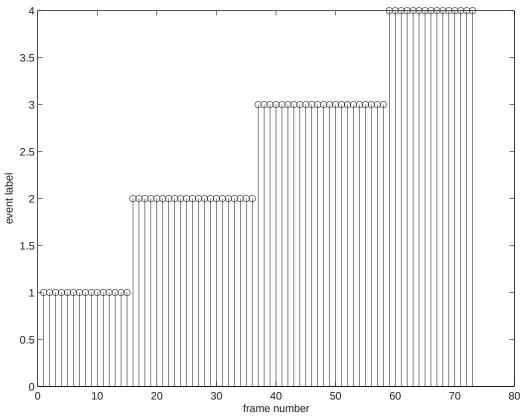


Figure 3.10: Shot segmentation result for sequence #9 of OSU database

3.5.4 Polarity switch

We have also done some preliminary test with polarity switch detection. Fig. 3.13 plots the calculated e_{avg} for all ten testing sequences. It clearly indicates the negative value for #3, which is the only one with polarity switch among ten in OSU database. However when applying the proposed heuristic strategy to more challenging situations (e.g., sequences with polarity switch in WVU database), we find that our strategy suffers from performance degradation in terms of increased miss detection rate. Polarity switch in WVU database is more difficult to handle because: 1) Pedestrians with and without polarity switch simultaneously appear in a sequence or even occasionally in one image; 2) pedestrians are partially occluded by the fence at the front. For example, Fig. 3.14 shows two frames from a sequence containing missed pedestrians. This sequence is composed of 60 consecutive frames sampled at 30 frames/second recording three walking pedestrians. After applying our detection algorithm with a constant set of parameters to this sequence, we get a detection rate of 88% (21 pedestrians are missed), and a false positive rate of 2% (3 falsely detected pedestrians). We also find that 15 out of the 21 missed detected pedestrians belong to the first 19 frames all of which contain the mixture of polarity switched and normal pedestrians (due to a glass-body building in the scene), and only 6 miss detections belong to the rest 41 frames in which all pedestrians are polarity switched. One plausible explanation lies in the heuristic strategy we proposed in Section

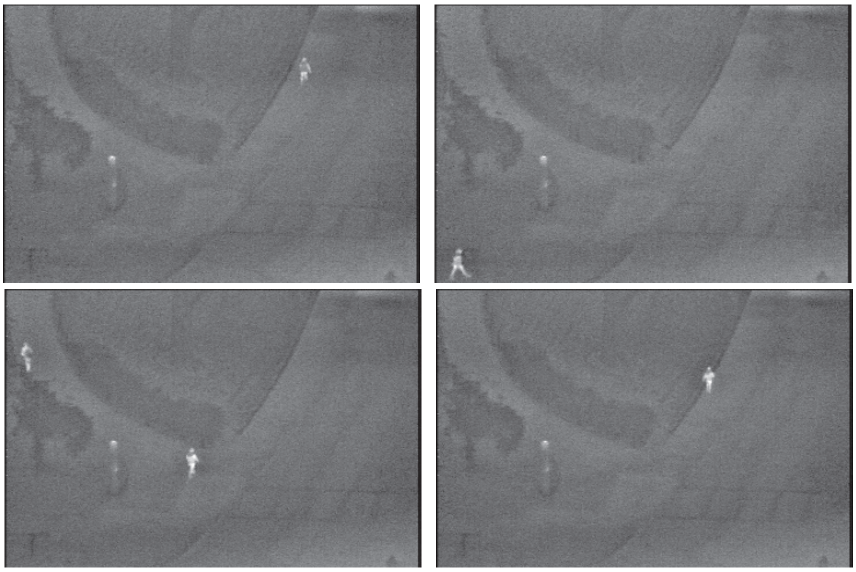


Figure 3.11: Frames No. 1, 16, 37, 59 in sequence #9 - they are the starting frames of new shots.

3.3 to handle the phenomenon of polarity switch. In our heuristic strategy, we propose to take the absolute value on the normalized foreground layer. Although such strategy does compensate for the polarity switched pedestrians to some degree, it switches the polarity of noise components in the foreground layer as well. Whenever the background mosaicing result contains errors, pedestrians in the foreground could be confused with the modified noise, which leads to the events of miss detection or false alarm regardless of parameter settings.

3.6 Concluding Remarks

This chapter presents a pedestrian detection and tracking algorithm for infrared imagery using the appearance cue via a layered representation.

Our layered representation separates infrared imagery into two layers: the background layer (still), and the foreground layer (moving). This layered representation structure significantly facilitates the pedestrian detection and tracking tasks by reducing the interference of non-pedestrian objects. To accommodate the panning of thermal sensors, we

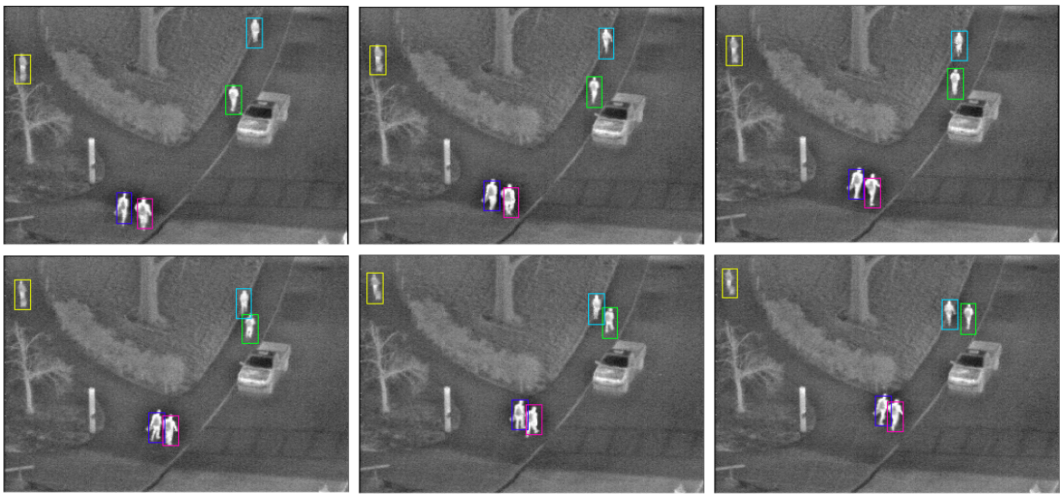


Figure 3.12: Tracking results for frames No. 13-18 in sequence #5. (Bounding boxes with the same color represent the same pedestrian tracked at successive instances.)

propose a generalized EM algorithm is proposed for dynamic background mosaicing, in which the background registration and foreground mask estimation procedures are conducted in an iterative fashion. In section 3.5.1, this background mosaicing algorithm has been verified to be sufficient to provide a satisfying background/foreground separation for pedestrian detection and tracking with a moderate computational complexity.

Based on the layered representation, we have presented a pedestrian detection algorithm in which we propose to exploit the appearance cue of pedestrians by a PCA based algorithm. To meet the challenges of infrared imagery (e.g. "polarity switch", thermal signature variations, and heavy noise), we have carefully modified the original "eigenface" algorithm. Compared to the benchmark shape-based pedestrian detection algorithm in [35] using the OSU thermal pedestrian database, our appearance-based approach have achieved a comparable overall performance and a noticeable better sensitivity. This result supports our argument that the appearance cue is also important for IR imagery. To serve the purpose of multi-scale pedestrian detection, we applied three sets of PCA templates with various sizes sequentially to IR images containing variable sized pedestrians. This heuristical scheme is verified by experimental results on the WVU infrared image database.

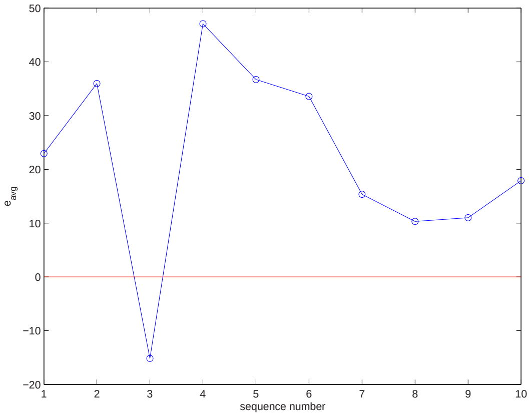


Figure 3.13: Polarity switch detection result.

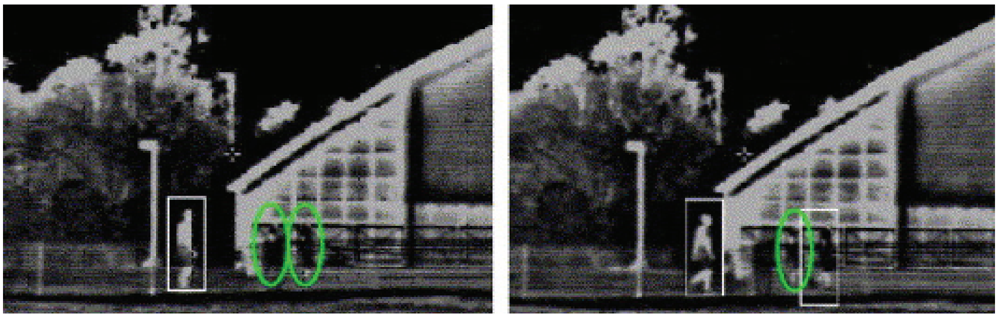


Figure 3.14: Miss Detection (green circle) due to polarity switch and fence blocking.

We also studied the problem of pedestrian tracking. Because the thermal image sequences in the OSU database are temporally non-uniformly sampled, we proposed to divide a thermal image sequence into serval shots by a Hausdorff-distance based shot segmentation algorithm. Tracking tasks are conducted within each shot. The pedestrian tracking task is formulated as a graph-matching problem between two successive IR frames by exploiting both the appearance similarity and geometry proximity of each individual pedestrians. To reduce the computational complexity for solving this graph-matching problem, we have proposed to match isolated pedestrians first to reduce the solution space.

Experiments conducted on OSU and WVU databases demonstrate that our algorithm performs well for challenging situations regardless of camera motion and distance. However, the proposed algorithm still has some limitations that can be improved in the future. We summarize them as follows: First, due to the limitation of PCA based approach, our pedestrian detection scheme can not handle the interference of moving objects well. Indeed, in [92], we have proposed a promising approach to screen non-pedestrian moving object using shape information via support vector machine (SVM). However due to the limited training examples in current OSU and WVU database, we have not been able to fully test the shape-based classifier. In the future, we shall work on integrating the shape-based classifier into the existing pedestrian detection scheme by collecting more data with moving objects. Second, "Polarity switch" is one of the most challenging cases in infrared imagery. It has negative impact on the accuracy of our algorithm. Currently, the cause of this phenomenon is still poorly understood. We believe that more experiment data and a better understanding of the physical mechanism shall provide effective schemes against this hostile phenomenon. Third, our current pedestrian detection scheme does not exploit any temporal information of IR image sequences. We believe motion information provided by the tracking algorithm can help further improve the detection performance (i.e., from static to dynamic). Within each shot, the motion cue can be exploited to resolve the ambiguity with overlapped pedestrians.

Chapter 4

Accurate Video Alignment Using Phase Correlation

4.1 Introduction

Despite recent advances in sensor technology, the spatial and temporal resolution of video cameras remain limited. Although higher resolutions can be achieved using camera array and high-speed cameras, the cost remains high. An alternative solution is to integrate/fuse visual information acquired by multiple standard cameras (e.g. multi-view video sequences). Such a computational approach has the advantage of cost efficiency and has many potential applications such as automatic video based surveillance [36,37], video metrology for athletic events [38], video-based modeling and rendering of 3D scenes [39], and tele-immersion [40].

Multi-view video sequences of the same dynamic scene are subject to spatiotemporal displacements. The spatial displacement is due to different camera positions. While the temporal displacement originates from the fact that the cameras may not be able to start recording at exactly the same temporal instance. Therefore, fusing such non-synchronized multi-view sequences requires the knowledge of their spatial and temporal relationships. Video alignment is a technique to fulfill this requirement by establish the correspondence in time and space among different sequences of the same dynamic scene [93].

In recent years, a flurry of algorithms for aligning video sequences have emerged.

In [41], a three-step approach using a set of corresponding feature points is proposed. In [45,93], parametric spatial and temporal alignment is obtained by iterative refinement. In [42], alignment is achieved by matching trajectories of moving objects. In [43], a nonlinear temporal warping function is introduced to compute temporal alignment. In [44], a linear video synchronization method is proposed to simultaneously align multiple sequences.

All the aforementioned techniques have achieved very good spatiotemporal alignment performances on multi-view sequences. For example, most of these techniques are able to reach integer frame accuracy for estimating the relative temporal displacement, and some of them (e.g. [44, 45]) do claim for subframe accuracy. However, to our best knowledge, there is no published work (before ours) that has ever conducted quantitative evaluations on their temporal alignment results. This situation might be due to the difficulty to measure true temporal displacements without using any specially designed synchronization instruments.

Accurate temporal alignments (especially at the subframe level) of multi-view sequences indeed are beneficial to certain video applications. For instance, in [94], a temporal super-resolution algorithm is developed to handle the motion aliasing problem by fusing several multi-view sequences based on the knowledge of their relative temporal displacements with subframe accuracy.

Motivated by the above observations, in this chapter, we present a highly accurate approach toward space-time video alignment using 3-D phase correlation. To evaluate the accuracy of the proposed algorithm, we have invented a simple but effective way to measure the unknown temporal displacement by using supplementary audio information that is recorded together with the video clips by the video cameras[1] while they are shooting a dynamic scene. Since audio signals can be sampled at a much higher rate (usually hundreds of times higher) than the video clips, the alignment of these audio signals provides an accurate enough estimate of temporal displacements between video clips. Experiments conducted on five pairs of multi-view sequences show that the temporal alignment results of the proposed algorithm lie within ±0.1 frames to the ground

[1] Note that in nowadays the audio recoding function has been a common feature of popular digital video camera brands.

truth provided by the audio signals.

4.1.1 Contributions

We summarize the contributions of our work into the following two aspects:

Video alignment via 3D phase correlation. We present a novel and highly accurate approach toward space-time video alignment via 3D phase correlation. Inspired by the subpixel image registration technique in [46], we propose to generalize the 2D phase correlation algorithm in [46] into 3D for the temporal alignment purpose. When combining it with existing spatial alignment techniques (e.g., [95] and [96]) under an iterative framework, we show that accurate temporal alignments can be achieved between multi-view sequences regardless of illumination difference and camera motion. we also explain how to achieve subframe accuracy using phase correlation, which outperforms cross-correlation based alignment (e.g., [44], and [45]).

Subframe ground truth via supplementary audio. As we have discussed, the difficulty to obtain the true temporal displacement prevents us as well as other researchers from evaluating the accuracy of proposed video alignment algorithms at the subframe level. Although an accurate synchronization can be achieved by using certain specially designed timing instruments, the associated design and instrumentation cost is not trivial. The second contribution of this work is providing a simple and effective method that does not require any extra instrument to achieve reliable and highly accurate estimates of true temporal displacements between multi-view sequences. In our method, we make a novel use of the audio recording function of video cameras, and record together with the video sequences a piece of audio signal while the cameras are shooting a dynamic scene. Since, audio signals are usually sampled at a much higher frame rate than video signals, the alignment of audio signals via 1-D phase correlation can provide the ground-truth with the accuracy of 0.0014 frame distance in our experiment settings.

The rest of this chapter is organized as follows: In section 4.2.1, we present our assumptions and the mathematical formulation for the multi-view video alignment problem. In section 4.2.2, we present how to estimate the temporal displacement via 3-D phase correlation. In section 4.2.3, we briefly describe the procedure for spatial alignment via existing image gradient based approach in [95,96]. In section 4.2.4, An iterative spatiotemporal sequence alignment scheme is presented to enhance the spatiotemporal alignment performance. In section 4.2.5, we describe the procedure to obtain subframe ground-truth for temporal displacements via supplementary audio information. Simulation results and associated discussions are presented in section 4.3. Concluding remarks and future work are presented in section 4.4.

4.2 Video Alignment Using 3-D Phase Correlation

4.2.1 Problem Formulation

Without loss of generality, we focus on the alignment of two video sequences here. The two cameras satisfy the following assumptions:

1) Spatially, two cameras are kept close to each other (not necessarily still), and the distance between camera centers is negligible compared to the camera-to-scene distance (planar scene);

2) Temporally, two cameras have the same sampling rate but are not synchronized;

We note that the first assumption is to assure that two video sequences have significantly overlapped field of view (FOV) which contains the moving target of interest. No assumption is made about the calibration of video cameras - i.e., the inner and outer optical parameters of two cameras might slightly differ.

Let f_1 and f_2 be two input multi-view sequences. Denote (x, y, t) and (x', y', t') the spatiotemporal correspondent points (voxels [45]) in f_1 and f_2 respectively. The mathematical model link these two sequences can be written as:

$$f_1(x, y, t) = f_2(x', y', t') + v(x, y, t) \tag{4.1}$$

Where $v(x, y, t)$ denotes the noise/illumination variations.

According to our assumptions, the temporal and spatial constraints between two correspondent voxels (x, y, t) and (x', y', t') can be described as follows:

Temporal translation constraint:

$$t' = t + \Delta t \tag{4.2}$$

Spatial projection constraint:

$$H \times \vec{P'} = \vec{P} \tag{4.3}$$

where H is an instantaneous 3×3 homography matrix and $\vec{P} = [x, y, 1]^T$ is the homogeneous coordinate of the spatial component of (x, y). Note that the above spatial projection constraint holds for our planar scence assumption [45].

Given f_1 and f_2, we need to resolve the uncertainty of both H (spatial) and Δt (temporal). Note that the spatial and temporal alignments are intertwisted in such a way that knowing one will directly facilitate the estimation of the other. Therefore in the subsequent sections we shall describe the algorithms for estimating Δt and H respectively. Then we present an iterative procedure in which the estimates of Δt and H are jointly refined.

4.2.2 The 3-D Phase Correlation Based Temporal Alignment

In this section we discuss how to achieve the temporal alignment via 3-D phase correlation. Similar to the 2D case, we define the phase-correlation between f_1 and f_2 as

$$C(x, y, t) = \mathcal{F}^{-1}[\frac{F_1^* F_2}{|F_1^* F_2|}] \tag{4.4}$$

where F_1, F_2 are the 3D Fourier transform of f_1, f_2. It is easy to observe that when $f_2 = f_1(x + \Delta x, y + \Delta y, t + \Delta t)$, the phase correlation $C(x, y, t)$ takes the form of Dirac function $\delta(x - \Delta x, y - \Delta y, t - \Delta t)$. Similar to sub-pixel image registration [46], we can estimate the subframe displacement in the phase-correlation domain.

Let $f_1(x, y, t)$ and $f_2(x', y', t')$ be generated by down-sampling two higher-resolution signals linked by integer translations of x_0,y_0, and t_0. Then the fractional displacements can be expressed as: $\Delta x = \frac{x_0}{M}$,$\Delta y = \frac{y_0}{N}$, and $\Delta t = \frac{t_0}{K}$ where M,N, and K denote the down-sampling factors along three dimensions. Using similar derivation to [46], we can show:

$$C(x,y,t) \approx \frac{sin(\pi(Mx - x_0))}{\pi(Mx - x_0)} \frac{sin(\pi(Ny - y_0))}{\pi(Ny - y_0)} \frac{sin(\pi(Kt - t_0))}{\pi(Kt - t_0)} + w(x,y,t) \qquad (4.5)$$

where $w(x,y,t)$ is a zero-mean Gaussian random variable modeling the interference noise (e.g., due to non-overlapping regions).

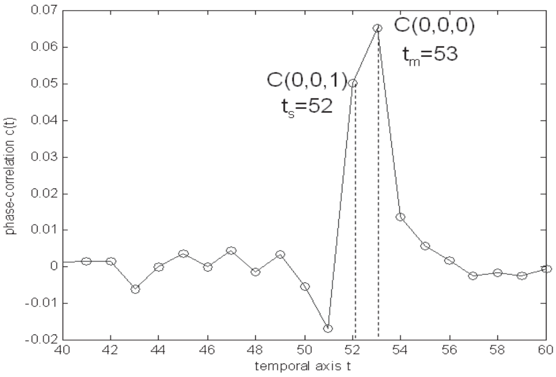


Figure 4.1: Subframe estimation from main and side peaks of phase-correlation function.

Eq. (4.5) basically shows that the energy spreading of $C(x,y,t)$ from the main peak observes the sinc function. Using the same technique as [46], we can fit the model of Eq. (4.5) around the locations where signal energy is mostly concentrated at main-peak t_m and side-peak t_s (shown in Fig. 4.1). For instance, we can have the main peak located at $(0,0,0)$ and its closest side-peak along the t-dimension at $(0,0,1)$ by change of variables. Then we obtain

$$C(0,0,0) = \frac{sin(\pi x_0)}{\pi x_0} \frac{sin(\pi y_0)}{\pi y_0} \frac{sin(\pi t_0)}{\pi t_0} \qquad (4.6)$$

$$C(0,0,1) = \frac{sin(\pi x_0)}{\pi x_0} \frac{sin(\pi y_0)}{\pi y_0} \frac{sin(\pi(K - t_0))}{\pi(K - t_0)} \qquad (4.7)$$

It follows from (4.6) and (4.7) that

$$\Delta t = \frac{t_0}{K} = \frac{C(0,0,1)}{C(0,0,1) \pm C(0,0,0)}, \tag{4.8}$$

where the ± ambiguity can be resolved by imposing the constraints that Δt is in the range of [-0.5,0.5] and has the same sign as $t_s - t_m$.

4.2.3 The Image Based Spatial Alignment

Although temporal alignment can be efficiently handled by the phase-correlation based method, it is not appropriate for spatial alignment. Two-parameter translational models are too limited to characterize the multi-view geometric relationship and consequently, spatial alignment given by phase-correlation is often suboptimal. Therefore, we resort to more sophisticated models such as 2D planar homography matrix H (refer to Eq.(4.3)) with eight free parameters (scale invariant):

$$H = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{bmatrix} \tag{4.9}$$

In this work, we have adopted hierarchical techniques presented in [95] and [96] to estimate the homography matrix H for each pair of frames (under further assumption of both cameras being fixed, we can estimate only one H for the whole sequence).

For the clarity and completeness purposes, in this section, we briefly describe the technique in [95] for estimating the homography matrix H. Assuming the temporal displacement Δt between two sequences f_1 and f_2 is estimated via the 3-D phase correlation algorithm in previous section, the homography matrix H linking the two correspondent frame pairs $f_1(x,y,t)$ and $f_2(x,y,t+[\Delta t])$ ($[\Delta t]$ is the nearest integer of Δt) at any instance t is estimated by minimizing the sum of squared intensity errors:

$$E = \sum_{(x,y)} e^2 = \sum_{(x,y)} (f_2(x',y',t+[\Delta t]) - f_1(x,y,t))^2 \tag{4.10}$$

where the expressions of x' and y' can be derived from Eq.(4.3) and Eq.(4.9):

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1}, \qquad y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}. \tag{4.11}$$

To minimize Eq.(4.10) with respect to the unknown parameters $\vec{m} = [m_0, m_1, ..., m_7]$, the hierarchical Levenberg-Marquardt iterative nonlinear minimization algorithm [97] is applied for its numerical stability and fast convergence properties. Note that since the implementation detail of the hierarchical Levenberg-Marquardt algorithm has been covered in [95] and [96], we do not repeat the whole procedure here.

Once the homography maxtrix H at each instance t has been estimated, the spatial alignment of f_1 and f_2 is achieved by warping the sequence f_2 towards f_1 using Eq. (4.3). Since some pixels (x', y') may fall between the sampling grids of f_1, a bilinear interpolation technique is adopted to compute the pixel values at the correspondent grade points.

4.2.4 The Joint Spatiotemporal Sequence Alignment

As we have mentioned, the spatial and temporal alignments are intertwisted procedures: On the one hand, the spatial alignment rely on the knowledge of temporal displacement; On the other hand, the temporal alignment is inaccurate without compensating the spatial displacements which are usually non-translational (refer to Fig. 4.5). Therefore, in this section, we present an joint spatiotemporal sequence alignment scheme in which the temporal and spatial alignments are iteratively refined.

Putting the temporal and spatial alignment procedures together, we summarize our spatiotemporal alignment algorithm in Fig. 4.2. As shown in Fig. 4.2, the iterative procedure starts with estimation of the temporal displancement. After the initial temporal alignment, the set of homography matrices $\{H_k\}$ are computed from the spatial alignment procedure. Then the temporal alignment procedure is resumed for one original sequence (f_1) and the spatially compensated version of the other sequence (f_2') which is achieved via the homography transform defined in Eq.(4.3). Finally, the iterative procedure stops when the temporal alignment can not be further refined. Note that during the iterations only the integer part of temporal alignment result $[\Delta t_i]$ is fed back into the loop to improve the homography estimation (no temporal interpolation is involved). For

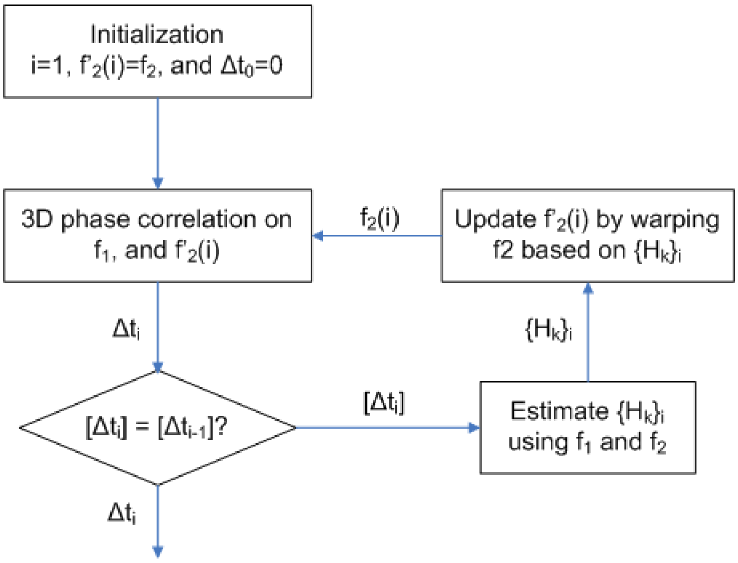

Initialization
i=1, f'2(i)=f2, and Δt0=0
3D phase correlation on f1, and f'2(i)
f2(i)
Update f'2(i) by warping f2 based on {Hk}i
Δti
{Hk}i
[Δti]
[Δti] = [Δti-1]?
Estimate {Hk}i using f1 and f2
Δti

Figure 4.2: Flow chart of the joint spatiotemporal alignment algorithm

most sequences, we have found that convergence is reached after only one spatial and two temporal alignment steps.

4.2.5 Subframe Ground-Truth via Supplementary Audio

In view of difficulty with evaluating the accuracy of subframe video alignment, we come up with a novel approach based on the observation that supplementary audio is sampled at a much higher rate than video. As shown in 4.3, we can apply 1D phase correlation based alignment technique to audio signals and obtain the temporal distance (measured by the number of audio samples). It is easy to see that integer-sample accuracy of audio sampled at 22,050Hz corresponds to 0.0014-frame accuracy of video sampled at 30Hz. Therefore, audio-based alignment conveniently provides the subframe ground truth to validate the accuracy of video alignment techniques.

Indeed the idea of using audio information to obtain accurate estimates of temporal displacements of video signals is both effective and cost efficient. the cost efficiency comes from the fact that it does not require any extra instrumentation and is very easy to conduct. It exploits the audio recording function owned by most commercial camcorders.

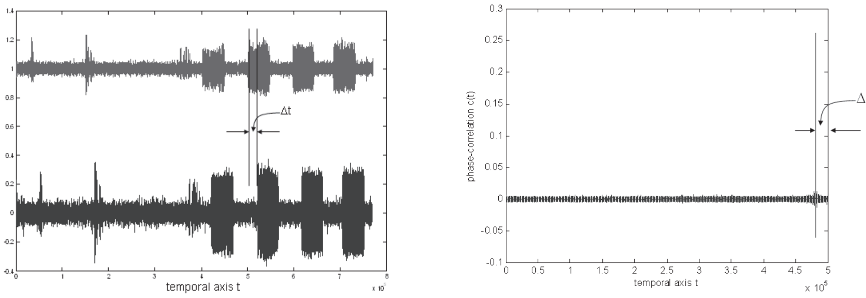


Figure 4.3: Illustration of audio alignment to obtain subframe ground truth.

What one needs to provide is a piece of audio signal that will be recorded by the cameras while they are shooting a specific dynamic scene. The effectiveness of this technique is generally guaranteed by the fact that the sampling rates of audio signals (22,050Hz) are much higher than video signals (30Hz). One may concern that the latencies between the video and audio signals inside the two cameras could differ due to different inner parameters of the cameras. This difference however should be negligible in our work, since we have used two cameras of the identical model (Canon S2-IS) throughout all the experiments.

4.3 Simulation Results and Discussions

In this section, we use experimental results to demonstrate the performance of the proposed video alignment techniques. Two sets of test sequences are captured under ideal (*walk*, *ball*, $flag$) and nonideal ($flag1$, $flag2$) conditions respectively. Specifically, two cameras have varying illumination conditions in $flag1$ and in $flag2$, one camera is kept still and the other moves.

First, it is well known from [46] that phase-correlation is more accurate than cross-correlation for subpixel image registration. Similarly, we want to show that phase-correlation is preferred to cross-correlation adopted in [45] for subframe video alignment. Fig. 4.4 compares the phase-correlation function and cross-correlation function for $flag1$. It can be seen that the peak of phase-correlation function is sharper than that of cross-correlation function, which has been observed in [46] for 2-D images.

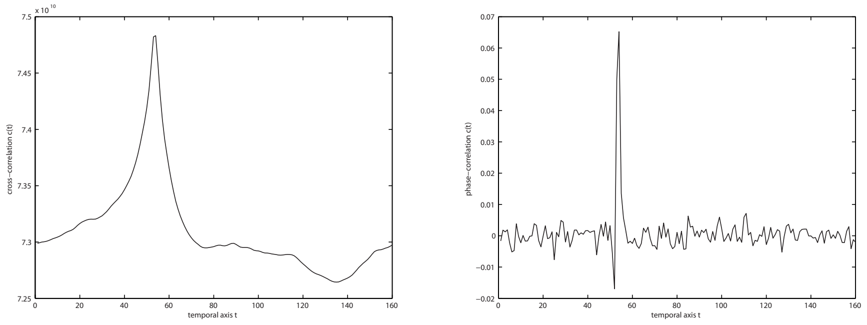


Figure 4.4: Comparison between cross-correlation (left) and phase-correlation (right) for *flag*1.

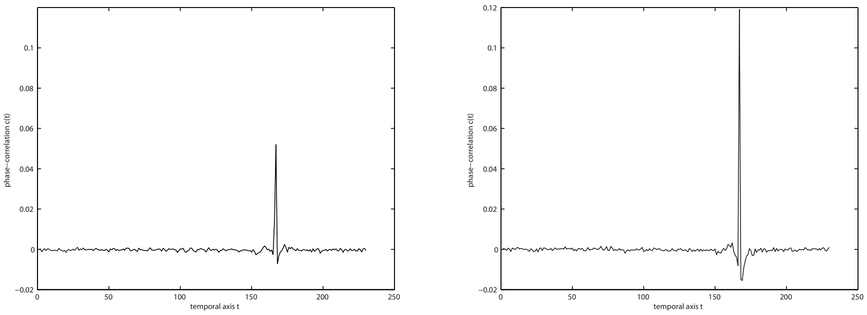
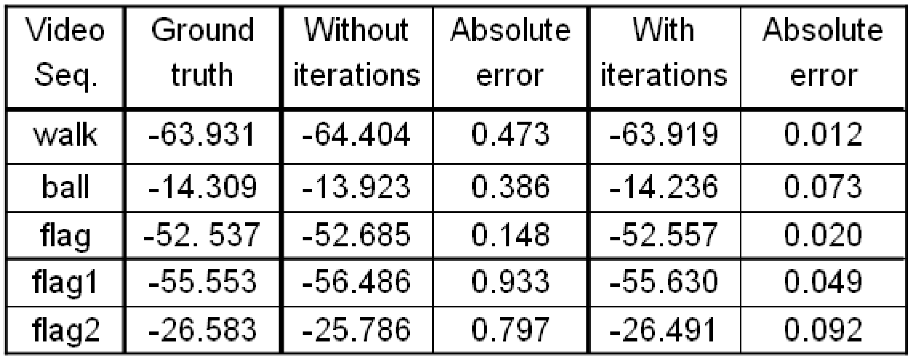


Figure 4.5: Phase-correlation function along the temporal axis before (left) and after (right) spatial alignment for *walk*.

Second, we want to demonstrate how spatial alignment sharpens the peak of phase-correlation function and therefore improves the accuracy of subframe temporal alignment. Fig. 4.5 compares the phase-correlation functions before and after spatial alignment for the *walk* sequence. To facilitate the visual inspection, they have been displayed at the same range - the improvement of peak magnitude is obvious. Such increased energy concentration at the main and side peaks is a direct evidence of improved alignment accuracy [46].

Finally, we report the alignment results for the two data sets in Table 4.1. For both data sets, we observe that our method can achieve temporal accuracy within the range of ±0.1 frame compared to the ground truth. It should be noted that 3D phase-correlation achieves highly accurate alignment regardless of motion rigidity. Such result suggests



Video Seq.	Ground truth	Without iterations	Absolute error	With iterations	Absolute error
walk	-63.931	-64.404	0.473	-63.919	0.012
ball	-14.309	-13.923	0.386	-14.236	0.073
flag	-52. 537	-52.685	0.148	-52.557	0.020
flag1	-55.553	-56.486	0.933	-55.630	0.049
flag2	-26.583	-25.786	0.797	-26.491	0.092

Table 4.1: Video synchronization results

that accurate alignment of multiview video does not necessarily require the estimation of motion trajectory (i.e., independent of the complexity of motion in each sequence). For the non-ideal data set, we have found that our technique achieves higher accuracy for $flag1$ than $flag2$ because phase-correlation is robust to photometric distortions than geometric distortions. Fig. 4.6 shows two spatially aligned video frames selected from the $flag$ and the $walk$ sequences using the estimated homography matrices. It is clear to observe that the spatial correspondence between the misaligned video sequences is not purely translational.

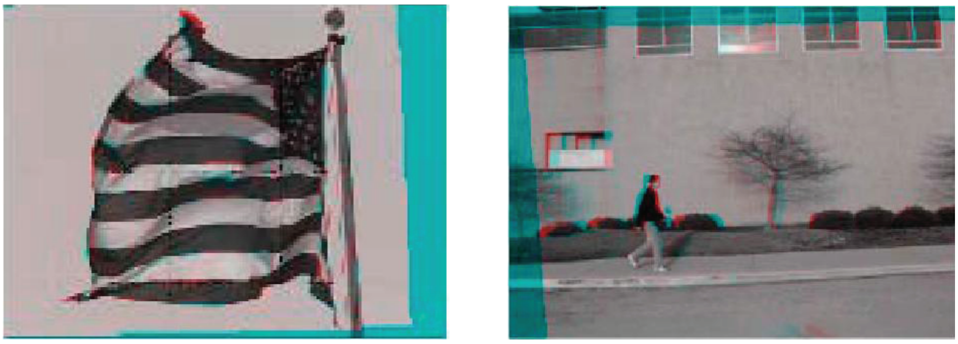


Figure 4.6: Spatial alignment results (Left: $flag$ sequence, Right: $walk$ sequence).

4.4 Concluding Remarks

In this chapter, we have studied the problem of multi-view sequence alignment. To achieve an accurate temporal alignment, we have generalized the 2-D phase correlation algorithm in [46] for 3-D video data. Since the spatial alignment between multi-view video sequences is constrained by the homographic transformation (assuming planar scene), we adopted the procedures in [95,96] to estimate the homography matrices. We verified that after combining the spatial and temporal alignment procedures in an iterative fashion, very accurate estimate of temporal alignment can be achieved. Experimental results show that our scheme is robust to both illumination variations and camera motions.

Currently, the proposed algorithm is based on the assumption of 2-D planar scenes under which the spatial relationship of multi-view sequences is reduced to the homographic transformation. This assumption is valid for large camera-to-scene distances. When the distance becomes close, our current algorithm does not guarantee to produce satisfying alignments because of the effect of scene depth variations. Indeed in this case the spatial constraint of multi-view sequences becomes the more general 3-D perspective transformation. In the future, we shall investigate on developing algorithms to deal with 3-D scenes. Besides, our current algorithm assumes an identical temporal sampling rate for the two cameras. In the future, we shall explore on a more general case in which the temporal alignment of multi-view sequences could have a linear relationship.

Chapter 5

Conclusions and Future Work

In this thesis we choose to study three topics in video coding and computer vision. These topics cover a wide range of video applications ranging from the conventional monocular and binocular applications within the visible spectrum to the applications beyond the visible spectrum. For each selected problem, we have investigated the fundamentals behind and proposed our own solutions.

Intra coding in block-based hybrid video coding systems is essentially a still image coding problem. In chapter 2, we identify that the quad-tree based image representation structure adopted by many image coding schemes (e.g., wavelet based image coders and H.264/AVC intra coding) fail to take into account the geometric constraint of edges in natural images and therefore lead to a suboptimal R-D performance. Based on this observation, we have introduced the concept of geometry-adaptive intra prediction (GAIP) to exploit the redundancy along edge orientations. In the GAIP algorithm, an image block is separated into two regions by an arbitrarily oriented line segment, and a set of prediction schemes have been designed to model each partitioned region. We have also studied the problem of modeling texture patterns of natural images by exploiting the non-local information and developed a geometry-adaptive intra displacement prediction (GAIDP) algorithm by applying the proposed geometry-adaptive block partitioning structure to the existing intra displacement prediction (IDP) algorithm. Simulations have shown that at low bit rate regime, the intra coding performance of H.264/AVC is significantly improved after being incorporated with the proposed GAIP and GAIDP algorithms.

Despite the demonstrated power of modeling natural images, the proposed algorithms can still be improved. First, the current block partitioning edge is modeled by a line segment. This linear model is too simple to efficiently represent arbitrarily shaped image edges. Therefore in the future, partitioning curve models of higher orders should be explored. Second, in our current implementation, the optimal partitioning edge is determined via an exhaustive searching procedure which is highly intensive in terms of computation. For real time applications, fast edge searching algorithms based on the statistics of decoded information shall be investigated. Finally, due to the complex nature of texture patterns, we believe that an adaptive fusion of multiple texture descriptor is a promising direction of modeling texture patterns.

Object detection and tracking has been extensively studied in visible spectrum. For infrared imagery, we are facing new challenges because of a different sensing modality. In chapter 3, we have proposed a pedestrian detection and tracking algorithm via layered representation. To separate the foreground layer and the background layer in the presence of the camera panning motion, a generalized expectation maximization (GEM) procedure is developed to iteratively conduct the global background registration and foreground mask estimation. We argue that for the pedestrian detection purpose, the appearance cue of pedestrians in infrared imagery is equally important as the shape cue which has been exploited in the literature. We have developed a multi-scale principle component analysis (PCA) technique to detect pedestrians with various sizes. Simulations on both the OSU and WVU thermal image databases justify the importance of the appearance cue. To facilitate tracking task, we propose a shot segmentation technique based on the Hausdorff-distance measure. Within each shot, tracking is formulated as a graph matching problem by exploiting both the appearance similarity and the geometric approximate of individual pedestrians.

Due to the limitation of PCA based approach, the current pedestrian detection technique is not sufficient to distinguish between pedestrians and non-pedestrian moving objects on the foreground layer. To address this problem, in the future we can collect more examples of various non-pedestrian objects, and explore the possibility of combining a shape-based classifier with our appearance based approach. Besides, our current tracking scheme defined on two successive frames does not fully exploit the motion information. Indeed, motion is a very important cue for object tracking tasks. We believe that the robustness and accuracy of current tracking scheme can be further improved by exploiting the motion trajectories of pedestrians. “Polarity Switch” is a hostile phenomenon for detection and tracking tasks in thermal imagery. To fight against it, heuristic methods have been proposed (including ours) in literature. We believe the solution to this problem lies behind the understanding of the imaging mechanisms of thermal sensors.

Video alignment is a procedure to facilitate the fusion of multi-view sequences. For video alignment, we identify that the temporal and spatial alignments are intertwined. In chapter 4, we have proposed to conduct temporal and spatial alignments in an iterative framework. To obtain accurate temporal alignments, we have generalized the 2-D phase correlation algorithm to 3-D. We also provide a novel way of obtaining the ground truth of temporal displacements by using auxiliary audio signals with much higher sampling rates. Currently, our algorithm is based on the assumptions of planar scenes and identical temporal sampling rates. In the future, we shall explore the situations of more complex 3-D scenes and non-identical frame rates.

References

[1] S. Liu and M. Hayes, “Segmentation-based coding of motion difference and motion field images for low bit-rate video compression,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992, pp. 525–528.

[2] P. Salembier, “Segmentation based video coding system allowing the manipulation of objects,” *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 60–74, 1997.

[3] K. Aizawa, C. S. Choi, H. Harashima, and T. S. Huang, “Human facial motion and synthesis with application to model-based coding,” in *Proceedings of Motion Analysis and Image Sequence Processsing*, 1993, pp. 317–348.

[4] D. E. Pearson, “Developments in model based video coding,” in *Proceedings of IEEE*, 1995, pp. 892–906.

[5] P. Eisert and B. Girod, “Model-based coding of facial image sequences at varying illumination conditions,” in *Proceedings of 10th IMDSP Workshop 98*, 1998, pp. 119–122.

[6] G. Karlsson and M. Vetterli, “Three dimentional sub-band coding of video,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 1100–1103.

[7] D. Taubman and A. Zakhor, “Multirate 3d subband coding of video,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 572–588, 1994.

[8] G. Cote, B. Erol, M. Gallant, and F. Kossentini, “H.263+: Video coding at low bit rates,” *IEEE Transactions on Circuits and Systemes for Video Technology*, vol. 8, no. 7, pp. 849–866, 1998.

[9] T. Halbach, “Performance comparison: H.26l intra coding vs. jpeg2000,” in *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 4th meeting*, 2002.

[10] M. N. Do, P. L. Dragotti, R. Shukla, and M. Vetterli, “On the compression of two-dimensional piecewise smooth functions,” in *Proceedings of IEEE International Conference on Image Processing*, 2001, pp. 14–17.

[11] V. Chandrasekaran, M. B. Wakin, D. Baron, and R. G. Baraniuk, “Surflets: A sparse representation for multidimensional functions containing smooth discontinuities,” in *Proceedings of IEEE International Symposium on Information Theory*, 2004, pp. 563–563.

[12] S. Mallat, *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

[13] S. L. Yu and C. Chrysafis, “New intra prediction using intra-macroblock motion compensation,” in *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 3rd meeting*, 2002.

[14] T. K. Tan, C. S. Boon, and Y. Suzuki, “Intra prediction by template matching,” in *Proceedings of IEEE International Conference on Image Processing*, 2006, pp. 1693–1696.

[15] S. Kondo, H. Sasai, and S. Kadono, “Tree structured hybrid intra prediction,” in *Proceedings of IEEE International Conference on Image Processing*, 2004, pp. 473–476.

[16] J. Balle and M. Wien, “Extended texture prediction for h.264 intra coding,” in *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 31st meeting*, 2007.

[17] B. Leibe and et al., “Pedestrian detection in crowded scenes,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 878–885.

[18] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings of IEEE International Conference on Computer Vision*, 2003, pp. 734–741.

[19] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proceedings of IEEE International Conference on Computer Vision*, 1998, pp. 555–562.

[20] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1997, pp. 193–199.

[21] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, "Shape-based pedestrian detection," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2000, pp. 215–220.

[22] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4:real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 22, no. 8, pp. 809–831, 2000.

[23] D. Gavrila and J. Geibel, "Shape-based pedestrian detection and tracking," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2002, pp. 8–14.

[24] I. Kakadiaris and D. Metaxas, "Model-based estimation of 3d human motion," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 22, no. 12, pp. 1453–1459, 2000.

[25] J. Deutscher, A. Blake, I. Reid, and O. Oxford, "Articulated body motion capture by annealed particle filtering," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 126–133.

[26] Y. Song, X. Feng, and P. Perona, "Toward detection of human motion," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 810–817.

[27] A. Mohan and T. Poggio, “Example-based object detection in images by components,” *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 23, no. 4, pp. 349–361, 2001.

[28] R. Cutler and L. Davis, “Robust real-time periodic motion detection analysis and applications,” *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 22, no. 8, pp. 781–796, 2000.

[29] U. Franke, D. Gavrila, S. Gorzig, F. Linder, F. Paetzold, and C. Wohler, “Autonomous driving goes downtown,” *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 6, pp. 40–48, 1998.

[30] L. Zhao and C. E. Thorpe, “Stereo- and neural network-based pedestrian detection,” *IEEE Transaction on Intelligent Transportation Systems*, vol. 1, no. 3, pp. 148–154, 2000.

[31] H. Nanda and L. Davis, “Probabilistic template based pedestrian detection in infrared videos,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2002, pp. 15–20.

[32] F. Xu, X. Liu, and K. Fujimura, “Pedestrian detection and tracking with night vision,” *IEEE Transaction on Intelligent Transportation Systems*, vol. 6, no. 1, pp. 63–71, 2005.

[33] M. Yasuno, N. Yasuda, and M. Aoki, “Pedestrian detection and tracking in far infrared images,” in *Proceedings of IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, 2004, pp. 125–131.

[34] Y. Owechko, S. Medasani, and N. Srinivasa, “Classifier swarms for human detection in infrared imagery,” in *Proceedings of IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, 2004, pp. 121–127.

[35] J. Davis and M. Keck, “A two-stage approach to person detection in thermal imagery,” in *Proceedings of Workshop on Applications of Computer Vision, IEEE OTCBVS WS Series Bench*, 2005.

[36] L. Zelnik-Manor and M. Irani, "Event based analysis of video," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 123–130.

[37] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 22–29.

[38] I. Reid and A. Zisserman, "Goal directed video metrology," in *Proceedings of the 4th European Conference on Computer Vision*, 1996, pp. 647–658.

[39] T. Naemura, J. Tago, and H. Harashima, "Real-time video based modeling and rendering of 3d scenes," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 66–73, 2002.

[40] S. Vedula, S. Baker, and T. Kanade, "Spatio-temporal view interpolation," in *Proceedings of the 13th Eurographics workshop on Rendering*, 2002, pp. 65–76.

[41] G. P. Stein, "Tracking from multiple view points: self-calibration of space and time," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 521–527.

[42] Y. Caspi, D. Simakov, and M. Irani, "Feature-based sequence-to-sequence matching," in *Proceedings of IEEE Workshop Vision Modeling Dynamical Scenes*, 2002.

[43] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood, "View-invariant alignment and matching of video sequences," in *Proceedings of IEEE International Conference on Computer Vision*, 2003, pp. 939–945.

[44] R. L. Carceroni, F. L. C. Padua, G. A. M. R. Santos, and K. N. Kutulakos, "Linear sequence-to-sequence alignment," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 746–753.

[45] Y. Caspi and M. Irani, "A step towards sequence-to-sequence alignment," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. 682–689.

[46] H. Foroosh, J. B. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 188–200, 2002.

[47] Y. Li and K. Sayood, "Lossless video sequence compression using adaptive prediction," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 997–1007, 2005.

[48] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[49] A. Sadka, *Compressed Video Cmmunications.* John Wiley and Sons, 2002.

[50] "Iso/iec cd 11172: Coding of moving pictures and associated audio for digital storage media at 1.5 mbits/s," 1991.

[51] "Iso/iec jtc1/sc29/wg11 cd 11172: Generic coding of moving pictures and associated audio," 1993.

[52] "Itu-t h.261: Video codec for audiovisual services at px64 kbits/s," 1993.

[53] "Itu-t h.263: Video coding for low bit rate communications," 1998.

[54] ITU-T, *ITU-T Recommendation H.264*, 2005.

[55] I. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia.* John Wiley and Sons, 2003.

[56] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h.264/avc video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[57] A. Cohen, I. Daubechies, O. G. Guleryuz, and M. T. Orchard, "On the importance of combining wavelet-based nonlinear approximation with coding strategies," *IEEE Transactions on Information Theory*, vol. 48, no. 7, pp. 1895–1921, 2002.

[58] O. Divorra, P. Yin, C. Dai, and X. Li, "Geometry-adaptive block partitioning for video coding," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007, pp. 657–660.

[59] E. M. Hung, R. L. D. Queiroz, and D. Mukherjee, “On macroblock partition for motion compensation,” in *Proceedings of IEEE International Conference on Image Processing*, 2006, pp. 1697–1700.

[60] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, 1st ed. Prentice-Verlag, 2000.

[61] “Jsvm 6 software,” in *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q. 6*, 2006.

[62] G. Bjontegaard, “Calculation of average psnr differences between rd-curves,” in *VCEG-M33, ITU-T SG16 Q.6, VCEG 13th meeting*, 2001.

[63] L. Y. Wei and M. Levoy, “Fast texture synthesis using tree-structures vector quantization,” in *Proceeding of SIG-GRAPH 2000*, 2000, pp. 479–488.

[64] M. Ashikhmin, “Synthesizing natural textures,” in *Proceedings of ACM Symposium on Interactive 3D Graphics*, 2001, pp. 217–226.

[65] K. Sugimoto and et al., “Inter frame coding with template matching spatio-temporal prediction,” in *Proceedings of IEEE International Conference on Image Processing*, 2004, pp. 465–468.

[66] T. Wedi and H. Musmann, “Motion- and aliasing-compensated prediction for hybrid video coding,” *IEEE Transactions on Circuits and Systemes for Video Technology*, vol. 13, no. 7, pp. 577–586, 2003.

[67] O. Divorra and P. Yin, “Geometry adapted frames partition for video coding,” in *Technical Report, Thomson*, 2006.

[68] P. List, A. Joch, G. Bjontegaard, and M. Karczewicz, “Adaptive deblocking filter,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 614–619, 2003.

[69] T. Tan, G. Sullivan, and T. Wedi, “Recommended simulation common conditions for coding efficiency experiments,” in *VCEG-AE10, ITU-T SG16 Q.6, VCEG 31st meeting*, 2007.

[70] F. Hampson and J. Pesquet, "Motion estimation in the presence of illumination variations," *Signal Processing: Image Communication*, vol. 16, no. 4, pp. 373–381, 2000.

[71] N. Jojic and B. Frey, "Learning flexible sprites in video layers," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 199–206.

[72] H. S. H. Tao and R. Kumar, "Object tracking with bayesian estimation of dynamic layer representations," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 24, no. 1, pp. 75–89, 2002.

[73] P. M. Q. Aguiar and J. M. F. Moura, "Figure-ground segmentation from occlusion," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1109–1124, 2005.

[74] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[75] J. Wang and E. Adelson, "Layered representation for motion analysis," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1993, pp. 361–366.

[76] S. Ayer and H. Sawhney, "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and mdl encoding," in *Proceedings of IEEE International Conference on Computer Vision*, 1995, pp. 777–784.

[77] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillancen," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[78] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice-Hall, 2003.

[79] V. Philomin, R. Duraiswami, and L. Davis, "Pedestrian tracking from a moving vehicle," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2000, pp. 350–355.

[80] P. Burt and E. Adelson, "A multiresolution spline with application to image mosaicking," *ACM Transactions on Graphics*, vol. 2, no. 4, pp. 217–236, 1983.

[81] C. Kuglin and D. Hines, "The phase correlation image alignment method," in *Proceedings of International Conference on Cybernetics and Society*, 1975, pp. 163–165.

[82] C. Dai, Y. Zheng, and X. Li, "Layered representation for pedestrian detection and tracking in infrared imagery," in *Proceedings of IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, 2005, pp. 13–20.

[83] S. T. Chow and J. J. Pupich, "Flir image enhancement by automatic low frequency gain limiting," in *Technical Report*, 1978.

[84] V. Popovici and J. P. Thiran, "Face detection using an svm trained in eigenfaces space," in *Proceedings of the 4th International Conference on Audio- and Video-Based Biometric Person Authentication*, 2003, pp. 925–928.

[85] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 20, no. 1, pp. 23–38, 1998.

[86] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, no. 1, pp. 10–28, 1993.

[87] G. A. K. D. P. Huttenlocher and W. J. Rucklidg, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 15, no. 9, pp. 850–863, 1993.

[88] W. J. Rucklidge, "Efficiently locating objects using the hausdorff distance," *International Journal of Computer Vision*, vol. 24, no. 3, pp. 251–270, 1997.

[89] C. Wren, A. Azarbayejani, T. Darrell, and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 19, no. 7, pp. 780–785, 1997.

[90] A. Shokoufandeh and S. Dickinson, "Graph-theoretical methods in computer vision," *Theoretical Aspects of Computer Science: Advanced Lectures*, pp. 148–174, 2002.

[91] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions via graph cuts," in *Proceedings of IEEE International Conference on Computer Vision*, 2001, pp. 508–515.

[92] C. Dai, Y. Zheng, and X. Li, "Pedestrian detection and tracking in infrared imagery using shape and appearance," *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 288–299, 2006.

[93] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 24, no. 11, pp. 1409–1424, 2002.

[94] E. Shechtman, Y. Caspi, and M. Irani, "Space-time super resolution," *IEEE Transactions on Pattern Analysis and Machine Intellegence*, vol. 27, no. 4, pp. 531–545, 2005.

[95] R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 22–30, 1996.

[96] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proceedings of the 2nd European Conference on Computer Vision*, 1992, pp. 237–252.

[97] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.