EFFICIENT H.264 VIDEO CODING WITH A WORKING MEMORY OF OBJECTS

A Thesis

presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

WENQING DAI

Prof. Zhihai (Henry) He, Thesis Supervisor

December 2009

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

EFFICIENT H.264 VIDEO CODING
WITH A WORKING MEMORY OF OBJECTS

presented by Wenqing Dai,

a candidate for the degree of Master of Science,

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

Professor Zhihai (Henry) He

Professor Marjorie Skubic

Professor Ye Duan

To my parents, Jinfeng Dai and Li Zhang, for their everlasting love and support.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to my advisor Professor Zhihai He. I thank him for offering me the precious opportunity to become one member of video research lab. Without his excellent guidance, great patience and consistent support, I would not be able to complete my research.

I am also extremely grateful to Professor Marjorie Skubic, for her excellent guidance on my research and project, and support through all my course work. I thank Professor Ye Duan for his helpful suggestions during the review of this thesis. I would like also to thank Professor Tina Smilkstein for offering me the teaching experiences during my Master study. I would like to thank Ms. Shirley Holdmerier for her assistance and great patience on my application.

I thank the members of the Video Processing and Communication Lab: Xi Chen, York Chung, Jay Eggert, Xiwen Zhao, Zhongna Zhou, Xin Li , and visiting student Yongfei Zhang, Yunsheng Zhang. Their great support and help made my life and work at MU a precious experience.

I thank all my friends who cheered me up when I was down, offered me great help and shared great time with me. Especially I would express my most sincere thankfulness to Jia Yao for her great support and care to make me never give up on my dream.

At last, I would like to give my sincerely deepest gratitude to my parents Jinfeng Dai and Li Zhang who, from the day I was born, give me their everlasting love to support me and love me. Thank my entire family members, who are at the other side of the earth, for their never-stop love and support on me.

ABSTRACT

Efficient spatiotemporal prediction to remove the source redundancy is critical in video coding. The newest international standard H.264 video coding introduces several advanced features, such as multiple-frame motion prediction and spatial intra prediction [1], which significantly improve the overall coding efficiency. In this work, we focus on efficient H.264 video coding for video monitoring and surveillance. The video camera, mostly stationary, watches the surveillance scene continuously, compresses the video streams which are then transmitted to a remote end for information analysis or archived in a storage device. In these types of video monitoring and surveillance scenarios, the video frame rate is often set relatively low and the activities of persons in the scene often exhibit strong patterns which might repeat at different spatiotemporal scales. In this work, we aim to develop efficient methods to exploit this type of long-term source correlation to improve the overall video compression efficiency. We propose a working memory approach for efficient temporal prediction in H.264 video coding. After video frames are encoded, objects are extracted, analyzed, and indexed in a dynamic database which acts as a working memory for the H.264 video encoder. At the same time, silhouettes are evaluated by using different compression configurations and comparing with ground truth. During the encoding process, objects with similar spatial characteristics are retrieved from the working memory and used for motion prediction of objects in the current video frame. This approach extends the multiple-frame estimation and provides a more generic framework for spatiotemporal prediction of video data. Our experimental results on indoor activity monitoring video data demonstrate that the proposed approach is able to save the coding bit rate by up to 35% with a small computational overhead.

TABLE OF CONTENT

Chapter

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

Introduction

1.1 Overview

Efficient spatiotemporal prediction to remove the source redundancy is the key component in video coding. H.264 video coding introduces several advanced features, such as multiple-frame motion prediction and spatial intra prediction [1], which significantly improve the overall coding efficiency.

In this work, we propose a working memory approach to exploit the long-term source redundancy for efficient H.264 video coding of activity monitoring and surveillance videos. We assume that the video at each moment only has very few persons in the surveillance scene and the persons may appear repeatedly in the scene at different spatiotemporal scales. This assumption does hold in most activity monitoring and surveillance videos. The proposed approach builds up on object detection, content analysis, and image retrieval. More specifically, as shown in Figure 1.1, after a video

frame is being encoded and reconstructed at the encoder side, using adaptive background modeling and silhouette extraction technique developed in our previous work [2], we detect and extract persons from the frame. We extract shape and texture features to describe the object. We index the object and its features in a database, called a *working memory*. During H.264 video encoding, we extract features from the input frame and use these features to retrieve the objects from the working memory. We expect that these retrieved objects will have the highest similarity to objects in the current frame. We then use these objects to construct a reference picture for motion prediction of the input frame. The motion compensated residual picture is then encoded with conventional H.264 video coding. We also develop a memory management module to determine which subset of objects should be maintained in the working memory. Our experimental results demonstrate that the proposed method is able to save the coding bit rate by up to 35% with a small computational overhead.

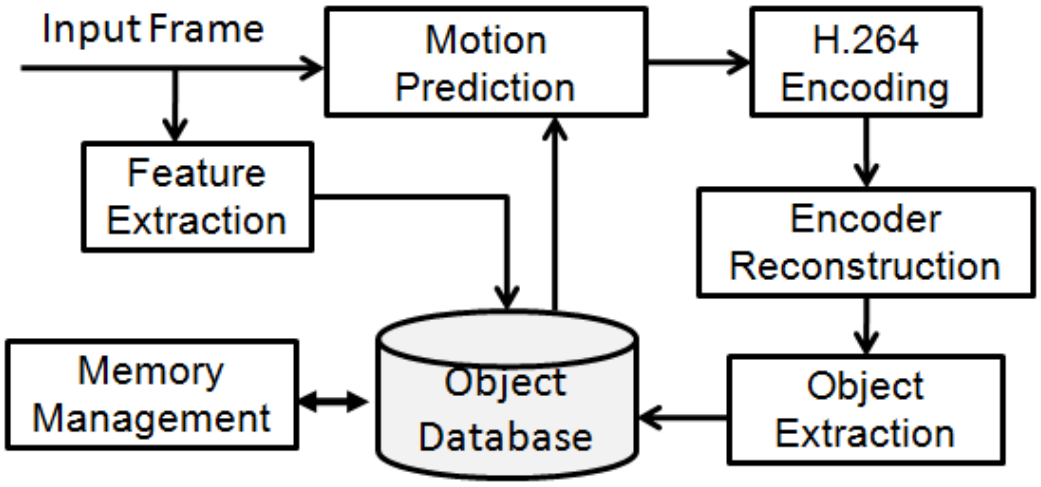


Figure 1.1: Overview of the proposed approach.

1.2 Motivation

In this work, we attempt to develop efficient methods to exploit this type of long-term source correlation to improve the overall video compression efficiency. Multiple-frame motion estimation has been introduced in the H.264 video coding to explore repeated motion in videos [3]. However, its performance degrades significantly at lower frame rates. Immediate previous frames may not be the best prediction reference for the current video frame. For example, Figure 1.2 shows an example of an in-home activity monitor video at a frame rate of 2 frames per second (fps). The top row shows four consecutive video frames, Frames 306, 307, 308, and 309. For each of these four frames (denoted by F_n, $n = 306, 307, 308, \text{and } 309$), we use the following brute-force approach to find its best match in the previous reconstructed frames: Let $\{\hat{F}_k | 1 \le k \le n-1\}$ be the set of previous reconstructed frames. We use each reconstructed frame $\hat{F}_k$ as reference to perform motion prediction of frame F_n and let SAD_k^n be the total SAD (sum of absolute difference) of the corresponding residual picture after motion compensation. The reconstructed frame which yields the minimum SAD_k^n is considered as the best match to frame F_n. The bottom row of Figure 1.2 shows the best matches for Frames 306, 307, 308, and 309, which are reconstructed frames 244, 82, 83, and 222, respectively. The corresponding minimum SAD values are shown in the second column of Table 1.1. The third column shows the minimum SAD if we use the immediate previous frame $\hat{F}_{k-1}$ as reference for motion prediction. We can see that, if we use the best match from the past history, the motion prediction residual can be significantly reduced, which will cost much less bits during compression. This suggests that it is possible to significantly improve the H.264 video coding efficiency by exploiting this type of long-term source redundancy for

activity monitoring and surveillance videos.

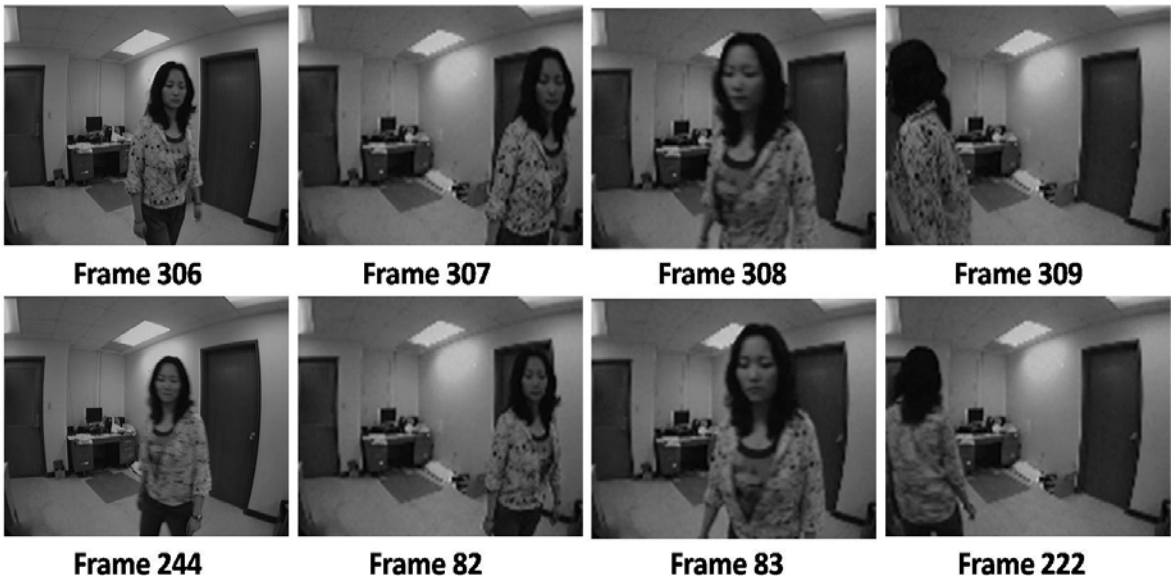


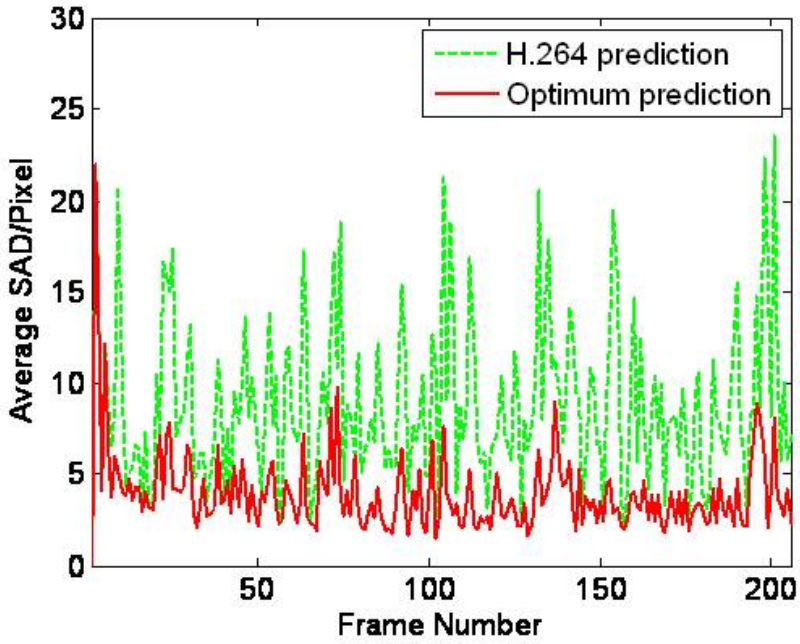
Figure 1.2. Example video frames 306-309 from Sequence2 and their best match in previously reconstructed video frames.

Table 1.1

The minimum SAD comparison between best matches and the previous frame as motion prediction reference.

Frame	Minimum SAD with the	
	Best Match	Previous Frame
306	5.157	13.235
307	6.882	15.091
308	6.372	18.308
309	8.013	21.308

To verify our observation and assumption, we compare the SAD (sum of absolute difference) of the motion compensated difference picture after motion prediction. Figure 1.3 to Figure 1.5 shows the SAD of each frame obtained by these three methods for Video_1, Video_2, and Video_3, respectively. (To show the results clearly, we split the figure into two parts, each showing one half of video frames.) We can see that the SAD obtained by the optimum prediction is much smaller than that of the conventional H.264 motion prediction.



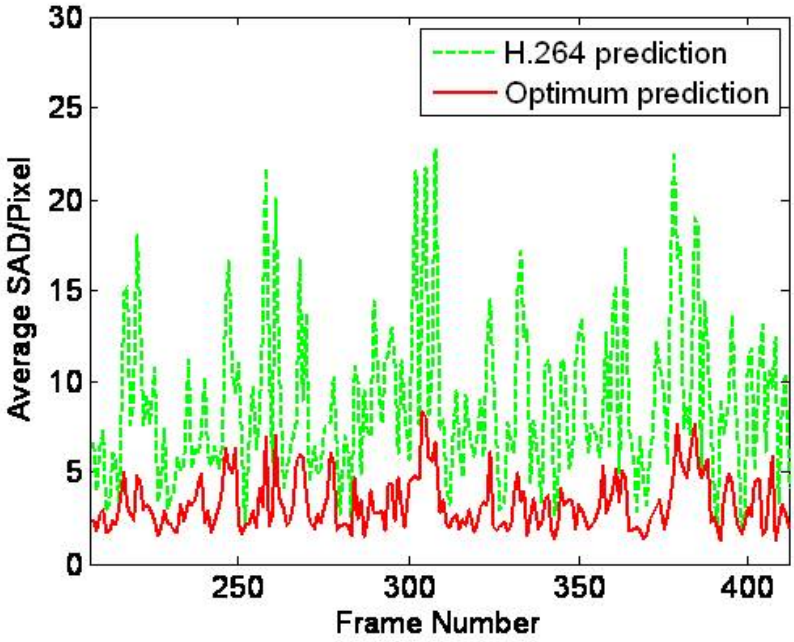
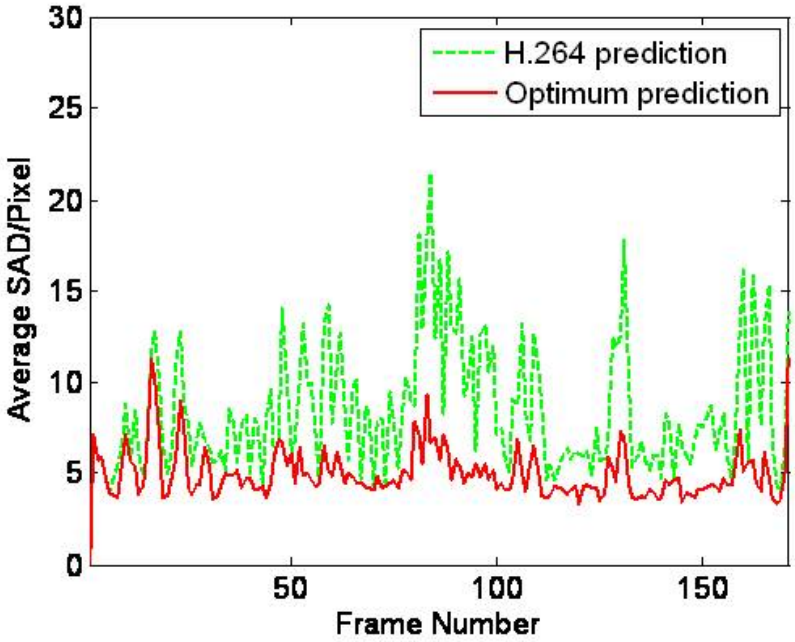


Figure 1.3: Average residual SAD comparison on Video_1 with H.264 motion prediction and optimum search.



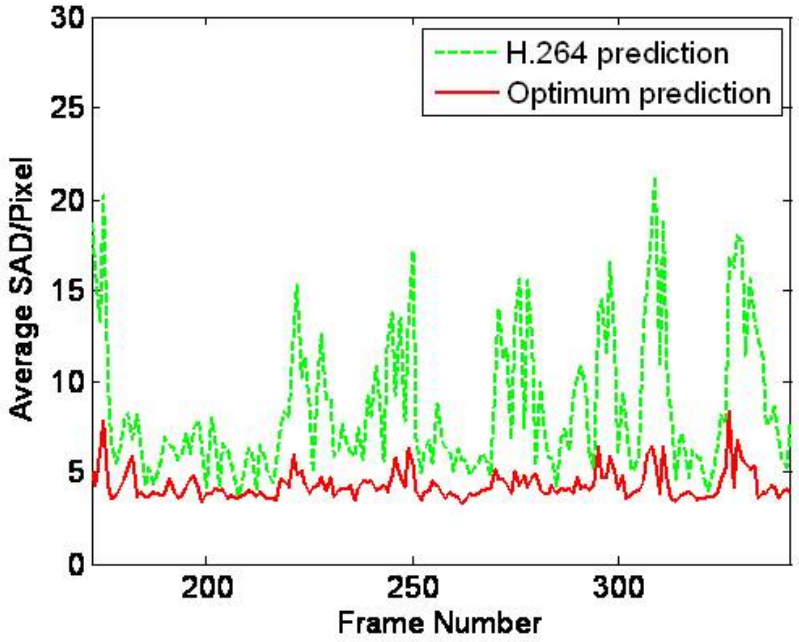
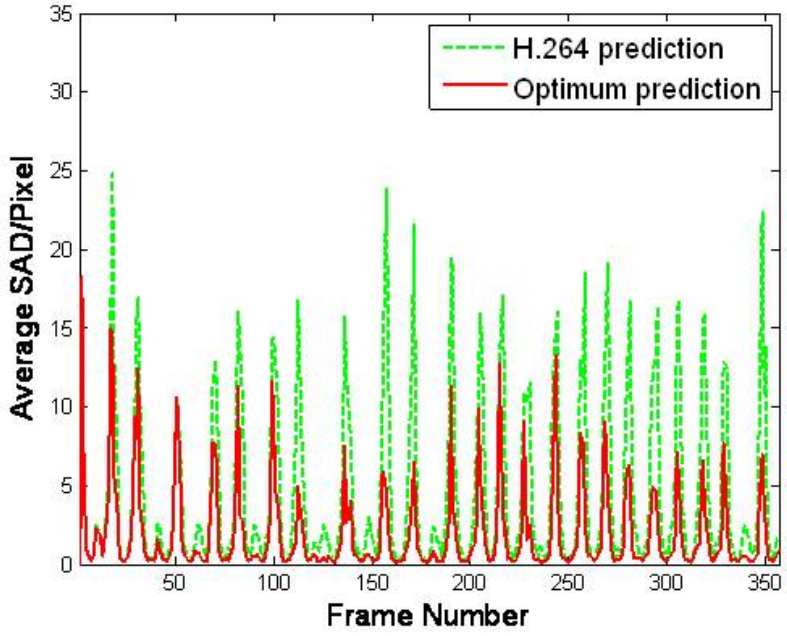


Figure 1.4: Average residual SAD comparison on Video_2 with H.264 motion prediction and optimum search.



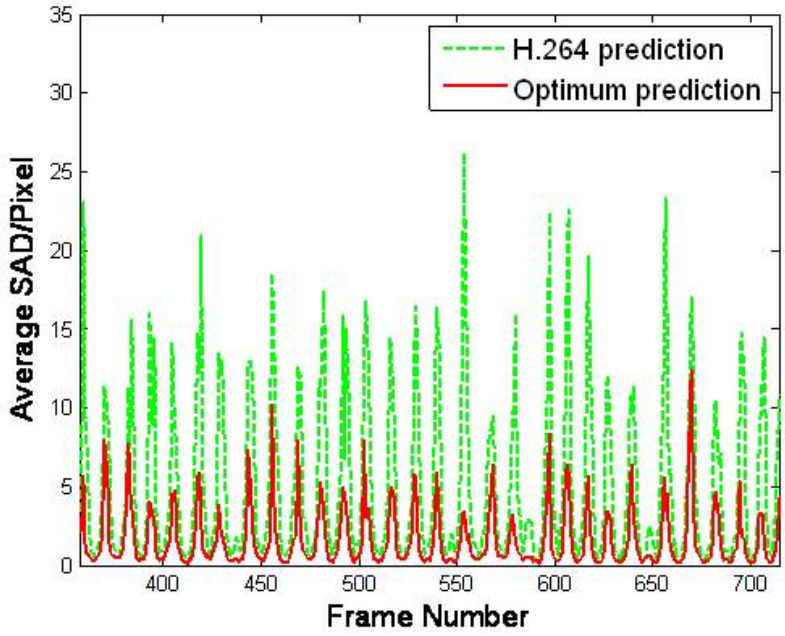


Figure 1.5: Average residual SAD comparison on Video_3 with H.264 motion prediction and optimum search.

Now, the challenge is how to find the best match for each frame to be encoded. As we know, motion estimation is computationally intensive, especially with multiple reference frames as in H.264 [3-5]. In our case, we need to search all previous frames to find the best match. The computational complexity becomes prohibitive as more and more frames are encoded and reconstructed for motion prediction references. Additionally, how to select certain number frames to be stored in the working memory is another challenge in this research.

1.3 Major Contribution

The main contributions of this thesis are as follow:

- Summarize the impact on silhouettes extraction introduced by video compression.

- Development of efficient feature based object retrieval and matching algorithm. We demonstrate that by using silhouettes extraction, centroid, histogram correlation and histogram of dimension, we could find best match frame without performing brute full search in order to achieve better coding efficiency compared to conventional H.264.

- Development of working memory design applied to H.264 encoder. With limited buffer size, the results show higher coding efficiency can be achieved compared to H.264 encoding with small overhead.

1.4 Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 reviews the background and real application of this work. More advanced features related to H.264 coding are summarized. After the background review, some typical algorithms for improving multiple reference frames coding efficiency for H.264 are introduced. Some image retrieval techniques used for object matching are explained as well.

Chapter 3 explains how the video compression has a significant impact on silhouette extraction. We then explain how silhouette extraction is used in our H.264 working

memory approach. At last, a set of results are provided and analyzed.

Chapter 4 presents the main techniques used in this research for object retrieval and matching. The algorithm which is based on Centroid, histogram correlation and histogram of dimension techniques will be formally introduced. Extensive experimental results and comparisons of this method are provided.

Chapter 5 reviews the H.264 encoding and introduces working memory management design. The working memory is applied to H.264 encoder to improve coding efficiency. The integrated design will be further explained. Comparisons made among different working memory sizes are discussed as well.

Chapter 6 summarized the studies presented in this thesis. Conclusions are provided. Future work and directions are discussed.

Chapter2

Background and Related Work

2.1 Background

In this work, we focus on efficient H.264 video coding for video monitoring and surveillance. The video camera, mostly stationary, watches the surveillance scene continuously, compresses the video streams, which are then transmitted to a remote end for information analysis or archived in a storage device. For example, in our on-going research project on eldercare [2], we deploy a video camera to monitor elderly (often aged over 85) people's activities at home continuously for automated functional assessment and safety enhancement, such as detecting falls and abnormal situations which might indicate changes in health conditions. In these types of video monitoring and surveillance scenarios, the video frame rate is often set very low, such as 2-5 frames per second, which is sufficient for human tracking, activity analysis, and scene understanding. The activities of persons in the scene often exhibit strong patterns which might repeat at different spatiotemporal scales. For example, every morning about

6:30am, the person gets up and walks to bathroom. Ten minutes later, he will walk out of the bathroom towards the kitchen for breakfast. During the morning time, the person is often more active, walking around the home, doing exercises, cleaning, preparing meals, etc. Therefore, our goal is to develop efficient methods to exploit this type of long-term source correlation to improve the overall video compression efficiency.

2.2 H.264 Video Coding

The key in efficient video compression is efficient spatiotemporal prediction to remove the data redundancy in the spatiotemporal domain. The H.264 video coding standard introduces several advanced features, such as multiple-frame motion prediction, variable block size motion compensation and sub-pixel motion estimation [1], which have significantly improved the coding efficiency.

One of our central tasks in this work is to find the best reference frame for current frame at a low computational complexity. The best reference frame is selected from previous decoded N frames. To further reduce the spatiotemporal redundancy, H.264 uses short and long term reference frame for more accurate motion prediction. In H.264 pictures that are encoded or decoded and available for reference are stored in the Decoded Picture Buffer (DPB). All available reference frames are marked as short term reference picture or long term reference picture. Short term reference pictures will be removed from DPB by an explicit command or when the DPB is full. The frame marked as long term reference will only be removed by an explicit command, which means long term reference pictures can be utilized as the reference frames which are not only within the

small search window. To be noted, a new innovation in H.264/AVC allows the motion-compensated prediction signal to be weighted and offset by amounts specified by the encoder [1], which will dramatically improve coding efficiency for scenes with gradual transitions such as fades. As seen in prior MPEG standards, a single reference picture is used in P frame and prediction is not scaled. In B frame, it uses two pictures as reference and the prediction is composed by equally averaging the weighting factors. Comparing to previous standards, H.264 associates the weighting factor with reference picture index which is efficient for multiple reference frame management. In H.264 explicit mode, a weighting factor and offset may be coded in the slice header for each allowable reference picture index; in H.264 inexplicit mode, the weighting factors are derived based on the relative picture order count (POC) distances of the two reference pictures [6].

2.3 H.264 Multiple Reference Frame Motion Estimation

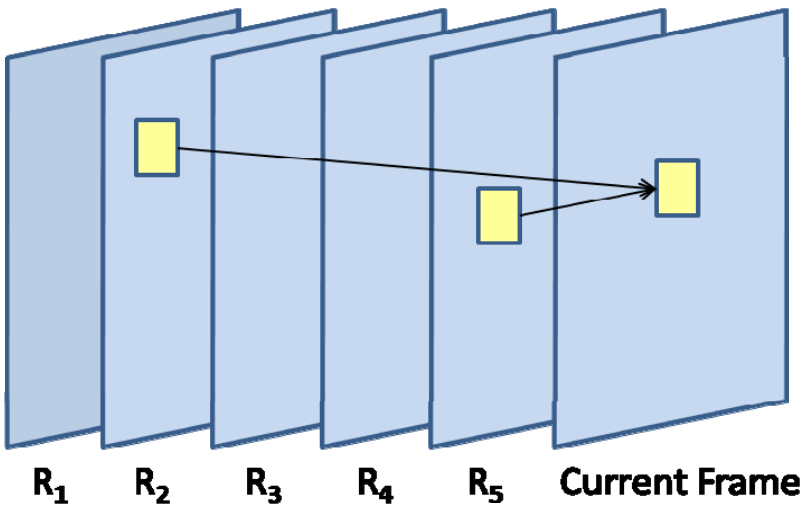


Figure 2.1: Overview of multiple reference frame motion prediction

H.264 uses multiple reference frames to achieve better prediction in many conditions.

Figure 2.1 gives an overview of H.264 multiple reference frame motion estimation. However, multi-frame motion predict dramatically increases the computation complexity of the encoder [7]. Several methods have been proposed to reduce the computational complexity. For example, the unsymmetrical-cross multi-hexagon-grid search (UMHexagonS) has been proposed in [7]. Su and Sun [3], Hsiao et al [8] and Duanmu et al [9] attempted to reduce the complexity using continuous tracking techniques to provide a good starting point for motion search. Methods based on tracking will likely fail in case of occlusions. Wiegand et al [10] introduced a new motion search order based on the triangle inequality for long-term motion prediction. An adaptive motion search scheme with early termination and zero-block detection has been developed in [11]. Huang et al [12] utilized the information from the previous frame to determine whether the motion search on the remaining reference frames is needed or not. [13] examined available MV and SAD information so as to terminate the multi-frame motion estimation procedure. Wang et al [14] exploited the spatial correlation between neighboring blocks to choose the best reference frame for the current block. Sohn and Kim [15] determined the number of reference frames using the correlation between the block of current frame and that of previous frame. Kapotas and Skodras [16] considered the Lagrangian cost for reference frame selection. We can see that existing methods have been exploring the source correlation between neighboring frames for fast and efficient motion prediction. Because of the dramatic increase of computational complexity in multi-frame motion prediction, typically, up to 5 frames are used for motion prediction in practical H.264 video encoding [7]. This small window of reference frames limits our capability in exploring long-term source correlation in video data, especially in surveillance videos.

2.4 Image Retrieval

In this work, we consider all of the previous decoded frames as a data base. This is in nature of image retrieval problems: to find an image from the data base with highest similarity. Figure 2.2 provides an overview about image retrieval.

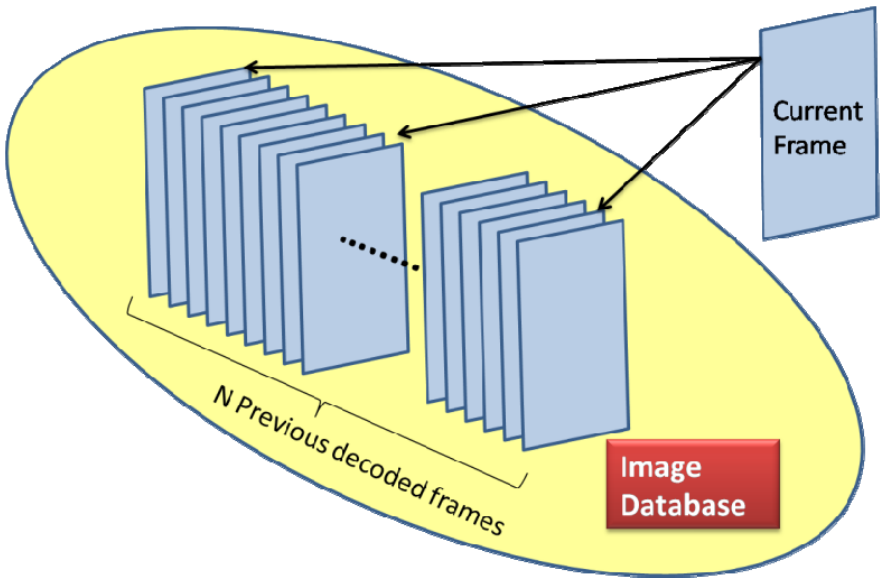


Figure 2.2: Overview of image retrieval

With the advances in computer technologies, there has been an explosion in the amount and complexity of digital images being generated. How to access the vast amount of data is a key challenge to allow people to browse, search and retrieval efficiently. Based upon this fact, image retrieval has been a very active topic since the 1970s. Date back to 1970s, in traditional database people annotated image with a set of predefined key words which are stored with the corresponding images and will be matched with user's queries, for example Chang's Query-by pictorial-example [17] and Pictorial Data-Base Systems [18]. The former one used a relational query language introduced for manipulating queries regarding pictorial relations as well as conventional relations [17]. The latter one proposed a pictorial data base which is a collection of sharable pictorial data encoded in

various formats. A pictorial database system, or PDBS, provides an integrated collection of pictorial data for easy access by a large of number of users. But TBIR has to manually annotate the images so it is not practical to manually do the annotation on the extremely large number of images. The human perception differences will make the keywords different even which are used to describe the same image. As well as some low level features like color, texture and shape cannot be well captured by textual keywords. All these lead people to find the new method during 1990's, which is called content-based image retrieval (CBIR). CBIR is a very active research area for past decades. There are a large amount of researchers currently working on CBIR. Main research issues in CBIR include feature extraction, dimensionality reduction, relevant feedback, etc. Because CBIR is most close to this research, we will focus on more aspects in CBIR. Figure 2.3 shows a basic framework of a CBIR system [19].

Color is one of the most widely used features of the great majority of content-based image retrieval systems. The color feature is relatively robust to background complications and independent of image size and orientations. [20] introduced Chabot, which basically is an image database which adds color feature with manually annotated keywords to search the image. Swain [21] introduced a technique called Histogram Intersection, which matches model and image histograms and allows real-time indexing into a large database of stored models. Stricker [22] proposed two color indexing techniques. One is using cumulated color histogram which has slightly better performance than color histogram but significantly more robust with respect to the quantization parameter of the histograms. The other approach is that the similarity

function which is used for the retrieval is a weighted sum of the absolute differences between corresponding moments. Smith and Chang [23，24] also proposed to identify the regions within images that contain colors from predetermined colorsets. By searching over a large number of color sets, a color index for the database is created in a fashion similar to that for file inversion which allows very fast indexing of the image collection by color contents of the images.

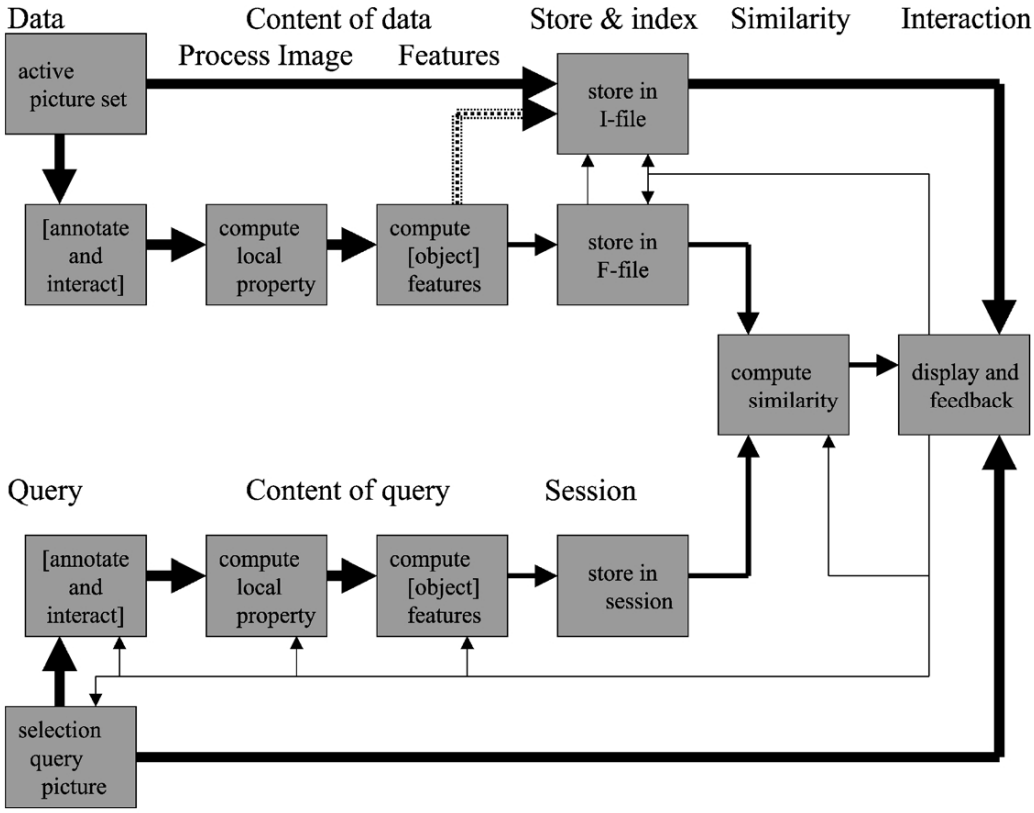


Figure 2.3: Framework of CBIR system

Texture is another feature which is commonly used by CBIR system. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment [25]. Haralick got textural features derived from the

angular nearest-neighbor gray-tone spatial-dependence matrices at early 1970s[25]. Gotlieb further extended this idea and derived a general model for analysis and interpretation of experimental results in texture analysis when individual and groups of classifiers are being used. They proposed to use six representative classifiers which are second angular moment f1, contrast f2, inverse difference moment f5, entropy f9, and information measures of correlation I and II, f12 and f13, and it could give a systematic study of the discrimination power of all 63 combinations of these classifiers on 13 samples of Brodatz textures [26]. Later on, Tamura proposed to represent texture by six visual texture properties which are coarseness, contrast, directionality, linelikeness, regularity and roughness [27]. All of the six properties are visually meaningful. The Query by Image Content (QBIC) project studies the method to extend and complement text-based retrievals by querying and retrieving images and videos by content. Queries can be performed using attributes such as colors, textures, shapes and object positions [28]. MARS (Multimedia Analysis and Retrieval System) is a system that supports similarity and content-based retrieval of images based on a combination of their color, texture, shape and layout properties [29, 30]. CBVQ developed by J. R. Smith focused on color and texture region and used binary set representations of color and texture, respectively [31, 32].

Shape representations depending on applications may require transformation invariant. There are a lot of researches which had been done in this area. Fourier descriptor is one major achievement in this area. It utilizes the Fourier transformed boundary as the shape feature. Rui proposed a Modified Fourier Descriptor and a new distance metric for

describing and comparing closed planar curves for shape matching in content based image retrieval system. Their method accounts for the effects of spatial discretization of shapes [33]. [34] generated an image “signature” for each database picture with respect to “key” objects. WebSeer system is based on statistical observations about the image content of the two types. The system uses image contents like colors and shapes to index images [35].

Although previous features could provide reasonable discriminating power in image retrieval, the false positives become more as the image collection sizes increase. Therefore, another method called color layout utilized both color feature and spatial relations came up. Rickman presented a novel image coding scheme which captures some of this locally correlated color information and improves the selectivity of the retrieval mechanism. Their technique used a histogram of features which represent frequently occurring local combinations color tuples occurring throughout the image [36]. Huang proposed a new image feature called the color correlogram and used it for image indexing and comparison. This feature distills the spatial correlation of colors and is both effective and inexpensive for content-based image retrieval [37].

It should be noted that the objective of content-based image retrieval is to find images from the database which are perceptually, conceptually, or semantically similar to the query image. It does not necessarily minimize the differences between these two images. Therefore, existing features used for image retrieval cannot be directly used in this research for finding the best motion match.

Chapter 3

Effects on Silhouettes Introduced by Video Compression

3.1 Overview

Extracting features to differentiate foreground objects from background is the first step of silhouettes. The silhouette extraction scheme is based on brightness distortion and chromaticity distortion. Therefore, the silhouette extraction quality will be greatly depending on the value of each pixel's change. Quantization, involved in image processing, is a lossy compression technique achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible. But the more the image is compressed, the more information you would lose due to the Quantization process. This chapter will give a detail explanation on this aspect. At last some comparisons are made between the conventional H.264 and H.264 without deblocking filter.

3.2 Silhouettes Extraction

In this work, we use the silhouette extraction method developed in our previous work [2] to extract persons from the video scene. For the completeness of presentation, we provide a brief review of this algorithm. Silhouette extraction, namely, segmenting a human body or objects from a background, is the first and enabling step for many high-level vision analysis tasks, such as video surveillance, people tracking and activity recognition [38-41]. We consider silhouette extraction as an adaptive classification problem. We utilize image features which are invariant to changes in lighting conditions. High-level knowledge is fused with low-level feature-based classification results to handle time-varying backgrounds changes.

We consider silhouette extraction as an adaptive classification problem. We utilize image features which are invariant to changes in lighting conditions. High-level knowledge is fused with low-level feature-based classification results to handle time-varying backgrounds changes. Extracting features to differentiate foreground objects from background is the first step of silhouette extraction. A basic requirement is that features should be invariant under brightness changes. Further, it should be effective in differentiating shadow from background. In this work, we use two features: *brightness distortion* and *chromaticity distortion*. More specifically, we extract features in the RGB color space [42]. For adaptive background update, we use the past Δ frames for background modeling. At each pixel location i, we compute the average values of its RGB components in the past Δ frames and denote them by vector E_i. We also calculate and standard deviations of the color components at each pixel. Let I_i be the pixel in the

current frame. As shown in Figure 3.1, we project the vector I_i onto vector E_i. We define brightness distortion α_i as:

$$\alpha_i = \arg\min_{\alpha_i} \|I_i - \alpha_i E_i\|^2 = \frac{\left(\frac{I_R(i)\mu_R(i)}{\sigma_R^2(i)}\right) + \left(\frac{I_G(i)\mu_G(i)}{\sigma_G^2(i)}\right) + \left(\frac{I_B(i)\mu_B(i)}{\sigma_B^2(i)}\right)}{\left(\frac{\mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{\mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{\mu_B(i)}{\sigma_B(i)}\right)^2}, \quad (1)$$

and chromaticity distortion as:

$$CD_i = \|I_i - \alpha_i E_i\| = \sqrt{\left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)}\right)^2}, \quad (2)$$

where $[I_R(i), I_G(i), I_B(i)]$ represent the values of red, green and blue components of the i^{th} pixel in the RGB color space. $[\mu_R(i), \mu_G(i), \mu_B(i)]$ and $[\sigma_R(i), \sigma_G(i), \sigma_B(i)]$ are the mean and standard deviation of these color components. This color model separates the brightness from the chromaticity components as shown in Fig. 4. It has been found that the chromaticity distortion is invariant under brightness changes [42]. Our foreground-background classification is based on the following two observations: (1) *image pixels in the background often have little change in their chromaticity distortion*; and (2) *shadow often causes brightness distortion but little chromaticity distortion*. Based on these two observations, we establish the following decision rules for foreground, background, and shadow detection: (1) if the chromaticity distortion CD_i is large, I_i is a foreground pixel; (2) if the chromaticity distortion is small and the brightness distortion is about 1.0, it is a background pixel; (3) if chromaticity distortion is small and the brightness distortion smaller than 1.0, it is a shadow pixel.

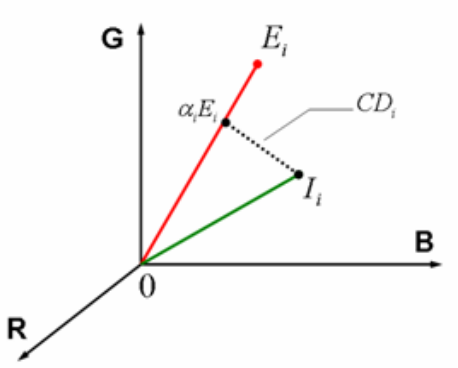


Figure 3.1 Illustration of brightness and chromaticity distortion.

In silhouette extraction within a dynamic video scene, we need to continuously update the background model by incorporating background changes. A commonly used method to update background is that, if an object or image area remains stationary for a certain period of time, it is considered to be background. Here, we use the past Δ frames to update the background model. For accurate silhouette extraction, we want Δ to be small so that the background can be quickly updated. However, when Δ is small, the human body could be easily updated as background if the person does not move for a while, for example, sitting still on a chair for a few minutes. To solve this problem, we propose to utilize high-level knowledge about human motion as a guideline to perform adaptive update of the background model.

Many sophisticated human tracking algorithms have been developed in the literature [43, 44, 45]. However, they often have high computational complexity. Here, to achieve low-complexity, we use a simple block-based motion estimation which has been extensively used in video coding [7]. More specifically, suppose that we have obtained the silhouette for frame n. We find a bounding box for the silhouette such that 95% of foreground

pixels are included. For each image block within the bounding box in the current frame n, we find its best match in the next frame $n+1$ using SAD (sum of absolute difference) as a distance metric. To speed up the motion estimation process, we use a fast algorithm called diamond search [46]. Once the motion vectors of all blocks are obtained after block-based motion estimation, we take their average to predict the human body position (or the center of its bounding box) in the next frame $n+1$. Those image blocks which contain the human body should be updated very slowly so that the human body won't be absorbed into the background. Those blocks outside the predicted body region can be updated much faster to make sure that new objects are quickly absorbed into the background. After background update and silhouette extraction, we update the dimension, height and width, of the bounding box in frame $n+1$.

Figure 3.2 to Figure 3.4 show the silhouette extraction and human tracking results for some sample frames of three in-door activity monitoring videos. It can be seen that this algorithm is able to obtain high-quality of human silhouettes and track persons accurately.

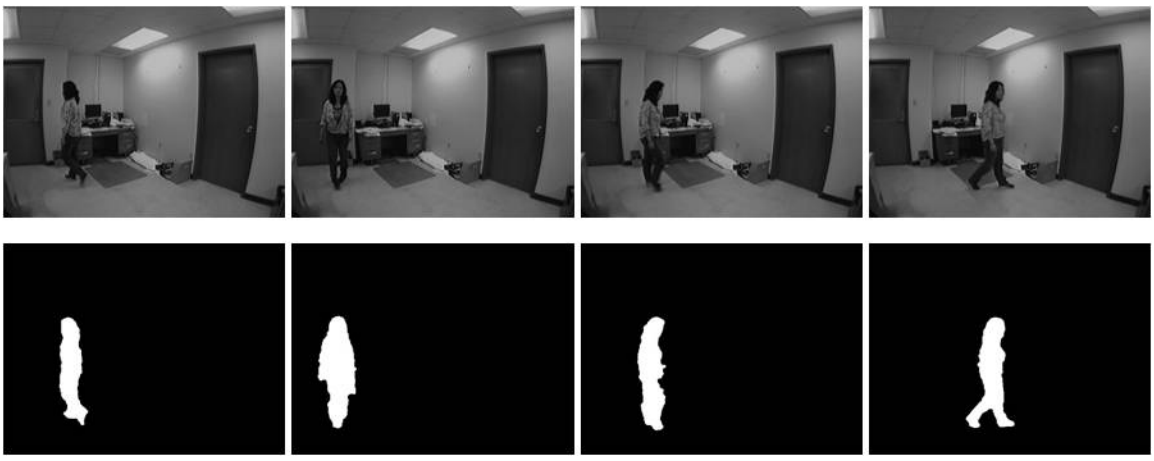


Figure 3.2: Silhouette extraction and human detection results for Video 1

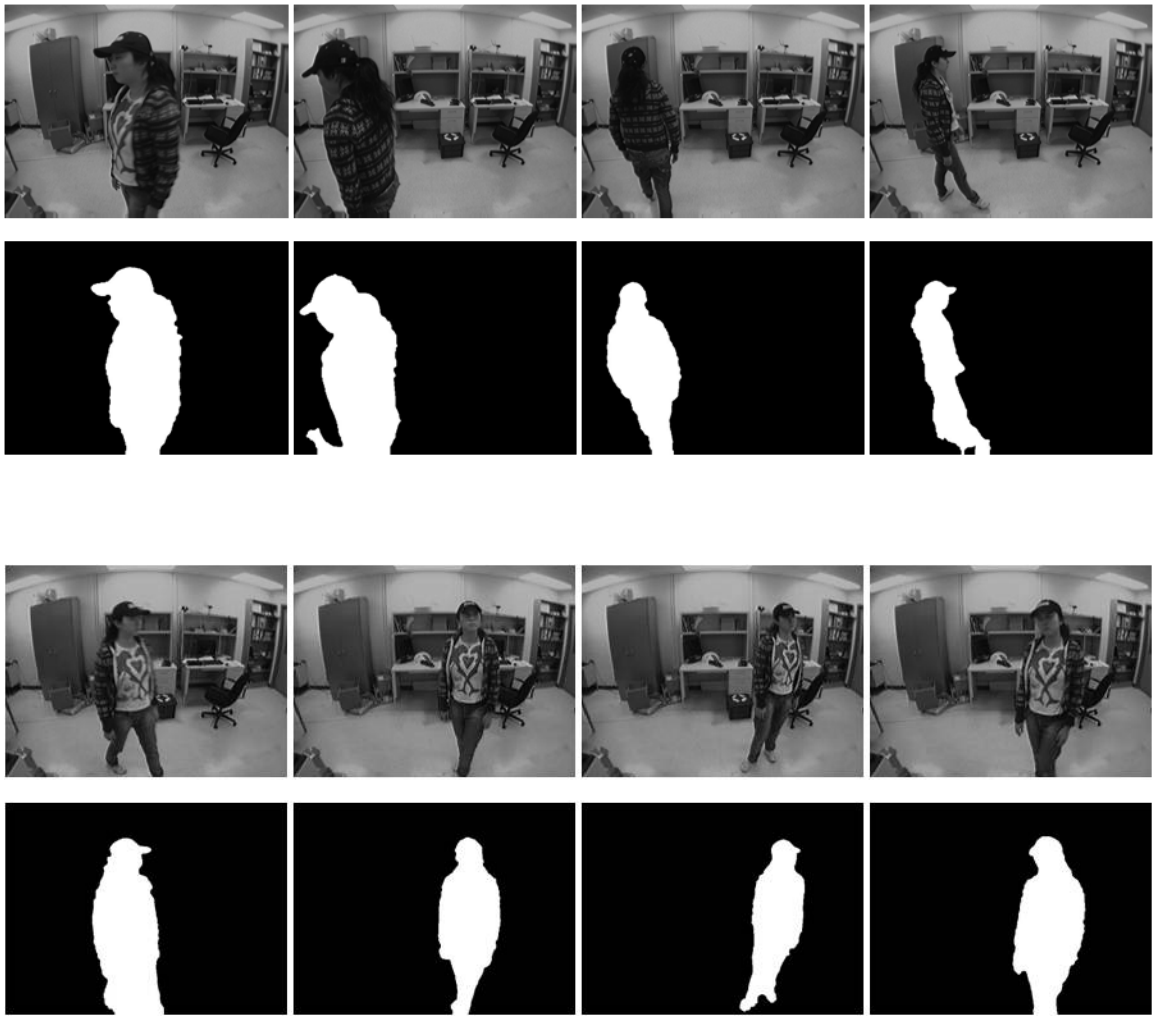


Figure 3.3: Silhouette extraction and human detection results for Video 2

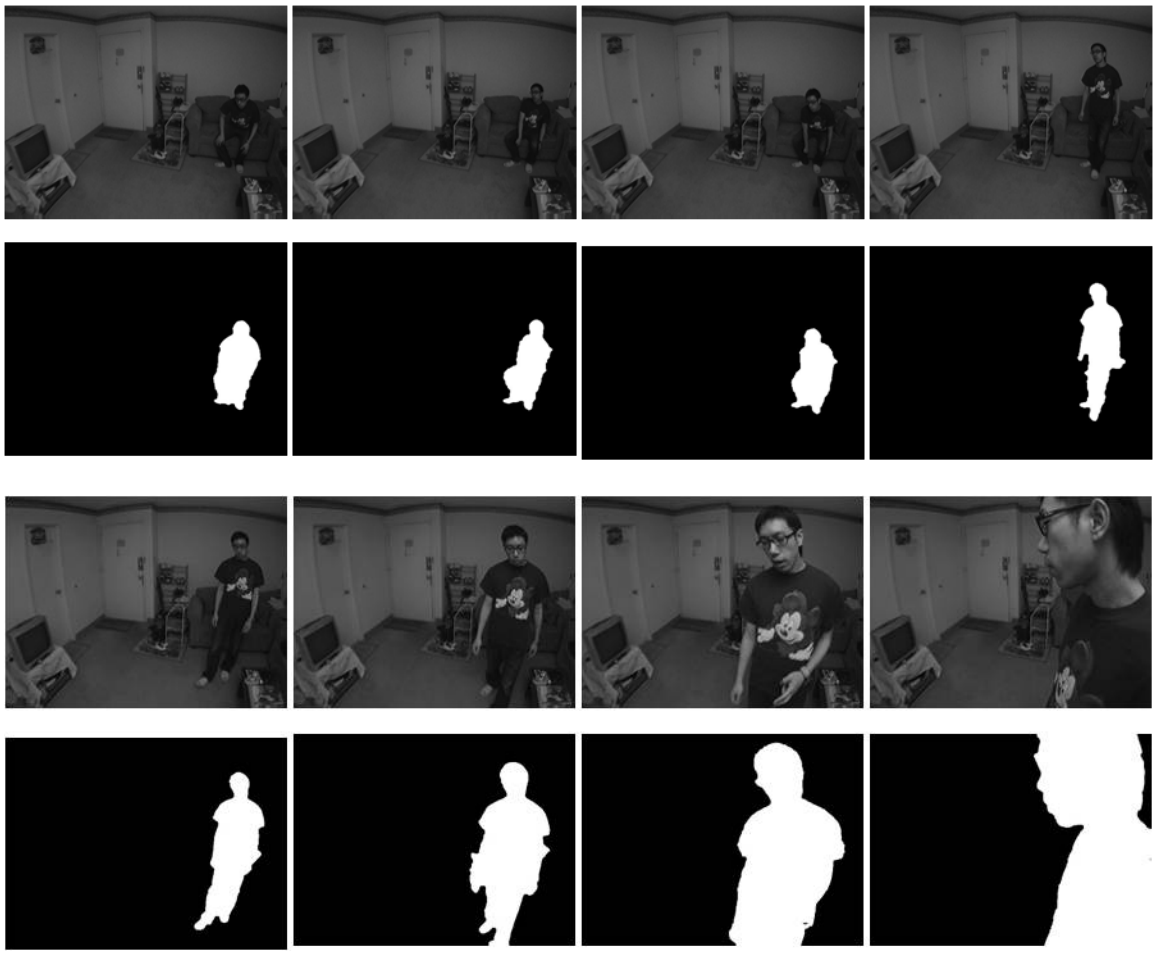


Figure 3.4: Silhouette extraction and human detection results for Video 3.

3.3 Silhouettes Extraction in Working Memory

In this research, we apply the silhouette extraction algorithm on the original video frame F_N and obtain the binary silhouette image. Using this binary image as mask, we segment the human object from the background and denote it by O_N. Let B_N be the corresponding background image constructed by the silhouette extraction algorithm. We then extract a set of visual feature, denoted by f_N, from O_N to characterize the human object. As illustrated in Figure 1.1, after frame O_N is encoded by H.264, we also apply the same

silhouette extraction algorithm to the encoder reconstruction $\hat{F}_N$ and obtain the reconstructed human object $\widehat{O}_N$. Let $\widehat{B}_N$ be the corresponding background image. We also extract a set of features, denoted by $\hat{f}_N$, to characterize $\widehat{O}_N$. Both $\widehat{O}_N$ and its feature vector $\hat{f}_N$ are indexed and stored into a database, called working memory in this work.

At this moment, let us assume that $\Omega = \{\widehat{O}_k | 1 \leq k \leq N-1\}$ are all available in the working memory when frame F_N is being encoded. We use feature f_N as query input to the working memory to retrieve the object which matches the current object $\widehat{O}_N$ best. We denote this best match by $\widehat{O}^*_{<N}$. Here, "$< N$" denotes frame numbers less than N. We overlay $\widehat{O}^*_{<N}$ on the background image $\widehat{B}_{N-1}$ to form a reference frame $\hat{F}^*_{<N}$ for motion prediction. We expect that, using $\hat{F}^*_{<N}$ as the motion prediction reference, the motion compensated difference will be minimized. For convenience, we refer to this type of motion prediction approach as working memory prediction (WMP). If the best match happens to be $\widehat{O}_{N-1}$, then $\hat{F}^*_{<N}$ will be exactly the previous frame $\hat{F}_{N-1}$. Therefore, the conventional H.264 motion prediction (P-frame) is a special case of the proposed WMP scheme.

3.4 Comparisons of Silhouettes with Different Video Compression Configurations

In practice, due to limited transmission bandwidth or storage space, videos are often compressed with JPEG, MPEG, or H.264 coding scheme. Therefore, it is necessary to understand the performance of silhouette extraction on compressed video frames and investigate how the compression artifacts could impact the performance of silhouette

extraction. Depending on different configurations in video compression scheme, the degree of performance degradation in silhouette extraction varies. In this section, we conduct extensive experiments to evaluate the impact of different image/video compression schemes on the silhouette extraction performance.

First, we configure the H.264 to the default settings. We tested on three set of sequences which have the ground truth for each sequence. In order to get different quality image from the video compression, we encoded each sequence with different quantization parameters (QP). QP, a setting in video coding that controls the quality of video compression. In H.264 as QP is increased, the quality of the video decreases. Then we extracted the silhouettes from the decoded images and compared with the ground truth. The results of sequence1 are selectively summarized from Figure 3.5 to Figure 3.8.

Second, we turned off the deblocking filter function in H.264 which might introduce more errors due to the edge effects caused by block based the motion estimation. Same as previous step, after we decoded the images which are compressed at different QP level, we extracted silhouettes of these sequences. Then, we compare the silhouettes results with the ground truth. Figure 3.9 to Figure 3.14 are the picked sample results for sequence 1.

Figure 3.5: Test sequence 1 Frame 80 (Original Image) and its silhouette (Ground Truth).

Figure 3.6: Test sequence 1 Frame 80 decoded image and its silhouette with H.264 QP=24 .

Figure 3.7: Test sequence 1 Frame 80 decoded image and its silhouette with H.264 QP=42

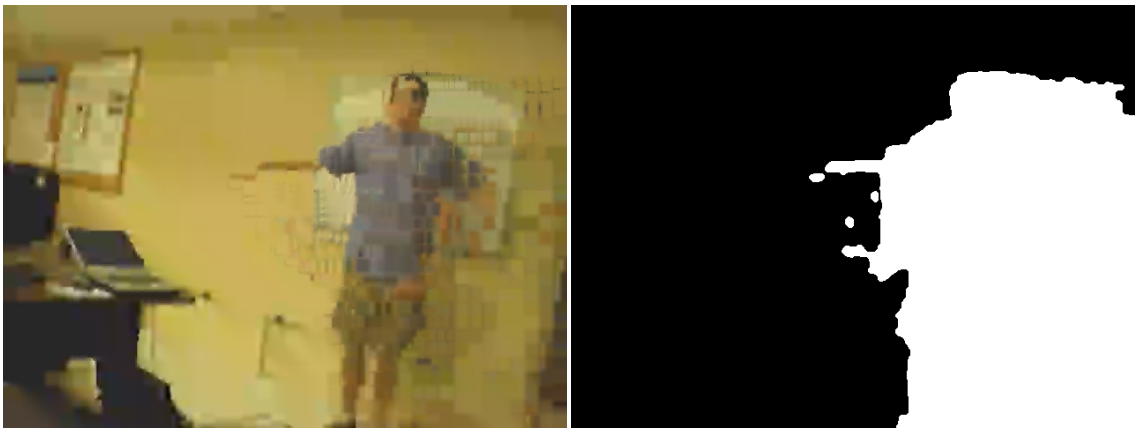


Figure 3.8: Test sequence 1 Frame 80 decoded image and its silhouette with H.264 QP=51

Figure 3.9: Test sequence 1 Frame 80 decoded image and its silhouette with H.264 QP=33 without deblocking filter

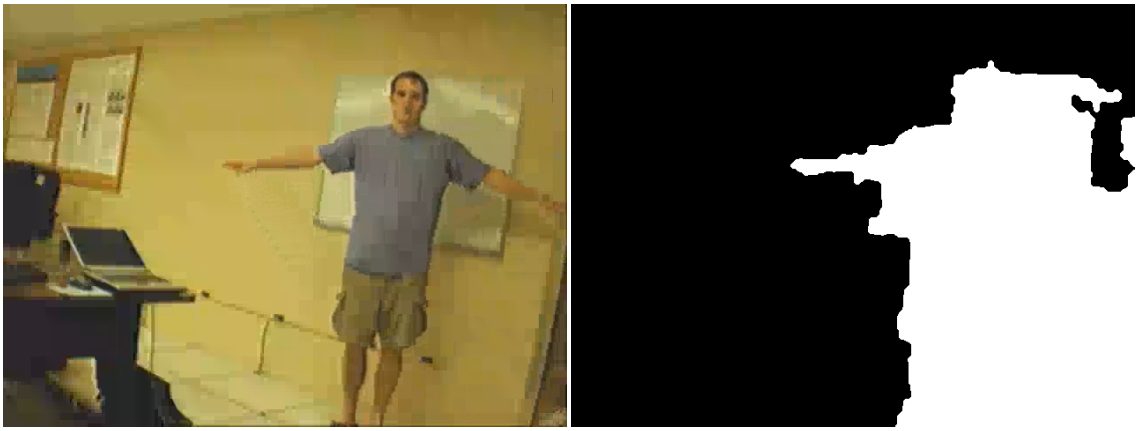


Figure 3.10: Test sequence 1 Frame 80 decoded image and its silhouette with H.264 QP=42 without deblocking filter

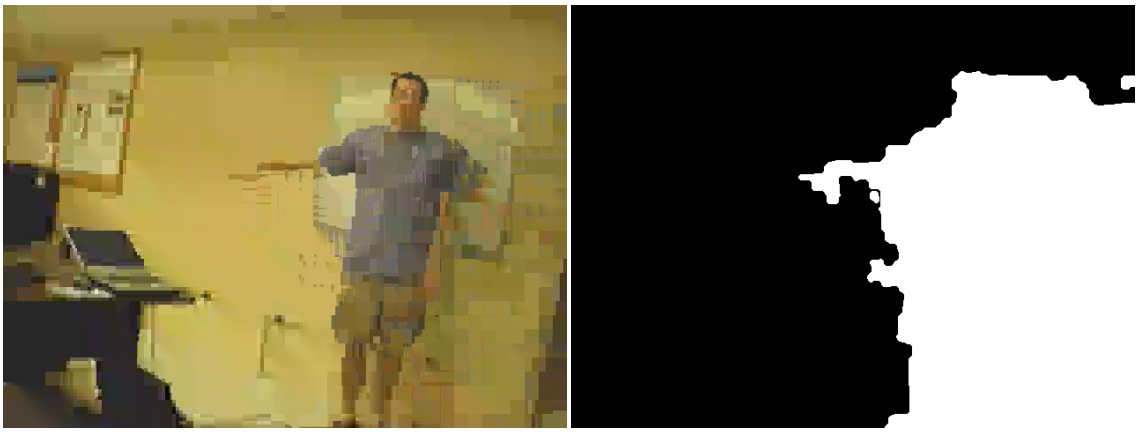


Figure 3.11: Test sequence 1 Frame 80 decoded image and its silhouette with H.264 QP=51 without deblocking filter

Figure 3.12: Test sequence3 Frame111 decoded image and its silhouette with H.264 QP=33 without deblocking filter

Figure 3.13: Test sequence3 Frame111 decoded image and its silhouette with H.264 QP=42 without deblocking filter

Figure 3.14: Test sequence 3 Frame 111 decoded image and its silhouette with H.264 QP=51 without deblocking filter

After we decoded the video and extracted the silhouettes, we compared each sequence to the ground truth whose value is denoted as N_{GT}. N_{GT} is the total pixel number per frame, which is the foreground of the ground truth. Then we calculate the average error rate R_{err} for each video sequence with n frames. We denote the average error rate as follow:

$$R_{err} = \frac{\sum_{f=1}^{n}(\frac{N_{ERR}}{N_{GT}} \times 100\%)}{n} = \frac{\sum_{f=1}^{n}(\frac{N_{BF} + N_{FB}}{N_{GT}} \times 100\%)}{n} \quad (3)$$

Where N_{ERR} is the total error pixel numbers per frame, which is composed of two type of errors: the error from false detecting background pixel as foreground pixel denoted as N_{BF} and the error from false detecting foreground pixel as background pixel denoted as N_{FB}. Figure 3.15 to Figure 3.17 show the picture PSNR v.s. error rates.

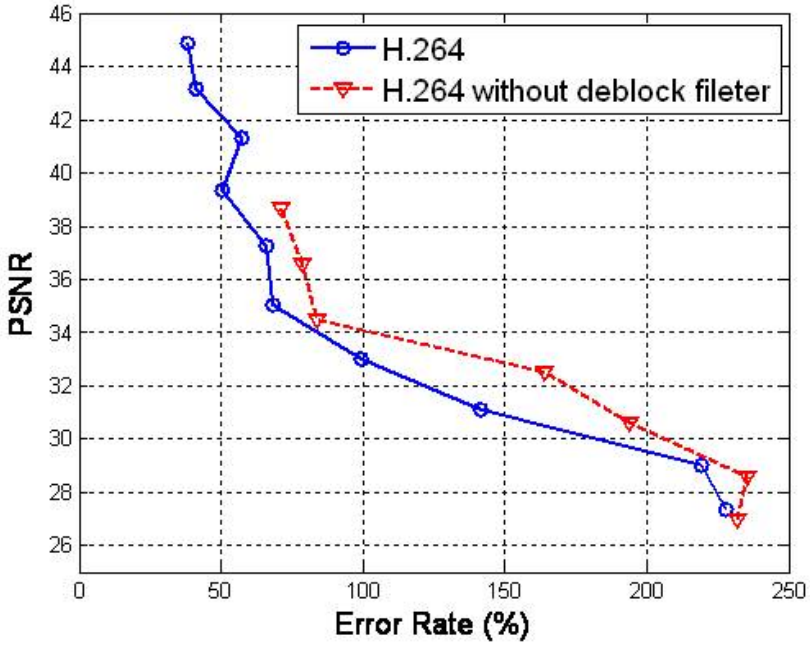


Figure 3.15: Error rate vs PSNR comparison between H.264 and H.264 without deblocking filter for Sequence 1

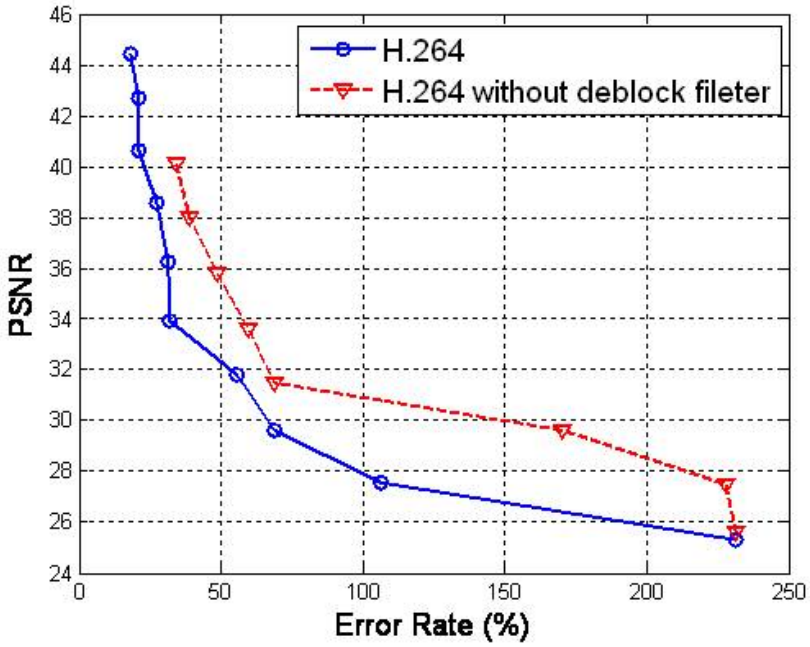


Figure 3.16: Error rate vs PSNR comparison between H.264 and H.264 without deblocking filter for Sequence 2

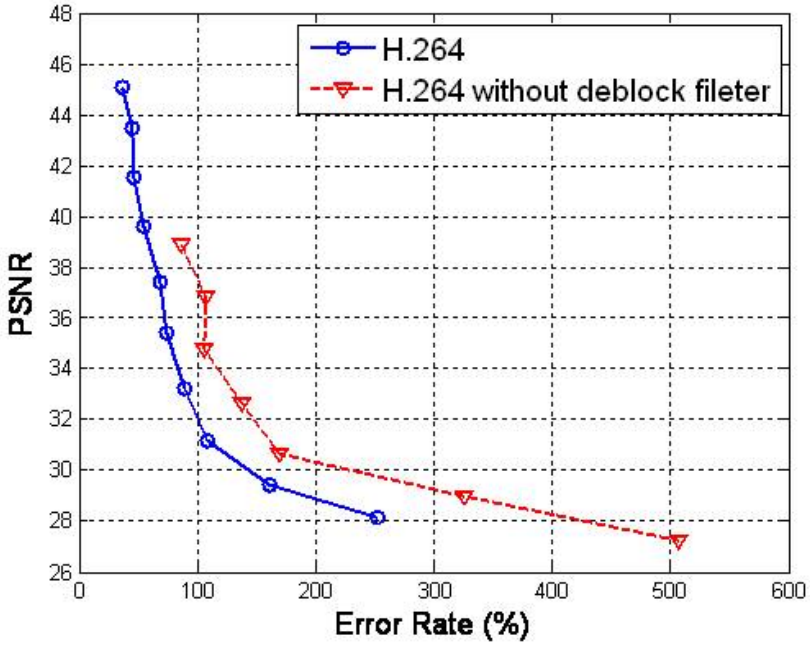


Figure 3.17: Error rate vs PSNR comparison between H.264 and H.264 without deblocking filter for Sequence 3

From the results, we can see that, as PSNR decreases (which means as QP increases), the quality of silhouettes decreases and the error rate of silhouette extraction increases. Especially after a certain level, the Error Rate of silhouettes increases dramatically. Comparing the silhouette results which are from the default H.264 settings to the silhouettes results which are from the H.264 without deblocking filters, the latter one will introduce more errors compared to the former in respect to the same PSNR. Therefore the silhouettes results from the latter one are worse than the silhouettes from the former one. The research helps us realize the relationships between the video compression techniques and the silhouette extraction technique. It is fundamental for achieving better coding efficiency by using this proposed algorithm.

Chapter 4

Feature Based Fast and Accurate Object Retrieval

4.1 Overview

As discussed in the previous chapter, we need to find the best object $\hat{O}^*_{<N}$ from the working memory such that the motion compensated difference between $\hat{F}^*_{<N}$ and F_N is minimized. We denote this difference by $d_M[\hat{F}^*_{<N}, \ F_N]$. Note that typically the background models at frames $N-1$ and N will be very close to each other, except sudden lighting condition changes. In this case, an effective encoding option is to terminate the motion prediction chain and to use an INTRA frame. Therefore, the major motion compensated difference will be from the foreground objects, e.g. persons. We have

$$d_M\left[\hat{F}^*_{<N}, F_N\right] = d_M\left[\hat{O}^*_{<N}, O_N\right]. \tag{4}$$

The process of finding the best match $\hat{O}^*_{<N}$ can be summarized as follows:

$$\hat{O}_\Omega = \arg \min_{\hat{O}_k \in \mathbf{\Omega}} \ d_M\left[\hat{O}_k, O_N\right] \tag{5}$$

To compute $d_M[\hat{O}_k, O_N]$, we need to perform motion prediction and compensation between $\hat{O}_k$ and O_N. As we know, motion estimation is computationally intensive. Furthermore, the size of the working memory, i.e., the total number of candidate objects in it, increases with the frame number N. Therefore, the computational complexity in finding the best match $\hat{O}_\Omega$ in (3) will become prohibitive when N is large. Now, the question is: *how to finding the best match $\hat{O}_\Omega$ in (3) with low computational complexity*?

In this research, we propose to explore a content-based image retrieval method. We will extract a set of features to describe $\hat{O}_k$ the reference object in the working memory and the current object O_N, respectively. We then attempt to find the best match in the feature space. As we know, feature-based matching has much low computational complexity than direct motion prediction. Essentially, the problem can be summarized as follows: *finding the best motion match without performing motion prediction*. This is a non-trivial task.

4.2 Feature Based Object Retrieval

When defining the features, we need to make sure that the best match in the feature space yields the minimum motion compensated difference. Based on our extensive simulation experience, we find that the following features are sufficient for our purpose: (1) body centroid; (2) histograms of horizontal and vertical dimensions, and (3) color histogram. More specifically, after silhouette extraction, we compute the centroid of the foreground pixels. We also scan the foreground image horizontally and vertically and record the

number of pixels in each row and column, as illustrated in Figure 4.1. We refer to this information as histograms of horizontal and vertical dimension. Figure 4.2 (b) and (c) show the histograms of horizontal and vertical dimensions for the silhouette image in Figure 4.2 (a). This provides a simple yet efficient characterization of the body shape and size of the silhouette which implies the distance between the person and the camera. The third feature, color histogram, describes the content inside the object.

(A) (B)

Figure 4.1: Body centroid and dimensions, and histogram of dimensions.

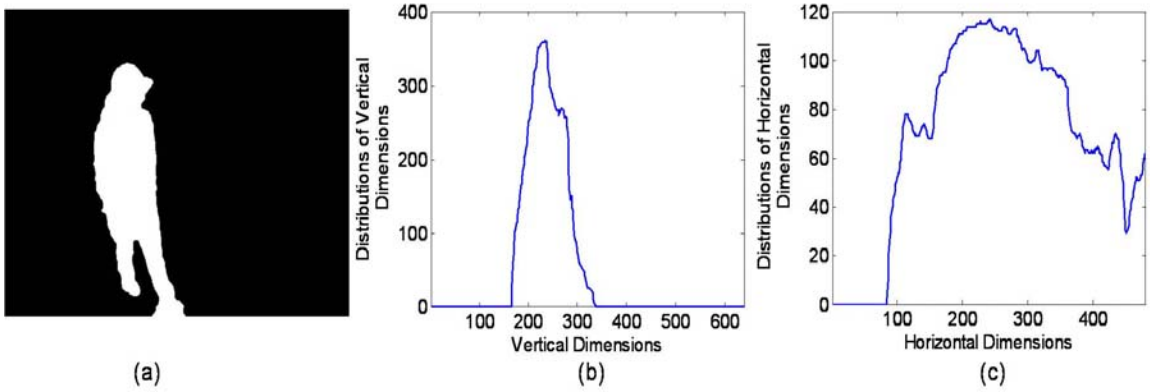


Figure 4.2: (a) Silhouetter image from sequence 2. (b) Distribution of vertical dimensions of image (a). (c) Distribution of horizontal dimensions of image (a).

Let C and R be the object from the current frame and the object to be matched in the working memory. Let $[X_C, Y_C]$ and $[X_R, Y_R]$ be their centroids. Let $p_h(y)$ and $p_v(x)$ be the distributions (or normalized histograms) of horizontal and vertical dimensions, respectively. Here, we use x and y to index horizontal and vertical pixel positions. For the centroid feature, we use their Euclidean distance, denoted by d_c. For the distributions of horizontal and vertical dimensions, we consider them as probability distributions and use the Kullback Leibler distance [47] which is defined as:

$$d_h[p_h^C(y), p_h^R(y)] = \sum_y p_h^C(y) \log \frac{p_h^C(y)}{p_h^R(y)}, \tag{6}$$

$$d_v[p_v^C(x), p_v^R(x)] = \sum_x p_v^C(x) \log \frac{p_v^C(x)}{p_v^R(x)}. \tag{7}$$

We then form the following distance metric for object retrieval from the working memory:

$$d(C, R) = \beta \cdot d_c + (d_h[p_h^C(y), p_h^R(y)] + d_v[p_v^C(x), p_v^R(x)]), \tag{8}$$

where β is a normalization factor on the centroid distance. The object in the working memory with the minimum distance from the current object O_N is the best match $\hat{O}_\Omega$ and used for motion prediction of the current frame.

Using this distance metric, we find the best match object $\hat{O}_{<N}^*$ from the working memory and overlay it on the background image $\hat{B}_{N-1}$ to form the reference frame $\hat{F}_{<N}^*$ for motion

prediction of the current frame F_N. From Figure 4.3 to Figure 4.5, we show the best matching object for video frames of 3 test videos. The top row is the frames to be encoded and the bottom row is the corresponding frames from which the best matching objects are extracted.

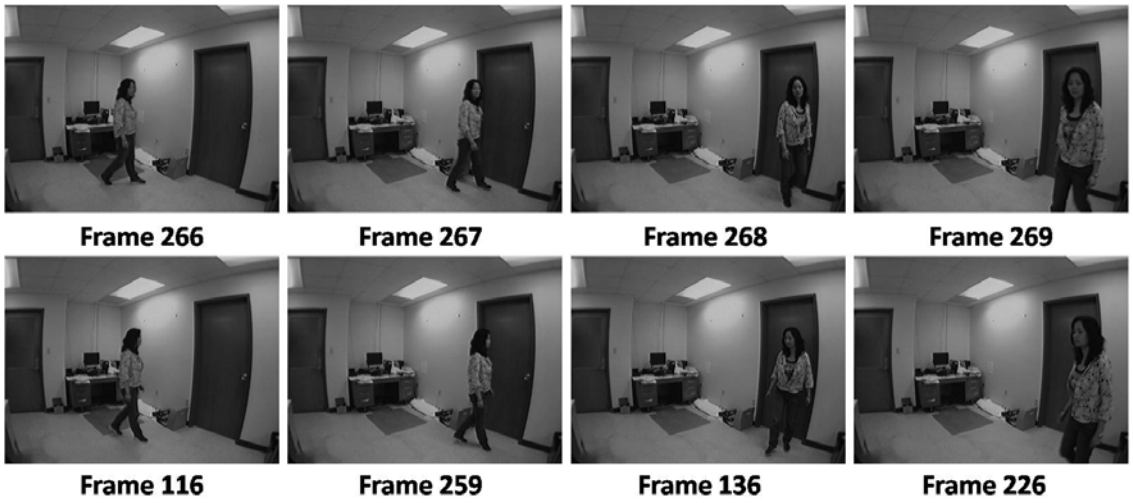


Figure 4.3: Best matching objects for video frame 266-260 of test video 1

Figure 4.4: Best matching objects for video frame 254-257of test video 2

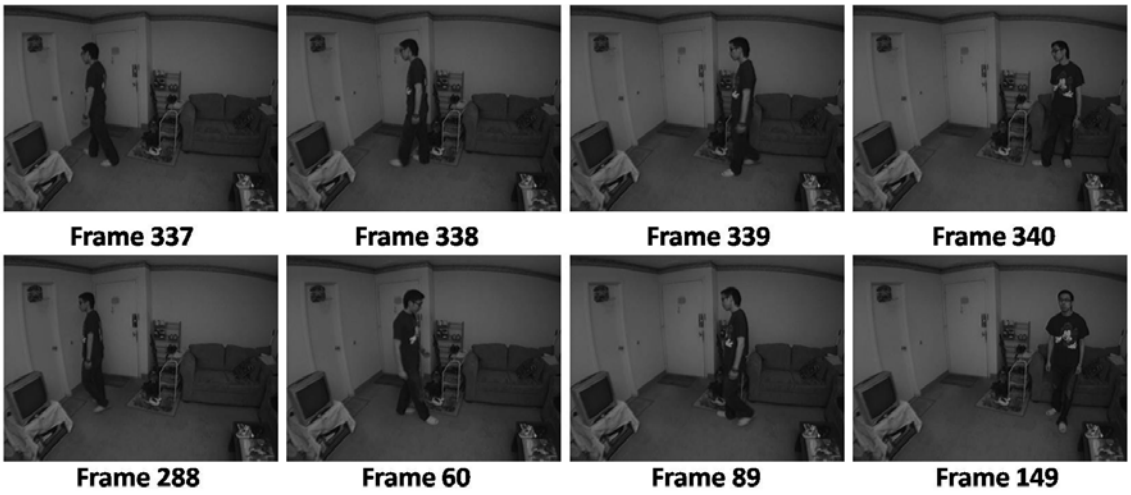


Figure 4.5: Best matching objects for video frame 337-340 of test video 3

4.3 Results and Analysis

We have implemented the proposed working memory prediction scheme in H.264 JM 15.1 [48]. We test the performance of the video encoder within the context of indoor activity monitoring. The three test videos, labeled as *Video_1*, *Video_2*, and *Video_3* are shown in Figure 4.3, Figure 4.4 and Figure 4.5, respectively. We use I and P frames with a GOP (group of pictures) size of 32. We turn off rate control and use a constant quantization parameter for all I and P frames to achieve near-constant video quality. We compare the performance of the following three encoding schemes: (A) conventional H.264 video encoding, (B) H.264 video encoding with optimum prediction which find the best match from all the previous reconstructed frames using brute-force motion search, and (C) H.264 video coding with working memory prediction.

First, we compare the SAD (sum of absolute difference) of the motion compensated difference picture after motion prediction. Figure 4.6 to Figure 4.8 show the SAD of each

frame obtained by these three methods for Video_1, Video_2, and Video_3, respectively. (To show the results clearly, we split the figure into two parts, each showing one half of video frames.) We can see that the SAD obtained by the optimum prediction is much smaller than that of the conventional H.264 motion prediction, and the SAD value obtained by our working memory prediction is very close to the optimum. This implies that the proposed low-complexity feature-based object retrieval scheme is able to accurately find the near-optimum motion match. These results are summarized in Table 4.1.

Next, we compare the encoding bit rates. We set the target video quality to be 35 dB by choosing a similar quantization parameter. Figure 4.9 to Figure 4.11 show the encoding bits of each frame when these three prediction methods are applied. The results are summarized in Table 4.2 and Table 4.3. By using the proposed working memory prediction scheme, we can achieve an average bit rate saving of 25-27%. The maximum bit saving on video frames can even go up to 77.5%. The proposed feature-based object retrieval scheme approaches the optimum performance, only about 0.1-5% of performance loss in bit saving. Figure 4.12 to Figure 4.15 show the rate-distortion (PSNR) comparison between the conventional H.264 video coding and this work on three test videos. We can see that the working memory prediction achieves about 1.2-1.5 dB improvement in average PSNR.

Table 4.1

SAD Comparison.

Video	Average SAD Saving Comparison (SAD/pixel)					Maximal SAD Saving Comparison (SAD/pixel)				
	H.264	Optimum Prediction		This work		H.264	Optimum Prediction		This work	
	SAD	SAD	Saving (%)	SAD	Saving (%)	SAD	SAD	Saving (%)	SAD	Saving (%)
1	8.79	3.61	59.94	4.39	50.08	10.80	1.48	86.29	1.48	86.29
2	8.46	4.65	45.01	5.16	39.02	15.89	4.27	73.15	4.27	73.15
3	4.67	1.84	60.67	3.73	20.10	8.09	0.80	90.08	0.80	90.08

Table 4.2

Bit rate saving in H.264 video coding

Video	Bit Rate Saving (%) from H.264			
	Optimum		Feature-based	
	Avg	Max	Avg	Max
1	32.5%	77.5%	27.2%	77.5%
2	30.8%	71.4%	25.7%	71.4%
3	16.0%	73.6%	25.9%	72.0%

Table 4.3

Bit rate comparison in H.264 video coding

Video	Bit Rate Comparison from H.264 (kbits/s at 30 Hz)	
	Optimum	Feature-based
1	265.99	182.70
2	462.68	326.84
3	163.40	126.50

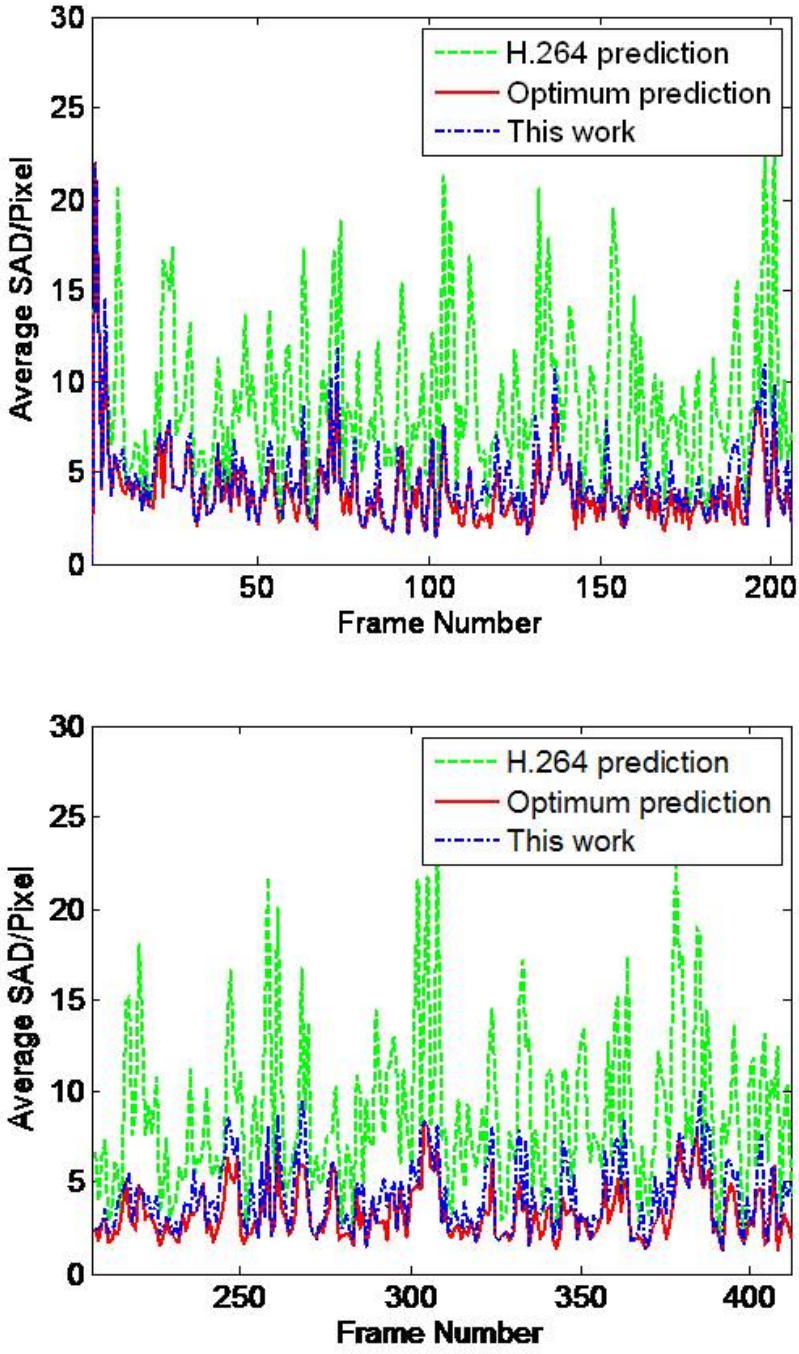


Figure 4.6: Average residual SAD comparison on Video_1 with H.264 motion prediction and optimum search and the proposed algorithm in this work.

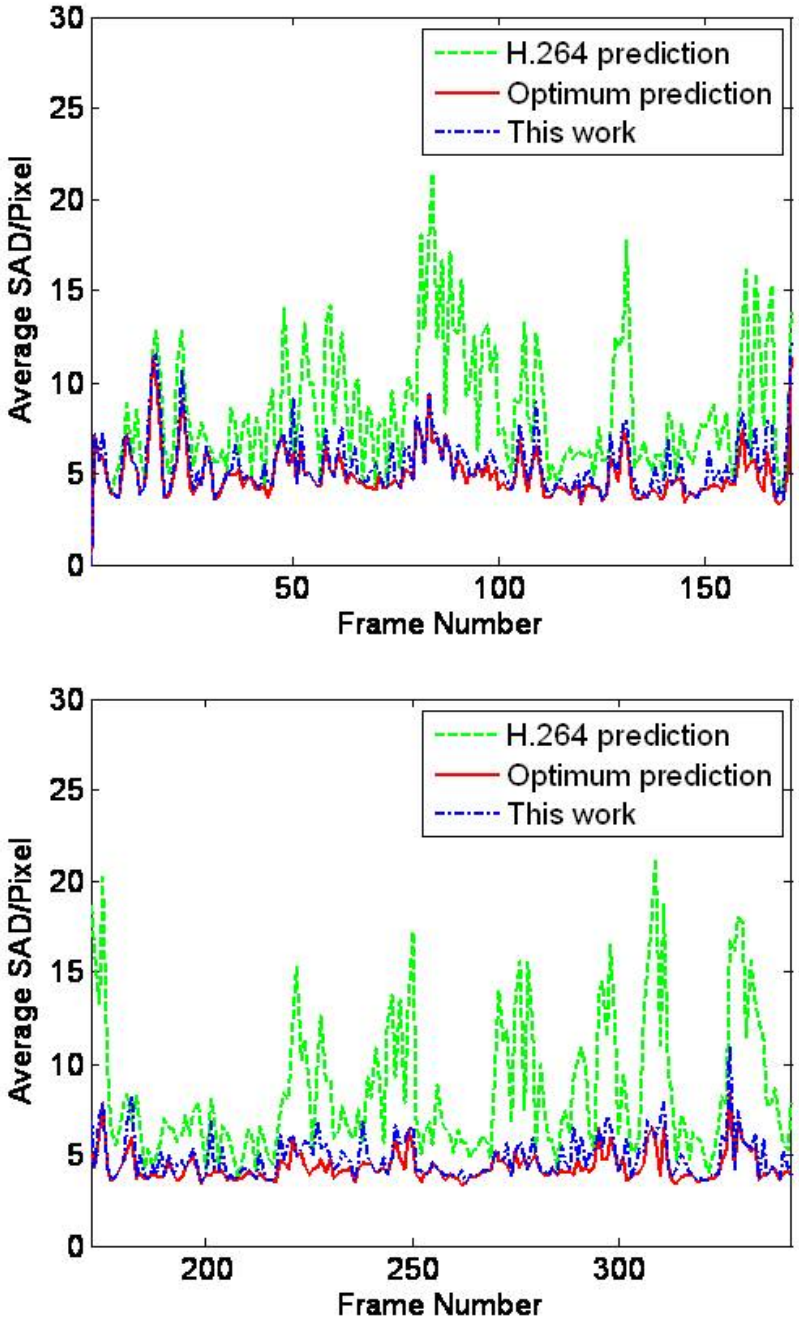


Figure 4.7: Average residual SAD comparison on Video_2 with H.264 motion prediction and optimum search and the proposed algorithm in this work.

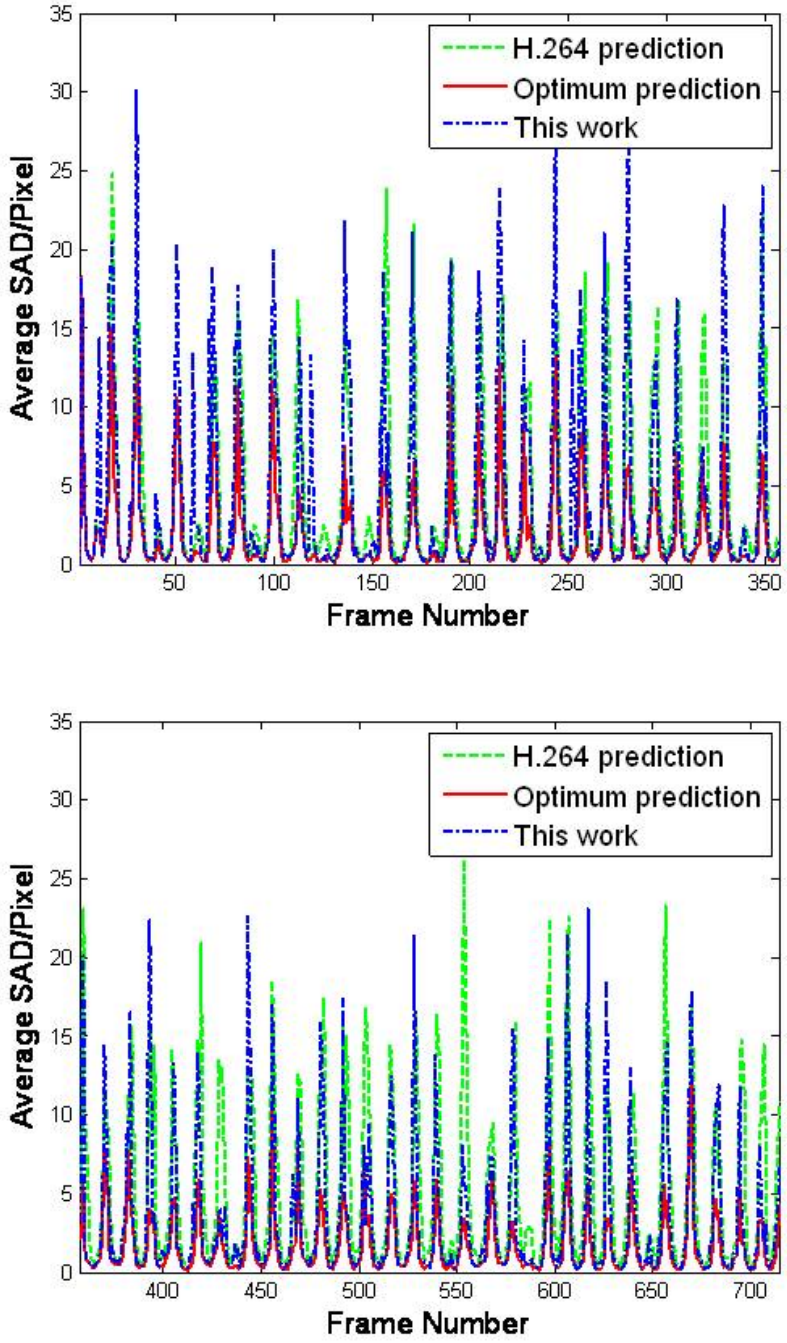


Figure 4.8: Average residual SAD comparison on Video_3 with H.264 motion prediction and optimum search and the proposed algorithm in this work.

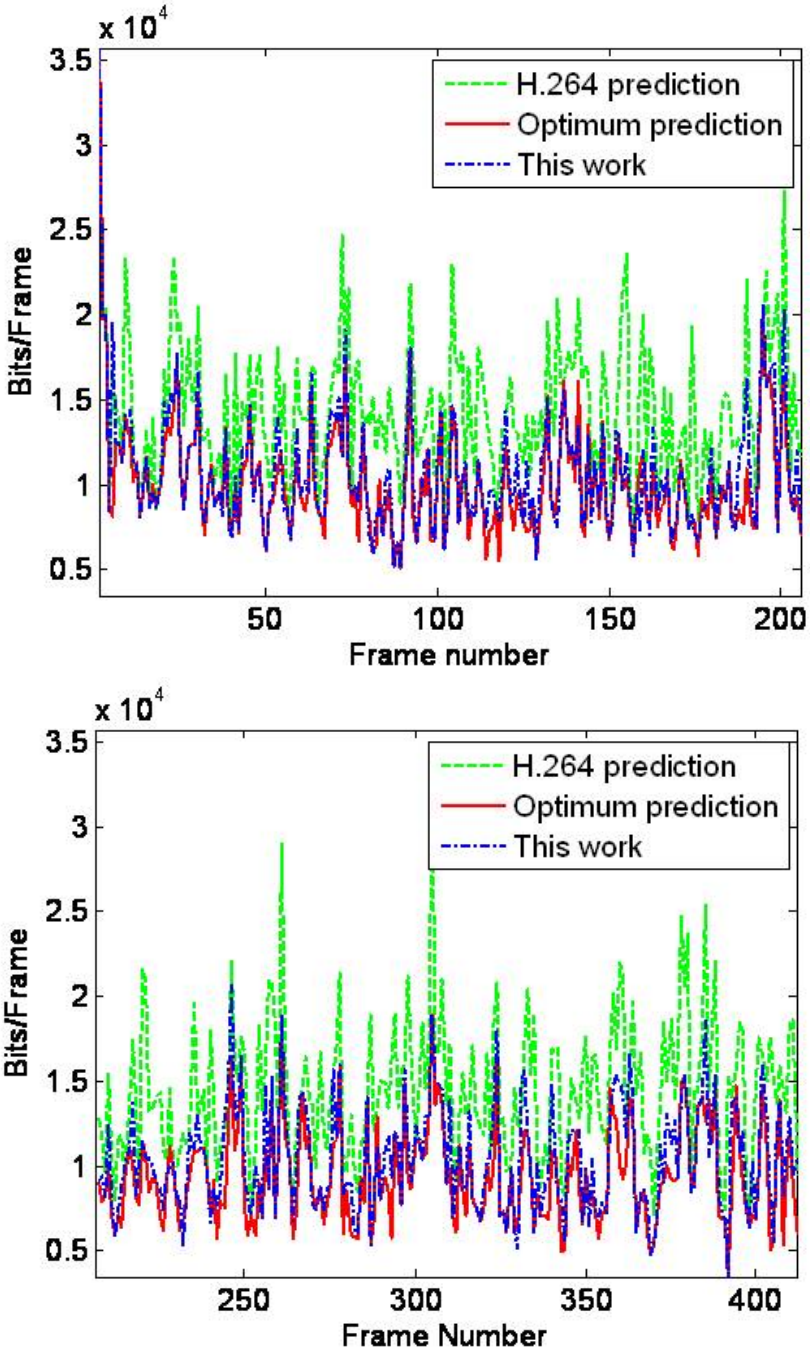


Figure 4.9: H.264 encoding bits rate comparison on Video_1 with H.264 motion prediction, optimum search, and the proposed algorithm in this research.

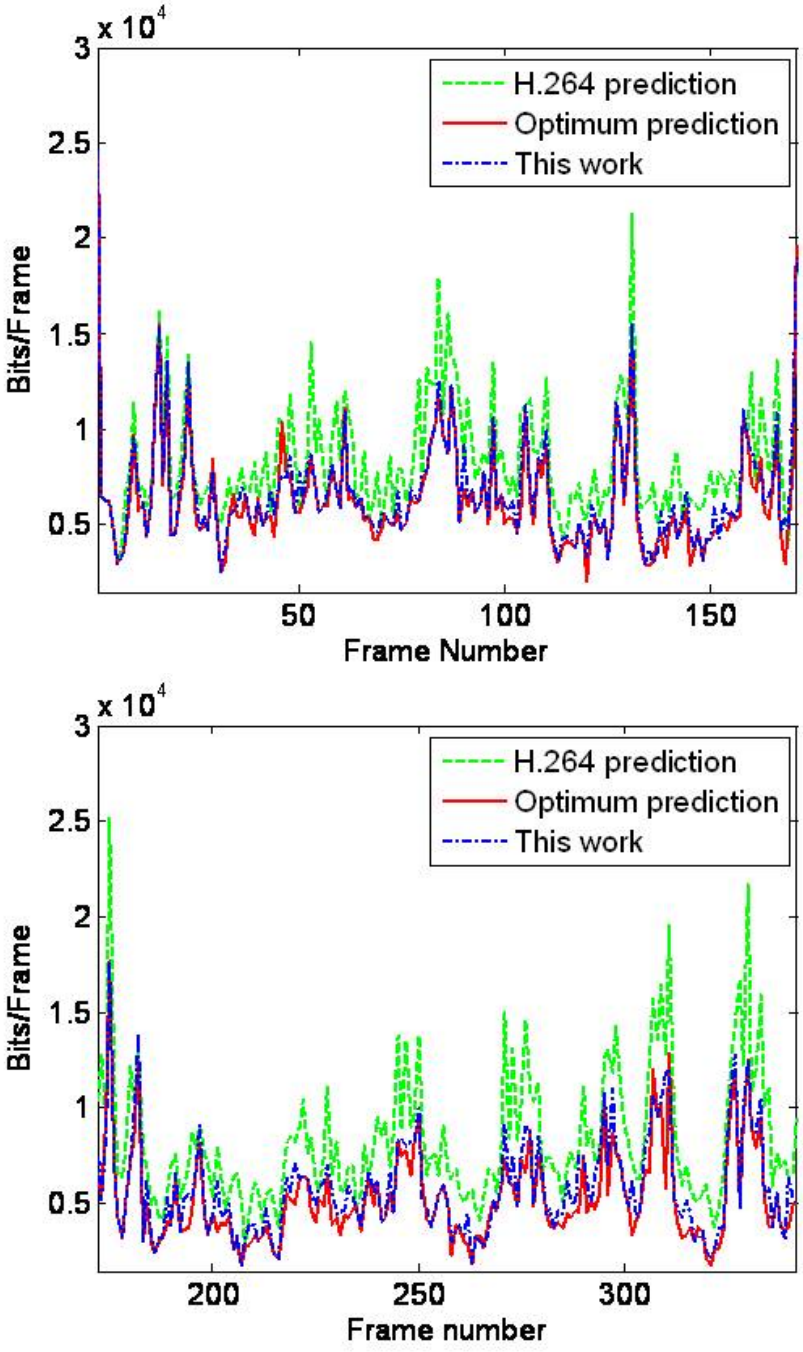


Figure 4.10: H.264 encoding bits rate comparison on Video_2 with H.264 motion prediction, optimum search, and the proposed algorithm in this research.

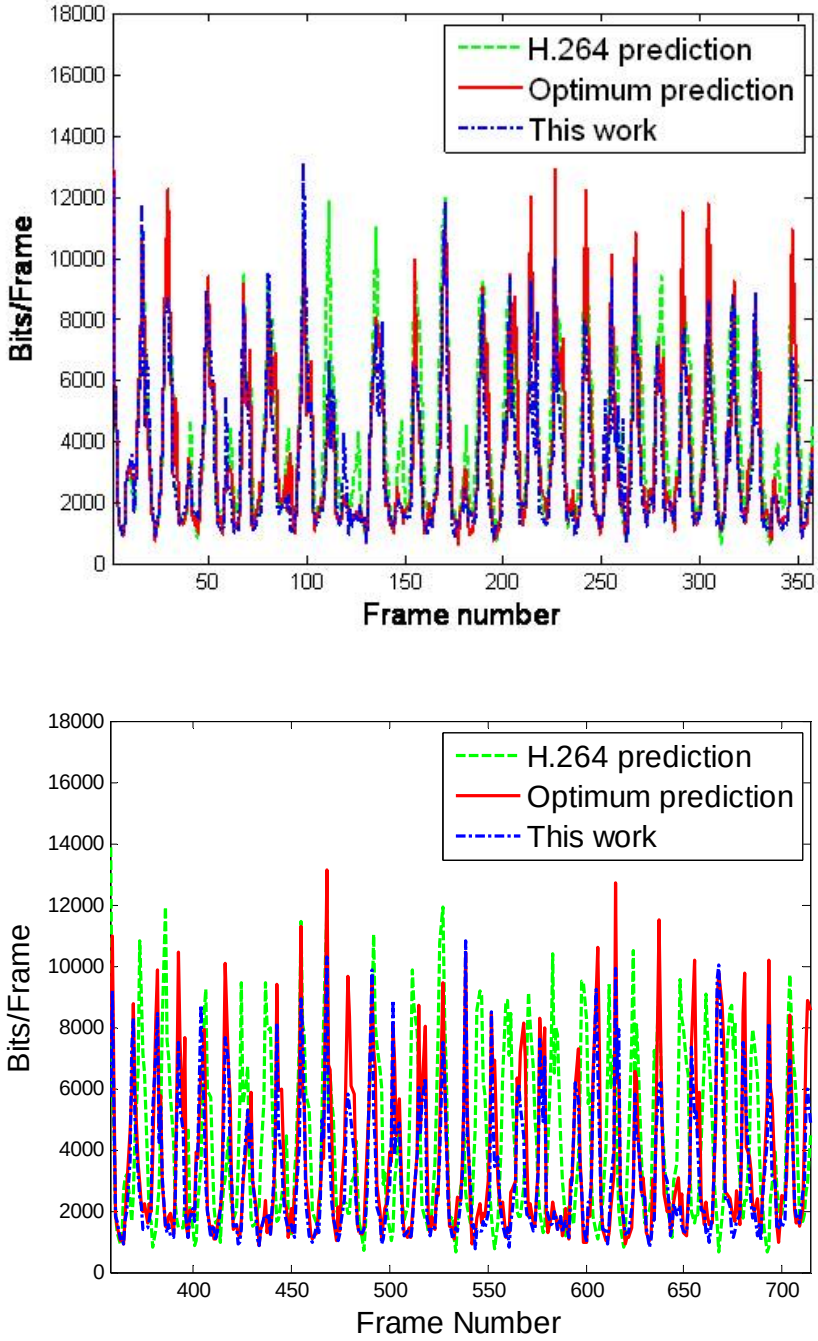


Figure 4.11: H.264 encoding bits rate comparison on Video_3 with H.264 motion prediction, optimum search, and the proposed algorithm in this research.

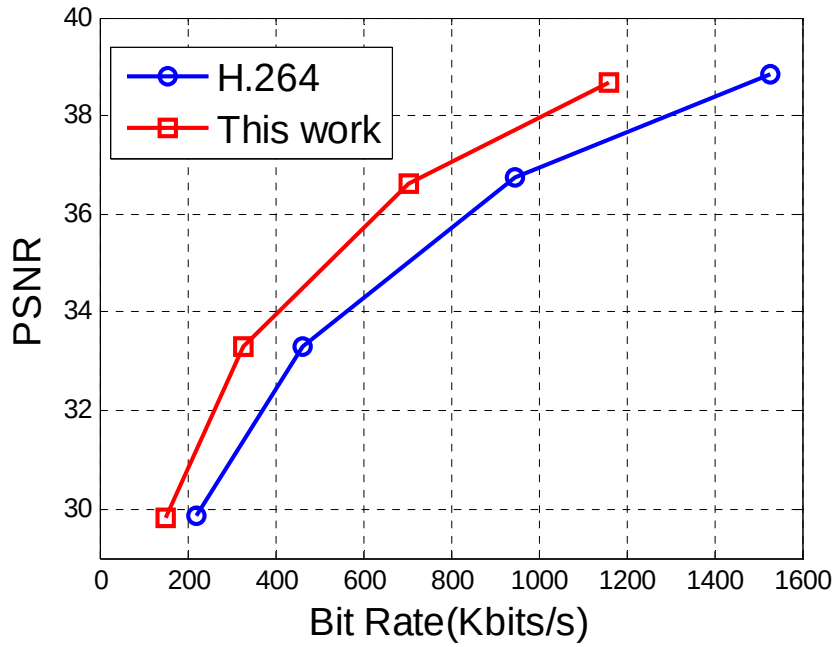


Figure 4.12: Rate-distortion performance comparison with conventional H.264 video coding on Video_1

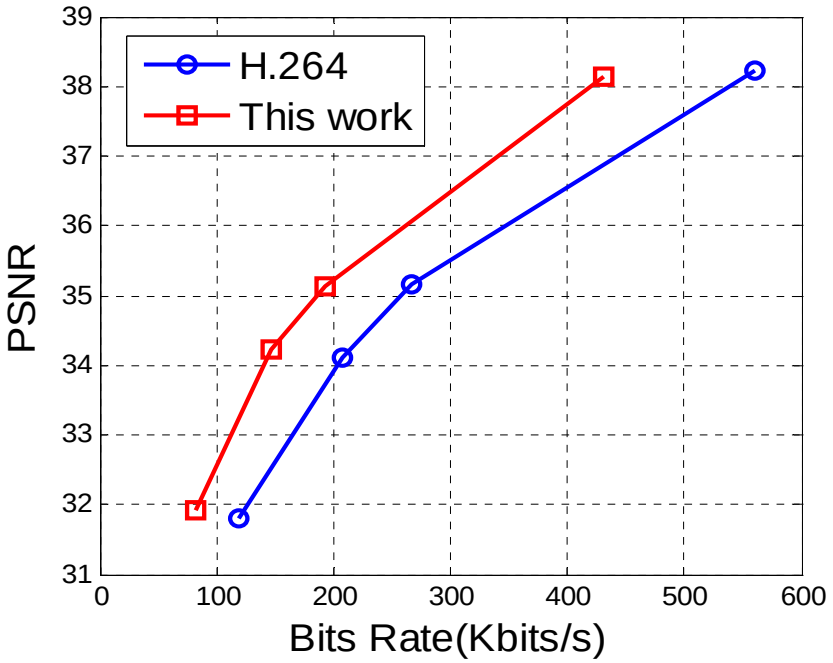


Figure 4.13: Rate-distortion performance comparison with conventional H.264 video coding on Video_2

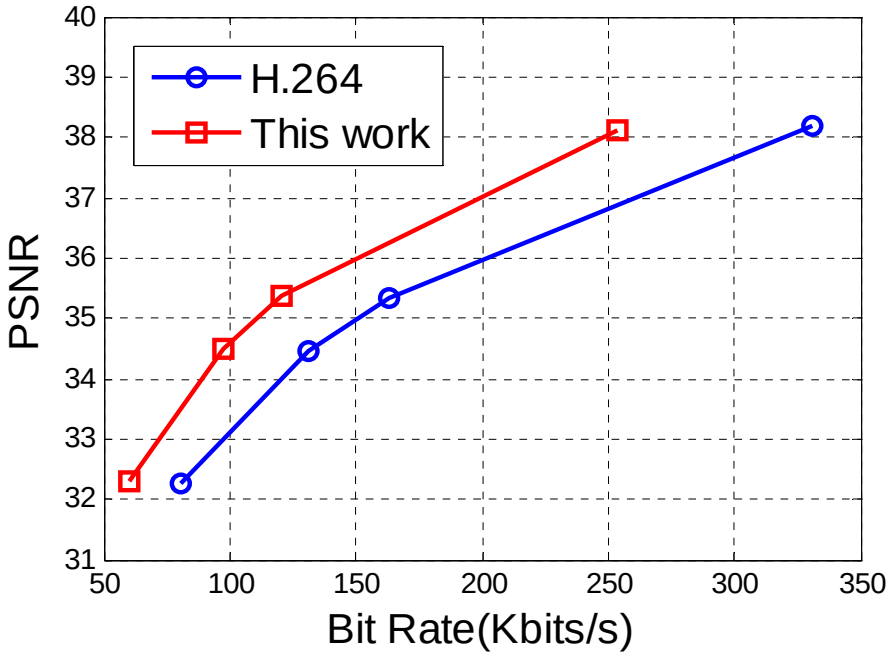


Figure 4.14: Rate-distortion performance comparison with conventional H.264 video coding on Video_3

Chapter 5

Working Memory Management

5.1 Overview

As more video frames are being encoded, more objects are being added into the working memory. This will require larger memory and higher implement cost. It will also increase the computational complexity in object retrieval. Therefore, there is a need to develop an efficient working memory management scheme to control the memory cost and complexity of the working memory. More specifically, we need to control the total number of objects in the working memory.

Our proposed scheme for working memory management is based on the following observations. First, there is no need to store objects that are very similar to each other. Second, objects maintained in the memory should cover different human actions, poses, or appearances as many as possible. They should be quite different from each other. Therefore, we propose to use the distance metric in (8) for dynamic working memory management. More specifically, when a new frame is being encoded, we retrieve the best

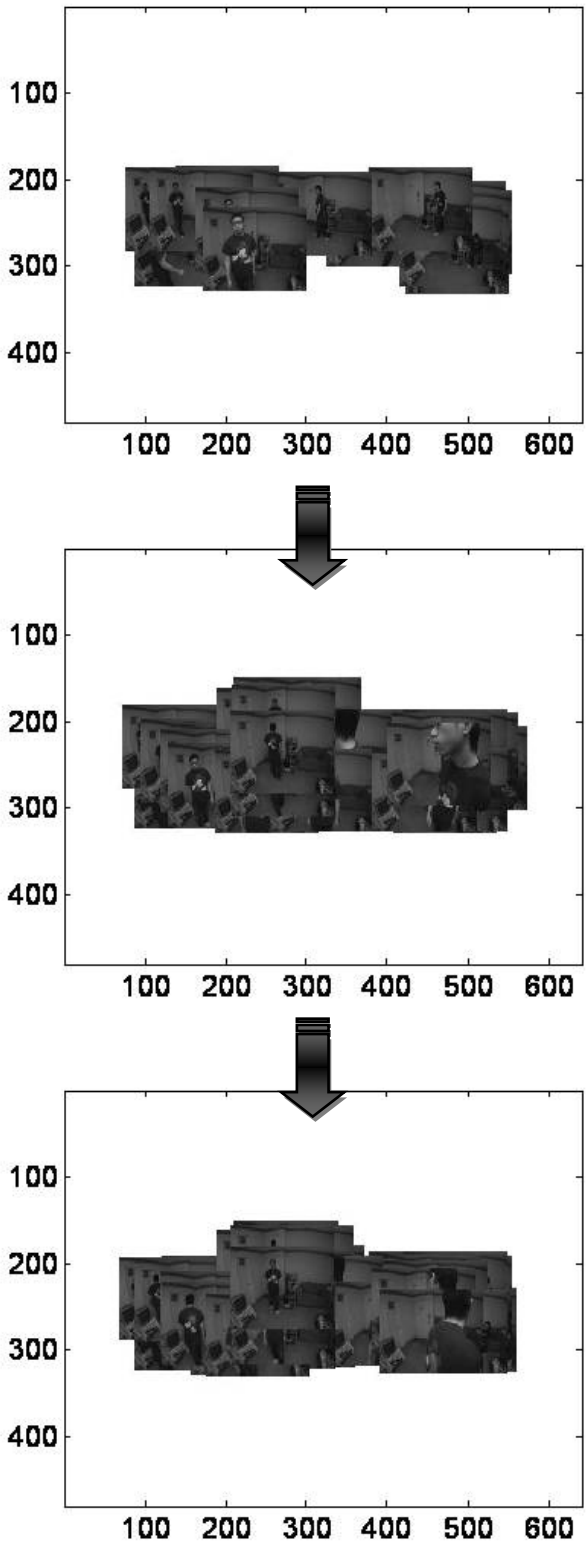


Figure 5.1: Illustration for work memory

matching from the memory for motion prediction. Therefore, we know the distance between the current object and each object in the working memory. In this way, we always have the distance between any two objects in the working memory. When the current object joins the working memory, we remove the object which has the smallest distance from other objects. Figure 5.1 shows an example where a maximum of 50 objects are being maintained in the working memory at 3 different time instances.

5.2 H.264 Video Coding with Object Retrieval and Matching

In this Section, we describe the H.264 video encoding scheme based on working memory prediction. Figure 5.2 shows a block diagram of the modified H.264 video encoder. The proposed encoding scheme has the following major steps. Let F_N be the current video frame.

Step 1. ***Silhouette extraction and foreground object detection***. Apply the silhouette extraction algorithm described in Section III to the current frame F_N and obtain the foreground object O_N.

Step 2. ***Object retrieval from the working memory***. From the foreground object O_N, we extract its features, namely, its centroid and histograms of horizontal and vertical dimensions. Using the distance metric defined in (7), we find the retrieve the best match object $\hat{O}_\Omega$ from the working memory Ω.

Step 3. ***Constructing the motion prediction reference***. Overlay the object $\hat{O}_\Omega$ on top of the background image $\hat{B}_{N-1}$ obtained from silhouette extraction of the

reconstructed frames to construct a working memory prediction $\hat{F}_{\Omega}$.

Step 4. ***H.264 encoding***. Using $\hat{F}_{\Omega}$ as the reference frame for motion prediction, encode the residual with the H.264 encoder, and reconstruct the current frame.

Step 5. ***Working memory management***. Using the silhouette as mask to extract the reconstructed foreground object $\hat{O}_N$. Extract its features $\hat{\boldsymbol{f}}_N$ and store them into the working memory. Using the procedure described in Section V to update the working memory.

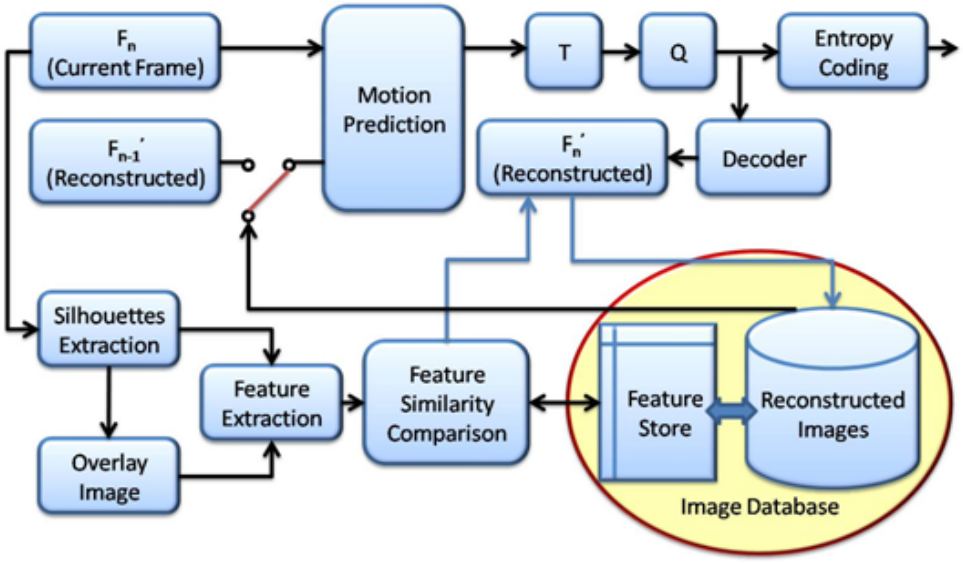


Figure 5.2: Overview of H.264 video encoding with working memory prediction

The major computational complexity of the proposed algorithm lies in silhouette extraction. Our current silhouette extraction algorithm is able to run 10-15 frames per second on 640×480 images. The feature-based matching process has low computational complexity, especially when the number of objects stored in the working memory is small. For example, in this work, we set the number in the range of [50, 300],

which is typically enough to cover the major actions, poses, or states of the human activities in the indoor environment. According to our estimation, the overhead computational complexity introduced by the proposed working memory prediction is less than 10% of the H.264 video encoding.

5.3 Results and Analysis

In Figure 5.3, we choose five different sizes of the working memory: 50, 100, 150, 200, and 713. The total number of frames is 1400. We can see that, as we increase the working memory size, the video quality is improved. In a typical setting, a working memory size of 100 objects will be sufficient and the performance loss is less than 0.3 dB.

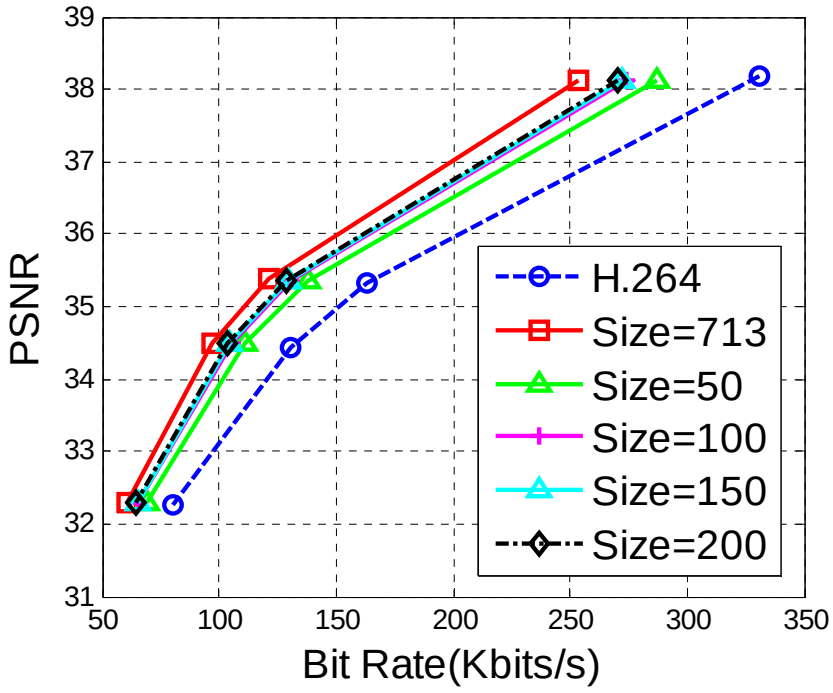


Figure 5.3: Experimental results with different sizes of working memory. This result is average on three videos.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this work, we have developed a working memory approach for efficient temporal prediction and H.264 video coding. We extract objects from the reconstructed video frames to form a knowledge base, just like us being able to remember persons that have appeared before when watching a movie. We developed a set of features for fast and accurate object retrieval. This approach extends the multiple-frame estimation and provides a more generic framework for spatiotemporal prediction of video data. Our experimental results on surveillance video data demonstrate that the proposed approach is able to save the coding bit rate by up to 35% with a small computational overhead.

6.2 Future Work

The work introduces a new concept for video prediction, using the past observation to construct a synthesized reference frame for more efficient motion prediction. The working memory can be considered as a knowledge base and the proposed approach can

be further extended to knowledge-based video encoding, which might provide a better approximation of the human visual system. In our next step, we shall explore more advanced computer vision methods, such as multi-person detection and tracking, and extend the proposed method for video scene with multiple persons. We would also like to investigate how the proposed method performs on other types of video data, such as sports and movies.

Reference

[1] T. Wiegand, G. J. Sullivan, G. Bjntegaard, A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans on Circuits Syst. Video Technolog.,* vol. 13, no. 7, pp. 560-576, 2003.

[2] Z. Zhou, X. Chen, Y. Chung, X. Han, J. Keller, and Z. He, “Activity analysis, summarization, and visualization for eldercare,” *IEEE Trans on Circuits Syst. Video Technolog.*, vol. 18, no. 11, pp. 1489-1498, 2008.

[3] Y. Su and M. T. Sun, “Fast multiple reference frame motion estimation for H.264/AVC,” *IEEE Trans on Circuits Syst. Video Technolog.,* vol. 16, no. 3, pp. 447-452, 2006.

[4] T. Kuo and H. Lu, “Long-term memory motion-compensated prediction,” *IEEE Trans on Circuits Syst. Video Technolog.,* vol. 18, no. 3, pp. 400-405, 2008.

[5] Y. Huang, B. Hsieh, S. Chien, S. Ma, and L. Chen, “Analysis and complexity reduction of multiple reference frame motion estimation in H.264/AVC,” *IEEE Trans on Circuits Syst. Video Technolog.,* vol. 16, no. 4, pp. 507-522, 2006.

[6] J. M. Boyce, “Weighted prediction in the H.263/MPEG AVC video coding standard,” in *Proc. IEEE Int. Symp. Circuits Systems*, 2004, pp. III - 789-92.

[7] S. Lin, C. Chang, C. Su, Y. Lin C. Pan, and C. Chen, “Fast multi-frame motion estimation and mode decision for H.264 encoders,” *IEEE Int. Conf. Wireless Networks, Communication and Mobile Computing*, 2005, pp. 1237-1242.

[8] Y. Hsiao, T. Lee, and P. Chang, “Short/long-term motion vector prediction in multi-frame video coding system,” *IEEE Int. Conf. Image Processing*, 2004, vol.3, pp. 1449-1452.

[9] C. Daunmu, M. O. Ahmad, and M. N. S. Swamy, “A continuous tracking algorithm for long-term memory motion estimation [video coding],” in *Proc. IEEE Int. Symp. Circuits Systems*, 2003, pp. II-356 - II-359.

[10] T. Wiegand, X. Zhang, and B. Girod, “Long-term memory motion-compensated prediction,” *IEEE Trans on Circuits Syst. Video Technolog.*, vol. 9, no. 1, pp. 70-84, 1999.

[11] X. Li, E Q. Li, and Y. Chen, “Fast multi-frame motion estimation algorithm with adaptive search strategies in H.264,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2004, vol.3, pp. iii - 369-72.

[12] Y. Huang, B. Hsieh, T. Wang, S. Chient, S. Ma, C. Shen, and L. Chen, “Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2003, vol.3, pp. III - 145-8.

[13] L. Shen, Z. Liu, Z. Zhang, and X. Shi, “An adaptive and fast H.264 multi-frame selection algorithm based on information from previous searches,” in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2007, pp. 1591 - 1594.

[14] H. Wang, L. Wang, and H. Li, “A fast multiple reference frame selection algorithm based on H.264/AVC,” *Third IEEE Int. Conf. Intelligent information Hiding and Multimedia Signal Processing*, 2007, pp. 525 - 528.

[15] J. Sohn and D. Kim, “Fast multiple reference frame selecation method using correlation of sequence in JVT/H.264,” *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. E89-A, pp. 744-746, 2006.

[16] S. Kapotas, and A. Skodras, “Fast multiple reference frame selection method in H.264 video encoding,” *Picture Coding Symposium*, 2007.

[17] N. Chang and K. Fu, “Query-by-Pictorial-Example,” *IEEE Trans. Software Engineering*, vol. SE-6, pp. 519-524, 1980.

[18] S. Chang and A. Hsu, “Image information systems: where do we go from here?” *IEEE Trans. Knoledge and Data Engineering*, vol. 4, pp. 431-442, 1992.

[19] X. Bao,” Image Retrieval Technologies: A Survey,” Oregon State University.

[20] V. Ogle and M. Stonebraker, “Chabot: retrieval from a relational database of images,” *IEEE Computer*, vol. 28, pp. 40-48, 1995.

[21] M. Swain and D. Ballard, “Color indexing,” *Int. Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.

[22] M. Swain and D. Ballard, “Similarity of color images,” in *Proc. SPIE Storage and Retrieval for Image and Video Database*, vol. 2, 1995, pp. 381-392.

[23] J. Smith and S. Chang, “Single color extraction and image query,” in *Proc. IEEE Int. Conf. Image Processing*, 1995, vol.3, pp. 528-531.

[24] J. Smith and S. Chang, “Tools and techniques for color image retrieval,” in *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Database IV*, vol. 2670, 1996.

[25] R. Haralick and K. Shanmugam, “Texture features for image classification,” *IEEE Trans. Sys., Man and Cyb.*, vol. 3, pp. 610-621, 1973.

[26] C. Gotlieb and H. Kreyszig, “Texture descriptors based on co-occurrence matrices,” *Computer Vision, Graphics, and Image Processing*, vol. 51, pp. 70-86, 1990.

[27] H. Tamura, S. Mori, and T. Yamawaki, “Textural features corresponding to visual perception,” *IEEE Trans. Sys., Man and Cyb.*, vol.8, pp.460-473, 1978.

[28] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, “Query by image and video content: the QBIC system,” *IEEE Computer*, vol. 28, pp. 23-32, 1995.

[29] T. Huang, S. Mehratra, and K. Ramchandran, “Multimedia Analysis and Retrieval System (MARS) project”, in *Proc. the 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval* , University of Illinois at Urbana-Champaign, 1996.

[30] M. Ortega, Y. Rui, K. Chakrabarti, S. Mehrotra, and T. Huang, “Supporting similarity queries in MARS”, in *Proc. the 5th ACM Int. Multimedia Conference*, Seattle, Washington, 1997, PP. 403-413.

[31] J. M. Smith and S. Chang, “Automated image retrieval using color and texture,” Technical Report CU/CTR 408-95-14, CTR, Columbia University, July, 1995.

[32] J. M. Smith and S. Chang, “Automated binary texture features sets for image retrieval,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 1996, vol.4, pp. 2239-2242.

[33] Y. Rui, A. She, and T. Huang, "Modified fourier descriptors for shape representation -- A practical approach", in *Proc. the 1st Int. Workshop on Image Databases and Multi Media Search*, Amsterdam, Netherlands, 1996.

[34] P. Huang, "Indexing pictures by key objects for large-scale image databases", *Pattern Recognition*, vol.30, pp. 1229-1237, 1997.

[35] M. Swain, C. Frankel, and V. Athitsos, “WebSeer: An image search engine for the world wide web,” , in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Amsterdam, Netherlands, 1997.

[36] R. Rickman and J. Stonham, “Content-based image retrieval using colour tuple histograms”, in *Proc. SPIE Storage and Retrieval for Still Image and Video Database IV*, vol. 2670, 1996, pp2-7.

[37] J. Huang, S. Kumar, M. Mirtra, W. Zhu, and R. Zabih, “Image indexing using color correlograms,” ", in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, San Juan, Pueto Rico, 1997, pp. 762-768.

[38] Ning, I., Harwood, D., Davis, L. S. (2000). W4: real-time surveillance of people and their activities, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, Aug., pp.809–830.

[39] T. Horprasert, D. Harwood, L. Davis, “A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection”, *Proc. IEEE ICCV’99 FRAME_RATE WORKSHOP*, Kerkyra, Greece, September 1999.

[40] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in Proc. CVPR, pp. 142-151, 2000.

[41] Dong Xu, Jianzhuang Liu, Xuelong Li, Zhengkai Liu, and Xiaoou Tang, “Insignificant Shadow Detection for Video Segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 1058-1064, 2005.

[42] T. Horprasert, D. Harwood, L. Davis, “A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection”, *Proc. IEEE ICCV’99 FRAME_RATE WORKSHOP*, Kerkyra, Greece, September 1999.

[43] Haritaoglu, I.; Harwood, D.; Davis, L. S. (2000). W4: real-time surveillance of people and their activities, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, Aug., pp.809–830.

[44] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, “Pfinder: Real-Time Tracking of the Human Body,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780-785, July 1997.

[45] F. Lv, J. Kang, R. Nevatia, I. Cohen and G. Medioni, “Automatic Tracking and Labeling of Human Activities in a Video Sequence,” IEEE Int’l Workshop on Performance Evaluation of Tracking and Surveillance, May 2004.

[46] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” *IEEE Trans. Image Processing*, vol. 9, pp. 287- 290, Feb. 2000.

[47] Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, 2nd Edition, Wiley-Interscience, 2006.

[48] H.264/AVC JM Reference: http://iphome.hhi.de/suehring/tml/.

Publications

[1] W. Dai, F. Wu, and Z. He, “Overview of the H.264/AVC video coding standard,” submitted to *IEEE Trans on Circuits Syst. Video Technolog.*, 2009.

[2] W. Dai, Y. Sun and Z. He, "Efficient H.264 video coding with a working memory of objects," *Picture Coding Symposium*, Chicago, IL, May 6-8, 2009, pp.1-4.

[3] Z. Zhou, W. Dai, J. Eggert, J. Giger, J. Keller, M. Rantz, and Z. He, “A real-time system for in-home activity monitoring of elders,” in *Proc. IEEE Aunual Int. Conf. Engineering in Medicine and Biology Society*, 2009, pp. 6115-6118.

[4] F. Wang, E. Stone, W. Dai, T, Banerjee, J. Giger, J. Krampe, M. Rantz, and M. Skubic, “Testing an in-home gait assessment tool for older adults”, in *Proc. IEEE Aunual Int. Conf. Engineering in Medicine and Biology Society*, 2009, pp. 6147-6150.

[5] F. Wang, E. Stone, W. Dai, M. Skubic, and J. Keller, “Gait analysis and validation using voxel data”, in *Proc. IEEE Aunual Int. Conf. Engineering in Medicine and Biology Society*, 2009, pp. 6127-6130.