NEW CHANGE DETECTION MODELS FOR OBJECT-BASED ENCODING OF PATIENT MONITORING VIDEO

by

Qiang Liu

B.S. in E.E., Xidian University, China, 1996

M.S. in E.E., Xidian University, China, 1999

Submitted to the Graduate Faculty of

the School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2005

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This dissertation was presented

by

Qiang Liu

It was defended on

April 8, 2005

and approved by

Robert J. Sclabassi, Ph.D., M.D., Professor

Mingui Sun, Ph.D., Associate Professor

Ching-Chung Li, Ph.D., Professor

J. Robert Boston, Ph.D., Professor

Luis F. Chaparro, Ph.D., Associate Professor

Jie Yang, Ph.D., Research Scientist, Computer Science, Carnegie Mellon University

Dissertation Advisors: Robert J. Sclabassi, Ph.D., M.D., Professor,

Mingui Sun, Ph.D., Associate Professor

NEW CHANGE DETECTION MODELS FOR OBJECT-BASED ENCODING OF PATIENT MONITORING VIDEO

Qiang Liu, PhD

University of Pittsburgh, 2005

The goal of this thesis is to find a highly efficient algorithm to compress patient monitoring video. This type of video mainly contains local motion and a large percentage of idle periods. To specifically utilize these features, we present an object-based approach, which decomposes input video into three objects representing background, slow-motion foreground and fast-motion foreground. Encoding these three video objects with different temporal scalabilities significantly improves the coding efficiency in terms of bitrate vs. visual quality.

The video decomposition is built upon change detection which identifies content changes between video frames. To improve the robustness of capturing small changes, we contribute two new change detection models. The model built upon Markov random theory discriminates foreground containing the patient being monitored. The other model, called covariance test method, identifies constantly changing content by exploiting temporal correlation in multiple video frames. Both models show great effectiveness in constructing the defined video objects. We present detailed algorithms of video object construction, as well as experimental results on the object-based coding of patient monitoring video.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I am deeply grateful to my advisors Robert J. Sclabassi, Ching-Chung Li, and Mingui Sun. It is their supports and guidances that have led me through the graduate study. I express my special thanks to Dr. Sun, who has been not only an advisor but also a friend. His creative thinking enlightens me on the whole research.

I thank other members on my committee - Dr. J. Robert Boston, Dr. Luis F. Chaparro and Dr. Jie Yang, for their insightful comments and valuable suggestions on my thesis. I also would like to thank Dr. Scheuer for granting the data collection access to the epilepsy monitoring unit.

Thanks to my fellow colleagues Lin-sen Pon, Rafael E. Herrera, Prophete J. Charles, Gusphyl Justin, Chengquan Hu, Bing Liu, Jian Xu, Paul Roche, Vijay Venkatraman for helping me with the research and for making such an enjoyable environment.

I thank my wife Zheng Ma for her love and support, for her patience in listening to my research story and for her suggestions that have accelerated my study. I thank my parents for their trust and love. I am so lucky to be their son. All the work is dedicated to my parents.

1.0 INTRODUCTION

This dissertation presents an *object-based* approach built upon novel *change detection* models to encode *patient monitoring video* (Fig. 1). In the following, we present why object-based coding benefits compression of patient monitoring video and how change detection approach can be designed and employed to construct video objects.

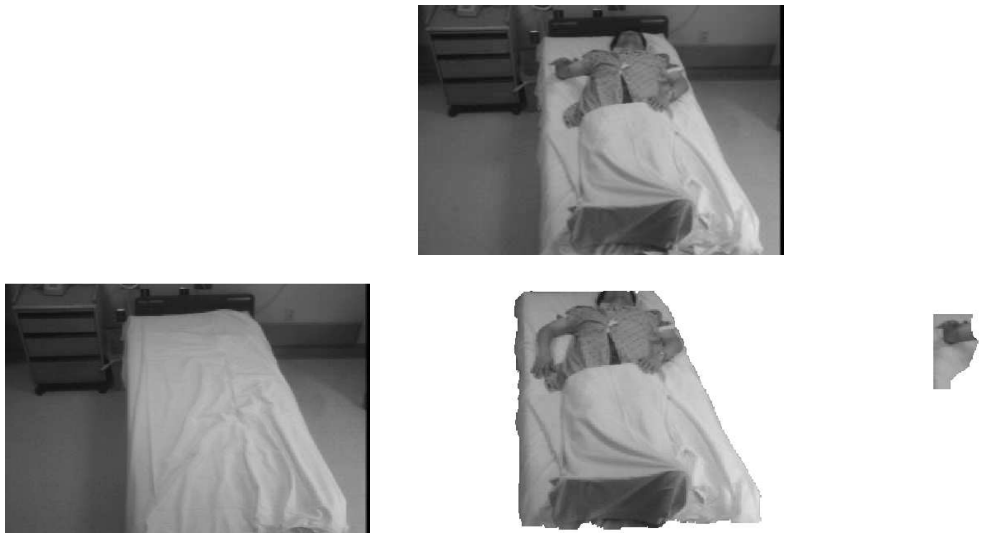

Figure 1: Patient monitoring video and object-based representation.

1.1 VIDEO REPRESENTATION VIA VIDEO OBJECTS

How information is processed depends on how it is represented. Because of this reason, video representation has long been a fundamental issue in computer vision community. In the past fifteen years, representing video as an ensemble of video objects has been actively studied. In this ensemble, a video object is defined as a set of pixels that share common semantic features in an image sequence. Primarily, this concept may build up a bridge that connects a pixel-based video processing system to a high-level image understanding system. The potential applications that are benefited from this concept include:

- Video coding — representation via video objects enables content-driven coding schemes that not only achieve higher compression, but also distinguish relevant features in the compressed bit stream.
- Video editing — segmenting a scene into video objects facilitates the manipulation of video contents. Separate objects can be assembled on-the-fly to form synthetic video streams.
- Multimedia database — object based representation also enables intelligent database search. Distributed storage of multimedia data is advanced.
- Copyright protection — decomposition of multimedia into objects may ease authorization of interactive or personalized content.

In summary, the regularity accommodated in video can be reflected by video objects collected in it. When represented in an object-based manner, the information contained in video is organized feature-wise so that higher level tasks can take advantage of it and provide more advanced facilities.

1.2 OBJECT-BASED VIDEO CODING

MPEG-4 standard [1, 2, 8] emerges as an immediate application of object-based video representation. This standard defines a video frame in terms of components. Each component,

called a video object plane (VOP), consists of a snapshot of a video object. Each VOP can be treated separately in a coding/decoding session.

This strategy is highlighted in Fig. 2 at a system level for an MPEG-4 encoder (top) and decoder (bottom). An input video frame is first decomposed into VOPs. Each VOP is encoded individually with a number of flexible choices, such as temporal, spatial, and quality scalabilities. The resulting elementary bit streams are multiplexed into a single bitstream in accordance with a well-designed protocol for transmission or storage. At the reception end, this bit-stream is demultiplexed and decoded. The composition unit combines the decoded VOPs and reconstructs the original video frames.

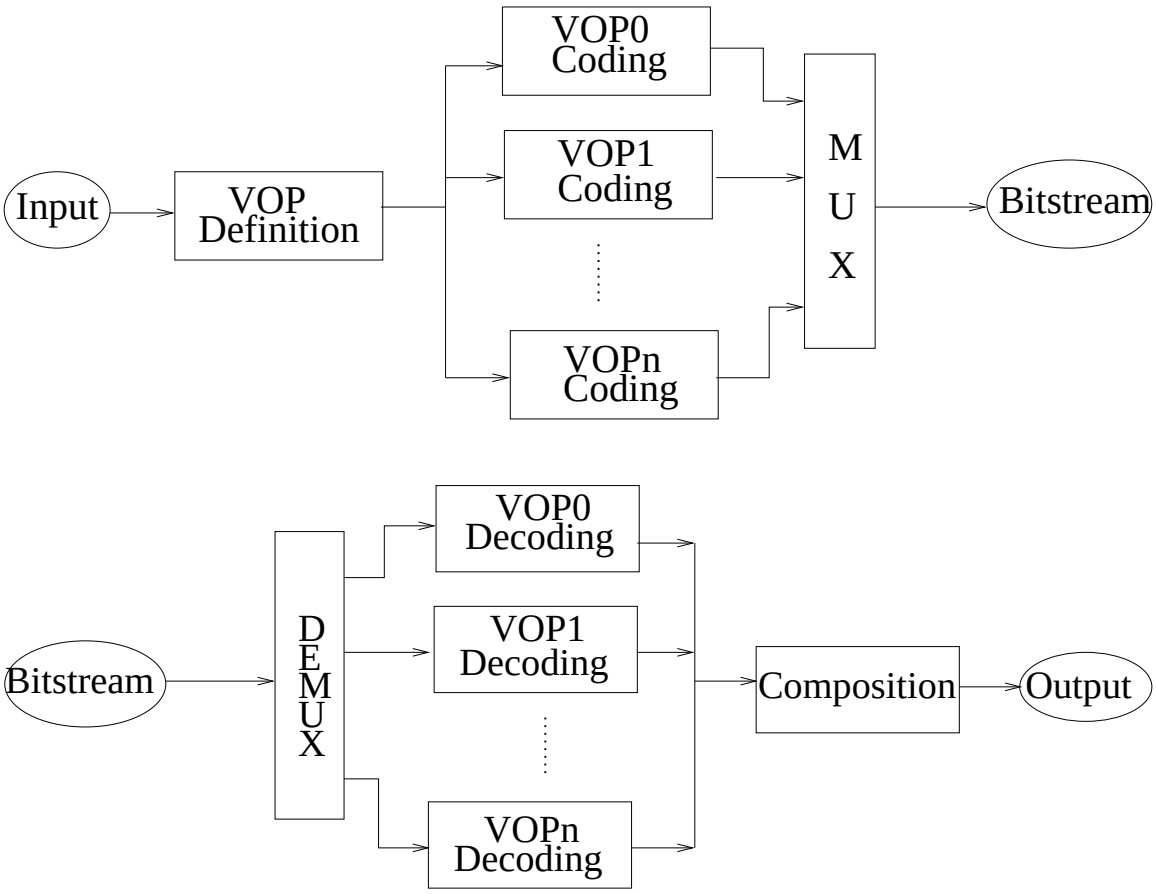

Figure 2: VOP based MPEG-4 encoder (top panel) and decoder (bottom panel).

The shape and location of a VOP are specified by an image called alpha plane. An alpha plane can be either a binary or a grey level type. The pixels in a binary alpha plane are either opaque (value 1) or transparent (value 0), which represent the inside and outside of a VOP respectively. Fig. 3 demonstrates a raw video frame (left), the VOP representing the fish (middle) and the binary alpha-plane of this VOP (right).

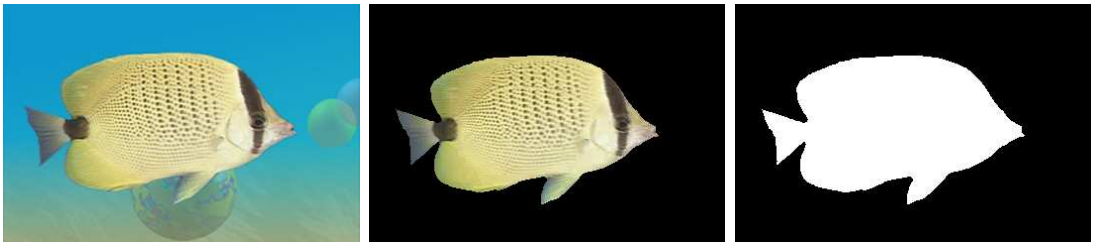

Figure 3: Original video frame (left), VOP of the fish (middle), and binary alpha plane of the fish VOP (right).

In the grey level case, an alpha plane can be interpreted in two ways, a segmentation mask or a transparency mask. In both cases, the pixel value can have a full range (usually 0 to 255). In the case of a segmentation mask, the pixel value indicates to which region the pixel belongs. In the other case, the pixel values represent the degree of transparency, e.g. from transparent (0) to opaque (255).

Although MPEG-4 provides a format of encoding and transmitting VOPs, the VOP construction, namely, how to decompose a video frame into video objects, is not described in the standard. It is the responsibility of the MPEG-4 users to construct VOPs [1, 2, 3, 12] to fit their own applications. **This is the research to which we devote the major effort**.

1.3 PATIENT MONITORING VIDEO

The investigations described in this dissertation aim at specific applications to patient monitoring video. A sample frame of this type of video is shown in Fig. 1. Our ultimate goal is to advance video coding systems utilized in patient monitoring. The specific aims

include improving coding efficiency of patient monitoring video and enhancing video archiving/retrieving facilities. The research was motivated in the following scenarios:

Video monitoring is commonly used in hospitals for clinical diagnosis. For example, video-EEG recording systems has been utilized to monitor epilepsy patients. This type of recording is usually conducted for a prolonged period of time (hours and days). Consequently, it produces a huge amount of data, mostly in the form of digital video. Compression of these video is necessary for both archiving and transmission purposes. Because of medical usage, high fidelity is required as well as high compression ratio.

Recently, digital recording systems based on general-purpose video coding standards (e.g. MPEG-2) have been utilized. These standards, designed for generic moving pictures, do not satisfy the special need for long-term monitoring under those requirements. As a result, they have performed in a sub-optimal way encoding patient monitoring video. For example, the Bio-Logic Digital Video-EEG System yields about 528 megabytes per hour, or 12.7 gigabytes per day to support a 352×240 video display. Due to limited storage, video files are usually stored in a temporary archive and then manually edited to discard most portions which otherwise should be kept for future reference.

Yet there exists a high potential to improve the compression performance, because patient monitoring video has the following features: 1) the camera position is usually fixed so that the background is almost static and there is hardly any global motion in the video; and 2) the movements of patient, when present, are mostly small and local because the location of patient is often restricted in certain area (e.g. in bed). Higher coding efficiency is expected if these features are utilized specifically in the design of a compression engine. For instance, the background regions in the video can be encoded with appropriately relaxed quality requirements (e.g. reduced spatial and temporal resolution), and only the region covered by patient needs to be encoded in the best quality.

In the light of object-based coding, the above thought can be formed in a more rigorous way that, a frame of patient monitoring video can be decomposed to at least two video objects, one representing recording environment and the other representing patient. This approach has the following advantages: 1) since background object may be considered static or very slowly changing, it does not have to be encoded at each video frame, therefore

leading to a reduction of frame rate; and 2) the object representing patient can be archived separately from background such that database searching and retrieving can be supported in a more content-driven fashion.

1.4 VIDEO OBJECT CONSTRUCTION IN THE LITERATURE

As previously mentioned, the major issue here is how to construct video objects from video frames. This problem is often referred as video segmentation [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. While human may easily identify objects in a video sequence, this task is still very difficult for computers to accomplish. One of the main problems is that a mathematical model of a video object is lacking. Therefore, the definition of a video object is usually vague, especially when a segmentation method tries to pursue the "semantics" in a general way. Nevertheless, various approaches to video segmentation have been reported. In this section, we briefly review some major techniques in the literature.

1.4.1 Automatic methods

Automatic methods try to segment a moving object from video frames without human supervision. In general, a video object is not necessarily moving in given frames. In these methods, however, motion is employed as a primary assumption of an object. Therefore, there are usually three components included in an automatic method: temporal segmentation to localize moving parts of an object, spatial segmentation to divide a video frame into regions, and fusion of the two results to form a final segmentation. Naturally, temporal segmentation provides a major clue of a moving object. However, it may not yield accurate segmentation of an object due to limitations of motion analysis algorithms. As a result, spatial segmentation is combined to improve the performance. According to the combination criteria, such video segmentation approaches can be categorized into two groups as follows:

- *Spatial homogeneity* based methods [17, 18] carry out spatial segmentation prior to temporal segmentation. Video frames are divided into homogeneous regions with respect

to color and texture, where image simplification (e.g. morphological filtering) followed by the watershed algorithms is usually applied. Temporal segmentation is performed on each region by calculating their motion activities. After that, regions with similar motion are grouped to form a video object. Although such approaches tend to detect object boundary well because of watershed algorithm, they are very computationally expensive. And, the assumption that "regions of an object possess analogous motion activities" may not be true for complex objects with elasticity, such as the human body.

- *Temporal transition* based methods [13, 14, 15, 56] utilize temporal segmentation results as the primary information. These methods calculate transition, defined as discontinuity in a signal, in temporal domain. This is usually accomplished by differentiating adjacent video frames or conducting motion estimation followed by a thresholding operation. A transition map, often a binary image, is provided to represent a rough segmentation of background and foreground. Spatial segmentation is then performed to deliver more accurate boundary. In [13], edges are detected in video frames and then registered to the transition map. The edges that belong to foreground are connected to form the contour of the moving object. Another approach [19, 20] applies edge detection directly to frame difference to obtain a "difference edge map", which is then refined to a "moving edge map" to provide the object contour. Besides edge detection, region-based spatial segmentation has also been proposed [14, 16], where regions are formed by watershed algorithm and those located in foreground are utilized to assemble a moving object. These temporal transition based methods are more efficient in exploiting motion information. However, they usually lack of robustness. Incompletion of a transition may result in considerable error in the final segmentation. For example, if any part of a contour is not detected, the entire region may be wrongly merged. Therefore, more potent temporal segmentation algorithms are needed.

In brief, the state of the art in automatic video segmentation still has to be improved for practical applications.

1.4.2 Semi-automatic methods

Due to the limitations of automatic methods, semi-automatic segmentation methods have been proposed [21, 22]. The concept of these methods is to introduce the definition of a video object from human supervision. As a consequence, the segmentation needs to be initialized by user and then followed by a tracking process. The initialized object is usually represented by the contour manually selected. This contour is updated from its initial shape to adapt to the motion estimation carried out between consecutive frames. Following that, a refinement operation is carried out to adjust the boundary according to spatial domain properties such as color and edge.

The weak points of these approaches include the following: first, because of the tracking-based nature, these approaches require re-initialization by user when the object is occluded or temporally dissapears; second, heavy deformation of a object can not be handled well by the available algorithms; and third, the computational complexity is usually high. As a consequence, these approaches are more suitable for offline applications, such as video editing.

1.4.3 Summary

Video object construction is a notably complex problem. The current techniques intend to combine image processing tools to establish segmentation in moving objects. However, because of the lacking of a theoretical model, the segmentation methods are founded upon a variety of assumptions and pre-set criteria. As a result, the generality and feasibility of the available methods have not yet been satisfactory. Furthermore, the high computational costs and the empirical parameters utilized in the image processing tools make these techniques unrealistic for certain practical applications.

The solutions to these problems rely on whether a clear definition of video object can be provided. It does not seem to be available in the neat future since the mechanism of high level processing in a human visual system is still not understood. This is also the major reason that semi-automatic methods utilize human supervision to describe an object. Although it is unrealistic to precisely define general video objects, it is achievable to give concrete

definition to a video object for specific applications. Face tracking [23] and iris detection [24, 25] are two applicable examples. In our research, we investigate specific definitions of video object in favor of video coding and archiving. Under such definitions, we provide a practical solution to extracting objects from *patient monitoring video*.

1.5 VIDEO OBJECT CONSTRUCTION VIA CHANGE DETECTION

1.5.1 What are the video objects in our approach?

In this dissertation, **three** video objects are defined based on the features of patient monitoring video. Normally, these video contains an idle environment and a patient with certain movements. A natural way would be prescribing the environment as one object and the contents related to the patient as the others. Therefore, we define the first object as *an image that contains only the scene of the environment*, and the second object as *the regions inside which the patient and the objects associated with him/her are included*. Note that the "objects" in this description may include the contents that are originally associated with the environment. For example, the bed where the patient rests may be deformed because of the patient's occupancy. In such a case, the deformed bed is considered as an entity associated with the patient, thus belonging to the second object. The third object is defined regarding the motion activity contained in the video. Noticing that normally only some body parts (not the whole patient) are involved in motion, we define the third object as *the regions that involve motion within a small time interval*. The substantial content of this object may be variant, such as "a moving hand" and "blinking eyes", all depending on what is moving in the time window. One realizes that this definition does not explore high-level semantics. Instead, it presents a mid-level semantic that delineates the moving objects in a general sense. As a result, this semantic does not directly enable object tracking functions. However, out of this definition, we do expect higher coding efficiency and archiving facilities by exploiting the motion information it represents.

1.5.2 How to construct the video objects?

Based on the three-object definition, we present a change-detection based approach to construct these video objects. The first video object, referred as VO_1 in further text, is relatively easy to obtain because of a static environment (monitoring room). In cases when the camera position is fixed during recording, a snapshot of the monitoring room can form this video object. If the camera is allowed to pan and tilt, the background scene can be updated online. The second video object, abbreviated as VO_2, is obtained by carrying out change detection between VO_1 and video frames. The outcome of a well designed change detection algorithm provides a binary mask representing regions undergoing essential content changes. Applying this mask on the video frame generates VO_2. The third video object, referred as VO_3, is constructed from multiple consecutive video frames, where change detection is executed to explore motion information. The regions that sustain motion through the video frames are grouped to form VO_3. In this dissertation, we present two novel change detection algorithms to generate VO_2 and VO_3 respectively. These algorithms are both robust and realistic for online applications on patient monitoring video.

1.5.3 Why change detection?

Change detection is a useful technique that distinguishes image differences caused by content changes from those by irrelevant disturbances. It has a broadband spectrum of applications [26, 27, 28, 29, 30, 31, 32] including video segmentation, where it forms a central unit of temporal segmentation that explores motion information [52, 51, 53, 55, 54, 64]. It should be noticed that the scope of change detection is beyond motion detection. When applied to successive video frames, the detected changes imply apparent motion, thus lead to the detection of moving pixels. In other disciplines, the interpretation of changes is application-specific.

Following such definition of change detection, one can see that the construction of VO_2 and VO_3 as previously defined directly leads to the application of change detection. The explicit benefits from change detection are the following:

- The computational complexity is much reduced when compared with other motion detection techniques, e.g. optical flow. This is of substantial importance to the overall performance of our video coding system.

- The appearance or disappearance of objects is identified, which is crucial for constructing VO_2, where the patient appears as a new object to the background.

- The assumption of rigid motion is not required, which is particularly useful in the construction of VO_3, because the motion generated by human body may be typically non-rigid.

Although change detection only provides binary results, it already satisfies the need of our segmentation tasks since the defined objects VO_2 and VO_3 require only binary segmentation masks. Therefore, complex transition and motion field calculation would be unnecessary.

1.6 PREVIOUS CHANGE DETECTION APPROACHES

The goal of a change detection algorithm is to classify image pixels into two sets, "changed" and "unchanged". The former denotes "there are significant differences between the images at the corresponding locations", and the latter denotes the opposite. The definition of "significant" is largely associated with human visual perception and may vary from application to application. In common cases, the image differences caused by relative motion between objects and camera, appearance/disappearance of objects, shape, color, and texture changes of objects, are considered to be "significant"; those caused by ambient and sensor noise, illumination variation, and registration error are "insignificant". It is by no means a trivial problem to guarantee the robustness to detect the changes of interest. Therefore, research on change detection has been carried out continuously for over twenty years [51, 52, 53, 54, 55, 14, 65, 66]. In this section, we present a systematic survey on these techniques.

1.6.1 Predictive model

The concept of this approach is to formulate the gray value intensity in a given region as a polynomial function of the pixel coordinates. A representative of this technique is the *quadratic picture function* model proposed by Hsu etc. [51]. They modeled an image as a mosaic of blocks where the intensity value was formulated as a second-order bivariate polynomial function of the pixel coordinates. Change detection is carried out by comparing corresponding block pairs in two images. If two blocks can be least-square fit by a same group of polynomial coefficients, then no change is detected between the two blocks. The alternative decision will be drawn if they are best fit by different polynomial coefficients. The examination is performed by a likelihood test derived by Yakimovsky [34], where the decision threshold is obtained by F-test. The major weak point with this technique is that the assumption that image intensity can be modeled as quadratic function is often violated in real scenarios. And the residuals from the polynomial fit may not be Gaussian distributed either. Therefore, the accuracy of the likelihood test is undependable.

1.6.2 Hypothesis testing

In this technique, whether a pixel is "changed" or "unchanged" is determined by choosing the hypothesis that best matches the observation and the prior knowledge. The *significance test* [53, 54] method developed by Aach etc. is a typical hypothesis testing approach. In this method, the statistics of noise is utilized to test whether the observed image difference is caused solely by noise. The null hypothesis in the test is that under the condition of "no change", the image difference can be modeled as a random variable that has a zero-mean Gaussian distribution with a known variance. The test of this hypothesis is carried out at each local region which is a spatial window centered at the testing pixel. The testing variable is defined as the local sum of squared difference of the pixel intensity normalized by the noise variance. This variable under the null hypothesis has a χ^2 distribution with the degrees of freedom equal to the number of pixels inside the local window. Therefore the decision threshold is determined by specifying a confidence level of the withholding of the null hypothesis. This approach performs change detection heuristically well if the local

window size and the confidence level are properly chosen. The weakness with this technique is that the testing is one-side, meaning that the knowledge of the alternative hypothesis is not utilized at all. As a result, this approach lacks of the sense of optimality.

1.6.3 The shading model

This technique intends to exclude illumination variation from "significant" changes by utilizing the shading model which formulates image intensity based on physical aspects of light reflection. With appropriate assumptions, the gray level intensity of a pixel is approximated by the product of the illumination of a physical surface point and its shading coefficient. This coefficient is determined by a number of factors, such as the reflectance of the surface material, and angles of striking and reflected lights [57]. If no change undergoes the physical structure of an object, the shading coefficient is assumed to be intact. Under such condition, the ratio of pixel intensities in two images becomes the ratio of illumination from the two corresponding physical locations. Since illumination can be approximated as a constant within regions that are sufficiently small, the pixel intensity ratios remain constant in the testing blocks under the condition of "no change". Based on this rationale, Skifstad etc. [52] suggested to test the variances of pixel intensity ratios within two given blocks. If the variance is smaller than a threshold empirically selected, then it is determined that the imaged object surfaces are in the absence of change. Durucan etc. [55] formulated the change detection from a point of view of linear dependence test. They formulated the hypothesis of "no change" as linear dependence between vectors of corresponding pixel intensities. The test is carried out by thresholding the determinants of Wrongskian matrices [55] that represent the linear dependence of the given vectors. Both Skifstad's and Durucan's approach are centered around the shading model, in which the illumination variation is dealt with reasonably well. However, the noise effects are not considered in these models. As a result, the thresholds utilized in these tests are chosen in an *ad hoc* manner.

1.6.4 Contextual consistency models

The above models are designed from different perspectives, but are common in one aspect that they are all *thresholding* based methods, i.e. the decisions are made by applying one chosen threshold to a well defined test statistic. While single-threshold approaches may be efficient from a computational point of view, they are subjected to a quandary of either causing false alarms when the threshold is not large enough, or missing detection of significant changes when the threshold is overestimated. The reason is that the change detection is performed locally at each pixel, but the single threshold to be applied is determined globally. In other words, this threshold is non-adaptive to the properties of a local region.

The concept of adaptive thresholds was introduced by Aach in [54]. He assumed that regions corresponding to moving objects are likely to have compact shape with smooth boundaries. Based on this assumption, a multiple-threshold approach was proposed where the thresholds are functions of not only intensity difference but also number of "border pixel pairs" that represented the degree of smoothness of region boundary. The intensity difference determined a so called "anchor threshold" and the "border pixel pairs" performed as a regulating factor that adjusted the threshold. Better results can be achieved if the threshold is increased/decreased when the contextual information of a local region reveals clues of "unchanged"/"changed" status of the pixel being tested. As a direct extension from significance test method, this method heavily depends on the "anchor threshold" chosen empirically in a deterministic nature. Consequently, the results in optimal sense are not expected in general. However, this approach can be extended to change detection methods in an optimization point of view.

Recently, optimization-based change detection methods have emerged for analyzing remote-sensing images [65, 66]. These methods utilized Markov random field (MRF) theory to enforce spatial-contextual constraints in the change detection process. The change detection mask is found by maximizing the associated *a posteriori* probability. In other words, the optimal result is obtained in maximum *a posteriori* (MAP) sense. We believe the MRF-based approaches have great potentials for image change detection problems. However, the available models that were specifically designed for satellite image analysis are not directly

transplantable to change detection in general video sequence. Therefore, this technique is worthy of further investigation for more general applications.

1.7 NEW CHANGE DETECTION MODELS

Based upon the review of the conventional methods, we see that there is a great margin to improve the current status of change detection techniques. While thresholding approaches have much less computational complexity, optimization methods may provide more robust results. Therefore, we explore both methods aiming at enhanced performance of the change detection algorithms such that more reliability is provided for video object construction.

In this thesis, we present two new change detection models designed for the construction of VO_2 and VO_3 as previously defined.

- The first model employs the MRF theory and the Mean Field Theory (MFT) to perform change detection in the MAP sense. In this model, novel energy functions are designed to reflect prior knowledge and contextual constraints on both the noise and the signal. An optimal change detection mask (CDM) is obtained by utilizing MFT to minimize the energy functions. This model is applied to the construction of VO_2.

- The second model that differs from the conventional frame-pair-based methods provides a thresholding-based approach to change detection by utilizing a group of video frames. The design of this model is based on a fact that the vector of pixel intensity across multiple frames tends to be highly correlated with its spatial neighbors at the presence of change. When applied to multiple consecutive video frames, this model accurately detects moving regions. For this reason, this model is implemented for constructing VO_3.

1.8 CONTRIBUTIONS

In this work, we present an object-based video coding system for the compression of patient monitoring video. Change detection is applied as the key technique to video object construction. We provide both theoretical analysis and experimental results to demonstrate the efficiency and effectiveness of this system. Specifically, the contributions of this work are:

- We present a novel three-layer structure that defines the video objects for coding. The three video objects that represent background, patient and moving body parts are encoded with different temporal scalability.
- We show by both statistical analysis and experimental results that the coding efficiency is improved significantly by the object-based coding approach.
- We contribute two new change detection approaches to constructing the defined video objects.
  - MRF-MFT method: this approach utilizes the Markov random field (MRF) theory and the mean field theory (MFT) to detect relevant changes between images. This approach differs from the conventional methods in that change detection is performed in an optimization process. Novel cost functions that reflect contextual constraints are defined, which show great effectiveness in detecting small changes.
  - Covariance test method: the novelty of this approach lies in the exploration on temporal correlation contained in successive video frames. This leads to the great robustness in detecting small changes between consecutive video frames.

  We show by experimental results that these two methods outperform the conventional approaches in terms of less false detections.

1.9 THESIS OUTLINE

The chapters of this proposal are organized as follows. In Chapter 2, we show how the coding efficiency is improved via the object-based approach. Both texture coding and shape coding are analyzed. In Chapter 3, we present two new change detection models that are

designed for constructing the video objects. In Chapter 4, the system implementation and experimental results of the object-based coding of patient monitoring video are reported. In the final chapter, we conclude this thesis and suggest future work.

2.0 OBJECT-BASED VIDEO CODING

2.1 INTRODUCTION

In this chapter, we show why coding efficiency is improved via object-based coding. Conceptually, this improvement is due to the selective coding of the video content, meaning that the bit rates are allocated based on the user's interests to the content. While the content of interest is assigned higher bit rate to maintain the fidelity, the coding of uninterested content can be omitted or relaxed, so that the overall bit rate is reduced and the essential quality is preserved. Lacking of this flexibility, frame-based coding approaches in general treat each pixel in the same manner, therefore, the content redundancy is not well exploited.

In the coding of patient monitoring video, the background contents are of far less interest than the foreground regions that contain the patient. In addition, there is a considerable portion of background region in a video frame. Coding the background in each video frame can be a significant waste of bandwidth. Indeed, the background contents only need to be coded occasionally when the background scene changes. At other times, only the initial background should be coded and the subsequent ones are negligible. Based on this concept, we analyze quantitatively the reduction in bandwidth (in terms of bit rate) by omitting the coding of the difference between the initial background and the subsequent ones.

Object-based coding has been described in MPEG-4, a lately developed video coding standard providing tools and algorithms for storage, transmission and manipulation of video data in multimedia environments. Unlike the previous standards, e.g. MPEG-1, MPEG-2, H261 and H263, MPEG-4 supports the coding of *video objects* which can be arbitrarily shaped. A video scene thus can be coded as a composition of video objects, each of which can be treated as an independent entity. In some coding applications [35, 36, 37], the video

object representing background may be coded only once, and the other objects are encoded through the time with possibly different scalabilities. At the receiving end, the decoded objects are repeatedly surmounted on the reconstructed background. Since only the objects of interest are coded with high quality, and they usually represent a small portion of the entire video, the bit rate of the encoded video stream can be significantly reduced.

In the coding process, a video object is represented by so-called *video object plane* (VOP) which is a snapshot of the video object at a time point. Two essential components are associated with a VOP, the intensities of the pixels in it and the shape of the VOP. Consequently, coding a video object involves two essential steps of texture coding and shape coding (except for a video object being an entire frame). An overview of the encoder kernel for each video object is outlined in Fig. 4, where shape coding and texture coding are carried out separately. The structure of the texture coding is called *hybrid coding* which exploits both spatial and temporal domain redundancy to code pixel intensities. Motion estimation and compensation [1] are carried out to utilize temporal correlation between adjacent video frames. The intensity residuals after motion compensation are coded by texture coding which is typically constructed by transform coding techniques [1, 80].

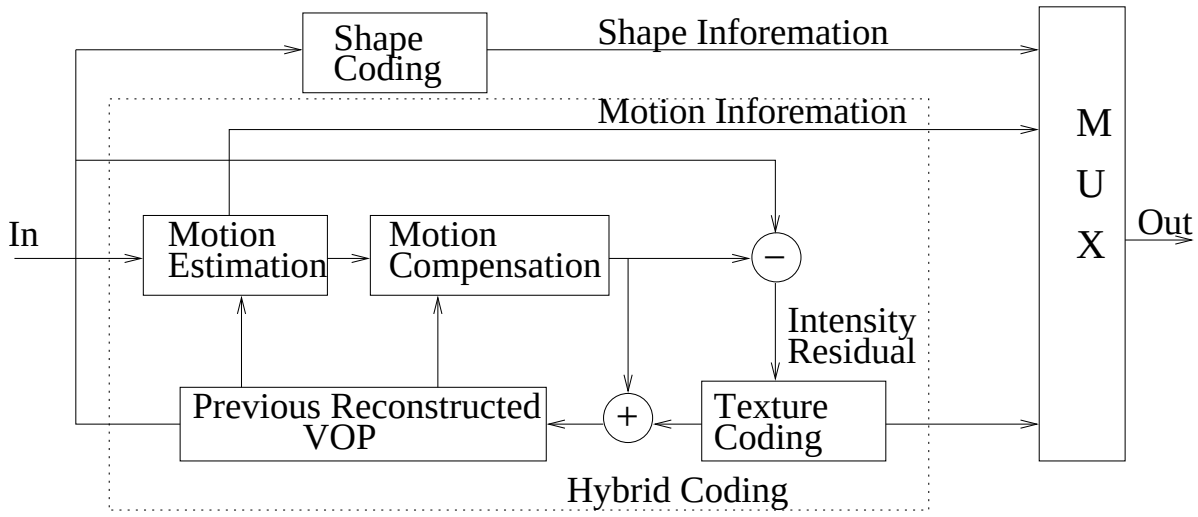

Figure 4: The structure of a VOP encoder.

It should be noticed that the same hybrid coding structure is utilized in the conventional frame-based coding techniques, e.g. MPEG-2 and H263. The only difference is that the

input of the hybrid coding is the entire frame, not a VOP. Therefore, for texture coding, we investigate the common hybrid coding mechanism aiming at a quantitative analysis on why and how the coding efficiency can be improved by manipulating VOPs. Shape coding on the other hand, is unique in object-based coding techniques. As a trade-off to the content-based functionalities, shape information needs to be coded. In contrast to frame-based coding, the extra bits allocated for shape coding raise concerns on the overall coding efficiency of object-based schemes. Therefore, for shape coding, we derive an estimate of coding an arbitrary shape and discuss that the overall performance is still superior to frame-based coding with respect to coding efficiency.

2.2 TEXTURE CODING

Transform coding has been the leading technique for coding image texture in the available compression standards and reported algorithms. A common structure of transform coding is outlined in Fig. 5, where three essential components are comprised: the transform, quantization and entropy coding. The transform can be either discrete cosine transform (DCT) or discrete wavelet transform (DWT), which decorrelates the input image into coefficients. Following the transform, the quantization block converts the transform coefficients into quantization indices (usually integers) which represent the scale level of the coefficients. At the final stage, these quantization indices are arranged into symbols and encoded by a Huffman or arithmetic coder *Ghanbari*.

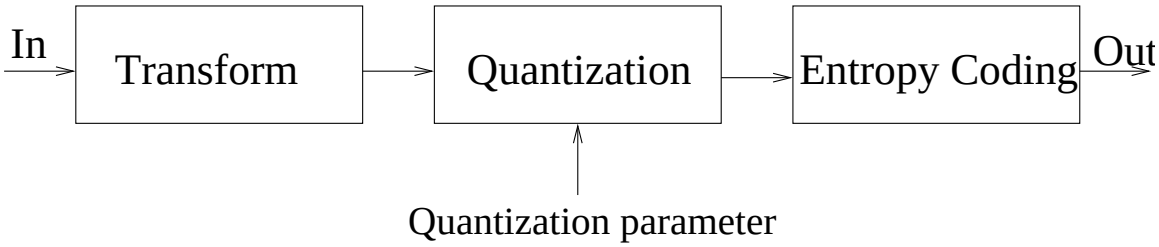

Figure 5: A generic structure of transform coding for image/video compression.

A critical parameter that controls the coding quality is the *quantization parameter*. In object-based coding, one can assign different quantization parameters to individual video objects to achieve quality control and bit rate allocation. In the following, we investigate the quantization schemes employed in the MPEG-4 standard to provide a quantitative analysis on these concerns.

2.2.1 MPEG quantization scheme

In image/video coding, a digital picture is partitioned into blocks (typically 8×8 pixels), which are transformed into matrices of coefficients in the same size. Each of the transform coefficients is quantized by a quantization step in the following form,

$$I_{u,v} = \text{Round}[\frac{C_{u,v}}{\Delta_{u,v}}], \tag{2.1}$$

where I, C and Δ denote the quantization index, transform coefficients and quantization step respectively, and u, v denote their indices in the corresponding matrix.

Essentially, the minimum average bits required to code $I_{u,v}$ can be estimated by its entropy,

$$B_{u,v} = -\sum_{i} P_{u,v}(i)\ \log_2 P_{u,v}(i) \tag{2.2}$$

where i is the integer value of $I_{u,v}$ and $P_{u,v}(i)$ is its probability.

Assuming $C_{u,v}$ has a probability density function (pdf) of $f_{C_{u,v}}(\cdot)$, one can obtain the probability $P_{u,v}(i)$ by

$$P_{u,v}(i) = \int_{(i-0.5)\Delta_{u,v}}^{(i+0.5)\Delta_{u,v}} f_{C_{u,v}}(x)dx. \tag{2.3}$$

Also, the quantization error can be calculated by

$$D_{u,v} = \sum_{i} \int_{(i-0.5)\Delta_{u,v}}^{(i+0.5)\Delta_{u,v}} (x - i\Delta_{u,v})^2 f_{C_{u,v}}(x)dx. \tag{2.4}$$

The actual step size $\Delta_{u,v}$ is associated with quantization matrices in MPEG standards, where two types of quantization matrices are provided, the intracoding and intercoding. The former is applied on the coefficients obtained without motion compensation, that is, the coefficients are transformed from pixel intensities directly. The latter is applied on the

coefficients transformed from intensity residuals after motion compensation. In MPEG-2 and MPEG-4, these quantization matrices are designed based upon human perceptual ability [1, 78, 80], given as follows,

$$Q_M^0 = \begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \tag{2.5}$$

$$Q_M^1 = \begin{bmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{bmatrix} \tag{2.6}$$

where Q_M^0 and Q_M^1 denote the intracoding and intercoding quantization matrix respectively. And the quantization step is determined in the following form,

$$\Delta_{u,v} = \begin{cases} 8, & \text{for a DC coefficient in an intracoding block} \\ \frac{2 \cdot q \cdot Q_M^0(u,v)}{16}, & \text{for an AC coefficient in an intracoding block} \\ \frac{2 \cdot q \cdot Q_M^1(u,v)}{16}, & \text{for an AC coefficient in an intercoding block} \end{cases} \tag{2.7}$$

where q is the quantization parameter shown in Fig. 5.

Note that while Q_M^0 and Q_M^1 are fixed, q is controllable. In object-based coding, quality control on different video objects can be realized by assigning different values to q with respect to each object.

Since the transform always performs decorrelation to an image block, the coefficients $C(u,v)$ can be approximately considered as independent random sources. Consequently, the number of bits required to encode a block are given by

$$B_M = \sum_{u,v} B_{u,v}, \tag{2.8}$$

and the corresponding quantization error is

$$D_M = \sum_{u,v} D_{u,v}, \tag{2.9}$$

where $B_{u,v}$ and $D_{u,v}$ are given in Eqs. 2.2 and 2.4 respectively.

2.2.2 Results and discussion

In frame-based coding, a significant portion of bit rate can be allocated for coding irrelevant contents (e.g. noise) even when they are presented as small amplitude samples. This is especially true if high fidelity is required on the coded pictures.

To show this result, we start from the assumptions on the transform coefficients and derive the estimated bit rate. The widely accepted assumptions on $f_{C_{u,v}}(\cdot)$, i.e.the pdf of the transform coefficients, are Gaussian and Laplacian.

2.2.2.1 For Gaussian distribution Employing the Gaussian assumption, one has

$$f_{C_{u,v}}(x) = \frac{1}{\sqrt{2\pi}\sigma_{u,v}} e^{-\frac{(x-\mu_{u,v})^2}{2\sigma_{u,v}^2}} \tag{2.10}$$

where $\mu_{u,v}$ and $\sigma_{u,v}^2$ are the mean and variance respectively. Usually, $\mu_{u,v}$ is assumed to be zero. Therefore, one has the probability $P_{u,v}(i)$ in Eq. 2.3 in the following form

$$\begin{aligned} P_{u,v}(i) &= \int_{(i-0.5)\Delta_{u,v}}^{(i+0.5)\Delta_{u,v}} \frac{1}{\sqrt{2\pi}\sigma_{u,v}} e^{-\frac{x^2}{2\sigma_{u,v}^2}} dx \\ &= \frac{1}{2}(\mathrm{Erf}[\frac{(i+\frac{1}{2})\Delta_{u,v}}{\sqrt{2}\sigma_{u,v}}] - \mathrm{Erf}[\frac{(i-\frac{1}{2})\Delta_{u,v}}{\sqrt{2}\sigma_{u,v}}]) \end{aligned} \tag{2.11}$$

where $\mathrm{Erf}(\cdot)$ is the "error function".

The close forms of Eqs. 2.2 and 2.4 are unavailable for the Gaussian case. The numerical results of them are shown in Figs. 6 and 7 respectively. The "bits per sample" are obtained by evaluating Eqs. 2.11 and 2.2 on random sequences with Gaussian distributions. The mean values are all zeros and the standard deviations are ranged from 0.5 to 4. The same settings apply to the results of the corresponding quantization distortions. The thin line plots (blue) in Fig. 6 delineate the numerical results of bits per sample vs. quantization steps at different standard deviations. It is seen that given a fixed standard deviation of the random sequence, the higher the quantization step, the lower the bits per sample. And, the bits per sample increase when the sequence to encode has a higher value of standard deviation.

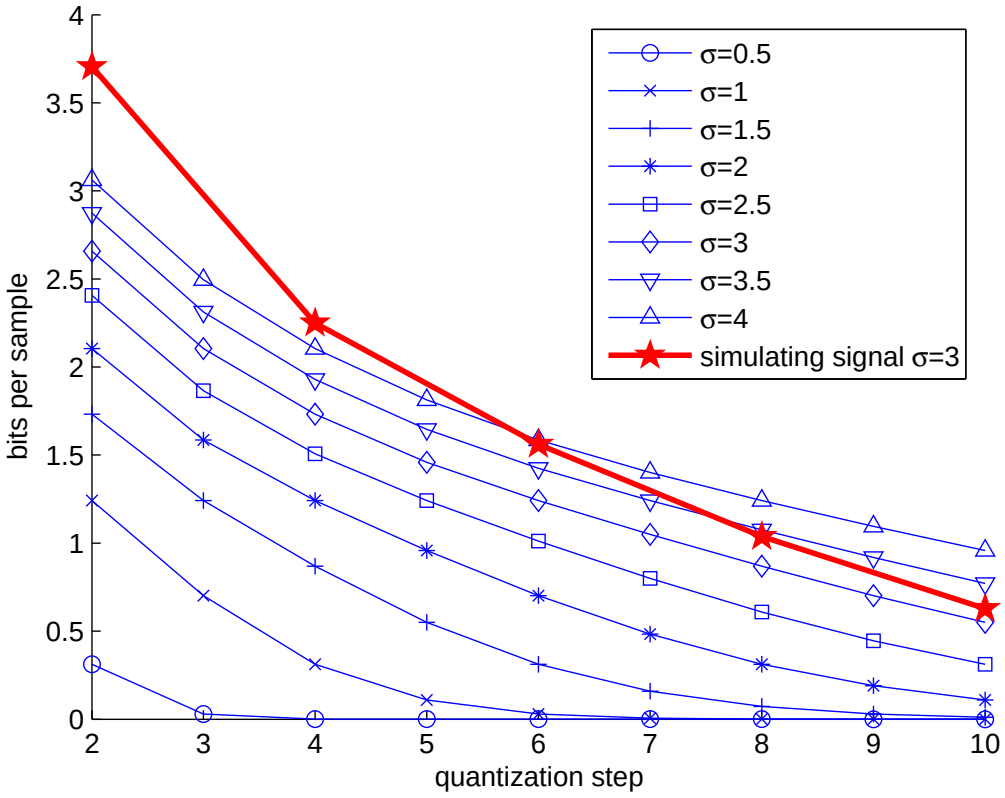

Figure 6: Bits per sample vs. quantization step for Gaussian distributed sequences with different standard deviations.

To examine the fitness of the analytical results, we present experimental results on some simulating data. The simulating video sequence was generated by using Matlab. The DCT transform coefficients had a Gaussian distribution with a mean of zero and a standard deviation approximately 3. This sequence was encoded by using ISO/IEC 14496(MPEG-4) Video Reference Software (version Microsoft-FDAMI-2.5-040207). The bits-per-sample resulted from this software package is plotted in Fig. 6 in thick (red) line. It is seen that the plot from simulating data is above the numerical result (diamond plot), meaning that the actual bits at given quantizations are slightly larger than the analytical ones. This is because the estimated bits-per-sample is derived from the entropy of the random source, which is the lower limit of the average code length.

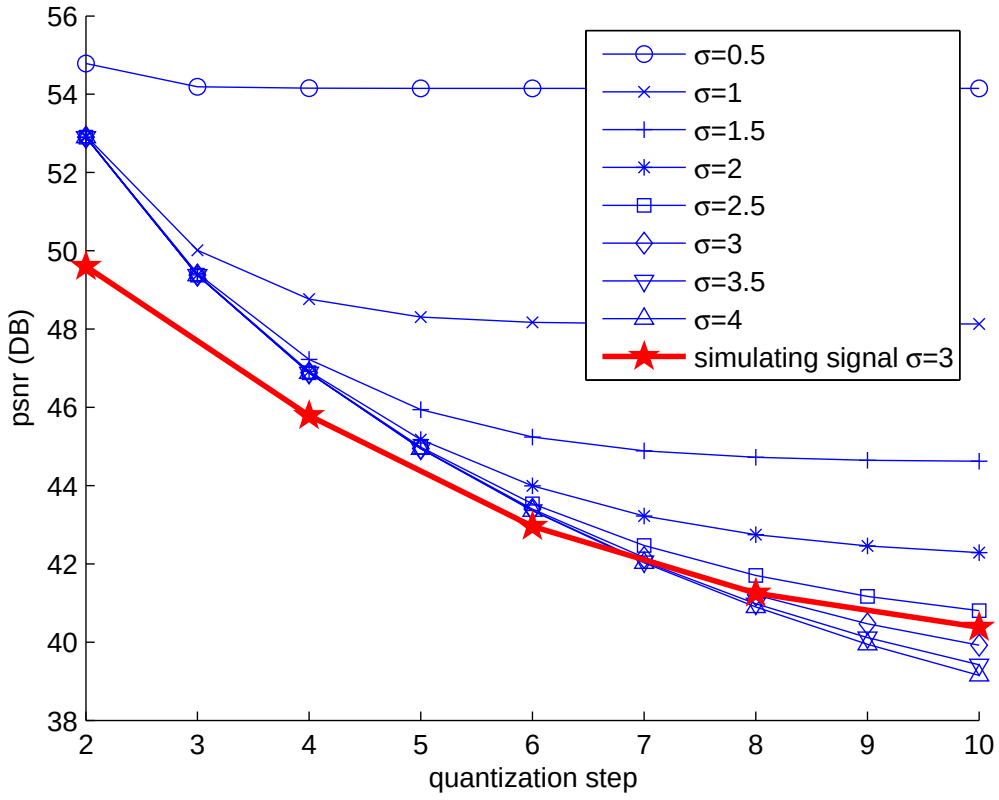

Figure 7: Distortion vs. quantization step for Gaussian distributed sequences with different standard deviations.

The corresponding quantization distortions in the form of "peak signal to noise ratio" (psnr), defined as $10\log_{10}\frac{255^2}{D}$ where D is the mean square error, are shown in Fig. 7. It is seen that the distortion becomes heavier as quantization step increases. And, sequences with larger standard deviation tend to have larger quantization errors. The experimental results (thick lines) matched the numerical plot (diamond thin line) well at comparably large quantization steps (i.e. beyond 5). The deviation at small quantization steps is due to the implementation in the MPEG-4 software package where the DCT transform coefficients are represented as integers. This extra rounding effect decreases as the quantization step become larger.

2.2.2.2 For Laplacian distribution

Employing the Laplacian assumption, one has

$$f_{C_{u,v}}(x) = \frac{1}{\sqrt{2}\sigma_{u,v}} e^{\frac{-\sqrt{2}(|x-\mu|)}{\sigma_{u,v}}} \tag{2.12}$$

where $\mu_{u,v}$ and $\sigma^2_{u,v}$ are the mean and variance respectively. Commonly, $\mu_{u,v}$ is assumed to have a value of zero. Then, the probability $P_{u,v}(i)$ in Eq. 2.3 has the following form,

$$\begin{aligned} P_{u,v}(i) &= \int_{(i-0.5)\Delta_{u,v}}^{(i+0.5)\Delta_{u,v}} \frac{1}{\sqrt{2}\sigma_{u,v}} e^{\frac{-\sqrt{2}|x|}{\sigma_{u,v}}} dx \\ &= \begin{cases} 1 - e^{-\frac{\Delta_{u,v}}{\sqrt{2}\sigma_{u,v}}} & \text{for } i = 0 \\ \frac{1}{2} e^{-\frac{(|i|-0.5)\Delta_{u,v}}{\sqrt{2}\sigma_{u,v}}} (1 - e^{-\frac{\sqrt{2}\Delta_{u,v}}{\sigma_{u,v}}}) & \text{for } i \neq 0 \end{cases} \end{aligned} \tag{2.13}$$

Let $\rho_{u,v} = e^{-\frac{\sqrt{2}\Delta_{u,v}}{\sigma_{u,v}}}$, the entropy for coefficient $C_{u,v}$ is given by

$$\begin{aligned} B_{u,v} &= -\sum_{i=-\infty}^{\infty} P_{u,v}(i)\ \log_2 P_{u,v}(i) \\ &= -P_{u,v}(0)\ \log_2 P_{u,v}(0) - 2\sum_{i=1}^{\infty} P_{u,v}(i)\ \log_2 P_{u,v}(i) \\ &= (1-\sqrt{\rho_{u,v}})\ \log_2 \frac{1}{1-\sqrt{\rho_{u,v}}} - \frac{1-\rho_{u,v}}{\sqrt{\rho_{u,v}}} \sum_{i=1}^{\infty} \rho^i_{u,v}[\ \log_2(\frac{1-\rho_{u,v}}{\sqrt{\rho_{u,v}}}\rho^i_{u,v}) - 1] \\ &= H(1-\sqrt{\rho_{u,v}}) + \frac{1}{1-\rho_{u,v}}[(1+\rho_{u,v})H(\sqrt{\rho_{u,v}}) + \sqrt{\rho_{u,v}}H(1-\rho_{u,v})] + \sqrt{\rho_{u,v}} \end{aligned} \tag{2.14}$$

where $H(\rho_{u,v}) = -\rho_{u,v}\log_2\rho_{u,v}$. And the corresponding quantization error is given by

$$D_{u,v} = \sum_{i=-\infty}^{\infty} d_{u,v}(i) \tag{2.15}$$

where

$$\begin{aligned} d_{u,v}(i) &= \int_{(i-0.5)\Delta_{u,v}}^{(i+0.5)\Delta_{u,v}} \frac{1}{\sqrt{2}\sigma_{u,v}} e^{\frac{-\sqrt{2}|x|}{\sigma_{u,v}}} (x - i\Delta_{u,v})^2 dx \\ &= \begin{cases} \sigma_{u,v}^2 - \frac{1}{4}\sqrt{\rho_{u,v}}(\Delta_{u,v}^2 + 2\sqrt{2}\Delta_{u,v}\sigma_{u,v} - 4\sigma_{u,v}^2) & i = 0 \\ \frac{1}{8}\sqrt{\rho_{u,v}}\rho_{u,v}^{|i|}[-\Delta_{u,v}^2 - 2\sqrt{2}\Delta_{u,v}\sigma_{u,v} - 4\sigma_{u,v}^2 + \\ \frac{1}{\rho_{u,v}}(\Delta_{u,v}^2 - 2\sqrt{2}\Delta_{u,v}\sigma_{u,v} + 4\sigma_{u,v}^2)] & i \neq 0. \end{cases} \end{aligned} \tag{2.16}$$

Therefore, one has

$$\begin{aligned} D_{u,v} &= d_{u,v}(0) + \sum_{i=1}^{\infty} d_{u,v}(i) \\ &= \sigma_{u,v}^2 - \frac{\sqrt{2}\Delta_{u,v}\sigma_{u,v}\sqrt{\rho_{u,v}}}{1 - \rho_{u,v}} \end{aligned} \tag{2.17}$$

where $\rho_{u,v} = e^{-\frac{\sqrt{2}\Delta_{u,v}}{\sigma_{u,v}}}$.

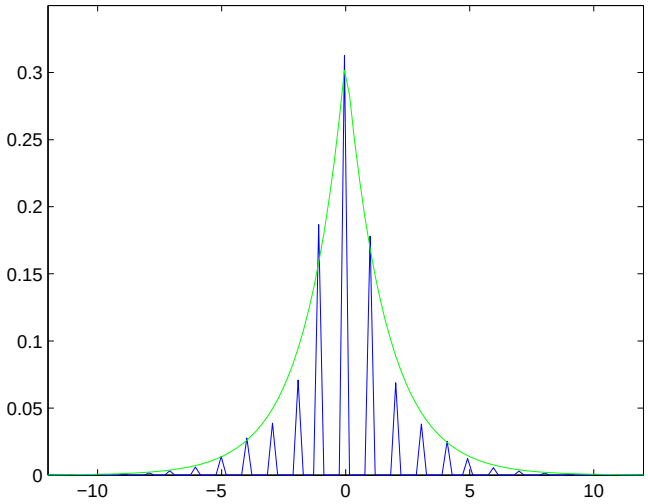

Figure 8: Histogram of DCT coefficients of image difference from real world data. The histogram can be approximately fit by a Laplacian pdf with a mean of 0 and standard deviation of 2.5.

We provide both the analytical and experimental results for Laplacian distributed sequences. The experimental results were obtained by coding real world image sequence. The test images contained merely background noises, e.g. device noise and ambient noise. The purpose was to show the cost of coding these disturbances. The histogram of the image differences is shown in Fig. 8, where a Laplacian pdf with a zero mean and a standard deviation of 2.5 approximated the histogram. In Figs. 9 and 10, the thin lines (in blue) show the analytical results of bits-per-sample vs. quantization step. The thick line (in green) shows the result of coding the real world data. We see that the bits-per-sample of the experimental result is larger than the analytical result (the square plot). This is because the analytical result represents the entropy of the source, the lower bound of the actual code length. We also see that the distortion plots of the experimental data match the analytical ones well.

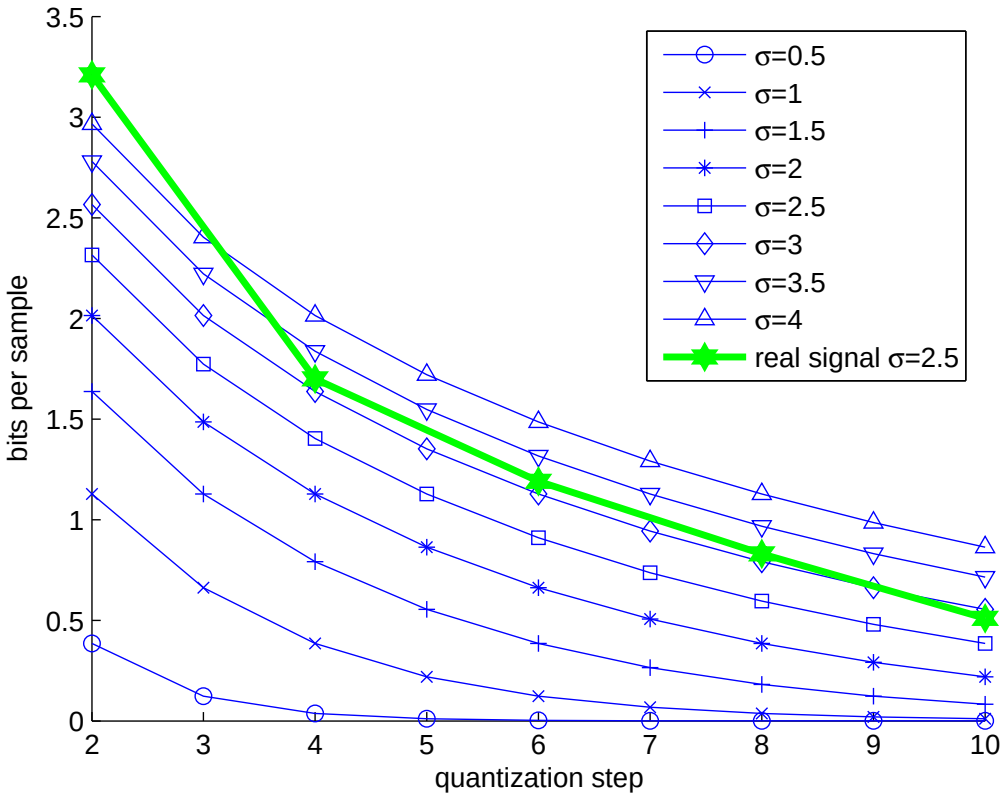

Figure 9: Bits per sample vs. quantization step for Laplacian distributed sequences with different standard deviations.

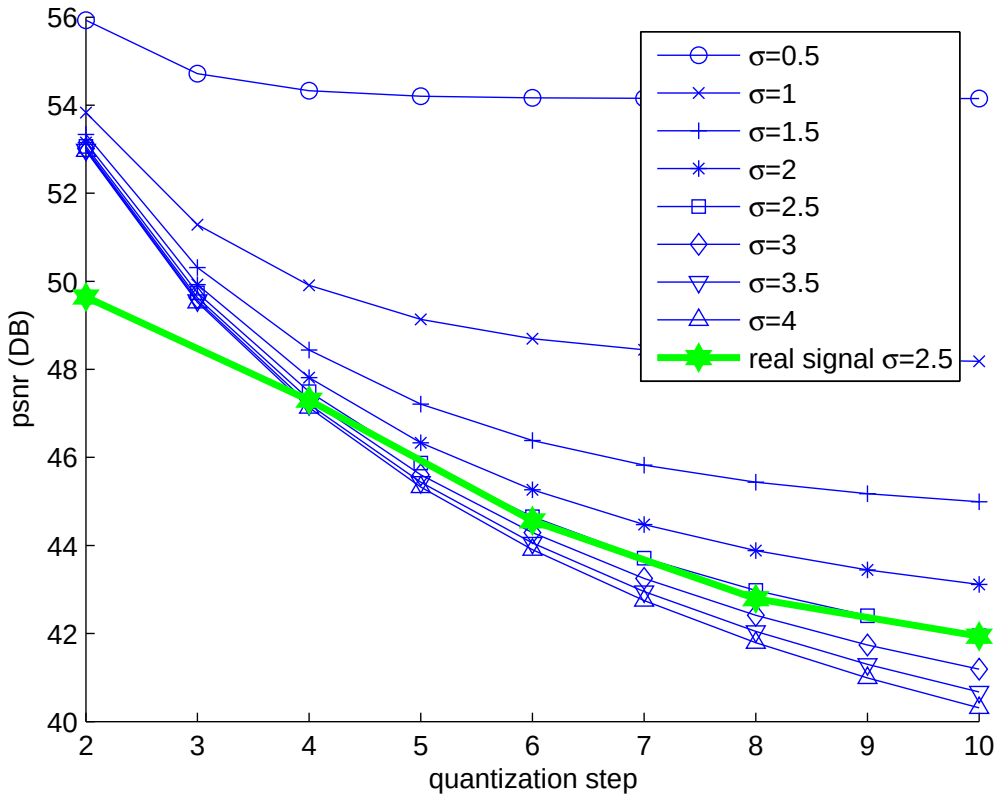

Figure 10: Distortion vs. quantization step for Laplacian distributed sequences with different standard deviations.

2.2.2.3 Discussion In the intracoding mode, $\sigma_{u,v}$, the variance of a DCT transform coefficient, varies with (u, v) (normally, the coefficients representing high frequency components have larger $\sigma_{u,v}$). Also, the entries of quantization matrix Q_M^0 (Eq. 2.5) vary with (u, v), as a consequence of which the quantization steps $\Delta_{u,v}$ (Eq. 2.7) also vary with (u, v). However, intracoding is only carried out on the initial video frames (I frames [1]) in a coding process. Most video frames (B and P frames [1]) are coded in the intercoding mode. For example, in MPEG-2, a group of 15 consecutive video frames contains only one I frame. And in MPEG-4, this group can consist of more than 300 frames while only containing 1 I frame. Therefore, normally, intracoding demands only a small amount of bit allocation. The

majority is from intercoding. The experimental results (thick lines) in Figs. 6, 7, 9, and 10 were all obtained from intercoded video frames. Since the amount of intracoded bits was far less than that from the intercoding, as in all the experiments, the plots approximated the average bits-per-pixel of coding the entire sequence.

For intercoding, since Q_M^1 is flat, i.e. all the entries have the same value, the quantization steps $\Delta_{u,v}$ are equal for all (u, v). Furthermore, the DCT coefficients $C_{u,v}$, transformed from residuals after motion compensation, are likely to have similar standard deviations $\sigma_{u,v}$. Approximately, one can assume that the coefficients $C_{u,v}$ are drawn from the same distribution. Henceforth, one has

$$\begin{aligned} B_M &= NB_{u,v} \qquad \forall u, v \\ D_M &= ND_{u,v} \qquad \forall u, v \end{aligned} \tag{2.18}$$

where N is the number of coefficients contained in a block, typically 64.

Therefore, in an object-based coding approach, the bit rate (bits per second) of coding the texture of a video object can be estimated by the following form,

$$R_{vo_i}^t(n) = \eta \sum_{k=nF_{vo_i}}^{(n+1)F_{vo_i}-1} \sum_{r,c \in \Phi_{vo_i}(k)} B_M(r, c, k) \tag{2.19}$$

where N is the number of pixels in a block, typically 64, F_{vo_i} is the frame rate, i.e. number of VOPs per second, n is time point in unit of second, (r, c) is the index of a block in spatial domain, $\Phi_{vo_i}(k)$ is the support region of the video object at frame k, and $\eta = \frac{\text{number of bits per pixel}}{8}$ is a parameter associated with the video format, for instance, if the video is grey level, then $\eta = 1$; if the video is in RGB format, then $\eta = 3$; and if YUV12 (e.g. CIF), $\eta = 1.5$, etc.. It should be noticed that F_{vo_i} is scalable. For instance, motionless video object can be assigned with a small F_{vo_i} to suppress the bit rate. Also, B_M is scalable by adjusting the associated quantization parameter q (e.g. Eq. 2.7). In brief, both B_M and F_{vo_i} are variables for different video objects.

Compared with object-based coding, frame-based coding has much less flexibility. Although frame rate and quantization parameter are still controllable, they are applied to the

entire frame, which implies that all pixels are treated equally no matter what content they represent. With all the previous derivations, we are able to show the redundancy in the frame-based coding which can be greatly reduced by the object-based approach. We utilize an example to show the potential improvement. Some sample video objects in the patient monitoring video are shown in Fig. 11, where (a) shows VOP_2 that represents the foreground and (b) shows VOP_3, the moving foreground. The VOP_3 was detected within a short time window equal to 0.5 seconds in this example. The 15 video frames in this duration can essentially be represented by the union of 1 image of the background, 1 VOP_2 and 14 VOP_3. Therefore, the background regions in the 14 P-frames were not coded. The bits saved from this can be calculated with Eqs. 2.14, 2.17 and 2.19. The normalized histogram of the DCT coefficients in the background area, transformed from the frame difference, is shown in Fig. 2.2.2.3. These non-zero coefficients in the background area were due to noise effect. The normalized histogram can be approximately fit by a Laplacian pdf with zero mean and a standard deviation of 1.5. Coding these coefficients at the quantization step of 10 results a 0.0829 bits/pixel according to Eq. 2.14. The bandwidth for coding all the pixels in the background area can be calculated by Eq. 2.19. The number of the background pixels equals to the total number of pixels less that of the pixels in VOP_3. That is, $720 \times 480 - 15518 = 330082$, in this example. The total number of the background pixels in the 14 P-frames is then 4621148, leading to 383093 bits for the coding. Therefore, in one second, it yields a bit rate of 766.2 Kbps (Kilo-bit per second), only for coding the background noise. This analytical result is compatible with our experimental results reported in the later chapters.

(a)

(b)

Figure 11: The sample VOPs obtained from a patient monitoring video. Each frame has a dimension of 720×480. The top panel shows a VOP_2 that represents the foreground. The bottom panel shows a VOP_3, which represents the parts of the foreground that were moving within a time interval of 0.5 seconds. There were 15518 pixels enclosed in VOP_3.

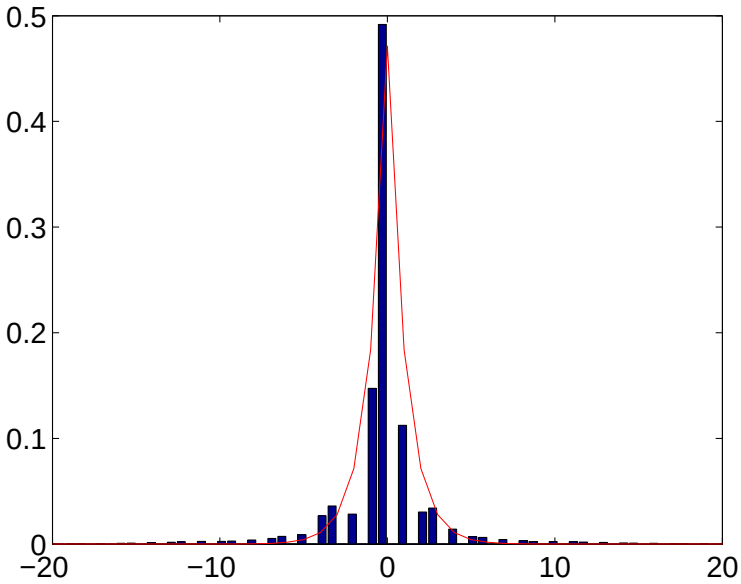

Figure 12: The normalized histogram (bar plot) of the DCT coefficients in the background area, which were transformed from the frame differences. It can be approximately fit by a Laplacian pdf with a mean of zero and a standard deviation of 1.5.

In summary, from the rate-distortion analysis on hybrid coding, we find that the uninterested content such as the background noise may create a significant amount of disbursement in bandwidth or storage. This expense can be greatly reduced by employing object-based coding, which discriminates disturbances from content of interest.

2.3 SHAPE CODING

2.3.1 Introduction

For an object-based coding scheme, the overall coding efficiency is the union of both the texture coding and the shape coding. Shape information is delineated by binary or grey scale images called "alpha planes", which represent single video object and multiple video

objects respectively. The coding of alpha planes is referred as shape coding. While grey scale alpha planes are encoded by DCT transform coding after motion compensation, similar to texture coding as previously described, the binary shape coding is unique in MPEG-4. During the development of MPEG-4 standard, a few methods for coding of the binary alpha planes have been proposed, including Chain coding [43, 47, 48], Quad-tree coding [42], Modified Modified Reed (MMR) coding [44], Content-based Arithmetic Encoding (CAE) [45, 46], Baseline-based coding [49] and Skeleton-based coding [50], etc.. All these methods are capable of being both lossless and lossy in the coding. Since the lossy mode makes the analysis of the overall distortion (i.e. the error caused by both shape coding and texture coding) rather complex, we are more interested in analyzing the lossless shape coding. In this section, we investigate the lossless mode aiming at an estimation of the bit rate of coding an arbitrarily shaped object.

2.3.2 Entropy estimation for contour coding

Let us consider a single solid (i.e. with no holes) region with arbitrary shape. Essentially, to encode the shape of such a region, one needs only to encode its contour. Therefore, in this scenario, the bit rate of binary shape coding can be estimated by the entropy of the arbitrarily shaped contour.

Entropy calculation for contour coding has been studied in the literature, including coding of 4-connected contour [47] and the performance of coding line drawings [48]. In this thesis, we provide a simple derivation of the entropy estimation for coding 4-connected contours.

To code a contour, one can start from coding the absolute position of any node on the contour, and then code the relative positions of the other nodes. For any node c_j on a 4-connected contour, its position relative to its predecessor c_{j-1} and to its successor c_{j+1} can be described by the edges connecting them. There are three types of edges going from c_j to c_{j+1}: "S" denoting "straight forward", "R" denoting "right turn" and "L" denoting "left turn", as shown in Fig. 13. Coding a sequence of nodes then equals to coding a string containing the three symbols.

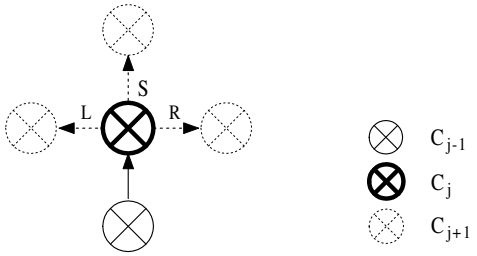

Figure 13: Three types of edges from c_j to c_{j+1}.

Assuming that the three types of edges appear equally often, one has an the entropy per node equal to $\log_2 3$. However, this estimation does not utilize the following constraints that, in order for the nodes to be on the contour of a region, there should be no node 4-connected to more than two others. This leads to the invalidation of "L-L" and "R-R" in the symbol string. Therefore, the only possible connections are the types shown in Fig. 14.

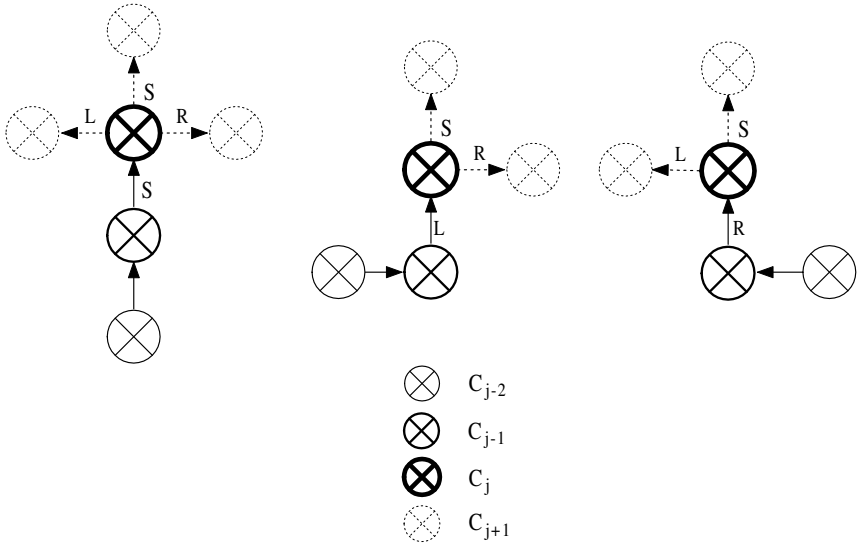

Figure 14: The possible configurations of edges from c_{j-1} to c_{j+1} under the constraints that "no node should be 4-connected to more than two others".

Henceforth, the number of possible edges connecting c_j and c_{j+1} has the following form,

$$\bar{n} = P(S) \cdot |\{S, L, R\}| + P(L) \cdot |\{S, R\}| + P(R) \cdot |\{S, L\}|, \tag{2.20}$$

where $\{\cdot\}$ denotes a set, $|\{\cdot\}|$ denotes the number of elements in a set, $P(\cdot)$ denotes the probability of the element being chosen. To obtain $P(\cdot)$, let us assume the three types of connections in Fig. 14 appear equally often in a contour, then in the three sets, i.e. $\{S, L, R\}$, $\{S, L\}$ and $\{S, R\}$, we see that $P(S) = \frac{3}{7}$, $P(L) = \frac{2}{7}$ and $P(R) = \frac{2}{7}$. Thus, we have

$$\begin{aligned} \bar{n} &= \frac{3}{7}|\{S, L, R\}| + \frac{2}{7}|\{S, R\}| + \frac{2}{7}|\{S, L\}| \\ &= \frac{17}{7}. \end{aligned} \tag{2.21}$$

And the entropy per node is

$$\begin{aligned} E_0 &= \log_2 \bar{n} \\ &= 1.28 \end{aligned} \tag{2.22}$$

It is seen that given the number of nodes on the contour, the entropy of a contour can be calculated. This entropy can be utilized to estimate the bit rate in coding the shape of a video object for 4-connected contours,

$$R^s_{vo_i}(n) = \sum_{k=nF_{vo_i}}^{(n+1)F_{vo_i}-1} 1.28 K_{4i}(k) \tag{2.23}$$

where n is time point in unit of second, K_4 is the length of a 4-connected contour and F_{vo_i} is the frame rate associated with the video object.

In practice, the contour of an object can be differentiated with respect to time, if temporal correlation is considered. Motion compensation can be carried out on contours in consecutive alpha planes. If the residual is less than a chosen threshold, which suggests that the two contours are similar, the residual will be coded, instead of the complete contour. Therefore, the length (number of nodes) of the coded contour can be much smaller than K_4. With this respect, we may consider Eq. 2.23 as an upper bound of the bit rate in shape coding.

2.4 DISCUSSION ON THE OVERALL CODING EFFICIENCY

We've discussed that in texture coding, object-based coding exploits content redundancy and is superior to frame-based coding in terms of bit rate reduction. However, shape coding, which is unique to object-based coding, has to be performed. The overall bit rate R_{vo} for object-based coding is the union of R^t_{vo} and R^s_{vo}, which denote the bit rate for texture coding and shape coding of video objects respectively. For frame-based coding, the bit rate, denoted by R_f, is equivalent to the addition of R^t_{vo} and R^u_f (the bit rate of the texture coding of uninterested content). Essentially, to compare the coding efficiency of object-based coding and frame-based coding, we only need to compare R^s_{vo} with R^u_f.

Similar to Eq. 2.19, R^u_f can be expressed in the following form,

$$R^u_f(n) = \eta \sum_{k=nF}^{(n+1)F-1} \sum_{r,c \in \Phi_u(k)} B_M(r,c,k), \tag{2.24}$$

where n is time point in seconds, F is the number of frames per second, Φ_u is the support region of uninterested content, N is the number of pixels in a block, r, c are the coordinates of a block, $B_M(r,c,k)$ is the number of bits of coding a block (given in Eq. 2.18), and η is the parameter associated with video format. Assuming that the frame difference, caused by noise, is statistically stationary, we have B_M as a constant, and R^f_u can be approximated by

$$\begin{aligned} R^u_f(n) &= \eta \frac{1}{N} B_M \sum_{k=nF}^{(n+1)F-1} S_u(k) \\ &= \eta \frac{1}{N} B_M F \bar{S}_u(n) \end{aligned} \tag{2.25}$$

where N is the number of pixels in a block, S_u is the number of uninterested pixels in one video frame, and $\bar{S}_u$ is the average of S_u in one second. Let S_f denote the total number of pixels in a video frame, and $\bar{S}_{vo}$ the average number of pixels contained in video objects. We have $\bar{S}_u(n) = S_f - \bar{S}_{vo}(n)$.

Now, applying Eq. 2.23, we have

$$R^s_{vo}(n) = \sum_i \sum_{k=nF_{vo_i}}^{(n+1)F_{vo_i}-1} EK_i(k), \tag{2.26}$$

where $E = 1.28$ for a 4-connected contour, and K is the number of nodes on the contour. In general, F_{vo_i} can be different and less than F. Therefore, we have

$$R^s_{vo}(n) \leq EF\bar{K}(n), \tag{2.27}$$

where $\bar{K}$ is the average length of the contours enclosing all the video objects in one second.

We can denote the area enclosed in the contours with $\bar{S}_{vo}$, considering each pixel as a unit square. Let $\bar{S}_{vo}(n) = \alpha(n)\bar{K}^2(n)$, where α is defined as the compactness of shape. We know that α should be ranged $0 \sim \frac{1}{4\pi}$, where $\frac{1}{4\pi}$ is achieved when the boundary is a circle. Since the shape of a video object may be arbitrary, we may assume α to be uniformly distributed, which gives the mean $\bar{\alpha} = \frac{1}{8\pi}$.

To compare the coding efficiency, let $R^u_f(n) = EF\bar{K}(n)$, and according to Eqs. 2.25 and 2.27, we have

$$\begin{aligned}
& R^u_f(n) = EF\bar{K}(n) \\
\Rightarrow & \eta\frac{1}{N}B_M F(S_f - \bar{S}_{vo}(n)) = EF\bar{K}(n) \\
\Rightarrow & \alpha(n)\bar{K}^2(n) + \frac{E}{\eta\frac{1}{N}B_M}\bar{K}(n) - S_f = 0 \\
\Rightarrow & \bar{K}^*(n) = \frac{-\frac{NE}{\eta B_M} + \sqrt{(\frac{NE}{\eta B_M})^2 + 4\alpha(n)S_f}}{2\alpha(n)}.
\end{aligned} \tag{2.28}$$

When $\bar{K}(n) < K^*(n)$ we have $R^s_{vo}(n) < R^u_f(n)$.

To evaluate $K^*(n)$, we apply the mean value of $\alpha(n)$, i.e. $\bar{\alpha} = \frac{1}{8\pi}$, to Eq. 2.28 and obtain $\bar{K}^* = \frac{-\frac{NE}{\eta B_M} + \sqrt{(\frac{NE}{\eta B_M})^2 + 4\bar{\alpha}S_f}}{2\bar{\alpha}}$. As a realistic example, let $N = 64$, $E = 1.28$ (4-connected contour), $\eta = 1.5$ (YUV12), $S_f = 352\times288$ (CIF), and $B_M = 64\times0.083$, i.e. the experimental result shown in Fig. 9 at quantization step equal to 10. With all these settings, we obtain $\bar{K}^* = 1.47 \times 10^3$. The average area enclosed by $\bar{K}^*$ is 8.6×10^4, a 84.8% of S_f. Essentially, this predicts that the average cost of shape coding equals to that of coding the uninterested content (disturbances) which covers an area of 15.2% of the entire frame. When uninterested content is more than 15.2% in a frame, the cost of coding the disturbances is more than the cost of coding the shape information. In this scenario, object-based coding outperforms frame-based in the coding efficiency. For our application, we have observed that the patient

usually occupies a region less than half of the entire frame. Henceforth, we expect an easy improvement of the coding efficiency via object-based coding schemes.

3.0 NEW CHANGE DETECTION MODELS FOR VIDEO SEGMENTATION

3.1 INTRODUCTION

Change detection is carried out by comparing two or more images to distinguish their differences caused by changes of image contents from those by irrelevant disturbances. The applications of change detection are broad, including video segmentation [8, 13, 14], remote sensing [65, 66], medical diagnosis [26, 27, 28, 29], and traffic assistance [30, 31, 32], etc.. In video segmentation, change detection is usually utilized to calculate temporal transition by differentiating two adjacent frames. This transition usually leads to a preliminary segmentation result. The advantages of change detection over other transition detection techniques include: 1) low computational cost, 2) capability of handling object's appearance/disappearance, and 3) no requirements on rigid motion. However, the results from change detection do not usually provide precise segmentation masks. Postprocessing is necessary to refine the output of a change detection module. This is commonly conducted by combining spatial domain features, such as edges and color homogeneity, with the change detection result.

With all the concerns, much research effort has been devoted to developing change detection algorithms aimed at robustness [52, 51, 53, 55, 54, 64, 65, 66]. Promising results have been reported in recent literature. However, it is still an open problem for a change detector to gain sensitivity of small changes at the presence of considerable disturbances. In Chapter 1, the models in major categories have been reviewed. The conventional methods [51, 52, 53] detect changes by thresholding. As a global threshold is not sufficiently effective in terms of false detections, adaptive thresholding approaches and optimization methods

have been studied [54, 65, 66]. While thresholding approaches have much less computational complexity, optimization methods may be more resilient to noise. Therefore, we explore both branches for enhancements in change detection algorithms. In this chapter, we contribute two new change detection models which show significant improvements over the conventional ones.

3.2 CHANGE DETECTION BASED ON MRF AND MFT

3.2.1 Why Markov random field (MRF)?

The major reason to apply MRF to change detection is to incorporate contextual information in decision making. A dominant category of change detection is single-threshold-based approaches, which utilize certain test statistics adapted to noise and image models [52, 51, 53, 55, 64] to make decisions. A critical problem with these approaches is to determine the threshold. False alarms are caused when the threshold is not large enough, while signal is mis-detected if the threshold is overestimated. The reason is that the change detection is performed locally at each pixel, but the single threshold to be applied is determined globally. In other words, this threshold is non-adaptive to the properties of a local region. Better results can be achieved if the threshold is increase/decreased when the contextual information of the local region suggests the test pixel stay "unchanged"/"changed". A simple example would be the constraint of smoothness, which means that the neighboring pixels of a "changed"/"unchanged" pixel are likely to be in the same mood too.

MRF is a well known tool for modeling these contextual constraints. Considering a change detection mask (CDM) as a 2-D random array, making decision on each pixel becomes a problem of finding an appropriate configuration of the random field. The prior knowledge of both "unchanged" and "changed" regions can be enforced by the associated energy functions that are defined to represent the potential of a pixel being in the corresponding status ("unchanged"/"changed"). By minimizing these energy functions, the optimal CDM in the maximum *a posteriori* (MAP) sense can be obtained. In another word, change detection

can be carried out at a strict optimization point of view. In the literature, the optimization process can be performed by, for example, simulated annealing and iterative conditional mode algorithms [59], [60]. The former aims at providing the global extremum, but requires extensive computation; the latter reduces the computational cost, but may converge to a local extremum. We adopt the mean field theory (MFT) approach as studied recently in [68],[70], which trades off between these two approaches.

3.2.2 Background theories

Fundamentals of the MRF and the MFT are briefly introduced in this section.

3.2.2.1 Markov Random Field Theory in Change Detection Let $\bar{F} = \{F_{1,2}, ..., F_{i,j}, ..., F_{m,n}\}$ be a 2-D random array, where $F_{i,j}$, $1 \leq i \leq m$, $1 \leq j \leq n$, is a random variable at site (i, j). Let $\Omega = \{(i, j)|1 \leq i \leq m, 1 \leq j \leq n\}$ be the set of all sites. Let frame $\bar{f} = \{f_{i,j}, (i, j) \in \Omega\}$ be a realization of $\bar{F}$. Let $p(\bar{f})$ denote the joint pdf of $\bar{F} = \bar{f}$, where $p(\bar{f}) = p\{\bar{F} = \bar{f}\} = p\{F_{i,j} = f_{i,j}, (i, j) \in \Omega\}$. Then, with the same notation, $\bar{F}$ is an MRF if: (1) $p(\bar{f}) > 0, \forall \bar{f} \in \bar{F}$, and (2) $p(f_{i,j}|f_{\Omega'}) = p(f_{i,j}|f_{N_{i,j}})$, where $\Omega' = \Omega - (i, j)$, with symbol "$-$" denoting exclusion, and $N_{i,j} = \{(i', j')|(i-i')^2+(j-j')^2 \leq k, (i', j') \in \Omega'\}$, with k being a positive integer. $N_{i,j}$ defines the set of the k-th order neighboring sites of (i, j). With the definition of $N_{i,j}$, a clique, denoted by c, is defined as a set containing single or multiple sites that are connected within $N_{i,j}$, $(i, j) \in \Omega$. Fig. 15 illustrates an example of cliques of a first-order neighborhood, where c may be a collection of single-sites or double-sites. It was introduced in [61] that the joint pdf $p(\bar{f})$ may be approximated by the Gibbs distribution

$$p(\bar{f}) = \frac{e^{-\frac{1}{T}U(\bar{f})}}{\sum_{\bar{f}} e^{-\frac{1}{T}U(\bar{f})}}, \tag{3.1}$$

where T is a constant and U is an energy function of the MRF, given by

$$U(\bar{f}) = \sum_{c} V_c(\bar{f}) \tag{3.2}$$

with V_c being the clique potential or clique function. The V_c functions represent contributions to the total energy from single-site cliques, double-site cliques and so on. Note that (3.1)

and (3.2) reflect the fact that the joint probability density function $p(\bar{f})$ is determined by the local activities, namely, the clique potentials.

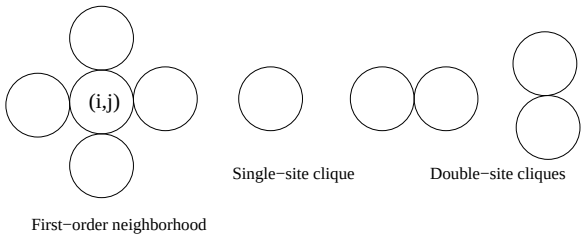

Figure 15: A first-order neighborhood system (first panel), single-site (second panel) and double-site cliques (third and fourth panels)

Considering the first-order neighborhood, we may rewrite (3.2) into the following form [63]

$$\begin{aligned} U(\bar{f}) = \sum_{(i,j)} \{ & V_{(i,j)}(f_{i,j}) + V_{\{(i,j),(i+1,j)\}}(f_{i,j}, f_{i+1,j}) \\ & + V_{\{(i,j),(i,j+1)\}}(f_{i,j}, f_{i,j+1})\}, \end{aligned} \tag{3.3}$$

where the first, second, and third term are single-site, horizontal double-site and vertical double-site clique potentials, respectively. Notice that for a double-site clique $\{(i,j),(i',j')\}$, the associated clique potentials $V_{\{(i,j),(i',j')\}}(f_{i,j}, f_{i',j'})$ and $V_{\{(i',j'),(i,j)\}}(f_{i',j'}, f_{i,j})$ are equal. Therefore, (3.3) may be rearranged into

$$\begin{aligned} U(\bar{f}) &= \sum_{(i,j)} \{ V_{c_1}(f_{i,j}) + \frac{1}{2} \sum_{(i',j') \in N_{i,j}} V_{c_2}(f_{i,j}, f_{i',j'}) \} \\ &= \sum_{(i,j)} U_{i,j}(f_{i,j}), \end{aligned} \tag{3.4}$$

where c_1 and c_2 are single-site and double-site cliques in the defined neighborhood, and $U_{i,j}(f_{i,j})$ is the energy function associated with site (i,j). As pointed out in [63], if $p(\bar{f})$ is a posterior distribution, minimizing the energy function $U(\bar{f})$ yields an *Maximum A Posteriori* (MAP) estimate of the joint pdf $p(\bar{f})$.

3.2.2.2 Mean Field Theory To make the MRF theory more practical, we need to introduce the MFT. From the description of the MRF, we know that the value assigned to a random variable in the MRF is affected by the values at its neighboring sites, which are further dependent on their neighbors. One way to calculate the interaction between one site and it neighbors is to apply the MFT [67][68], which assumes that the impacts from the neighbors can be approximated by an average field. Let us denote the mean field for site (i,j) by $f_{i,j}^{\text{mf}}$. As a result, if the first-order neighborhood is considered, one may write the energy function related to site (i,j) in the following form [68]

$$U_{i,j}^{\text{mf}}(f_{i,j}) = V_{c_1}(f_{i,j}) + \sum_{(i',j')\in N_{i,j}} V_{c_2}(f_{i,j}, f_{i',j'}^{\text{mf}}), \tag{3.5}$$

where $V_{c_1}(\cdot)$ and $V_{c_2}(\cdot,\cdot)$ are potential functions of single-site and double-site cliques respectively; and, $f_{i',j'}^{\text{mf}}$ is the mean field for $f_{i',j'}$. Then, the marginal distribution of the MRF at site (i,j) may be approximated by [68]

$$p(f_{i,j}) = \frac{1}{\sum_{f_{i,j}} e^{-\frac{1}{T}U_{i,j}^{\text{mf}}(f_{i,j})}} e^{-\frac{1}{T}U_{i,j}^{\text{mf}}(f_{i,j})}. \tag{3.6}$$

As seen from (3.4) and (3.5), the energy function is decomposed into local computations, where each site is treated independently. Therefore, the joint pdf $p(\bar{f})$ can be approximated by

$$p(\bar{f}) \approx \prod_{i,j} p(f_{i,j}) \tag{3.7}$$

Then, maximizing $p(\bar{f})$ is equivalent to maximizing each $p(f_{i,j})$, or, to minimizing the corresponding $U_{i,j}^{\text{mf}}(f_{i,j})$.

In order to evaluate $U_{i,j}^{\text{mf}}(f_{i,j})$, the mean field values $f_{i',j'}^{\text{mf}}$ at the neighboring sites (i',j') within $N_{i,j}$ must be computed. The general way to calculate a mean field value is by the following form

$$f_{i,j}^{\text{mf}} = \sum_{f_{i,j}} f_{i,j} \cdot p(f_{i,j}). \tag{3.8}$$

Note that (3.8) requires the evaluation of $p(f_{i,j})$, henceforth, $U_{i,j}^{\text{mf}}(f_{i,j})$. Therefore, the computation of the mean field value is usually carried out by iteration that stops when the change of the results from two consecutive iterations is sufficiently small.

3.2.3 MRF Change Detection Method

3.2.3.1 MAP-MRF in Change Detection Let us denote the CDM by $\bar{H} = \{H_{1,2}, ..., H_{i,j}, ..., H_{m,n}\}$, and $\bar{h} = \{h_{1,2}, ..., h_{i,j}, ..., h_{m,n}\}$ a configuration of $\bar{H}$, where $h_{i,j} \in \{-1, 1\}$, $(i, j) \in S$ with "-1" denoting "unchanged" and "1" denoting "changed". Then, given two frames $\bar{f}^{(0)}$ and $\bar{f}^{(1)}$, our goal is to find the optimal $\bar{h}^*$ in the MAP sense, such that

$$\begin{aligned} \bar{h}^* &= \text{argmax}_{\bar{h}} p(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)}) \\ &= \text{argmax}_{\bar{h}} \frac{p(\bar{f}^{(1)}|\bar{f}^{(0)}, \bar{h}) \cdot p(\bar{h}|\bar{f}^{(0)})}{p(\bar{f}^{(1)}|\bar{f}^{(0)})} \\ &= \text{argmax}_{\bar{h}} p(\bar{f}^{(1)}|\bar{f}^{(0)}, \bar{h}) \cdot p(\bar{h}|\bar{f}^{(0)}) \end{aligned} \tag{3.9}$$

Applying MRF assumption on both $\bar{F}$ and $\bar{H}$, maximizing $p(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)})$ with respect to $\bar{h}$ is equivalent to minimizing its energy function $U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)})$. This, as suggested by (3.9), can be accomplished by minimizing the energy functions $U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)})$ and $U(\bar{h}|\bar{f}^{(0)})$, which are associated with $p(\bar{f}^{(1)}|\bar{f}^{(0)})$ and $p(\bar{h}|\bar{f}^{(0)})$, respectively. $U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)})$ addresses the potential of the likelihood between $\bar{f}^{(1)}$ and $\bar{f}^{(0)}$ with the knowledge of $\bar{h}$, i.e. whether the sites are changed. And, $U(\bar{h}|\bar{f}^{(0)})$ is always considered to represent the spatial domain constraints, e.g., the smoothness or similarity between neighboring sites. Therefore, a general form of the prior model of these energy functions is

$$U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)}) = \gamma_f U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)}) + \gamma_h U(\bar{h}|\bar{f}^{(0)}) \tag{3.10}$$

where γ_f and γ_h are regularization parameters. The larger the regularization parameter values, the more the corresponding constraint is emphasized.

Equivalently, we can write (3.10) by

$$U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)}) = \gamma_f [U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)}) + \gamma U(\bar{h}|\bar{f}^{(0)})], \tag{3.11}$$

where $\gamma = \frac{\gamma_h}{\gamma_f}$. It is noticed that to minimize $U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)})$ with respect to $\bar{h}$ is equivalent to minimizing $U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)}) + \gamma U(\bar{h}|\bar{f}^{(0)})$. Therefore, we define the energy function in the following form,

$$U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)}) = U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)}) + \gamma U(\bar{h}|\bar{f}^{(0)}). \tag{3.12}$$

In order to design the above energy functions, one needs to employ the prior knowledge. In our application, the prior knowledge includes the distribution of the frame difference in the absence/presence of changes and the assumption of the similarity between immediate sites (pixels). There are no specific routines in designing potential functions. In general, as indicated in [62], the formulation of a potential function should keep consistency with the prior knowledge: if the formulation of the regions in a clique tends to be consistent with the prior knowledge, the value of the energy function decreases; otherwise, the value increases.

In change detection, we interpret $U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)})$ as the sum of single-site clique potentials, which is

$$\begin{aligned} U(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)}) &= \sum_{c_1} V_{c_1}(\bar{f}^{(1)}|\bar{h}, \bar{f}^{(0)}) \\ &= \sum_{i,j} V_{c_1}(f_{i,j}^{(1)}|h_{i,j}, f_{i,j}^{(0)}) \end{aligned} \tag{3.13}$$

where, V_{c_1} is selected to be

$$V_{c_1}(f_{i,j}^{(1)}|h_{i,j}, f_{i,j}^{(0)}) = -ln(p(d_{i,j} \mid h_{i,j})) \tag{3.14}$$

which is the negative of the natural logarithm of the pdf of the absolute frame difference $d_{i,j} = |f_{i,j}^{(1)} - f_{i,j}^{(0)}|$ at site $(i,j) \in S$, given the knowledge of $h_{i,j}$. Therefore, if $d_{i,j}$ is consistent with the prior belief, the conditional probability will be high. As a result, its logarithm value will be low, and vice versa, as required by the design rules. Choosing the natural logarithm is instinctive. First, more penalty would be assigned to smaller probability, e.g., when probability is close to zero, the value of energy function would be extremely large. Second, considering $p(\bar{f}^{(1)}|\bar{f}^{(0)}, \bar{h} = -1)$, which is equivalent to the pdf of frame difference caused by noise, we may assume $p(\bar{f}^{(1)}|\bar{f}^{(0)}, \bar{h} = -1) = \prod_{i,j} p(d_{i,j} \mid h_{i,j} = -1)$, or, $\prod_{i,j} Z_{i,j} \cdot e^{-\frac{1}{T}V_{c_1}(f_{i,j}^{(1)}|h_{i,j}=-1, f_{i,j}^{(0)})} = \prod_{i,j} p(d_{i,j} \mid h_{i,j} = -1)$, where $Z_{i,j}$ are normalization constants. Furthermore, if the noise distribution $p(d_{i,j} \mid h_{i,j} = -1)$ also has an exponential form, such as Gaussian and Laplacian, we may reasonably take the natural log on both sides of the above equation to get the potential function. For the case of $h_{i,j} = 1$, i.e., with the presence of change, the independence assumption may not hold in general. However, this assumption can be accepted as a reasonable simplification to trade off computational complexity [66].

Therefore, the above reasoning may also apply to the case $h_{i,j} = 1$. The collection of prior knowledge will be described in section 3.2.4.

The other energy function $U(\bar{h}|\bar{f}^{(0)})$ in (3.10) addresses the contextual constraints on the neighboring sites. This can be explained as follows: with the knowledge of $\bar{f}^{(0)}$, we want to obtain $\bar{h}$ that complies with the properties of $\bar{f}^{(0)}$, for example, the continuity of $\bar{h}$ if we assume that $\bar{f}^{(0)}$ is smooth. Based upon this reasoning, we define

$$\begin{aligned} U(\bar{h}|\bar{f}^{(0)}) &= \sum_{i,j} \sum_{c_2 \subset N_{i,j}} V_{c_2}(\bar{h}|\bar{f}^{(0)}) \\ &= \sum_{i,j} \{\frac{1}{2} \sum_{(i',j') \in N_{i,j}} V_{c_2}(h_{i,j}, h_{i',j'})\} \end{aligned} \tag{3.15}$$

where c_2 is a double-site clique in a first-order neighborhood $N_{i,j}$ at site $(i,j) \in S$. The scaling factor $\frac{1}{2}$ has been explained in (3.3) and (3.4). The clique potential $V_{c_2}(\cdot,\cdot)$ is defined as

$$V_{c_2}(h_{i,j}, h_{i',j'}) = -ln(1 - 0.5|h_{i,j} - \lambda \cdot h_{i',j'}|) \tag{3.16}$$

where $\lambda \in (0,1)$ is a constant representing the impact of site (i',j') on site (i,j) . The reasons behind this design are : (1) we want the state of site (i,j) to agree with its neighboring sites; (2) the logarithm form is consistent with that in (3.14). The term $1 - 0.5|h_{i,j} - \lambda \cdot h_{i',j'}|$ acts as a probability of the random variable at site (i,j) when its value agrees with those at its neighboring sites. Therefore, this definition also follows the design rules stated previously.

To minimize $U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)})$, we must evaluate the clique potential functions. A question now is how to calculate $V_{c_2}(h_{i,j}, h_{i',j'})$. As mentioned previously, we may apply MFT to simplify this calculation. If the first-order neighborhood system is assumed, we have the following approximation

$$U(\bar{h}|\bar{f}^{(0)}) \approx \sum_{i,j} \sum_{(i',j') \in N_{i,j}} V_{c_2}(h_{i,j}, h^{\mathrm{mf}}_{i',j'}) \tag{3.17}$$

where

$$V_{c_2}(h_{i,j}, h^{\mathrm{mf}}_{i',j'}) = -ln(1 - 0.5|h_{i,j} - \lambda \cdot h^{\mathrm{mf}}_{i',j'}|). \tag{3.18}$$

Combining (3.10) $\sim$ (3.18), we have

$$U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)}) \approx \sum_{i,j} U^{\mathrm{mf}}_{i,j}(h_{i,j}|f^{(0)}_{i,j}, f^{(1)}_{i,j}) \tag{3.19}$$

where

$$U_{i,j}^{\mathrm{mf}}(h_{i,j}|f_{i,j}^{(0)}, f_{i,j}^{(1)}) = -ln(p(d_{i,j} \mid h_{i,j}))- \\ [\gamma \sum_{(i',j')\in N_{i,j}} ln(1 - 0.5|h_{i,j} - \lambda \cdot h_{i',j'}^{\mathrm{mf}}|)]. \tag{3.20}$$

Essentially, to minimize $U(\bar{h}|\bar{f}^{(0)}, \bar{f}^{(1)})$, we only need to evaluate $U_{i,j}^{\mathrm{mf}}(\cdot)$ at each site (i, j), and choose $h_{i,j}$ between -1 and 1 to render a smaller value of $U_{i,j}^{\mathrm{mf}}(\cdot)$.

3.2.4 The MRF Change Detection Algorithm

Eq. (3.20) requires evaluation of $p(d_{i,j}|h_{i,j}), (i, j) \in S$. Instead of collecting the pdf for each site, we utilize the same pdf, denoted by $p(d|h)$, for all sites, where d and h have the same sample spaces as $d_{i,j}$ and $h_{i,j}$ respectively. This choice is motivated from a practical point of view, since it would be extremely expensive to allocate memory for $p(d_{i,j}|h_{i,j})$ for each $(i, j) \in S$. When $h(i, j) = -1$, this approximation can be justified because the value differences of unchanged sites are driven by noise, which is usually considered to be independently and identically distributed. For moving pixels, the above assumption is not true in general. However, if we assume that each pixel may experience the same or similar amounts of motion, the validity of using $p(d|1)$ for all the sites is also justifiable.

To train $p(d| - 1)$, we utilize the video segments containing motionless scenes. This is relatively easy to accomplish in many applications, such as in surveillance and teleconference videos. In general, it is difficult to train $p(d|1)$; however, it is possible to train a prototype for specific applications. Practically, we adopt the following strategy to calculate $p(d|1)$: first, $p(d|1)$ is initialized to be a uniform distribution across the entire range of its sample space, i.e. $p(d|1) = \frac{1}{L+1}, d \in [0, L]$ for a discrete case; then, starting with the initial value, we adapt $p(d|1)$ during a detection process, using the following equation

$$p^{(r)}(d|1) = (1 - \epsilon \cdot \rho) \cdot p^{(r-1)}(d|1) + \epsilon \cdot \rho \cdot p_{d|1}^{(r)}, \tag{3.21}$$

where $p^{(r)}(d|1)$ and $p^{(r-1)}(d|1)$ are the pdf $p(d|1)$ adapted from frame 1 to frames r and $r-1$ respectively, $p_{d|1}^{(r)}$ is the pdf of the "changed" pixels contained in frame r, ρ is the ratio of the number of "changed" pixels to the total number of pixels in that frame, and $\epsilon \in (0, 1)$ is a control parameter. The term ρ reflects the intuition that the more "changed" pixels

there are, the more $p(d|1)$ should be adapted. Parameter ϵ is designed to control the rate of adaptation.

An important question now is how the mean field value $h_{i,j}^{\mathrm{mf}}, (i,j) \in S$ is evaluated. As mentioned before, the mean field value is usually computed iteratively until it converges. As described in 3.2.2.2, with the local energy function $U_{i,j}^{\mathrm{mf}}(h_{i,j}|f_{i,j}^{(0)}, f_{i,j}^{(1)})$, $h_{i,j}^{\mathrm{mf}}$ can be evaluated by

$$h_{i,j}^{\mathrm{mf}} = \sum_{h_{i,j}} h_{i,j} \cdot \frac{e^{-\frac{1}{T}U_{i,j}^{\mathrm{mf}}(h_{i,j}|f_{i,j}^{(0)}, f_{i,j}^{(1)})}}{\sum_{h_{i,j}} e^{-\frac{1}{T}U_{i,j}^{\mathrm{mf}}(h_{i,j}|f_{i,j}^{(0)}, f_{i,j}^{(1)})}}. \tag{3.22}$$

Applying (3.20), we have

$$e^{-\frac{1}{T}U_{i,j}^{\mathrm{mf}}(h_{i,j}|f_{i,j}^{(0)}, f_{i,j}^{(1)})} = (p(d|h) \cdot (\prod_{(i',j')\in N_{i,j}} [1 - 0.5(h_{i,j} - \lambda h_{i',j'}^{\mathrm{mf}})])^{\gamma})^{\frac{1}{T}}. \tag{3.23}$$

Note that the computing time can be greatly reduced by using (3.23). The iteration continues until the following condition is satisfied:

$$\frac{1}{m \cdot n} \sum_{i,j} |h_{i,j}^{\mathrm{mf}}(k+1) - h_{i,j}^{\mathrm{mf}}(k)| < \theta \tag{3.24}$$

where, k is the index of iteration, $m \cdot n$ is the total number of pixels, and $\theta \in (0,1)$ is a chosen threshold.

With these assumptions and simplifications, we present an algorithm to implement the proposed model as follows:

- Step 1 : Load $p(d|-1)$ and initialize $p(d|1) = 1/256$, for $d = 0, 1, ..., 255$; Assign values to γ, λ, ϵ and θ.
- Step 2 : Take two frames $\bar{f}^{(0)}$ and $\bar{f}^{(1)}$, and calculate $\bar{d} = |\bar{f}^{(0)} - \bar{f}^{(1)}|$; Initialize mean field values $\bar{h}^{mf}$, where for each pixel (i,j), $h_{i,j}^{\mathrm{mf}} = 0$.
- Step 3 : For each pixel (i,j), evaluate (3.20) with $h_{i,j} = -1$ and 1, and calculate the new mean field value by (3.22) and (3.23).
- Step 4 : Evaluate the difference between the new mean field value and the previous one as defined in (3.24); If the difference is less than θ, then go to next step, otherwise go to step 3.

- Step 5 : For each pixel, if the local energy $U^{\mathrm{mf}}_{i,j}(h_{i,j} = -1|f^{(0)}_{i,j}, f^{(1)}_{i,j}) > U^{\mathrm{mf}}_{i,j}(h_{i,j} = 1|f^{(0)}_{i,j}, f^{(1)}_{i,j})$, then label pixel (i, j) "unchanged", otherwise "changed".
- Step 6: Update $p(d|1)$ by (3.21); Finish if all the frames are done, otherwise go to step 2.

3.2.5 ILLUMINATION INVARIANT APPROACH

In the previous sections, we have presented an MRF-MFT model to identify changes exclusively due to noise. The disturbance caused by illumination changes have not been addressed. This type of disturbance usually appears in images as visually noticeable changes, but are most of the time uninteresting and should be discriminated or excluded by a change detection algorithm. Recently, research [55] has been conducted to develop approaches with "illumination-invariant" features. In the following, we describe a new construction of an illumination-invariant change detection algorithm by using the proposed MRF-MFT model.

3.2.5.1 Shading Model The shading model [52, 58] formulates the gray level intensity of an image as the product of the illumination of a physical surface and its shading coefficients,

$$f_{i,j} = I_{i,j} S_{i,j}, \tag{3.25}$$

where (i, j) is a particular pixel representing a point on the physical surface, $f_{i,j}$ is the obtained intensity, $I_{i,j}$ is the illumination, and $S_{i,j}$ is the shading coefficient at (i, j). The shading coefficient is determined by a number of factors, such as the structure of physical surface, reflectance of the material, and angles of striking and reflected lights. A typical formulation of the shading coefficient was provided by Phong [57].

It is usually assumed that, for two given images containing the same objects, if there is no change in the physical structure of the object, the shading coefficient at the given location on two images are identical, i.e.,

$$S^{(0)}_{i,j} = S^{(1)}_{i,j}, \tag{3.26}$$

where the superscripts denote image indices. In addition, the illumination $I_{i,j}$ usually varies slowly in the spatial domain, which leads to the assumption that $I_{i,j}$ does not change within a small local region.

3.2.5.2 Illumination Invariant MRF-MFT Change Detection Considering both the shading model and noise, we may formulate the intensity at pixel (i, j) in image k by

$$f_{i,j}^{(k)} = I_{i,j}^{(k)} S_{i,j}^{(k)} + \eta_{i,j}^{(k)}, \tag{3.27}$$

where $\eta_{i,j}^{(k)}$ are assumed to be *i.i.d.* random variables due to noise. Therefore, the image difference can be modeled by

$$\hat{d}_{i,j} = (I_{i,j}^{(1)} S_{i,j}^{(1)} - I_{i,j}^{(0)} S_{i,j}^{(0)}) + (\eta_{i,j}^{(1)} - \eta_{i,j}^{(0)}). \tag{3.28}$$

Under the null hypothesis, namely, the object surface does not change, we have $S_{i,j}^{(0)} = S_{i,j}^{(1)}$, which leads to

$$\hat{d}_{i,j} = I_{i,j}^{(1)} S_{i,j}^{(1)} (1 - \mu_{i,j}) + (\eta_{i,j}^{(1)} - \eta_{i,j}^{(0)}), \tag{3.29}$$

where $\mu_{i,j} = I_{i,j}^{(0)} / I_{i,j}^{(1)}$ denotes the ratio of illumination on pixel (i, j) in the two images. If there is no illumination change, then $\mu_{i,j} = 1$.

In order to extend the previously described model with consideration of illumination, let us define an adjusted image difference to reflect the illumination change

$$e_{i,j} = |f_{i,j}^{(1)} - \frac{1}{\mu_{i,j}} f_{i,j}^{(0)}|. \tag{3.30}$$

Under the null hypothesis, we have

$$e_{i,j} = |\eta_{i,j}^{(1)} - \frac{1}{\mu_{i,j}} \eta_{i,j}^{(0)}|. \tag{3.31}$$

Now, the single clique function defined in (3.14) is changed to

$$V_{c_1}(f_{i,j}^{(1)} | h_{i,j}, f_{i,j}^{(0)}) = -ln(p(e_{i,j} | h_{i,j})). \tag{3.32}$$

If $\mu_{i,j}$ can be evaluated, so can the corresponding clique functions. A simple way is to use the image intensity values to estimate $\mu_{i,j}$. To do that, let us define

$$F_{i,j}^{(k)} = \frac{1}{M} \sum_{(p,q) \in W_{i,j}} f_{p,q}^{(k)}, \qquad k = 1, 2 \tag{3.33}$$

to compensate the noise effect, where $W_{i,j}$ is a window centered at pixel (i,j), and M is the number of pixels included in $\Omega_{i,j}$. If M is sufficiently large, we have

$$F_{i,j}^{(k)} \approx \frac{1}{M} \sum_{(p,q)\in W_{i,j}} I_{p,q}^{(k)} S_{p,q}^{(k)}, \qquad k = 1, 2. \tag{3.34}$$

Considering that the illumination is usually a slow changing variable in the spatial domain, we may assume it a constant within $W_{i,j}$. Consequently, we have

$$F_{i,j}^{(k)} \approx \frac{1}{M} I_{i,j}^{(k)} \sum_{(p,q)\in W_{i,j}} S_{p,q}^{(k)}, \qquad k = 1, 2. \tag{3.35}$$

Then we can use $F_{i,j}^{(k)}$ to obtain an estimated $\mu_{i,j}$ by the following,

$$\hat{\mu}_{i,j} = \frac{F_{i,j}^{(0)}}{F_{i,j}^{(1)}} = \frac{I_{i,j}^{(0)}}{I_{i,j}^{(1)}} \cdot \frac{\sum_{(p,q)\in W_{i,j}} S_{p,q}^{(0)}}{\sum_{(p,q)\in W_{i,j}} S_{p,q}^{(1)}} = \mu_{i,j} \cdot \frac{\sum_{(p,q)\in W_{i,j}} S_{p,q}^{(0)}}{\sum_{(p,q)\in W_{i,j}} S_{p,q}^{(1)}}. \tag{3.36}$$

Under the null hypothesis, $\frac{\sum_{(p,q)\in W_{i,j}} S_{p,q}^{(0)}}{\sum_{(p,q)\in W_{i,j}} S_{p,q}^{(1)}} = 1$, henceforth, $\hat{\mu}_{i,j} = \mu_{i,j}$.

As a result, we have

$$p(e_{i,j}|h_{i,j} = -1) = p(|\eta_{i,j}^{(1)} - \frac{1}{\hat{\mu}_{i,j}} \eta_{i,j}^{(0)}|). \tag{3.37}$$

Therefore, if the distribution of $\eta_{i,j}^{(k)}$ is known, $p(e_{i,j}|h_{i,j} = -1)$ can be evaluated. Because $\eta_{i,j}^{(k)}$ represents a noise variable, for simplicity, let us assume it obeys a Gaussian distribution with a zero mean and a variance of δ_η^2. Then, the function $\eta_{i,j}^{(1)} - \frac{1}{\hat{\mu}_{i,j}} \eta_{i,j}^{(0)}$ also has a Gaussian distribution with a zero mean and variance equal to $(1 + \frac{1}{\hat{\mu}_{i,j}^2})\delta_\eta^2$. Consequently, we have

$$p(e_{i,j}|h_{i,j} = -1) = \begin{cases} \frac{1}{\sqrt{2\pi(1+\frac{1}{\hat{\mu}_{i,j}^2})\delta_\eta^2}} & \text{if } e_{i,j} = 0, \\ \frac{2}{\sqrt{2\pi(1+\frac{1}{\hat{\mu}_{i,j}^2})\delta_\eta^2}} e^{-\frac{e_{i,j}^2}{2(1+\frac{1}{\hat{\mu}_{i,j}^2})\delta_\eta^2}} & \text{if } e_{i,j} > 0, \\ 0 & \text{otherwise} \end{cases} \tag{3.38}$$

Applying (3.38) to (3.32), we have the single clique function for "unchanged" pixels. For the "changed" case, $p(e_{i,j}|h_{i,j} = 1)$ can be calculated following the same procedure as described in 3.2.4, namely, being trained online. The adaptation of $p(e_{i,j}|h_{i,j} = 1)$ is still formulated by (3.21) except that that image difference d is replaced by e.

3.2.6 Experiments

As described previously, five controlling parameters T, γ, λ, ϵ, and θ are required. Table 1 lists typical values of these parameters, which were chosen experimentally and utilized for all the test sequences. In the following, we describe these parameters individually.

Table 1: Typical control parameters.

parameter	T	γ	λ	ϵ	θ
value	2	1	0.99	0.5	0.05

- T is called "temperature" in MRF based methods, e.g. simulated annealing algorithm [59]. This parameter determines the spread of the Gibbs distribution. The larger the T, the more it spreads. In simulated annealing, T is gradually decreased. However, as suggested by [69], a fixed T is able to render a satisfactory result while reducing the computational cost. Therefore, a constant T was utilized throughout our experiments.
- γ is a regularization parameter to balance the constraints introduced by different clique potentials. In our application, a large γ value emphasizes the smoothness constraint.
- λ models the impact between neighboring sites. In (3.16), $h_{i,j} - \lambda h_{i',j'}$ is utilized to represent the difference between neighboring sites (i, j) and (i', j'). The value of λ controls the degree of impact from (i', j').
- ϵ is utilized to control the adaptation of the pdf of d in the presence of change. The larger the value of ϵ, the more the pdf adapts to each CDM, and the faster the adaption to test data. However, considering the risk of false detection, we assign ϵ a moderate value.
- θ provides a stop threshold in the calculation of the mean field values.

3.2.6.1 Synthetic Data To evaluate the new change detection method quantitatively, we generated a synthetic image sequence by using MATLAB in the following way: a circle (with a radius of 20, line width of 3, both in pixels, and gray level intensity of 5) is plotted

in a frame; then, white Gaussian noise with mean 127 and standard deviation 1.6 is added to each frame. It should be noted that the signal-to-noise (SNR) ratio of the synthetic data, defined as $20log\frac{\text{circle intensity}}{\text{noise standard deviation}}$, is less than $10dB$, which is much lower than the SNR in most natural videos. The coordinates of the origins were randomly generated. Two pairs of sample frames are shown in Fig. 16. Let us denote the ground truth CDM by $\bar{h}^{(r)}$, the detected CDM by $\bar{h}^{(t)}$, and the set of sites with false labels by $S_e = \{(i,j)|h_{i,j}^{(r)} \neq h_{i,j}^{(t)}, (i,j) \in S\}$. The error rate is then defined as

$$E_r = \|S_e\|/\|S\| \tag{3.39}$$

where $\|S_e\|$ and $\|S\|$ denote the number of sites in S_e and S, respectively.

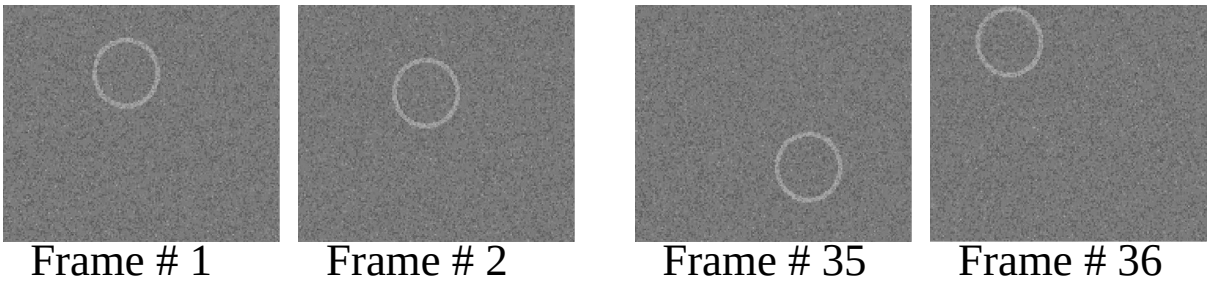

Figure 16: Two pairs of sample frames in the synthetic sequence: from left to right, frame 1, 2, 35 and 36 respectively.

Fig. 17 demonstrates the results of the synthetic data. The top row (a) shows the results obtained from frame 1 and 2. The left, middle and right panels in this row show the ground truth CDM, the detected CDM, and $p(d|h=-1)$ and the initial $p(d|h=1)$, respectively. Compared with the ground truth CDM, the detected CDM has visible false detections. However, with the adaption of $p(d|h=1)$, the false detections are reduced. As seen in Fig. 17(b), where the results were obtained from frame 35 and 36, the detected CDM (the middle panel) contains much less false detections. On the right panel it can be seen that $p(d|h=-1)$ was kept intact because of the assumption of stationary noise, but $p(d|h=1)$ was adapted to a bell-shaped function according to (3.21).

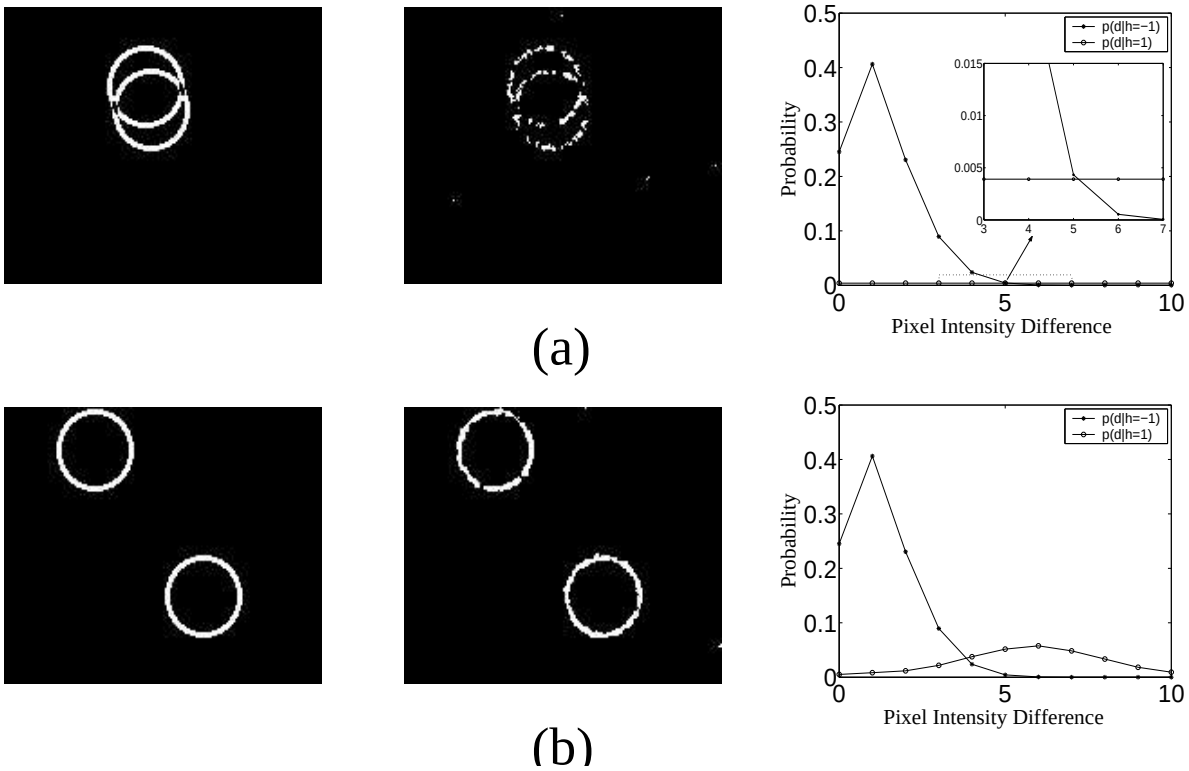

Figure 17: (a) The change detection results from frame 1 and 2. From left to right: the known CDM, the detected CDM and $p(d|h=-1)$ and initial $p(d|h=1)$, respectively. The subplot embedded in the right panel shows a close-look of the marked region (by the dash line). (b) The change detection results from frame 35 and 36. From left to right: the known CDM, the detected CDM and $p(d|h=-1)$ and $p(d|h=1)$ (adapted from frame $1 \sim 35$), respectively.

To demonstrate the robustness of the MRF approach, we compare it with two existing methods, "quadratic picture function" (QPF) method developed by Hsu etc. [51], and a novel method ("Method 3", abbreviated as M3 in the following) recently presented by De Geyter and Philips [71]. The parameters in the two methods were selected according to the original paper. In the QPF method, the threshold value of 5.76 was selected, corresponding to a significant level of 0.005. In the M3 method, the parameters α, β and z (see [71]) were set to 0.5, 0.9, and 3 respectively. The parameter k in M3 was tested from 2 to 5 and $k = 4$

was selected, which produced the best overall performance for the test sequences. These parameter values were utilized for all the test sequences (synthetic and natural). The results of QPF and M3 methods are illustrated in Fig. 18 (a) and (b) respectively. Compared with the CDM's shown in Fig. 17, these two methods appear to be more sensitive to the simulated noise. The error rates of the three methods are illustrated in Fig. 19, which shows that the MRF method performed better than the two existing methods in terms of less false detection. It is seen that the error rate of the MRF method decreases as frames 1 through 30 being processed, then becomes stable after that. The reason is that $p(d|h = 1)$ adapts gradually to the test data at the initial frames, and then becomes stationary. The adaptation speed is quite satisfactory for most common applications, as indicated by our results using other videos.

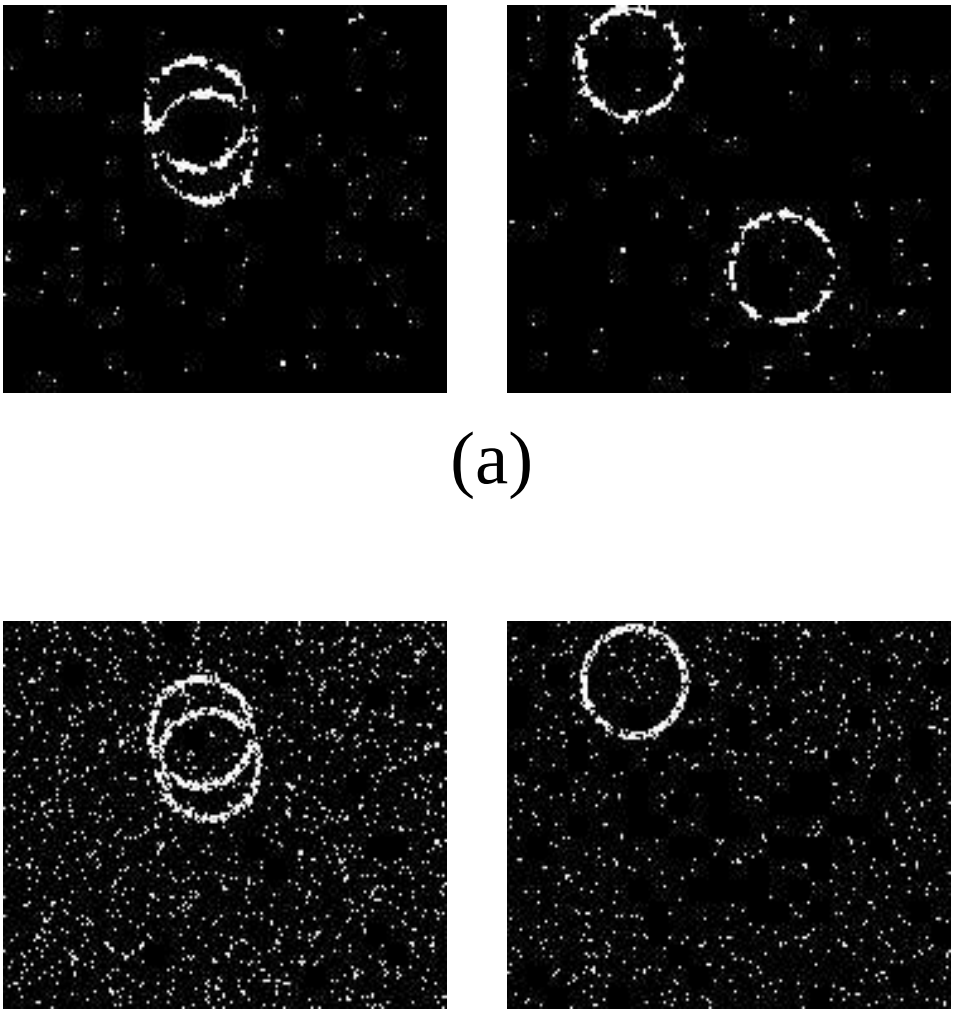

Figure 18: (a) The CDM's detected by "quadratic picture function" (QPF) method. Left panel: CDM from frame 1 and 2; Right panel: CDM from frame 35 and 36. (b) The CDM's detected by the method of De Geyter and Philips (M3 method). Left panel: CDM from frame 1 and 2; Right panel: CDM from frame 35 and 36.

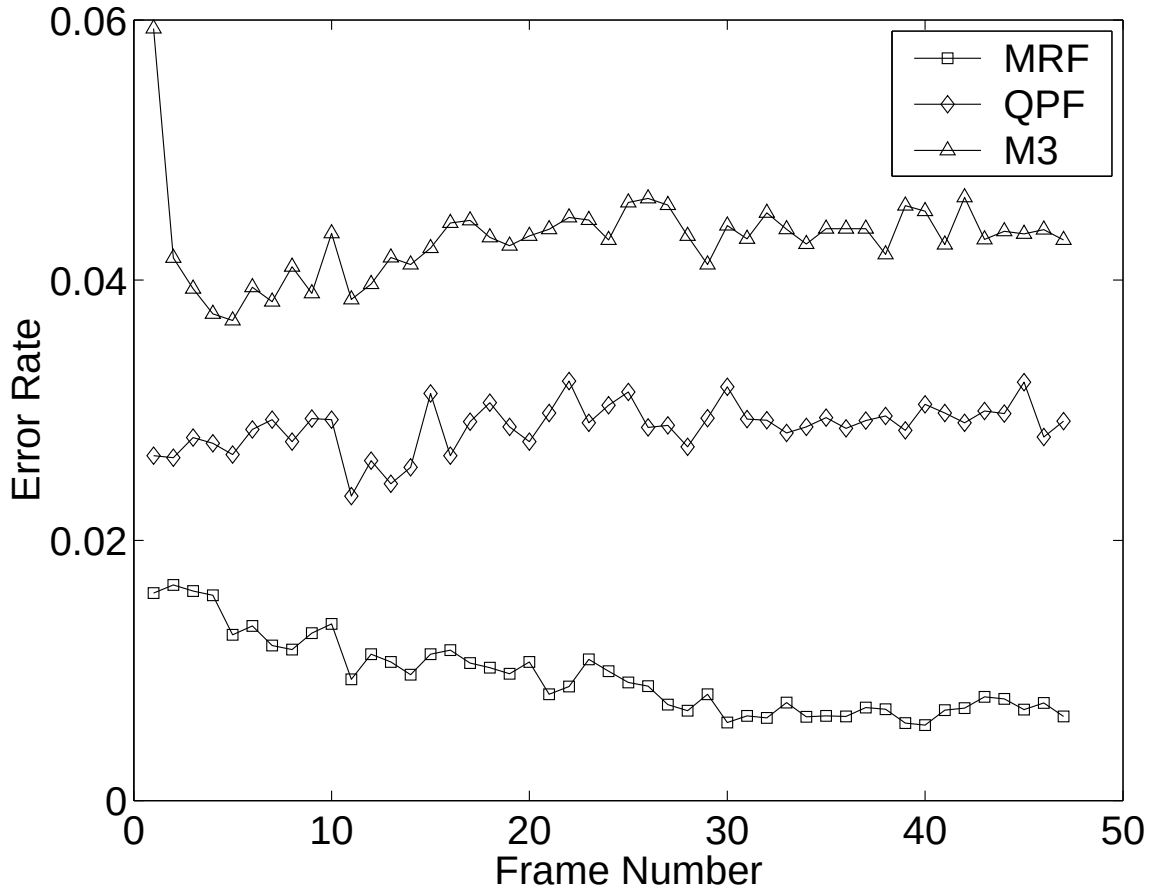

Figure 19: The error rates of our method (MRF), the "quadratic picture function" method (QPF) and the method of De Geyter and Philips (M3).

3.2.6.2 MPEG reference video In this section, experimental results on selected MPEG test sequences are presented. Change detection was carried on these sequences at a rate of 10 frame pairs per second. First, we report the experiment on *Mother & daughter* sequence by the proposed method. Fig. 20 shows frames 58 through 91 which contain both large motions (e.g. hand movement in frame 58 and 61) and small motions (e.g. chest and shoulder movements). The detected CDM's are shown in Fig. 21. It can be seen that the stationary background and the moving objects are well distinguished. The background area is quite clean, indicating that the MRF method is robust to the salt and pepper noise

contained in this sequence. Fig. 22 depicts the pdf's calculated from this sequence. While pdf $p(d|h = -1)$ was calculated from a background area that was manually selected, pdf $p(d|h = 1)$ was initialized and then adapted in the change detection process as described previously. Fig. 22 shows the pdf's calculated progressively at frames 1,60, 300, 600 and 900.

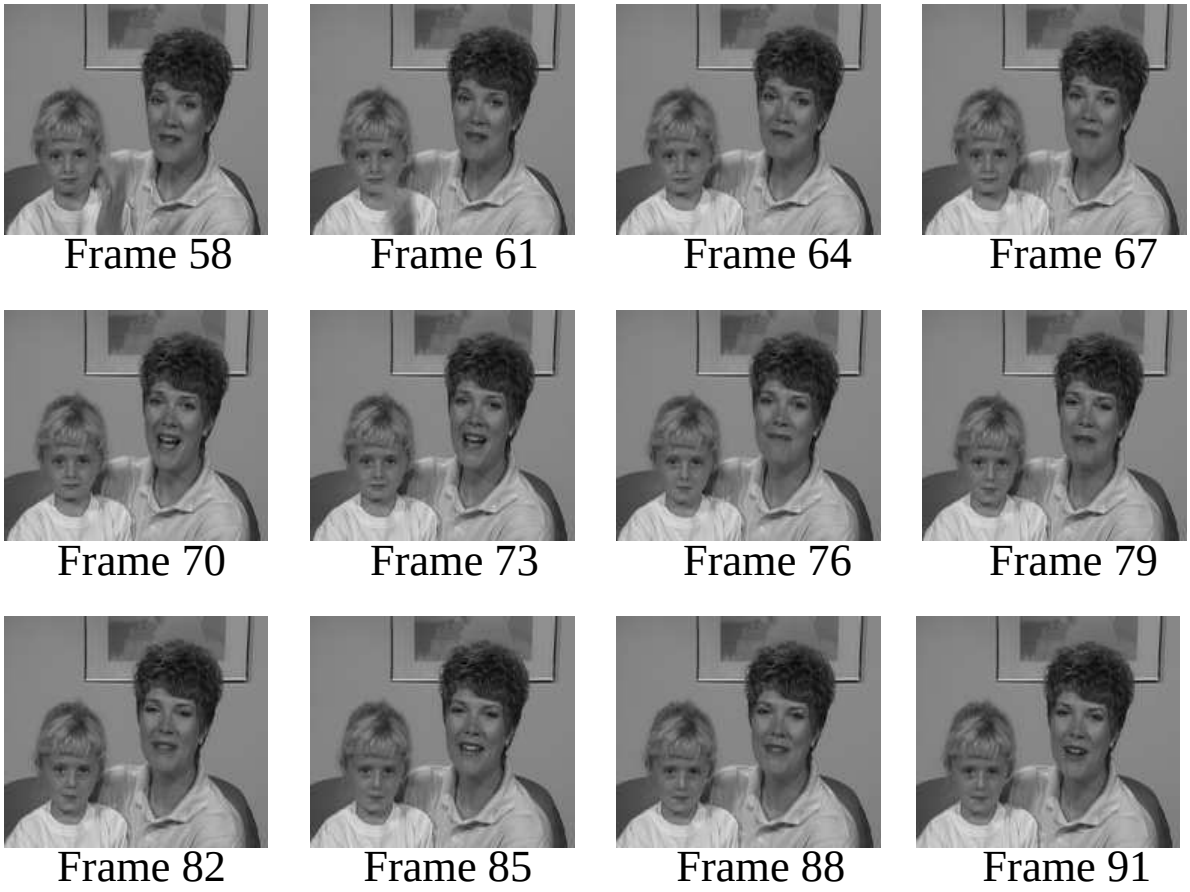

Figure 20: Frames 58 through 91 of *Mother & daughter* sequence.

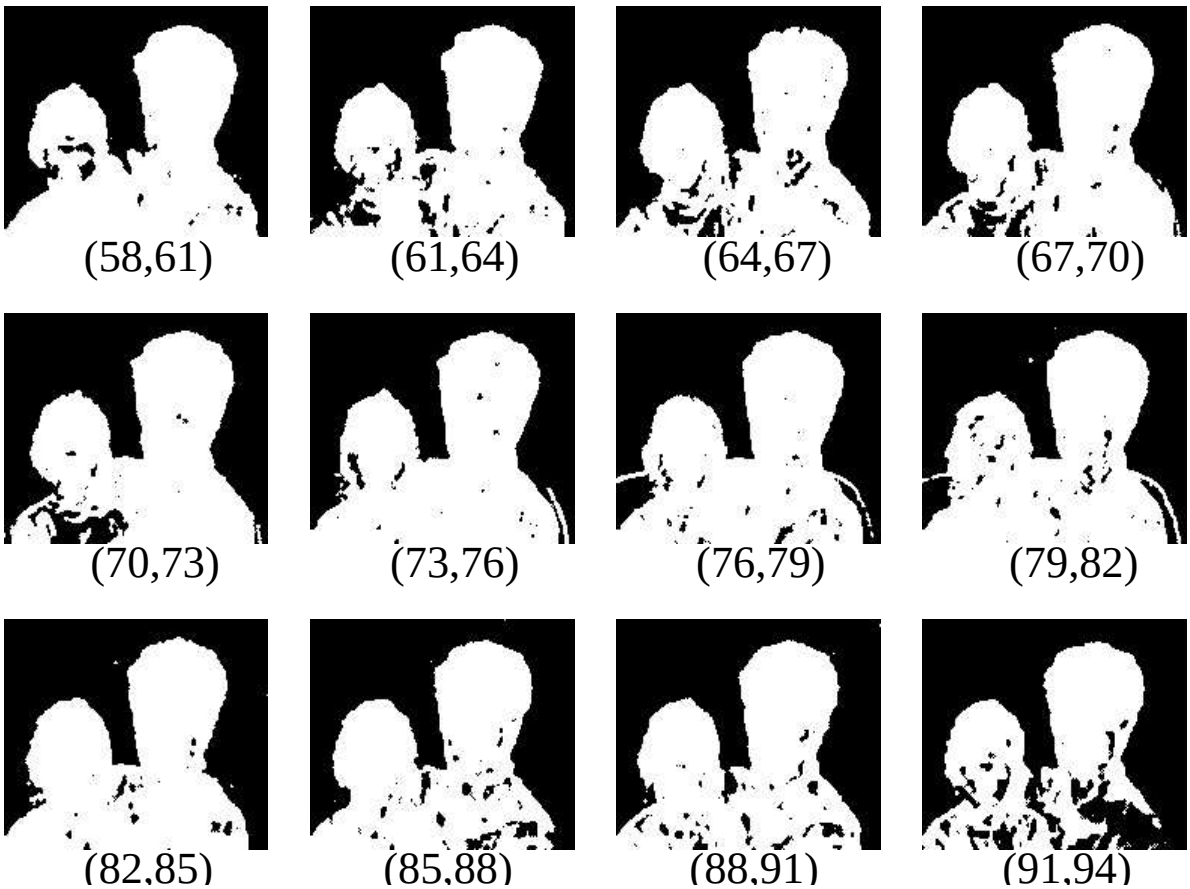

Figure 21: The detected CDM's from frames 58 through 91 of *Mother & daughter* sequence, using the parameter values listed in Table 1.

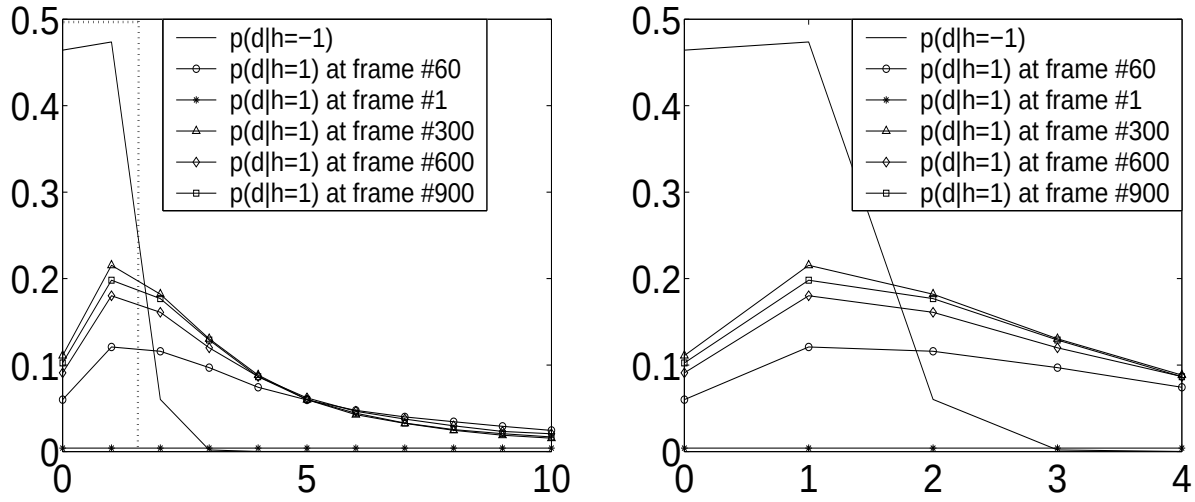

Figure 22: The pdf's obtained from *Mother & daughter* sequence: the left panel shows $p(d|h = -1)$, and $p(d|h = 1)$ at frames 1, 60, 300, 600 and 900; the right panel shows a close-look of the pdf's in the marked range (by the dash line) on the left panel.

Another test sequence is called *Hallway* that can also be found in the public domain. This sequence contains high level background noise. Sample frames are illustrated in Fig. 50, where the top panel shows frame 1 of *Hallway* sequence, and frame $25, 50, 100, 250, 275$ are shown on the bottom panel. It is seen that frame 1 contains only background scene, while the subsequent frames have appearances of new objects, including two walking persons and a suitcase placed at the left side of the hallway. The obtained CDMs by using the MRF-MFT algorithm described in Section 3.2.4 are illustrated in Fig. 24. One can see that the foreground was well separated from the background scene. The conditional probabilities required by the potential functions are shown in Fig. 25 (for the clearance of display, only part of the pdf's are displayed). The right panel shows a close-look of the pdf's on the left panel. The pdf of noise, i.e. $p(d|h = -1)$, was estimated from the intensity differences in manually selected regions, which contained no apparent changes. The pdf's of intensity differences caused by relevant changes were first initialized to be uniformly distributed within value range of $0 \sim 255$, then adapted in the process of change detection for the subsequent video frames. The convergence of the mean field value took 5.09 iterations in average.

Figure 23: Frames $1, 25, 50, 100, 250$ and 275 of *Hallway* sequence

Figure 24: The detected CDM's from the sample frames of *Hallway* sequence, with the parameter values listed in Table 1. The white ("1-pixel") regions denote "there are significant changes between the test image (containing moving objects) and the reference image (containing merely background scene)". It is seen that the significant changes caused by the moving subjects and the suitcase being placed in the hallway were well identified.

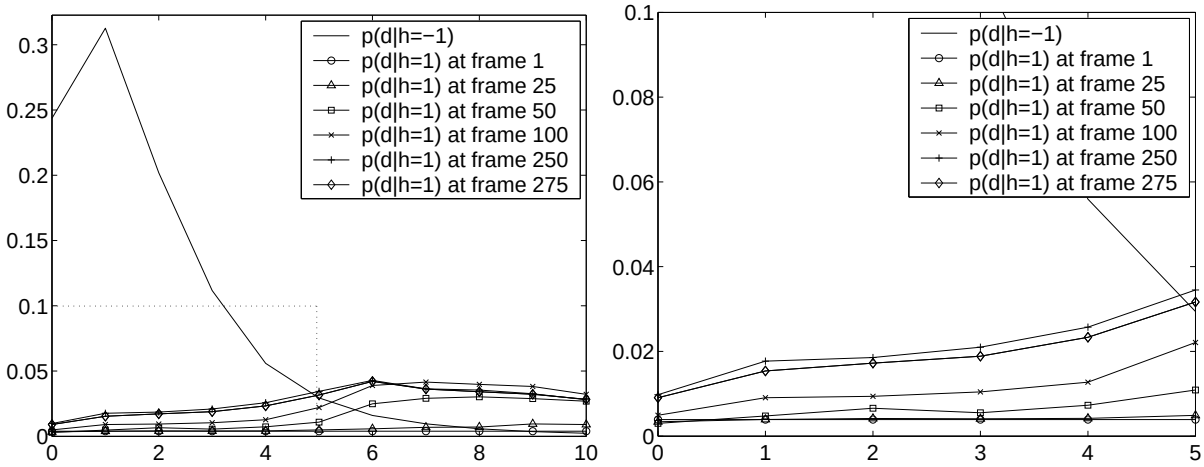

Figure 25: The pdf's obtained from *Hallway* sequence: the left panel shows $p(d|h = -1)$, and $p(d|h = 1)$ at the frames of 1, 25, 50, 100, 250 and 275; the right panel plots the pdf's in the marked range on the left panel, showing a close-look of the adaptation of $p(d|h = 1)$.

In the following, the comparisons with QPF and M3 methods are reported. Several representative change detection results on *Miss America*, *container*, *table tennis* and *News* are shown in Fig. 26-29. In the selected frames of *Miss America* (Fig. 26), the subject's head and body were moving to her left. It can be observed that the CDM's detected by the MRF approach reflected this motion, where changes in the face region were very well detected. The results from QPF captured most of the changes; however, the disturbance from noise appeared in the background area. The CDM's detected by M3 had even more missing detections and also suffered from background noise.

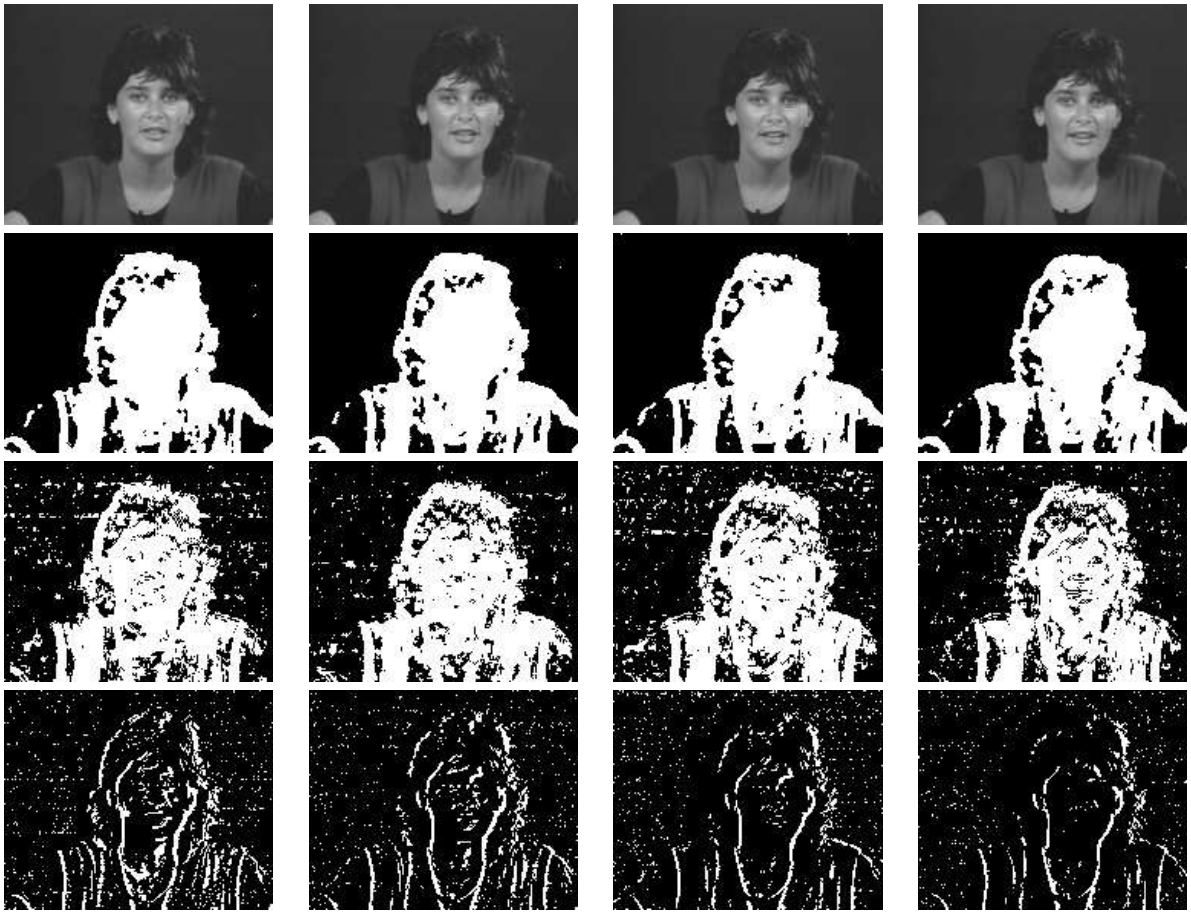

Figure 26: Experimental results of test sequence *Miss American*. From top to bottom: frames 75, 78, 81 and 84, CDM's detected by the MRF method, by the QPF method and by the M3 method.

The results on the *container* sequence, a typical outdoor video, are presented in Fig. 27. In the sample frames, the container was moving slowly to the right and two birds flew by quickly from the left to the right. It can be seen that all the three methods captured the changes caused by the flying birds. However, the motions of the container and rippling water were only well identified by the MRF method, which shows that the proposed method is more efficient in detecting small changes than the other two methods.

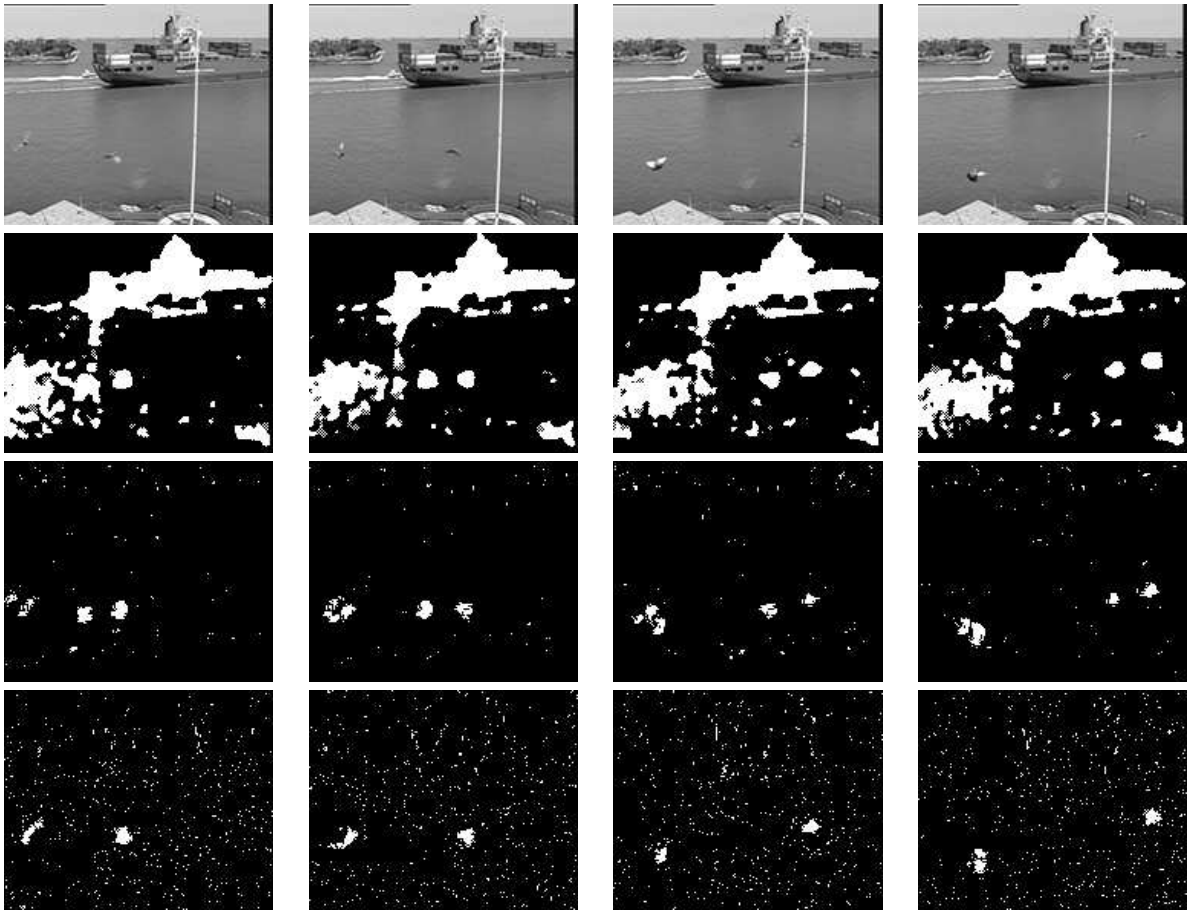

Figure 27: Experimental results on test sequence *Container*. From top to bottom: frames 252, 255, 258 and 261, CDM's detected by the MRF method, by the QPF method and by the M3 method.

The results on the *table tennis* sequence, which contains very fast motion, is shown in Fig. 28. Again, the MRF method detected moving regions more completely than the other two methods. The scenes selected in *News* sequence contain both small motion (e.g. face of the male journalist) and large motion (e.g. the spinning stage and dancers). It can be seen in Fig. 29 that, although all three methods were robust against background noise, the MRF approach was superior to the other two methods in detecting more complete changing regions, including both the journalists and the moving stage and dancers.

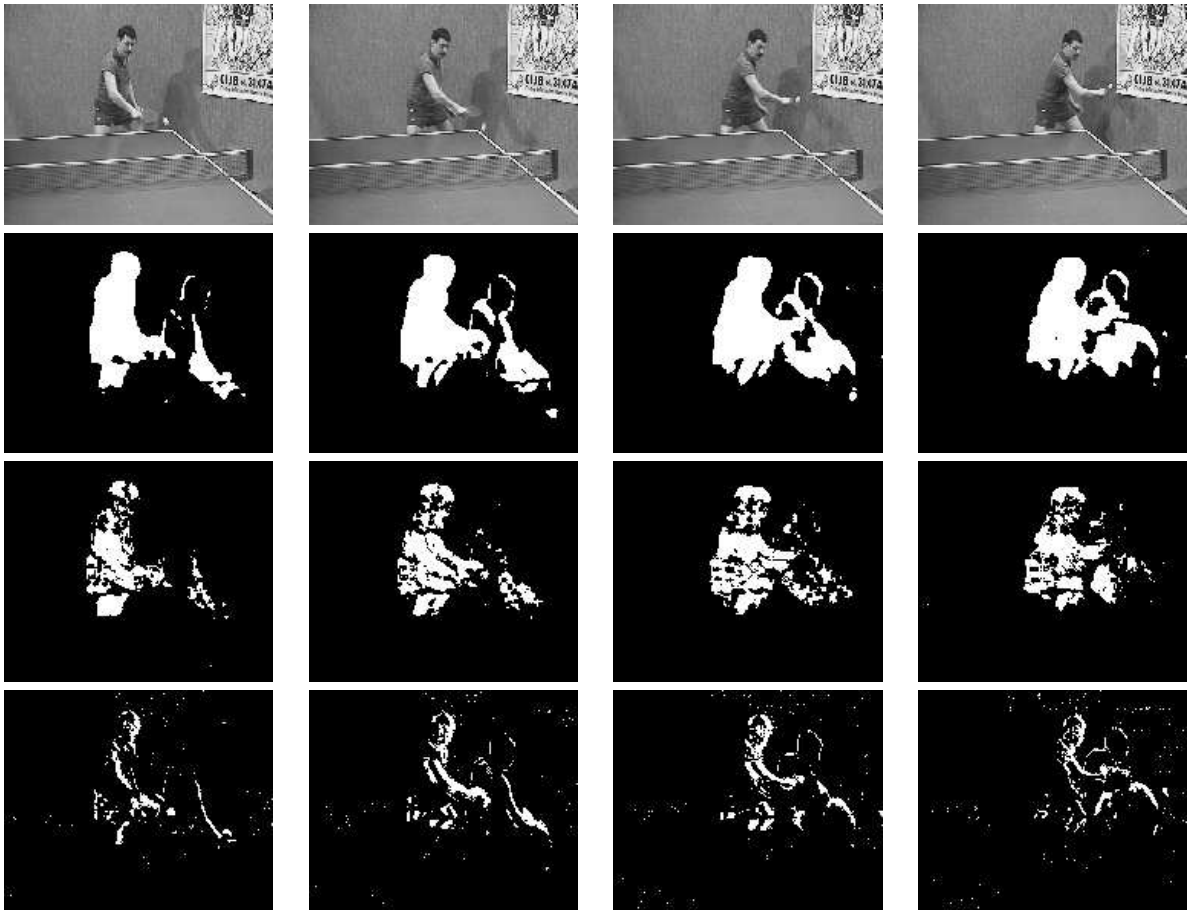

Figure 28: Experimental results of test sequence *Table tennis*. From top to bottom: frames 132, 135, 138 and 141, CDM's detected by the MRF method, by the QPF method and by the M3 method.

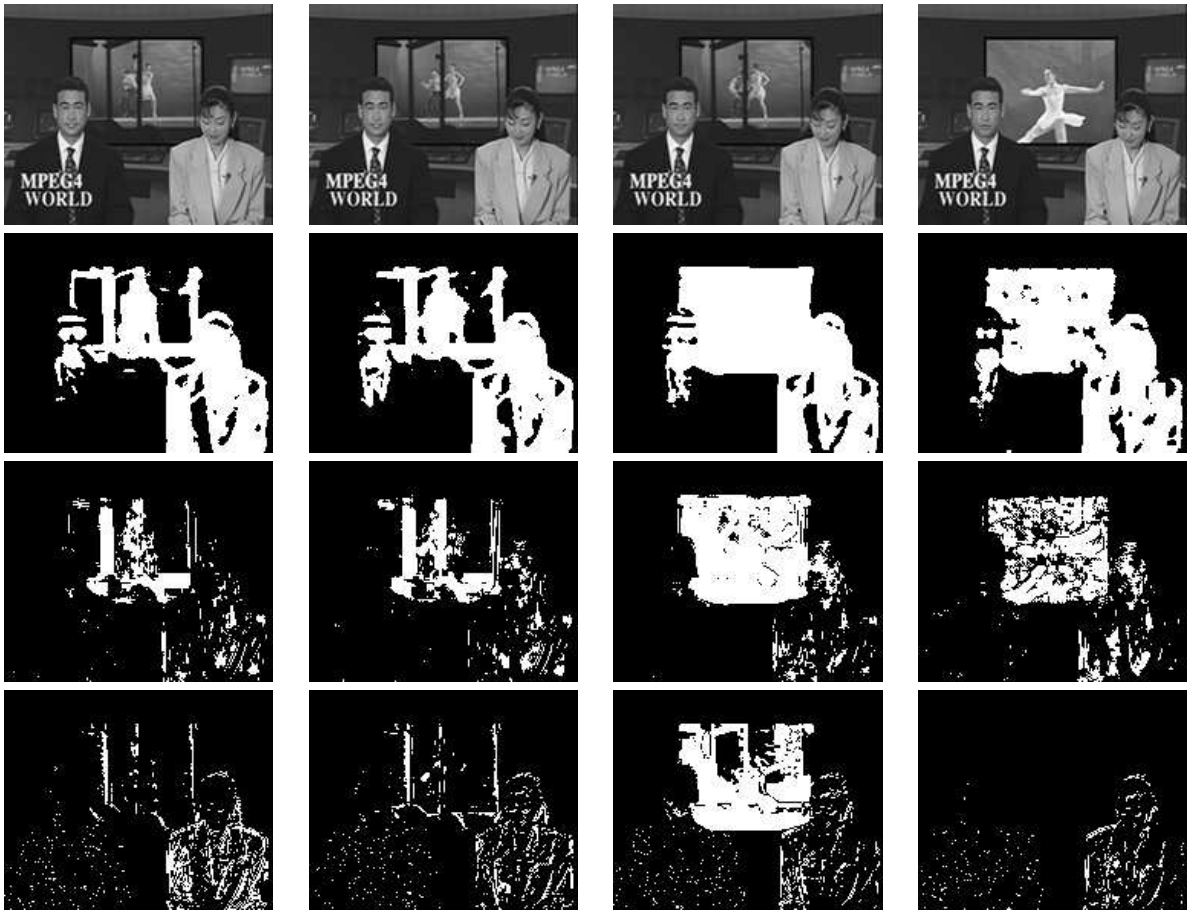

Figure 29: Experimental results of test sequence *News*. From top to bottom: frames 84, 87, 90 and 93, CDM's detected by the MRF method, by the QPF method and by the M3 method.

All the algorithms were implemented in C++ and compiled with Microsoft Visual C++ 6.0. Experiments were performed on an AMD Athlon 1900 (1.66 GHz) PC with 512M DDR2100 RAM. Among the three methods implemented, the M3 has the least computational complexity. The MRF requires iterations to compute the mean field, thus is slower than M3. The QPF required the most computation in all the experiments. In Table 2, the average computing time of each method on each testing sequence is listed. The computing time of the QPF and M3 is determined by the number of pixels contained in a video frame, therefore is largely fixed for all the testing sequence. The computing time of the MRF depends not

only on the spatial resolution, but also the number of iterations taken to compute the mean field values. In practice, if the time of computation is critical, a maximum number of iterations can be specified. For example, our system required an average of 9.4 milliseconds per iteration, so a maximum number of iterations of 10 was utilized in detecting changes in image sequence at 10 frames per second.

Table 2: Computational cost of MRF, QPF and M3 methods.

Test sequence	MRF (ave. loops/time)	QPF (time)	M3 (time)
Miss America	4.31 loops/40.51 ms	147.2 ms	5.56 ms
Container	6.49 loops/61.0 ms	147.2 ms	5.56 ms
Table Tennis	5.37 loops/50.48 ms	147.2 ms	5.56 ms
News	3.93 loops/36.94 ms	147.2 ms	5.56 ms

3.2.6.3 Patient monitoring video Next, we present experimental results on a video segment recorded at the Epilepsy Monitoring Unit at the University of Pittsburgh Medical Center. Sample video frames are shown in Fig. 30.

Figure 30: Experimental result on patient monitoring video WITHOUT illumination invariance function. The left panel shows a snapshot of the monitoring unit before the patient's occupancy. The middle panel shows a sample video frame at the presence of patient. The right panel shows the detected CDM by the proposed method without concerning illumination variation. It is seen that the CDM was affected by the illumination change, for instance, the marked polygonal regions in the background area.

We demonstrate the effectiveness of the illumination invariant approach described in Section 3.2.5. Firstly, we carried out experiment on the testing video without illumination invariance function. A typical result is illustrated in Fig. 30. The left panel shows a snapshot of the environment before the occupancy of the subject. The middle panel shows a video frame with the subject sitting in bed. Comparing these two images, we found that there were large intensity differences (in amplitude) contained in the background area. For example, the pixels in the marked regions on the right panel, which shows the CDM without concerning illumination variation, had a maximum intensity difference of 35. These intensity differences may be caused by shadow, light source change, and automated camera gain adjustment, which may all be considered as illumination variation. With the algorithm described in Section 3.2.5.2, these irrelevant disturbance can all be greatly reduced. This is demonstrated by our experimental results shown in Fig. 31, where the image containing the background scene, sample video frames, and the corresponding CDM are shown on the top, middle and bottom panels respectively. It is seen that the irrelevant changes in the background area were successfully eliminated, while the subtle changes, such as the distortion of the bed caused by the movements of the human subject, were retained.

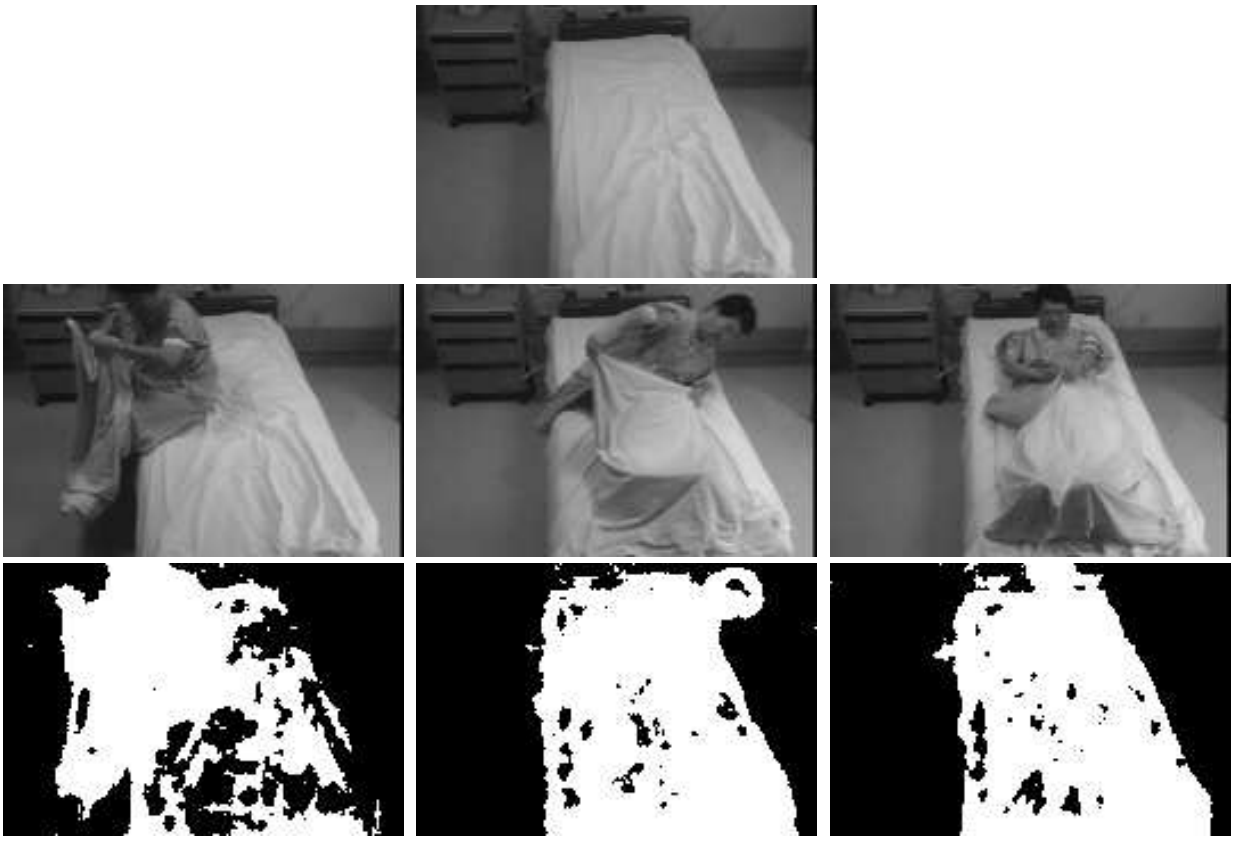

Figure 31: Experimental results based on algorithms described in Section 3.2.5 featured with illumination invariance. Top panel: the image containing the background scene; middle panels: sample video frames with presence of the subject; bottom panels: the corresponding CDM's. It is seen that the irrelevant changes in the background area were successfully eliminated, while the subtle changes of the bed caused by the movement of the subject were retained.

3.3 CHANGE DETECTION BY COVARIANCE TEST

In this section, we present another change detection model for moving pictures. The novelty of this approach lies in the way of exploiting the temporal correlation contained in consecutive video frames. In contrast to the previous methods that try to locate changes between *two* images, this new model detects changes among a group of video frames. In other words, this method utilizes multiple frames to locate moving pixels in these frames. A single CDM will be computed for the entire group of frames. This concept was motivated from two major concerns:

- The pattern that a pixel changes its intensity in the temporal domain may suggest whether the changes are due to actual motion or the affection by random noise
- It can benefit video coding by segmenting a group of frames simultaneously, because motion compensation requires to a reference of the moving object in adjacent (may not be immediate) frames. And, if "semantics" of video objects are not strictly required, the VOPs may share a common alpha plane, thus bandwidth needed for the shape coding can be reduced.

3.3.1 Pixel Vector

A pixel vector, denoted by $\vec{V}_{i,j}$, is composed of the intensity values of pixels at the same coordinates in all given group of frames. $\vec{V}_{i,j}$ is illustrated in Fig. 32, where $\vec{V}_{i,j} = [f_{i,j}(n)\ f_{i,j}(n+1)\ ...\ f_{i,j}(n+N-1)]'$, with N denoting the number of given frames. Let us model the pixel intensity by

$$f_{i,j}(k) = s_{i,j}(k) + \theta_{i,j}(k), \tag{3.40}$$

where $s_{i,j}(k)$ denotes the signal and $\theta_{i,j}(k)$ the noise. If pixel (i,j) does not change within the N frames, then $s_{i,j}(k)$ are identical for each $k \in [n, n+N-1]$. In this case, variations in the observed $\vec{V}_{i,j}$ are driven by $\theta_{i,j}(k)$. As a result, $\vec{V}_{i,j}$ and the noise have the same pattern of variation, which implies that if the noise has a known pattern of variation, then we can test $\vec{V}_{i,j}$ to determine whether the variation is caused by noise or true change. Based on this

concept, a new testing method is proposed by exploiting the temporal variations provided by vector $\vec{V}_{i,j}$.

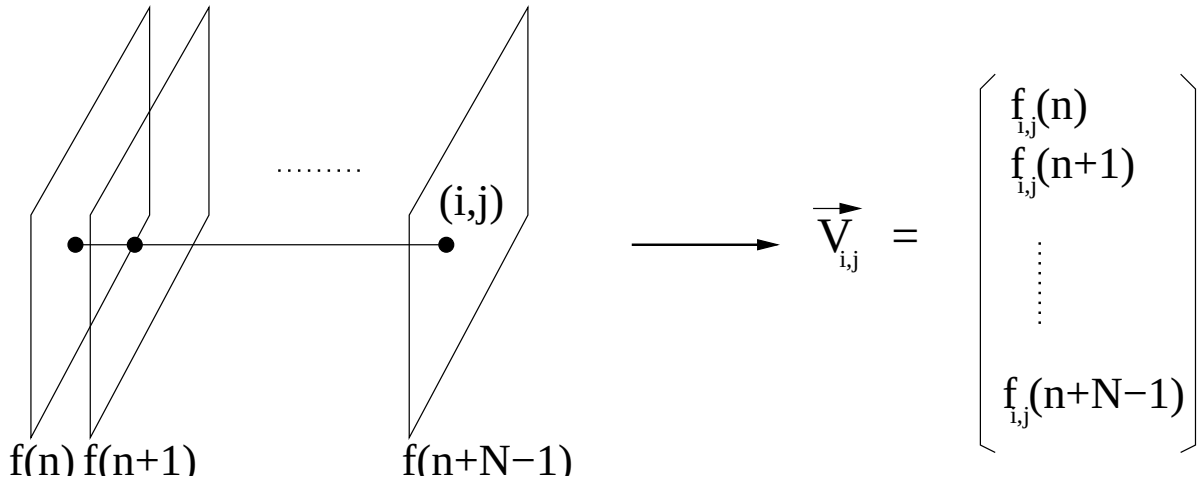

Figure 32: How to represent a group of frames by pixel vectors : a pixel vector $\vec{V}_{i,j}$ is composed of the intensity values of pixel (i, j) in all the N frames.

3.3.2 Pixel Covariance

Let us start with an observation of the pixel vectors. Fig. 33 shows two groups of pixel vectors: one group is centered at an "unchanged" pixel and the other one centered at a "changed" pixel. Each group consists of a test pixel (the centered one) and its 8-neighbors. Typically, we see that 1) the temporal variations of the "unchanged" test pixel had rather random patterns, while those of the "changed" test pixel were more regular; 2) the vector of the "changed" pixel was similar to at least one of the neighboring vectors, while the "unchanged" pixel was much less correlated with its neighboring ones; and 3) the variations of a "changed" pixel vector were likely to have larger amplitudes than those of an "unchanged" one.

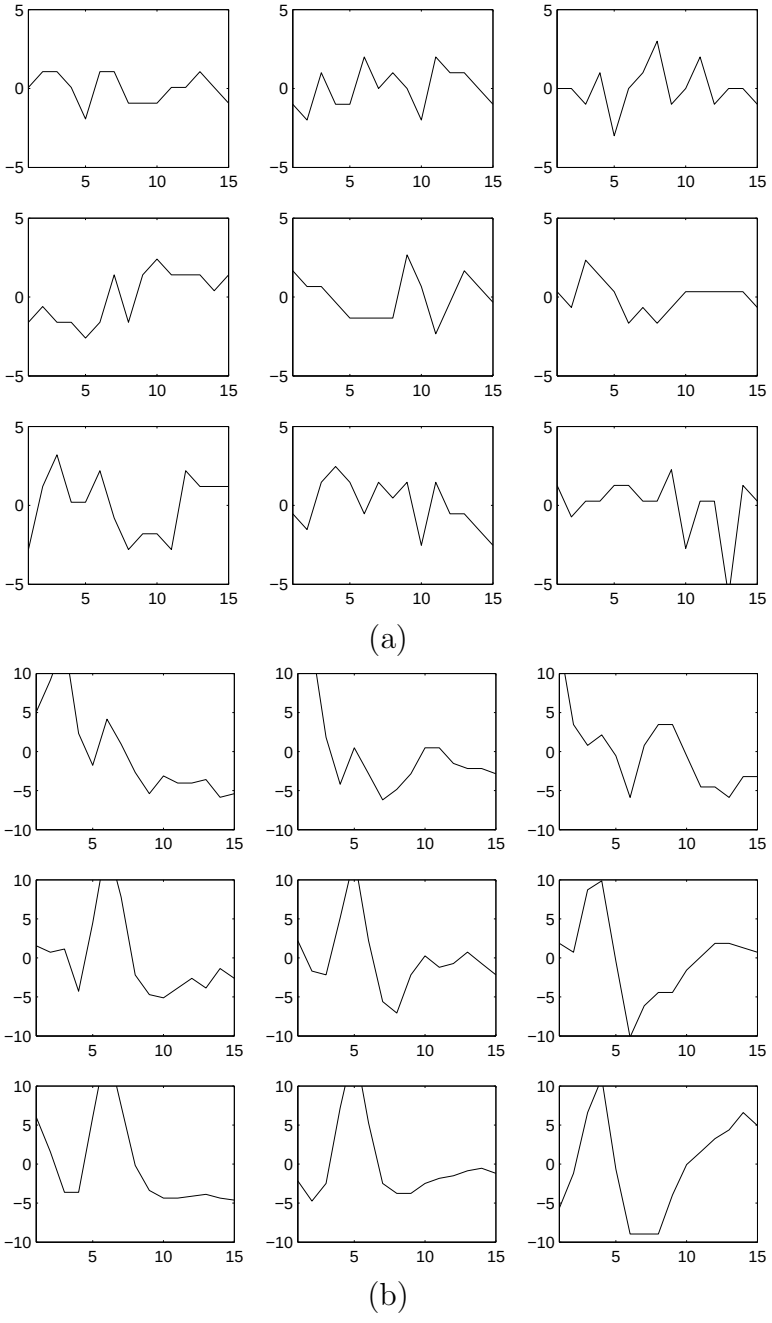

Figure 33: The pixel vectors of two clusters of pixels, where each subplot shows the demeaned intensity values of a pixel within a time window. In this example, the duration was 0.5 seconds, i.e. a time span of 15 video frames at a frame rate of 30 fps. On the top/bottom panel, the centered pixel is known as "unchanged"/"changed" and the other eight pixels are its immediate 8-neighbors.

The above observation is consistent with the assumption that under the hypothesis of "unchanged", pixel (i, j) has small covariance with its neighboring pixel (i', j'). To be more specific, let $\hat{\vec{V}}_{i,j}$ be the demeaned pixel vector,

$$\hat{\vec{V}}_{i,j} = \vec{V}_{i,j} - \overline{\vec{V}_{i,j}}, \tag{3.41}$$

where $\overline{\vec{V}_{i,j}}$ denotes the mean of the elements of $\vec{V}_{i,j}$, i.e., $\overline{\vec{V}_{i,j}} = \frac{1}{N}\sum_{k=n}^{n+N-1} f_{i,j}(k)$. Then, $\hat{\vec{V}}_{i,j}$ can be taken as a realization of $\theta_{i,j}(k)$. If we apply the following assumptions 1) ergodicity of $\theta_{i,j}(k)$, 2) $\theta_{i,j}(k)$ being stationary, 3) $\theta_{i,j}(k)$ being uncorrelated with $\theta_{i',j'}(k)$, and 4) N sufficiently large, the following approximation can be obtained,

$$E[\theta_{i,j}(k)\theta_{i',j'}(k)] \approx \frac{1}{N} < \hat{\vec{V}}_{i,j}, \hat{\vec{V}}_{i',j'} > \approx 0, \tag{3.42}$$

where E denotes expectation, and $< \cdot, \cdot >$ denotes vector inner product.

3.3.3 Covariance Test Algorithm

It is easy to see that if a pixel has low covariance with all its neighboring pixels, then the pixel is likely to be "unchanged". On the other hand, if a pixel has high covariance with at least one of its neighboring ones, then it tends to be a "changed" pixel. Based upon this concept, we design the covariance testing algorithm to decide pixel (i, j) "changed" or "unchanged":

- Let the pixel covariance be defined in the following form

$$C(\vec{V}_{i,j}, \vec{V}_{i',j'}) = \frac{1}{N} < \hat{\vec{V}}_{i,j}, \hat{\vec{V}}_{i',j'} >, \tag{3.43}$$

  where $\hat{\vec{V}}_{i,j}$ is defined in Eq. 3.41.
- With given intensity values of (i, j) in N consecutive frames, i.e., $f_{i,j}(n)$, $f_{i,j}(n+1)$, ..., $f_{i,j}(n+N-1)$, construct vector $\vec{V}_{i,j} = [f_{i,j}(n)\ f_{i,j}(n+1)\ ...\ f_{i,j}(n+N-1)]'$.
- Define $W_{i,j}$ as the neighborhood of (i, j), typically a 3×3 window centered at (i, j), and for each $(i', j') \in W_{i,j}$, and $(i', j') \neq (i, j)$, compute the pixel covariance $C(\vec{V}_{i,j}, \vec{V}_{i',j'})$ as defined in (3.43).

- Then, compute the maximum value of the pixel covariance within the 8-neighbors of pixel (i, j),

$$R(i,j) = \max_{\substack{(i',j') \in W_{i,j} \\ (i',j') \neq (i,j)}} |C(\vec{V}_{i,j}, \vec{V}_{i',j'})|. \tag{3.44}$$

- With a predetermined threshold τ, if $R(i, j) > \tau$, then label (i, j) as "changed", otherwise, "unchanged".

Now the question is how to determine the threshold τ. Practically, τ can be obtained in an off-line process:

- Collect N frames containing only stationary scene.
- Compute $R(i, j)$ for each pixel (i, j).
- Calculate the histogram of $R(i, j)$ and normalize it by the total number of pixels.
- Let $p(R)$ be the normalized histogram, then the threshold τ is obtained by solving

$$\ell = \int_0^{\tau} p(R)\ dR, \tag{3.45}$$

where $\ell \in [0, 1]$ is a specified significance level.

The parameters, i.e. N and ℓ, are chosen experimentally. In all the experiments, we set $N = 15$ and $\ell = 0.999$.

A sample histogram of R is shown in Fig. 34, which was obtained from a training video containing only stationary scene. A histogram of R in a testing video is shown in Fig. 35. The R with value less than 10 was mostly due to the "unchanged" pixel. The R of "changed" pixels has a larger value and a much wider range, e.g. valued over 5000. Accordingly, there are two humps shown in the testing histogram, one in the domain of $R < 10$ and the other in $R >= 10$. These two humps represent the two classes of pixels to be classified as "unchanged" and "changed", respectively.

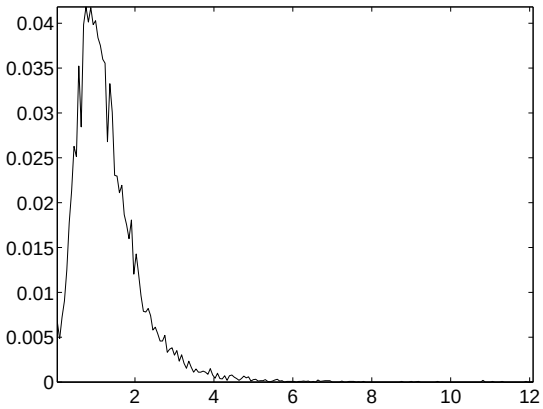

Figure 34: The normalized histogram of R as defined in Eq. (3.44) from collected video frames that contained only stationary scenes. This histogram was utilized as the $p(R)$ in Eq. (3.45) to determine the threshold τ.

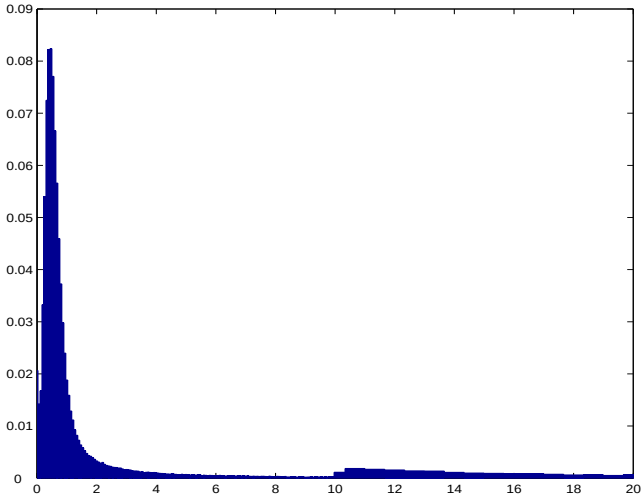

Figure 35: The normalized histogram of R in a testing video. There can be seen two humps in the histogram, one in the domain of $R < 10$ and the other in $R >= 10$. The former is due to the "unchanged" pixels, where the R is much smaller than that of the "changed" pixels.

3.3.4 Experimental Results

To evaluate this approach, we first present the experimental results of a simulating video (shown in Fig. 36), which was generated by the following steps:

- First, noise-free video frames were generated, where the foreground was a moving plate with intensity value of 15, a radius of 20 pixels, and the intensity of the background equal to 0. The plate was moving along one direction with a constant speed, both of which were generated randomly.
- Second, a video segment was collected by using SONY DCR-TRV20 camcorder and Sapphire Radeon 9000 VIVO (www.sapphiretech.com) video card. This video segment contained only stationary scenes, by which we assumed the intensity difference between frames was only due to device noise. In this case, the noise had a mean of 0 and a variance of 1.82.
- Third, for each pixel in the collected video, the intensity sequence was demeaned in temporal domain. The demeaned sequences, which represented the disturbance of noise, were added to the noise-free video frames. Then, in order to set the intensity values of the noisy video frames to $0 \sim 255$, these frames were added a DC intensity of 127.

To evaluate the results quantitatively, let us define the error rate of change detection. Let H_c denote the control and H_d the detected CDM. Then, the $S_e = \{(i,j)|H_c(i,j) \neq H_d(i,j), (i,j) \in S\}$ denotes the set of pixels with false detections. The error rate is then defined by $E_r = \|S_e\|/\|S\|$, where $\|S_e\|$ and $\|S\|$ denote the number of pixels in S_e and S respectively. By this definition, the error rate of the simulating data was 0.024%.

Experiments on real world video were also carried out. The results were obtained by utilizing the same values of ℓ and N as in the simulating video. In Fig. 37, the figures on the left column are sample frames of three natural video sequences, which are, from top to bottom, a patient monitoring video from Epilepsy Monitoring Unit (EMU) at the University of Pittsburgh, a home video recorded by SONY DCR-TRV20 camcorder, and a standard testing video named "Claire". In the patient monitoring video frame, the patient was sitting into the bed, thus causing the movements of his body, the sheet, and the bed. In the home video, the subject was sitting still and blinking his eyes. This experiment was set up to

test the approach on detecting small moving objects. In the *Claire* sequence (frame 290 was shown), the subject was speaking with her head moving constantly and body moving slightly. The plots on the right are the corresponding CDM's of the three video sequences. As can be seen, the expected movements included in the three video sequences were successfully detected by the covariance test method. In addition, we compare the results with another well known change detection method, called *significance test* model. The parameters required by this approach were selected according to the original paper, where the threshold was set to 52.61, corresponding to a significance level of 10^{-3} and a spatial window of 5×5. The results of both the covariance test and the significance test methods are shown in Fig. 38. One may observe from the results that the detection errors, including both the missed detection and the false alarms, were reduced by the covariance test method in contrast to the significance method.

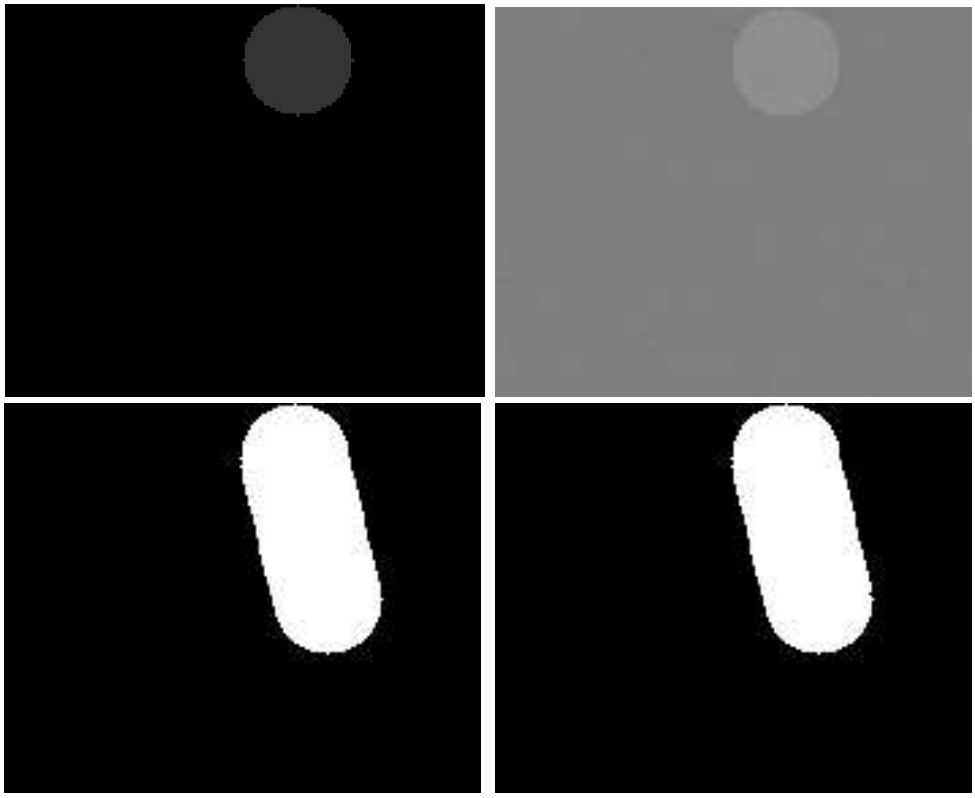

Figure 36: Experimental results of simulating video sequence. Top-left panel: a noise-free frame with black background (intensity 0), and a solid plate (radius 20 pixels and intensity value of 15) moving from top to bottom in the frame. Top-right panel: the noise-free frame added with collected noise (with variance 1.82) and a DC intensity 127. Bottom-left panel: the CDM obtained from the noise-free frames as the control to evaluate the covariance testing approach. Bottom-right panel: the CDM obtained by the covariance testing method, where N was 15, the significance level ℓ was 0.999 and the pdf of R shown in Fig. 34. All the video frames were in QCIF format (176×144 in pixels). There is only a difference of 6 pixels between the bottom-left and the bottom-right CDM's.

Figure 37: Experiments on real world data. The plots on the left column are sample frames from patient monitoring video (top), home video (middle), and standard testing video (bottom). The right column shows the corresponding CDM's detected by the covariance test approach.

Figure 38: Comparison with *significance test* method. The top panels show the test frames of patient monitoring video (left), home video (middle), and *Grandma* sequence (right). The middle panels show the moving objects in the corresponding test sequences, detected by the covariance test method. The bottom panels show the results detected by the significance test method. One may observe from the results that the detection errors, including both the missed detection and the false alarms, were reduced by the covariance test method in contrast to the significance test method.

3.3.5 Illumination invariant approach

So far, we've showed that the covariance test approach is highly effective in detecting meaningful changes and excluding disturbances due to noise. The illumination changes however, is also "meaningful" to this approach. The pixels that experience illumination variation, e.g. change of lighting condition and shadow effect, will be labeled as "changed" pixels. In order to rule out the infection by illumination variation, we combine the covariance test model with the *shading model* introduced in section 3.2.5.1.

Refer to Eq. 3.40, let us define the intensity of a pixel i in frame k by

$$f_{i,k} = s_{i,k} + \theta_{i,k}, \tag{3.46}$$

where $s_{i,k}$ denotes the signal and $\theta_{i,k}$ the noise. Applying the *shading model*, we above equation can be formed as

$$f_{i,k} = S_{i,k} I_{i,k} + \theta_{i,k}, \tag{3.47}$$

where $S_{i,k}$ denotes the shading coefficient and $I_{i,k}$ the illumination.

Now, considering the illumination variation, we define

$$I_{i,k+1} = \alpha_k I_{i,k}, \tag{3.48}$$

where α_k denotes the ratio of the illumination change. To compensate the illumination variation, let us define the "local intensity ratio" by

$$g_{i,k+1} = \frac{f_{i,k+1}}{\sum_{j \in N_i} f_{j,k+1}}, \tag{3.49}$$

where N_i is a neighborhood (a spatial window) of i. Applying Eqs. 3.47 and 3.48, we denote the denominator by

$$\begin{aligned} F_{i,k+1} &= \sum_{j \in N_i} (\alpha_k I_{j,k} S_{j,k+1} + \theta_{j,k+1}) \\ &= \sum_{j \in N_i} (\alpha_k I_{j,k} S_{j,k+1}) + \sum_{j \in N_i} \theta_{j,k+1} \end{aligned} \tag{3.50}$$

where the second term denotes the average of the noise variables in the spatial domain. When N_i is sufficiently large, this term is ignorable, under the assumption that these noise

variables are independently distributed in spatial domain. Therefore, we have $g_{i,k+1}$ in the following form,

$$g_{i,k+1} = \frac{I_{i,k}S_{i,k+1}}{\sum_{j\in N_i} I_{j,k}S_{j,k+1}} + \frac{\theta_{i,k+1}}{F_{i,k+1}}. \tag{3.51}$$

Considering that the illumination is a low frequency signal in the spatial domain, we can assume the $I_{j,k}, j \in N_i$ are identical to $I_{i,k}$, where N_i is a local window centered at i. As a consequence, we have

$$g_{i,k+1} = \tilde{S}_{i,k+1} + \tilde{\theta}_{i,k+1}, \tag{3.52}$$

where

$$\tilde{S}_{i,k+1} = \frac{S_{j,k+1}}{\sum_{j\in N_i} S_{j,k+1}}. \tag{3.53}$$

Essentially, the illumination factor has been canceled in Eq. 3.52. Therefore, carrying out change detection on $g_{i,k}$ instead of $f_{i,k}$ should provide the illumination-invariant function. One more concern is that the threshold which is determined offline can not be directly employed, because it is calculated from $f_{i,k}$. To compensate this, we notice that, under the null hypothesis, $\tilde{S}_{i,k}$ does not change with respect to k. Therefore, within a short time window, during which the physical surface corresponding to pixel i does not change, we have $\tilde{S}_{i,k}$ as a constant. Then, we define

$$\hat{f}_{i,k+1} = F_{i,k+1}(g_{i,k+1} - \bar{g}_i), \tag{3.54}$$

where $\bar{g}_i$ is the mean value of $g_{i,k}$ within the time window. With this definition, we see that under the null hypothesis, $\hat{f}_{i,k+1} = \theta_{i,k+1}$, which is the condition under which the threshold is determined. Henceforth, the covariance test algorithm can be applied on $\hat{f}_{i,k+1}$ to be illumination-invariant.

Next, we demonstrate the effectiveness of the approach with experimental results, as shown in Fig. 39. The parameters were set to the same values as in previous experiments, i.e. $N = 15$, $\ell = 0.999$. From the results, we see that the infection illumination variation was effectively removed. Yet, some homogeneous regions in foreground were miss detected. Nevertheless, the boundary of the foreground was accurately detected. Therefore, the holes inside can be boundary-filled.

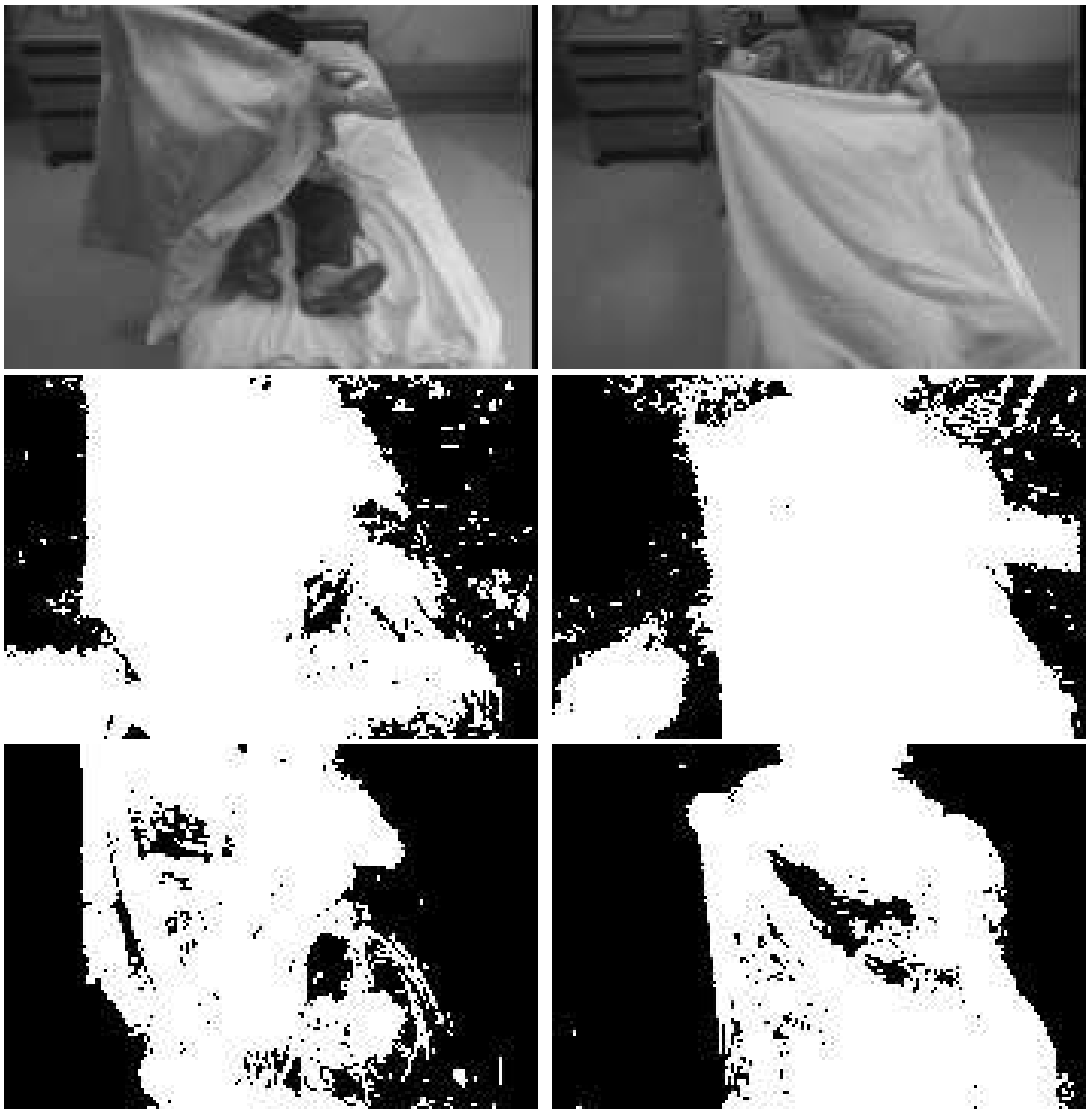

Figure 39: The experimental results of the illumination-invariant covariance test method. The top panels show the sample frames of the patient monitoring video. The middle panels show the CDMs detected without concerning illumination variations. One can see the shadows in the background area were detected as meaningful changes. The bottom panels show the results of the illumination-invariant covariance test approach. We see that the shadows in the background area were effectively removed. Yet, some smooth regions in foreground were also compromised. Nevertheless, the boundary of the foreground was detected accurately. Thus, the holes inside can be boundary-filled.

3.4 DISCUSSION ON ALTERNATIVE APPROACHES

The change detection models presented in previous sections are realized pixel-wisely in spatial domain. They play a role of preprocessing in the object-based coding system. The coding module is relatively separate from it. Therefore, some useful results that are generated in the coding process, such as motion vector and texture residual after motion compensation, are not available to the change detection module. In this section, we discuss change detection combined with the coding process.

3.4.1 Change detection based on motion vectors

In most of the video codecs, motion detection is carried out via block matching. The displacement between two matched blocks (in the test and reference video frame respectively) is called motion vector. The amplitude of a motion vector represents the intensity of the motion that the corresponding block undergoes. Thresholding on the amplitudes may distinguish the moving blocks from those motionless ones. A more advanced way is to classify the blocks according to homogeneity of the motion vectors, where both the amplitude and direction of the motion vectors are considered. The realization of these concepts is heavily dependent on the quality of motion estimation, i.e. the computation of the motion vectors. However, the motion estimation function realized in current video codecs does not provide a reliable motion field for the motion based segmentation.

In the following, we show some examples of the motion fields obtained via exhaustive block matching under mean square error criteria. A representative result of *container* sequence is illustrated in Fig. 40, where the motion vectors were obtained with 4×4 and 8×8 blocks. The mask image was achieved by thresholding the motion amplitudes, where the threshold was set to 1.

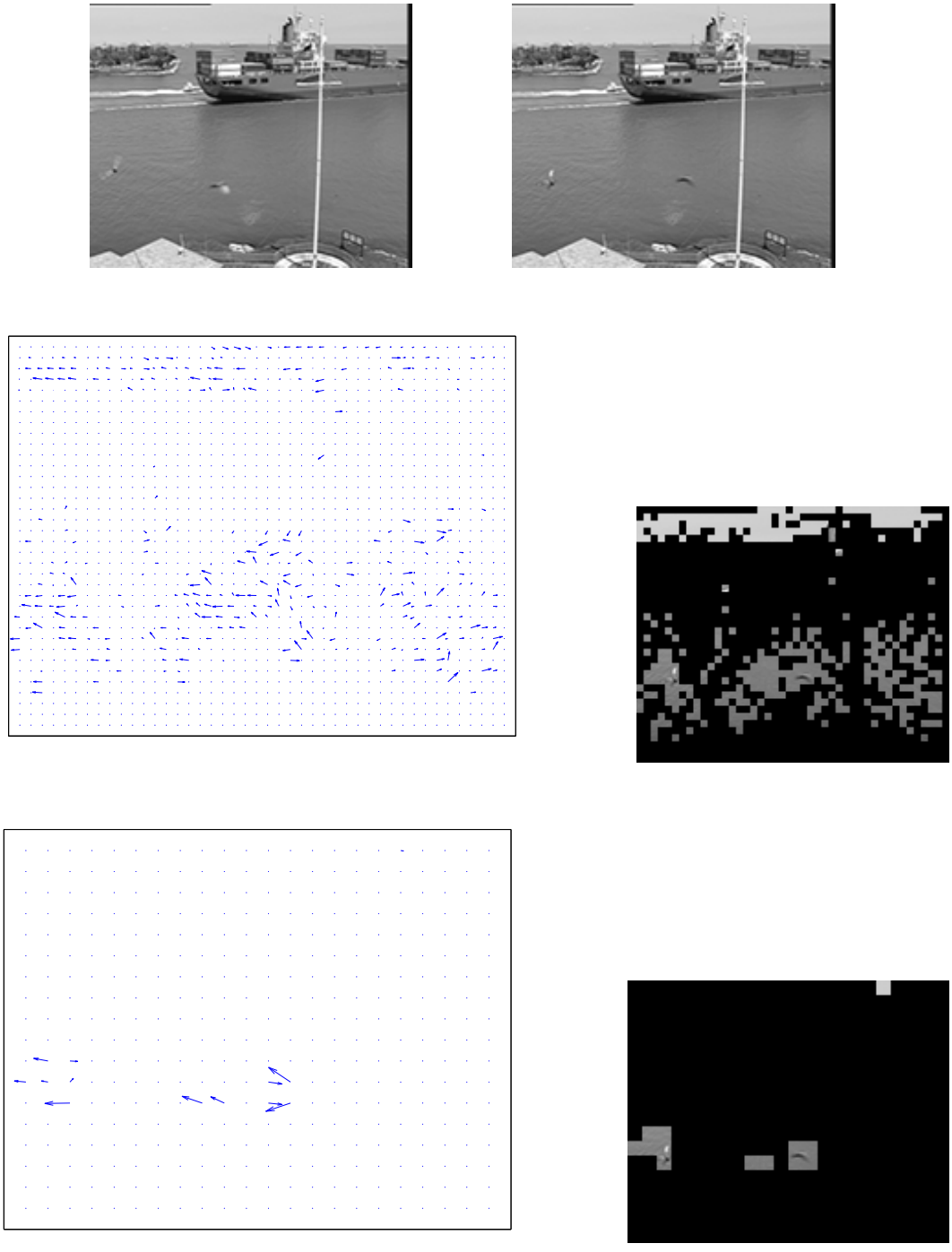

Figure 40: Block matching experiments on *container* sequence. Motion vectors were obtained via exhaustive search under mean square error criteria. Top panel: frames 252 and 255 of *container* sequence; Middle panel: motion vectors of 4×4 blocks and masked frame 255, where the mask was obtained by thresholding the motion vectors; Bottom panel: results of 8×8 blocks.

From the results, one may see several problems regarding the effectiveness of the motion-vector based approach,

- The slow motion of the container was not detected in either grid settings.
- Detection with small blocks were sensitive to noise, especially in homogenous regions, e.g. the sky was falsely labeled.
- Large blocks compromised spatial resolution and reduced the sensitivity of small motion, e.g. detection of the rippling was failed.

3.4.2 Statistical test on DCT blocks

We provide an alternative approach to combining the coding process for segmentation. This approach is built upon testing differences between DCT-blocks, where noise and illumination variations are excluded as irrelevant changes.

Considering two blocks in two video frames, we can formulate the intensity values by the following

$$x^{(k)} = I^{(k)} S^{(k)} + \theta^{(k)}, \ k = 1, 2 \tag{3.55}$$

where k is the image index, $x^{(k)}$ is an $N \times N$ matrix denoting the intensities of a block, $I^{(k)}$ is a scalar representing illuminance, $S^{(k)}$ is an $N \times N$ matrix representing reflectance of the patch surface, and $\theta^{(k)}$ is an $N \times N$ matrix denoting noise. It should be notified that (3.55) incorporates the *Shading Model* proposed by Phong [57]. Also, an assumption that the illuminance is uniformly distributed on the patch surface is employed.

Let $X^{(k)}$ denote the DCT block, i.e. the DCT coefficients of a block, which can be formulated by

$$X^{(k)} = M x^{(k)} M^T, \ k = 1, 2 \tag{3.56}$$

where M is the $N \times N$ transform matrix. Equivalently, we have

$$X^{(k)} = I^{(k)} M S^{(k)} M^T + M \theta^{(k)} M^T, \ k = 1, 2. \tag{3.57}$$

Essentially, the task of a change detection algorithm is to identify whether there is meaningful change between the two blocks given $X^{(k)}$. Usually, the hypothesis of "no change" can be

interpreted as $S^{(1)} = S^{(2)}$, which means that the patch surface does not change between the two images. Therefore, change detection can be carried out by testing the null hypothesis that $S^{(1)} = S^{(2)}$. Notice that the variations of $I^{(k)}$ and $\theta^{(k)}$ are considered to be irrelevant.

To perform the hypothesis test, one needs to have knowledge of $I^{(k)}$ and $\theta^{(k)}$. For the former, let us define

$$I^{(2)} = \gamma I^{(1)} \tag{3.58}$$

where γ denotes the ratio of the illuminance between two blocks. Then, instead of comparing $X^{(1)}$ and $X^{(2)}$, we can examine $\gamma X^{(1)}$ and $X^{(2)}$, where illumination variation is compensated. Now the question is how to obtain γ. An easy way is to use $X^{(1)}$ and $X^{(2)}$ to estimate it. To do that, let us consider the DC components of the two blocks, denoted by $X_{11}^{(k)}$, which can be formulated by

$$X_{11}^{(k)} = I^{(k)} M_1 S^{(k)} M_1^T + M_1 \theta^{(k)} M_1^T, \ k = 1, 2 \tag{3.59}$$

where M_1 is the 1st row of M, and M_1^T is its transverse. Let $\eta_{1,1}$ denote the second term on the right side of (3.59), namely, the noise variable. Then, since the entries $M_{1j} = \frac{1}{\sqrt{N}}$, $j = 1, 2, ..., N$, one has

$$\eta_{11}^{(k)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \theta_{ij}^{(k)}, \ k = 1, 2 \tag{3.60}$$

where $\theta_{ij}^{(k)}$ denotes an entry of noise matrix $\theta^{(k)}$. Assuming that $\theta_{ij}^{(k)}$ is independently identically distributed (*i.i.d.*) and has a Gaussian distribution with a mean of zero and a variance of σ^2, we have $\eta_{11}^{(k)}$ obeying the same distribution as θ_{ij}. Considering that, for natural images, $S^{(k)}$ has highly correlated entries, one may assume that $I^{(k)} M_1 S^{(k)} M_1^T$ usually has a much larger value than that of η_{11}. Therefore, we have

$$X_{11}^{(k)} \approx I^{(k)} M_1 S^{(k)} M_1^T, \ k = 1, 2. \tag{3.61}$$

As a result,

$$\gamma = \frac{I^{(2)}}{I^{(1)}} \approx \frac{X_{11}^{(2)}}{X_{11}^{(1)}} \frac{M_1 S^{(1)} M_1^T}{M_1 S^{(2)} M_1^T} \tag{3.62}$$

Under the null hypothesis, i.e. $S^{(1)} = S^{(2)}$, we have

$$\gamma \approx \frac{X_{11}^{(2)}}{X_{11}^{(1)}}. \tag{3.63}$$

Now, considering both the noise and illumination variation, let us define the difference between two DCT blocks as

$$\begin{aligned}\xi &= X^{(2)} - \gamma X^{(1)} \\ &= M(I^{(2)}S^{(2)} - \gamma I^{(1)}S^{(1)} + \theta^{(2)} - \gamma\theta^{(1)})M^T.\end{aligned} \tag{3.64}$$

Under the null hypothesis, one has

$$\begin{aligned}\xi &= M(\theta^{(2)} - \gamma\theta^{(1)})M^T \\ &= M\hat{\theta}M^T\end{aligned} \tag{3.65}$$

where $\hat{\theta} = \theta^{(2)} - \gamma\theta^{(1)}$. With the assumption on $\theta_{ij}^{(k)}$ stated previously, the entries of $\hat{\theta}$ are *i.i.d.* Gaussian random variables, with a mean of zero and a variance of $(1+\gamma^2)\sigma^2$.

In order to perform hypothesis test, we need to know the probability density function of the random variables of ξ on condition of the null hypothesis. Let ξ_{ij} denote the ijth entry of ξ, we have

$$\begin{aligned}\xi_{ij} &= M_i\hat{\theta}M_j^T \\ &= \sum_{k=1}^{N} M_{jk} \sum_{l=1}^{N} M_{il}\hat{\theta}_{lk}\end{aligned} \tag{3.66}$$

where M_i denotes the ith row of M and M_{ij} the ijth entry of M. Note that ξ_{ij} is a linear combination of *i.i.d.* Gaussian random variables, we have ξ_{ij} obeying Gaussian distribution. Also, since M is a unitary matrix, ξ_{ij} has the same mean and variance as $\hat{\theta}_{lk}$. Proof of the former is trivial. A proof of the latter is given as follows

$$\begin{aligned}E(\xi_{ij}^2) &= E(\sum_k M_{jk} \sum_l M_{il}\hat{\theta}_{lk} \cdot \sum_s M_{js} \sum_t M_{it}\hat{\theta}_{st}) \\ &= \sum_k M_{jk} \sum_l M_{il} \sum_s M_{js} \sum_t M_{it} E(\hat{\theta}_{lk}\hat{\theta}_{st}) \\ &= \sum_k M_{jk} \sum_l M_{il}M_{jk}M_{il}E(\hat{\theta}_{lk}^2) \\ &= \sum_k M_{jk}^2 \sum_l M_{il}^2 E(\hat{\theta}_{lk}^2) \\ &= E(\hat{\theta}_{lk}^2),\ l,k \in \{1,2,...,N\}\end{aligned} \tag{3.67}$$

where E stands for expectation. In addition, $\xi_{ij}, i, j \in \{1, 2, ..., N\}$ are independently distributed, which can be shown by the following

$$\begin{aligned} E(\xi_{ij}\xi_{gh}) &= E(\sum_k M_{jk} \sum_l M_{il}\hat{\theta}_{lk} \cdot \sum_s M_{gs} \sum_t M_{ht}\hat{\theta}_{st}) \\ &= \sum_k M_{jk} \sum_l M_{il} \sum_s M_{gs} \sum_t M_{ht} E(\hat{\theta}_{lk}\hat{\theta}_{st}) \\ &= \sum_k M_{jk} \sum_l M_{il}M_{hk}M_{gl} E(\hat{\theta}_{lk}^2) \\ &= \sum_k M_{jk}M_{hk} \sum_l M_{il}M_{gl} E(\hat{\theta}_{lk}^2) \end{aligned} \tag{3.68}$$

where, if $j \neq h$ or $i \neq g$, then $\sum_k M_{jk}M_{hk} = 0$ or $\sum_l M_{il}M_{gl} = 0$, therefore $E(\xi_{ij}\xi_{gh}) = 0$.

In brief, under the null hypothesis, $\xi_{ij}, i, j = 1, 2, ..., N$ are *i.i.d.* Gaussian random variables having the following conditional probability density function

$$p(\xi_{ij}|H_0) = \frac{1}{\sqrt{2\pi(1+\gamma^2)\sigma^2}} e^{-\frac{\xi_{ij}^2}{2(1+\gamma^2)\sigma^2}} \tag{3.69}$$

where H_0 denotes the null hypothesis.

Then, the hypothesis test can be carried out as follows. First, define

$$y = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\xi_{ij}^2}{(1+\gamma^2)\sigma^2} \tag{3.70}$$

as the measure of of the difference between two given blocks. Since $\frac{\xi_{ij}}{\sqrt{(1+\gamma^2)\sigma}}$ are *i.i.d.* standard normal random variables, y obeys a χ^2 distribution with N^2 degrees of freedom. Next, determine a threshold τ so that when $y > \tau$ the null hypothesis is rejected, otherwise established. And, τ is usually obtained by specifying the *significance level* denoted by α, such that

$$\alpha = P(y > \tau|H_0) \tag{3.71}$$

where P denotes probability.

We tested the method on *Hallway* sequence and a sequence *Car Toy* recorded by regular digital camera. The size of a DCT block was 8×8. The $significancelevel$ α was set to 10^{-6}. Fig. 41 shows the results of *Hallway* sequence, where (a),(b), (c) and (d) are frame

1, frame 250, the CDM and the masked video frame respectively. It is seen that the human subject and the suit case (placed by the wall) were well identified.

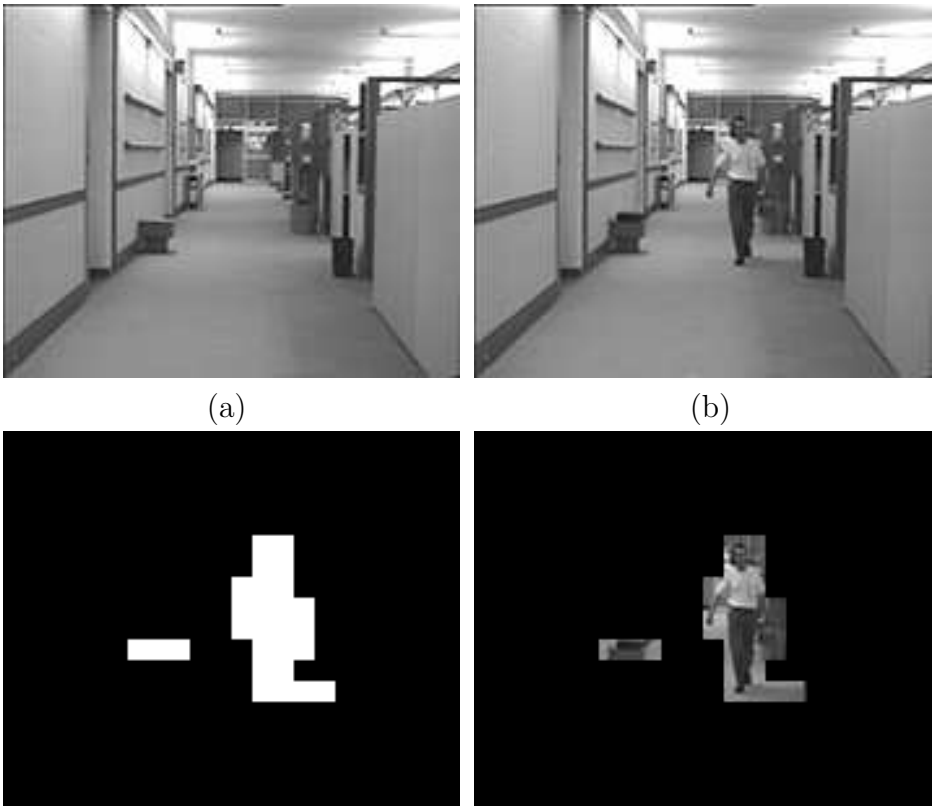

Figure 41: Results on *Hallway* sequence.

The two images shown in Fig. 42 (a) and (b) were taken under different lighting conditions. The purpose of this experiment was to test the illumination-invariant function of this DCT block test approach. Aside from the illumination change, these two images were only different in the toy car shown in (b). As expected, the toy car was identified, as shown in (c) and (d).

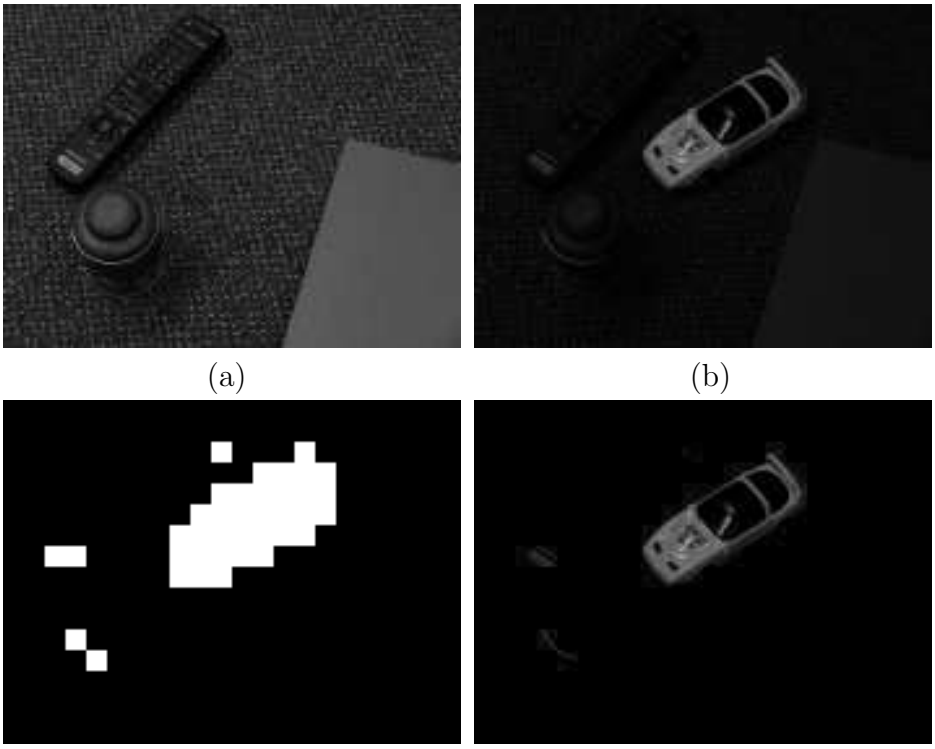

Figure 42: Results on *Car Toy* sequence.

3.4.3 Summary

In the above, we discussed two alternative change detection approaches to be combined with the coding process. The motion-vector based approach failed the segmentation purpose, because the block matching method did not provide reliable motion vectors. To improve the performance, a dense motion field needs to be computed (e.g. [72, 73]), which is very computationally demanding. The other approach that carries out statistical test on DCT blocks provided some promising results. We see however, the resolution of the CDM was compromised, i.e. the change detection was block-based instead of pixel-based. Further more, realizing these approaches needs to interfere with the codecs. The implementation therefore, may vary from codec to codec. With the evolution of the codecs, these embedded segmentation modules needs to be rebuilt. For these reasons, we adhere to the "prepro-

cessing" strategy to carry out change detection in spatial domain before the coding session. Nevertheless, it should be worthwhile to study the DCT domain change detection in future work.

4.0 IMPLEMENTATION AND EXPERIMENTS

4.1 INTRODUCTION

In this chapter, the details of the system construction are reported. First, we present video object construction schemes built upon the change detection algorithms. Next, we present a video coding system for patient monitoring, where the video object construction is embedded. To demonstrate the efficacy of the object-based coding system, we provide experimental results on the coding efficiency by comparing with frame-based coding systems. In addition, the objective evaluations on the visual quality of both video segmentation and video coding are reported. The limitations of the object-based coding system are discussed at the end of this chapter.

4.2 VIDEO OBJECT CONSTRUCTION

4.2.1 Three layer design

For the patient monitoring video, our strategy is to decompose a video frame into three video object planes (VOPs), defined as: 1) VOP_1, an image that contains only the scene of the environment, 2) VOP_2, *the regions inside which the patient and the objects associated with him/her are included*, and 3) VOP_3, *the regions that involve motion within a small time interval*. Essentially, these three VOPs can be considered as three layers which represent background, short-time stationary foreground, and moving foreground respectively. The background layer does not change over a long time period. The short-time stationary

foreground layer encloses the regions that are varied from the background layer and do not change in a short time window. The motionless body parts of the patient, the deformed bed and the re-positioned objects (e.g. a medical cart), may all be included in this layer. The moving foreground layer consists of the constantly changing regions, such as moving body parts of the patient and the associated entities that are caused to move by the patient.

Therefore, we construct three types of video object planes (VOPs) in the following way,

- VOP_1 represents the background layer. In our application, it is a snapshot of the patient monitoring room, without the presence of patient. VOP_1 has a rectangle shape, meaning that no shape coding is performed on it. Also, considering the background rarely changes, VOP_1 is encoded at a very slow frame rate, e.g. one frame per minute;
- VOP_2 represents the short-time stationary foreground. This VOP is constructed by applying change detection between VOP_1 and a test video frame. The MRF change detection algorithm is implemented to perform this task. VOP_2 is arbitrarily shaped, therefore shape coding of VOP_2 is required. Essentially, VOP_2 is composed of regions that contain substantial changes in contrast to the background. Therefore, in addition to the patient, the entities that are changed by the occupancy of patient are also included in VOP_2. It should be noticed that, within a short time window, only a small portion of the entities in VOP_2 are moving. Therefore, VOP_2 is encoded at a moderate frame rate, such as one VOP per second;
- VOP_3 comprises the moving foreground. This VOP is constructed by employing the covariance testing algorithm on a group of consecutive video frames. The entities moving in the corresponding period are included in VOP_3. This VOP is encoded at a regular frame rate, and shape coding is required.

As these three VOPs have different activity levels, they are encoded at different frame rates. The texture coding of each VOP is carried out at different time points. As shown in Fig 43, VOP_1 is encoded at $t_0^{'}, t_1^{'}, t_2^{'}, ...$, VOP_2 is encoded at $t_0, t_1, t_2, ...$, and VOP 3 is encoded at every frame. The alpha planes of VOP_1, VOP_2 and VOP_3 have life-spans of T_1, T_2 and T_3 respectively, which means within the time period T_1, T_2 and T_3, there will be only one alpha plane for VOP_1, VOP_2,and VOP_3 respectively. The values of T_1, T_2 and T_3 will

be determined experimentally. For simplicity, we plan to set T_2 equal to T_3. Then, for all the frames within the time interval T_3, only one alpha plane for VOP_3 needs to be constructed. This configuration greatly reduces the computational complexity of VOP construction, as well as the shape coding associated with VOP_3.

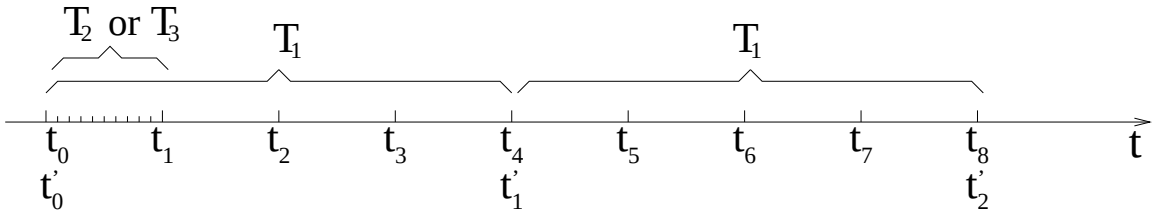

Figure 43: Time division for different VOPs. VOP_1 is coded at time points $t_0^{'}, t_1^{'}, t_2^{'}, ...$, VOP_2 is coded at time points $t_0, t_1, t_2, ...$, and VOP 3 is coded at every frame (the small divisions shown between t_0 and t_1). T_1, T_2 and T_3 denote the life-span of the binary alpha plane of VOP_1, VOP_2,and VOP_3 respectively. In other words, within the time period T_1, T_2 and T_3, the respective alpha planes for VOP_1, VOP_2, and VOP_3 do not change. Note that T_2 and T_3 are set equal, meaning that during the lift-span of VOP_2, only one alpha plane for VOP_3 is generated. This configuration simplifies the shape coding of VOP_3. Also, VOP_1 can be updated at fixed time points $t_0^{'}, t_1^{'}, t_2^{'}, ...$ to adapt to the changes of background.

4.2.2 Postprocessing on change detection masks

The alpha planes for VOP_2 and VOP_3 are obtained by postprocessing the change detection masks (CDM's). The postprocessing is performed in order to enforce the requirements of video object construction on the CDM's. Our major concern of postprocessing is to benefit texture and shape coding. In some applications, semantics of video objects are more preferred, which requires spatial domain features (e.g. edge, color homogeneity) to be utilized in postprocessing. Considering the computational complexity, we relax the semantics requirements on video segmentation. For simplicity, the postprocessing is carried out merely by filtering CDM's.

There are two types of regions in a CDM that need processing, the "0" regions in foreground and "1's" in background. The former is usually referred as "holes" and the latter as "islands". A conventional way to process these regions is to apply morphological operations, such as "opening" operation for filling holes and "closing" for eliminating islands. However, the structuring element utilized in a morphological operation is often determined in an *ad hoc* manner. To remove a large area, larger structuring element has to be used. This will not only degrade the precision of a segmentation mask, but also increase the computational cost.

In addition, the size of a hole or island is vague. For example, a small island may not necessarily be caused by noise. It can result from movements of a small object, such as an eye or a finger. Especially, in our application, eliminating an island means omitting the coding of the corresponding regions. A mistaken removal may not be tolerable. The same concern applies to holes in case that a hole results from false detection. For these reasons, we adopt a "safety" strategy that, 1) all holes should be filled, and 2) any island whose size is no smaller than a quarter block, i.e. 4×4 in pixels, should be retained.

Our approach is detailed in the following,

- First, a well-known connected components algorithm [78] is applied to the initial CDM to mark each separated "1" region. For each marked region, a tight rectangle frame that includes the region is formed. The top, left, right and bottom coordinates of the tight frame are recorded.

- In each tight frame, rows are scanned from left to right. The first and last swept "1" pixel are the left and right boundary points. All the pixels between the left and right boundary points are set to "1". These "1" pixels are the horizontal candidates for the final mask.

- Then, vertical candidates are obtained by the same operation that is performed column-wise. The intersection region of horizontal and vertical candidates form the final mask, also utilized as the binary alpha plane.

4.3 SYSTEM CONSTRUCTION

The structure of the coding system is shown in Fig. 44. Video frames are fed into two change detection blocks, the MRF change detection (MRF-CD) and the covariance test change detection (CT-CD), to generate segmentation masks for VOP_2 and VOP_3 respectively. The MRF-CD is controlled by $Timer_2$, which produces the CDM for constructing VOP_2 at an interval of T_2, as described previously. Similarly, the CT-CD is controlled by $Timer_3$ to generate the CDM for the segmentation of VOP_3. The video frames are buffered before being fed to CT-CD, since CT-CD operates on multiple video frames. At an interval of T_3, which is set equal to T_2, CT-CD generates a CDM. This CDM together with that generated by MRF-CD are postprocessed and then utilized as the alpha planes of VOP_3 and VOP_2 respectively.

Both texture coding and shape coding are carried out on these VOP_2 and VOP_3. The coding of VOP_1 is texture coding only, since VOP_1 is a rectangle video frame. The background scene in VOP_1 can be updated at a time interval of T_1 controlled by $Timer_1$. The algorithm of updating VOP_1 is further presented in section 4.6. In the coding blocks, each VOP is encoded at its own frame rate. The frame rates of VOP_1 and VOP_2 are controlled by $Timer_1$ and $Timer_2$ respectively. VOP_3 is encoded at the same frame rate as the input video. It should be noticed that during the coding process, each coded VOP is labeled with time stamps [79], which is determined by the associated frame rate. These times can be utilized to reconstruct the video frame from these VOPs. In the end, the three elementary streams are multiplexed and then output to storage or transmission.

The decoding system is shown in Fig. 45. The video stream is first demultiplexed. The three elementary streams are then decoded individually. The time stamps are extracted from the decoding process. The three VOPs are composed to form the final video frames for display. The composition is performed by means of overlaying: VOP_3 is surmounted on VOP_2 and then on VOP_1. Since these VOPs are encoded at different frame rates, they are rendered at different time points correspondingly. For the composition, VOP_1 and VOP_2 are held for time intervals T_1 and T_2 respectively after decoding.

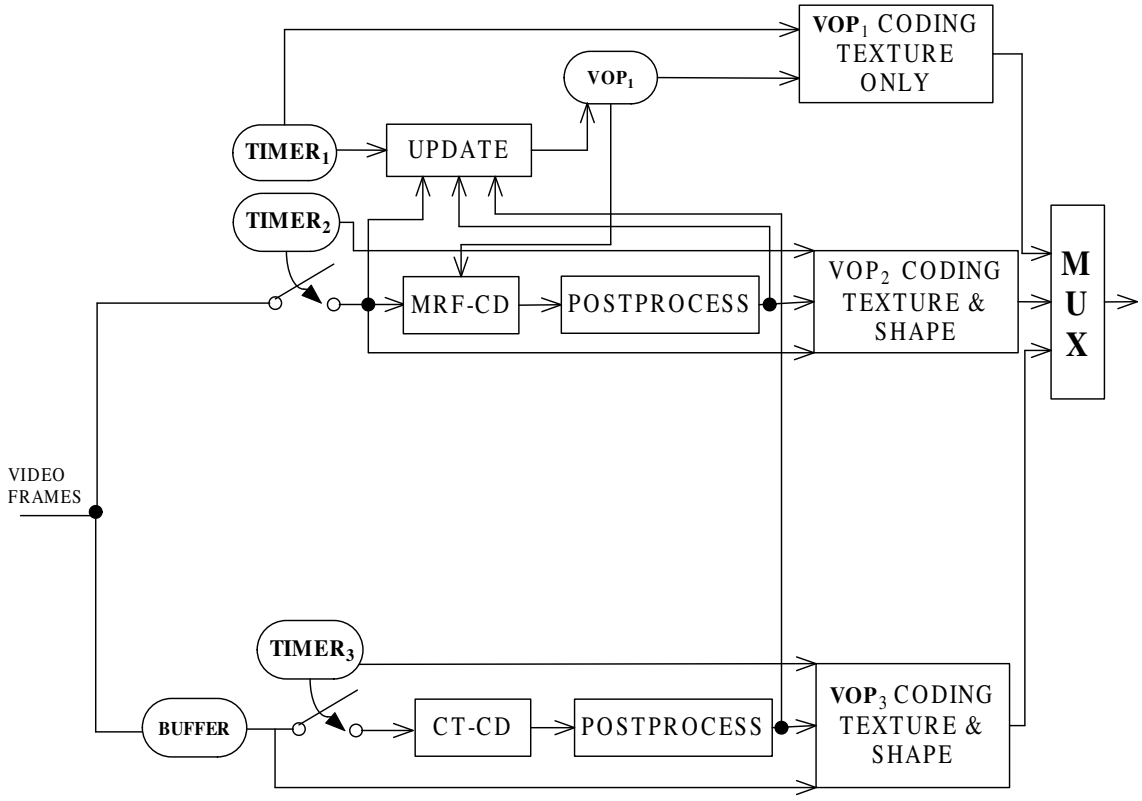

Figure 44: The block diagram of the object-based coding system.

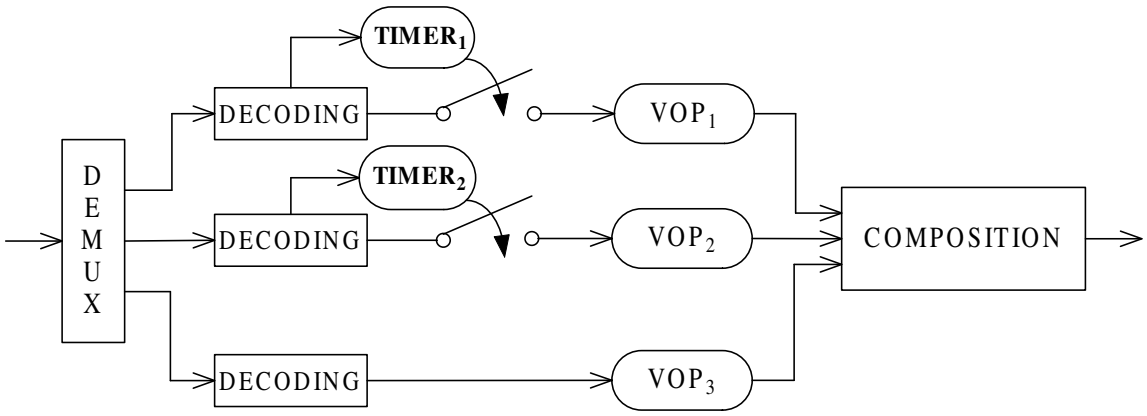

Figure 45: The block diagram of the decoding system.

The coding is concentrated on visual part. The implementation is based on adaptation of ISO/IEC 14496 video reference software package, version Microsoft-FDAMI-2.5-040207, which is available in the public domain. The functions of VOP coding, i.e. texture coding and binary shape coding, are implemented in this package. MPEG-4 simple profile, i.e. frame-based coding profile, is also included in the package. In our object-based coding system, we utilize the VOP coding routines. To compare the performance, we also carry out experiments with the simple profile. In section 4.5, we show the experimental results of both object-based coding and frame-based coding.

4.4 EVALUATION

We provide both the objective and subjective evaluation on the coding performance.

4.4.1 Objective measurement

A common measurement of coding quality is *peak signal to noise ratio* (PSNR), which is defined as $\text{PSNR} = 20\log\frac{255}{\text{noise standard deviation}}\text{DB}$. As a simple objective measurement, PSNR is much faster and cheaper than subjective assessments. Although popular because of its relative simplicity, the accuracy of PSNR is questionable in some cases. Therefore, more sophisticated evaluation methodology based on human visual perception is under investigation, e.g. [82, 83]. In this work, we utilize PSNR as a rough measurement of the coding quality. We evaluate our coding method by comparing with frame-based coding approach. The comparison is carried out by the following means:

- Constant quality: the quantization step is fixed in the coding routine. We compare the bit rate of the object-based coding with that of frame-based coding.
- Constant bit rate: coding is performed at a targeted bit rate. The PSNR of the object-based coding is compared with that of frame-based coding.

Normally, PSNR is calculated over the entire video frames. However, in our application, foreground regions and background regions are encoded at different scalabilities. Indeed, the

foreground region is the region of interest (ROI). Therefore, we calculate PSNR within the foreground region, called ROI-PSNR, as the measure of picture quality. Empirically [80], PSNR over 42 DB indicates "good" visual quality, meaning that "distortion is hardly seen". And PSNR over 45 DB means "excellent", i.e. "distortion is not viewable". We carry out the experiments aiming at high coding quality, namely, at ROI-PSNR over 40 DB.

4.4.2 Subjective measurement

There are four test methodologies utilized in the MPEG-4 video subjective tests: double-stimulus continuous quality scale (DSCQS), double-stimulus impairment scale (DSIS), double-stimulus binary vote (DSBV) and single stimulus (SS). The DSCQS method is typically employed for evaluations where the quality of the test sequence and that of reference are not much different. The DSIS method is usually applied for evaluating the annoyance of video impairment. The DSBV method is designed to evaluate the performance of a codec at the presence of long bursts of bit errors. And, the SS method is applicable when references are not available.

In our research, we aim at a high coding quality. The visual quality of coded video is expected to be close to the original sequence. Therefore, we carry out DSCQS test trials on our experimental video sequences. In the DSCQS method, each trial consists of a pair of stimuli: the reference sequence and the test. The two stimuli are each rendered twice in alternating fashion, with the order randomly chosen for each trial. Evaluating subjects are not informed of the ordering of the test and reference stimuli. Each stimulus is rated by a continuous quality scale. For each trial, two ratings are provided, one for the reference and the other for the test.

We provide evaluation results on the test sequences from six independent evaluators. The viewing conditions were set consistent with ITU-R Recommendation BT.814-1-1994.

4.5 EXPERIMENTAL RESULTS

In this section, we provide experimental results on several video sequences, including MPEG reference video and patient monitoring video. We show that the object-based coding approach outperforms frame-based coding with respect to the coding efficiency.

4.5.1 Head-shoulder sequence

(a) (b)

Figure 46: Frame 1 (a) and 50 (b) of *claire* sequence.

We first show the results on an MPEG benchmark sequence called *claire*. This head-shoulder sequence, as shown in Fig. 46, has simple background and low level noise. Motion activity is mostly contained in the face region. The subject body also generates slight motion. For this sequence, we utilized the first frame as the VOP_1. For every $N = 15$ frames, a VOP_2 was detected by applying MRF change detection approach. The frames between two VOP_2 were employed to calculate VOP_3 via covariance testing method. Several representative VOP_2 and VOP_3 are shown in Fig. 47.

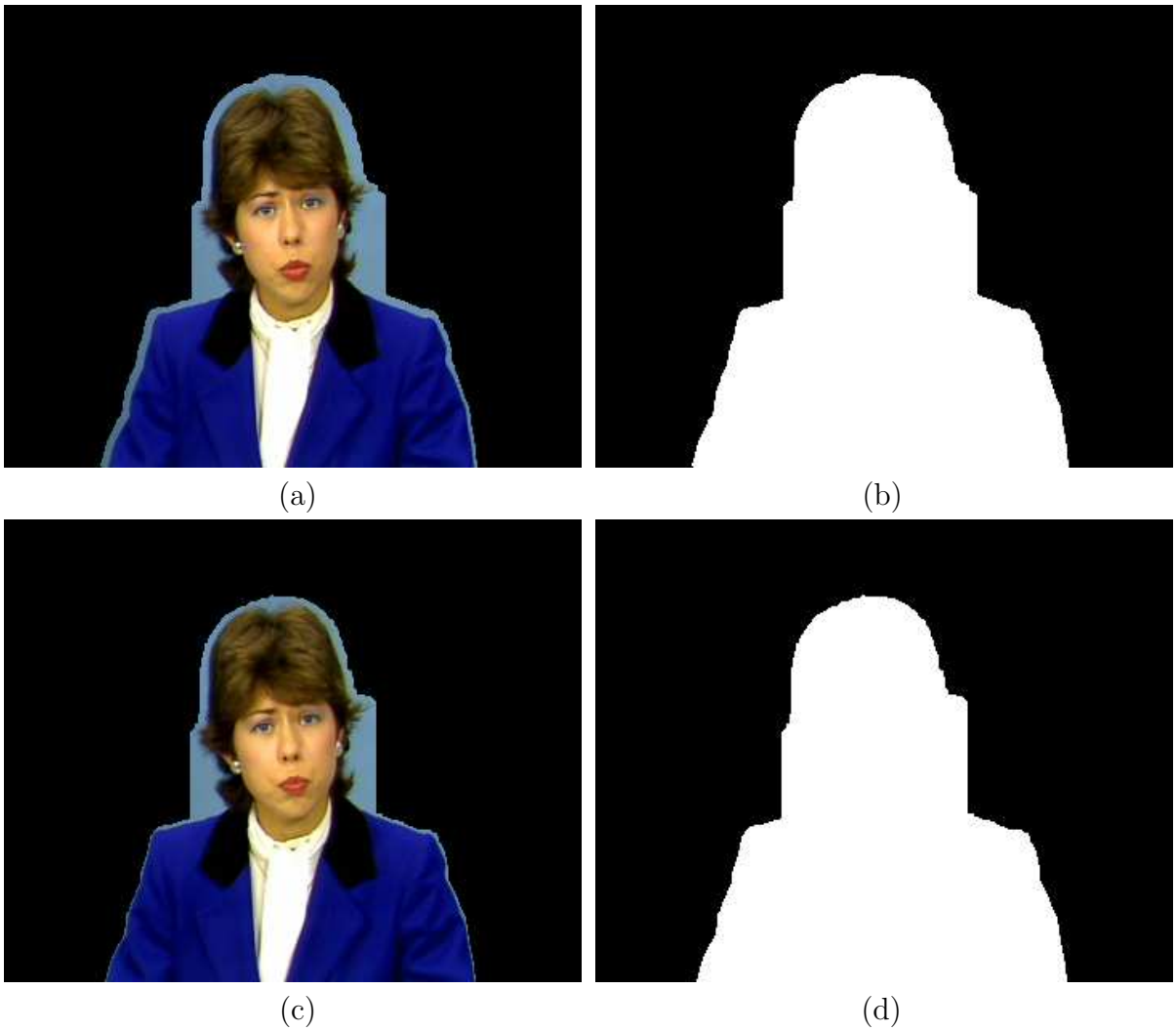

Figure 47: (a) and (b): The VOP_2 and its alpha plane at frame 45 obtained via MRF change detection. (c) and (d): The VOP_3 and the associated alpha plane at frame 55 generated by covariance test approach.

One can see that the foreground was successfully identified. It is also seen that some background regions were included in the VOPs. This is due to the simple postprocessing approach that was applied on the detected CDMs. Although more segmentation accuracy may be gained by furtherly utilizing spatial domain features, the segmentation results were already acceptable for the coding application.

The VOPs obtained via change detection methods are shown in Fig. 47. We see that there is not much difference between the shape of VOP_2 and that of VOP_3. This is because the whole body of the subject moved constantly in the video.

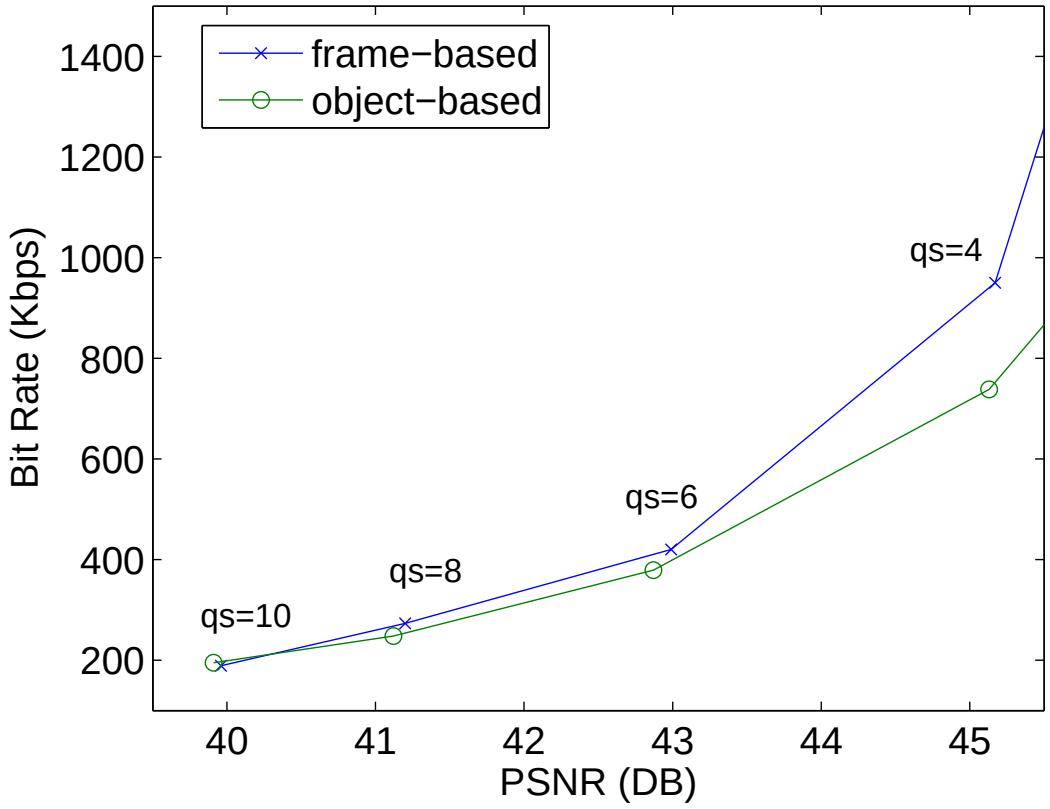

Figure 48: Comparison of object-based coding and frame-based coding results in constant quality mode. The "qs" stands for quantization step. For *Claire* sequence, object-based coding is slightly better than frame-based coding.

Next, we compare the object-based coding and frame-based coding results. The video is in Common Intermediate Format (CIF), i.e. each frame containing 288 lines and 352 pixels per line at 30 frames per second (fps). In Fig. 48, we show the coding results on *Claire* sequence in the *constant quality* mode. In this experiment, the same quantization steps were applied in frame-based and object-based coding procedures. We see that, for *Claire* sequence, the object-based coding is slightly better than frame-based coding. That the improvement is

not quite significant is due to two factors that, 1) the foreground takes a large portion of the video, and 2) the background is simple and contains low level noise (the standard deviation was estimated around 0.86). At large quantization step, the background difference between frames was removed, therefore similar to coding foreground only. However, one can see that, the higher the quality (higher PSNR) is, the more improvement is gained via object-based coding.

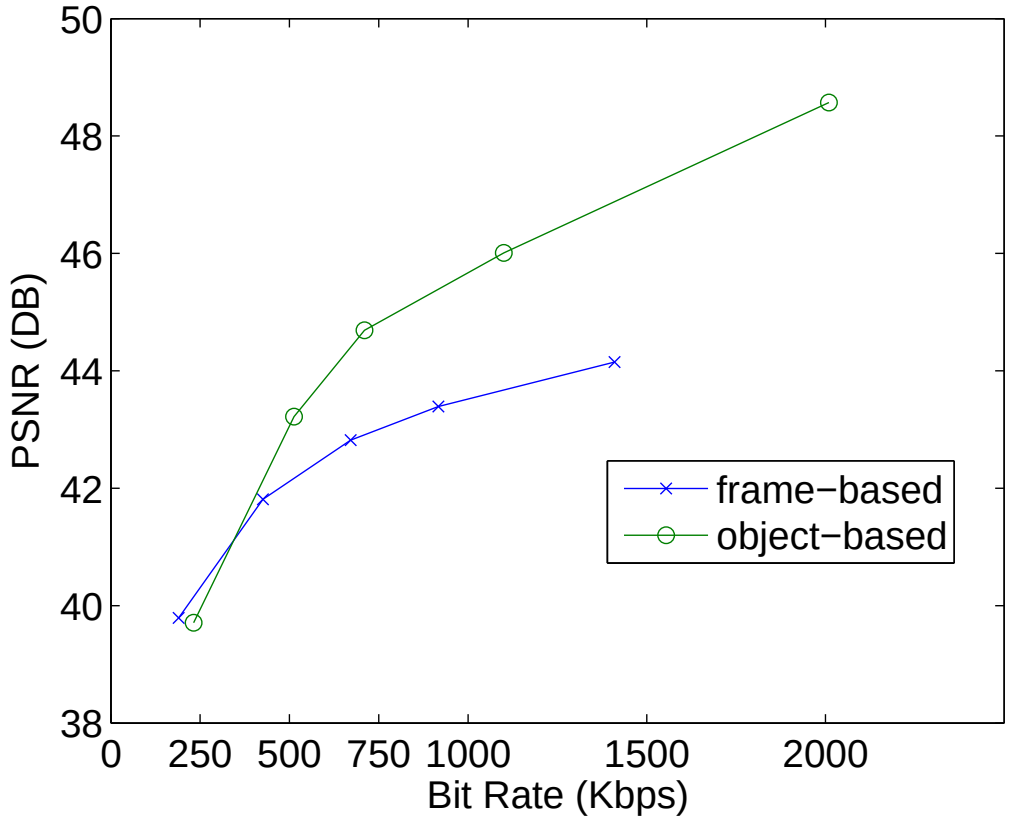

Figure 49: Comparison of object-based coding and frame-based coding results in constant bitrate mode.

The results in the *constant bitrate* mode are shown in Fig. 49. We see that at PSNR larger than 42 DB, object-based coding apparently outperforms frame-based coding. However, at 40 DB, where the bit rates were around 250 kbps, frame-based coding was slightly better. The reason is that at a target bitrate of 250 kbps, the quantization step was adjusted from

8 to 20 dynamically in the coding process. Therefore, the same reason as stated above also applies here.

4.5.2 Surveillance sequence

We show another MPEG benchmark sequence called *Hallway* in this section. Compared with *Claire* sequence, this sequence contains much higher noise (estimated noise standard deviation 3.2) and more complex background. The sample frames are shown in Fig. 50. The first video frame contains only background scene, thus was utilized as VOP_1. The construction of VOP_2 and VOP_3 was carried out with the same settings as those on *Claire* sequence.

(a) (b)

Figure 50: Frame 1 (a) and 120 (b) of *hallway* sequence.

Representative samples of VOP_2 and VOP_3 are shown in Fig. 51. One can see that the human subjects and the suit case placed in the hallway were well identified. The human subjects, as moving foreground, were included in both VOP_2 and VOP_3. The suit case was only included in VOP_2, because VOP_3 is meant to represent moving objects.

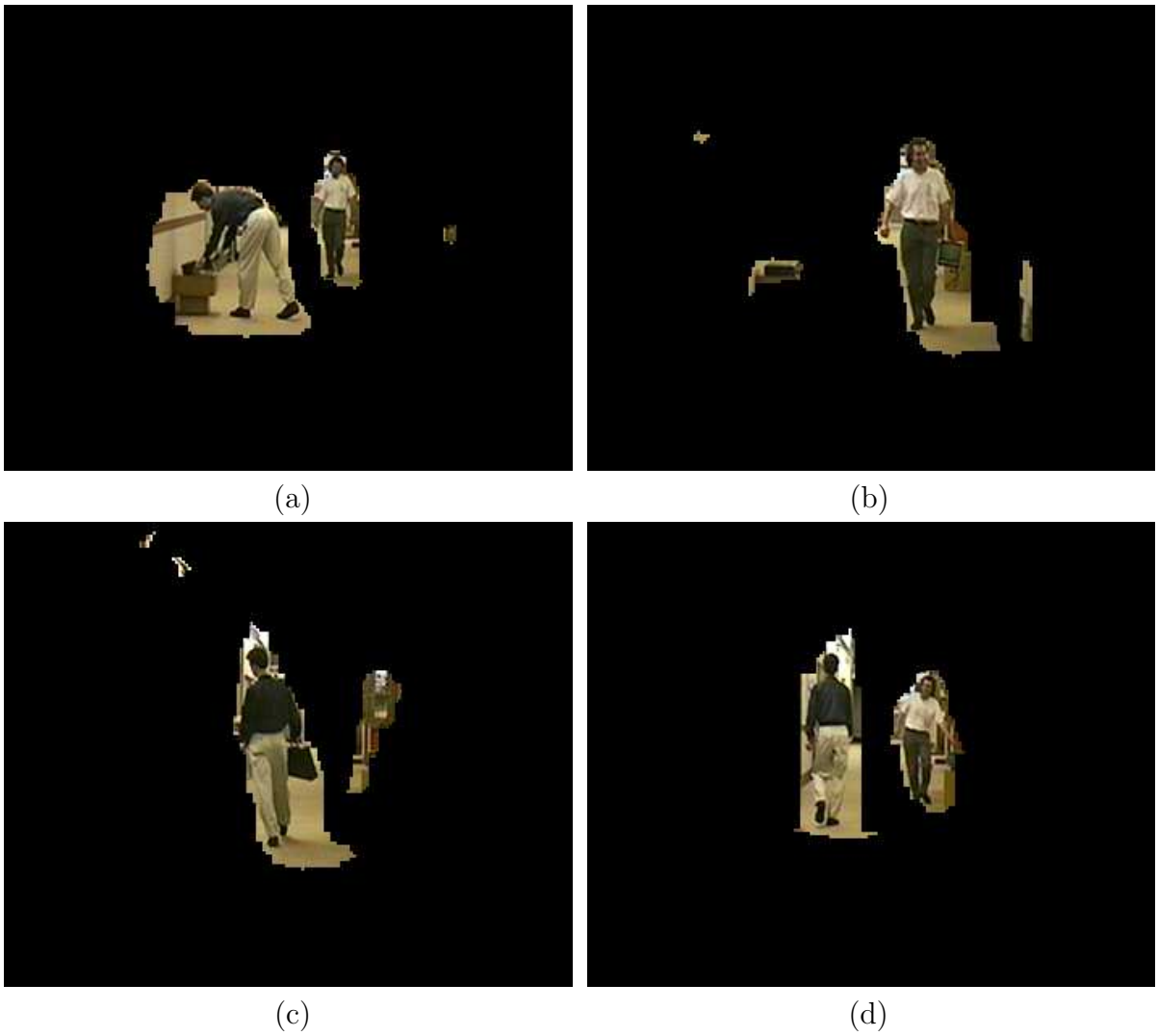

Figure 51: The VOPs of *hallway* sequence: (a)(b) the VOP_2 at frames 120 and 255, and (c)(d) the VOP_3 at frames 75 and 155.

The coding results in constant quality and constant bitrate modes are shown in Figs. 52 and 53, respectively. The test video is in CIF format. In contrast to the results of *claire* sequence, we see that for the *hallway* sequence, the object-based coding greatly improved the coding efficiency.

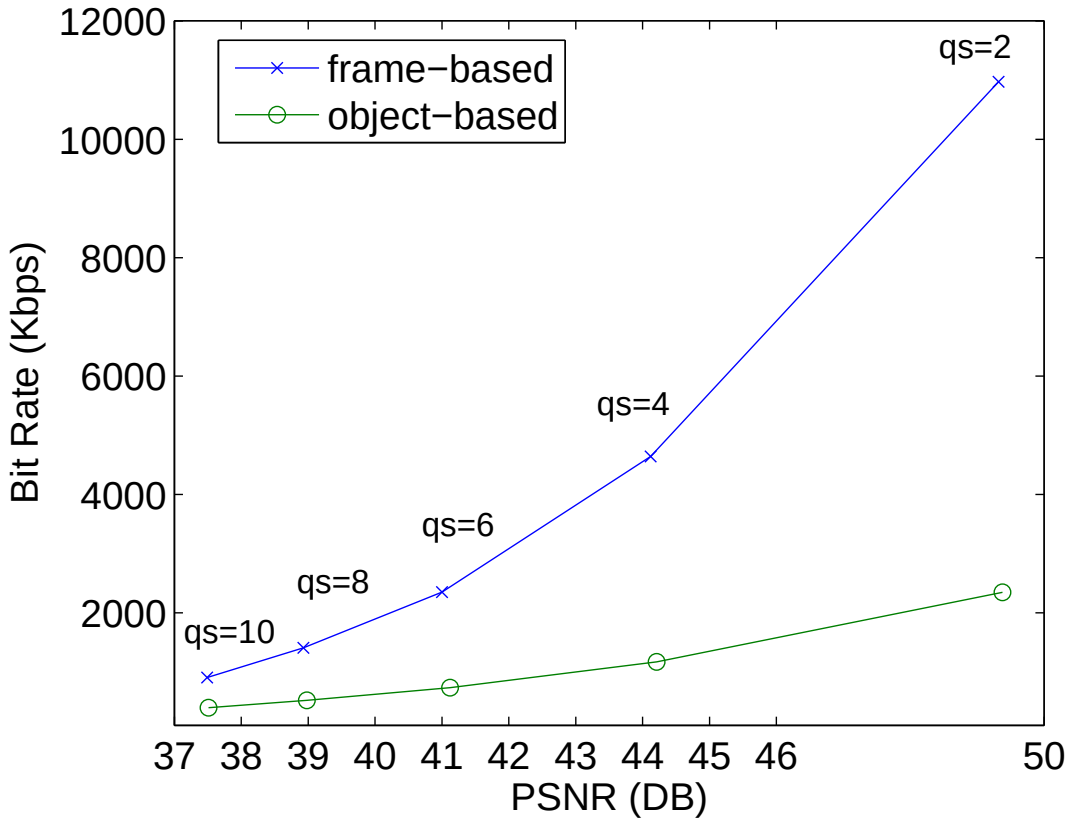

Figure 52: Coding results of *hallway* sequence. The object-based coding and frame-based coding results in constant quality mode are compared. The "qs" stands for quantization step. Object-based coding greatly outperforms frame-based coding.

This improvement is due to the fact that the noisy contents in the background region were omitted in the object-based coding approach. In the frame-based coding, however, these disturbances were not discriminated. And due to the relatively high noise level, they were not quantized to zero even at large quantization step. Therefore, a considerable amount of bits were wasted in the coding of them.

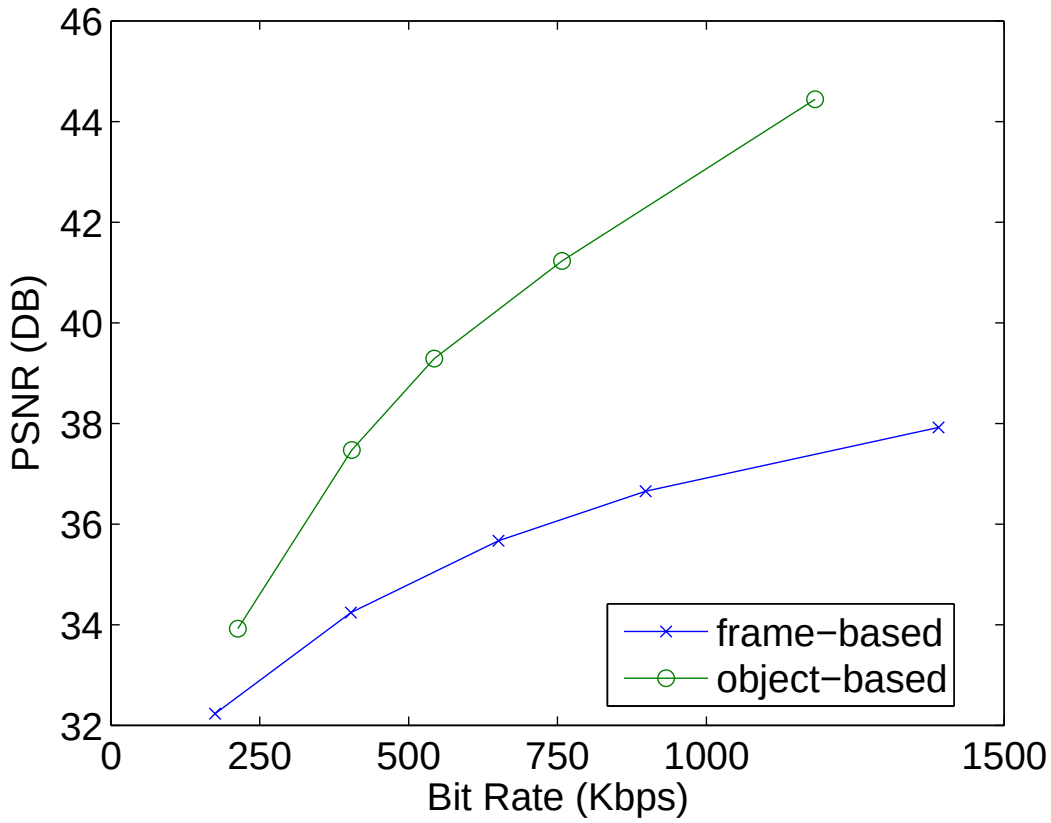

Figure 53: Coding results of *hallway* sequence in constant bitrate mode. The object-based coding and frame-based coding results are compared. For the *hallway* sequence, significant improvements were gained via object-based coding.

4.5.3 Patient monitoring sequence

Next, we present experimental results on a patient monitoring video sequence, recorded at the Epilepsy Monitoring Unit at the University of Pittsburgh Medical Center. Sample frames are shown in Fig. 54, where (a) shows a video frame that was shot before the patient made appearance. This frame was utilized as the initial VOP_1 containing only background scene. The video is in standard intermediate format (SIF), namely, with a spatial dimension of 352×240 and a frame rate of 30. And, this video sequence is in gray level.

(a) (b)

Figure 54: Sample frames of a patient monitoring video sequence. (a) A snapshot of the recording environment, which was utilized as VOP_1. (b) A video frame showing the patient.

The patient monitoring video is different from both of the MPEG sequences presented in previous sections. The noise level and structure complexity in the background of patient monitoring video lie in between those of the *Claire* and *hallway* sequences. And, the motion activity contained in patient monitoring video is usually less than that in both of them. Typical samples of VOP_2 and VOP_3 of patient monitoring video are shown in Fig. 55, where foreground contents were included in VOP_2, and moving foreground was identified as VOP_3. It is seen that the bed was detected as part of VOP_2. This is because the bed was deformed by the patient, thus differed from the bed in the background scene. With the movements of the patient, the bed made changes accordingly. Therefore, the bed was also counted as a part of the foreground. The VOP_3 contained only moving objects, especially those that had motion activity within a short time period (e.g. half second). We see that VOP_3 contained only small regions, which is because of the small motion generated by the patient. This fact leads to a considerable reduction of the coding bit rate.

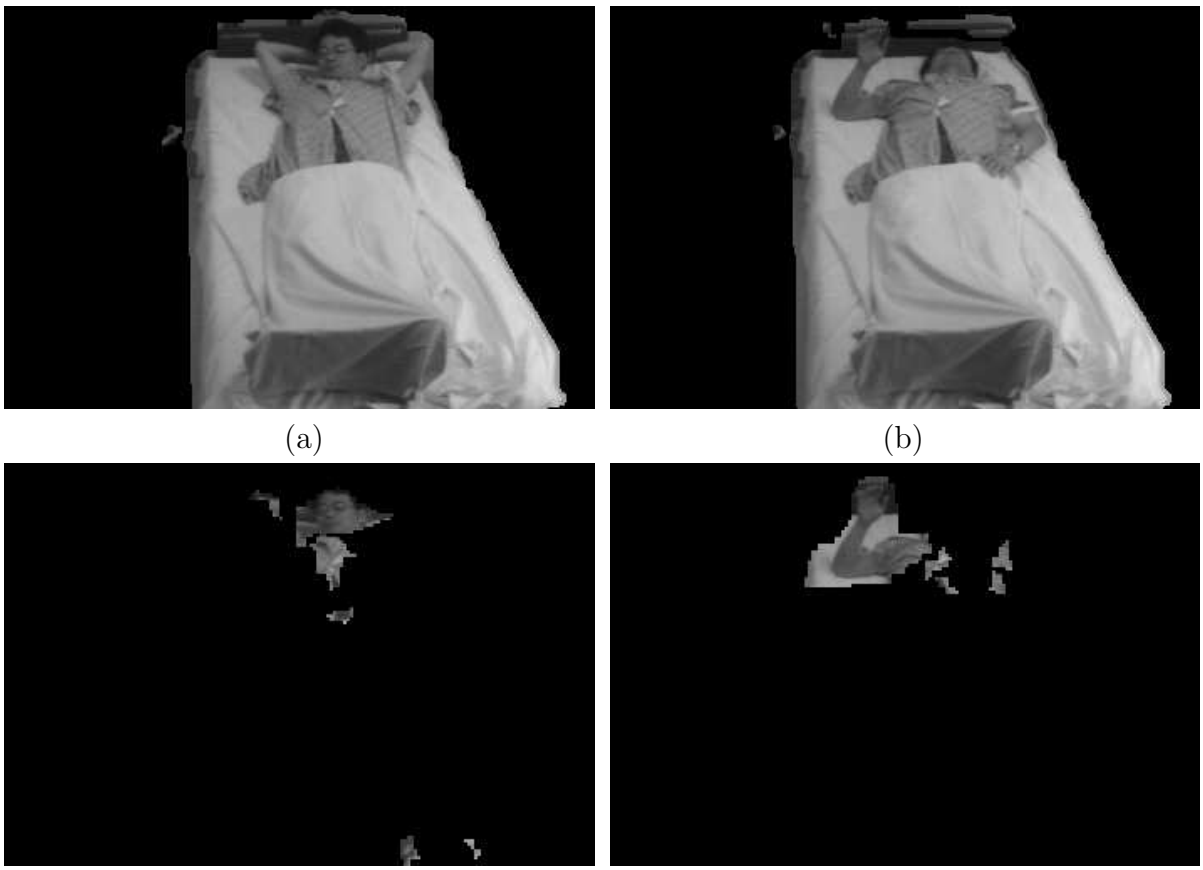

Figure 55: The VOPs of patient monitoring video sequence: (a)(b) samples of VOP_2, where the patient and the bed were included, and (c)(d) the VOP_3 samples, representing moving body parts.

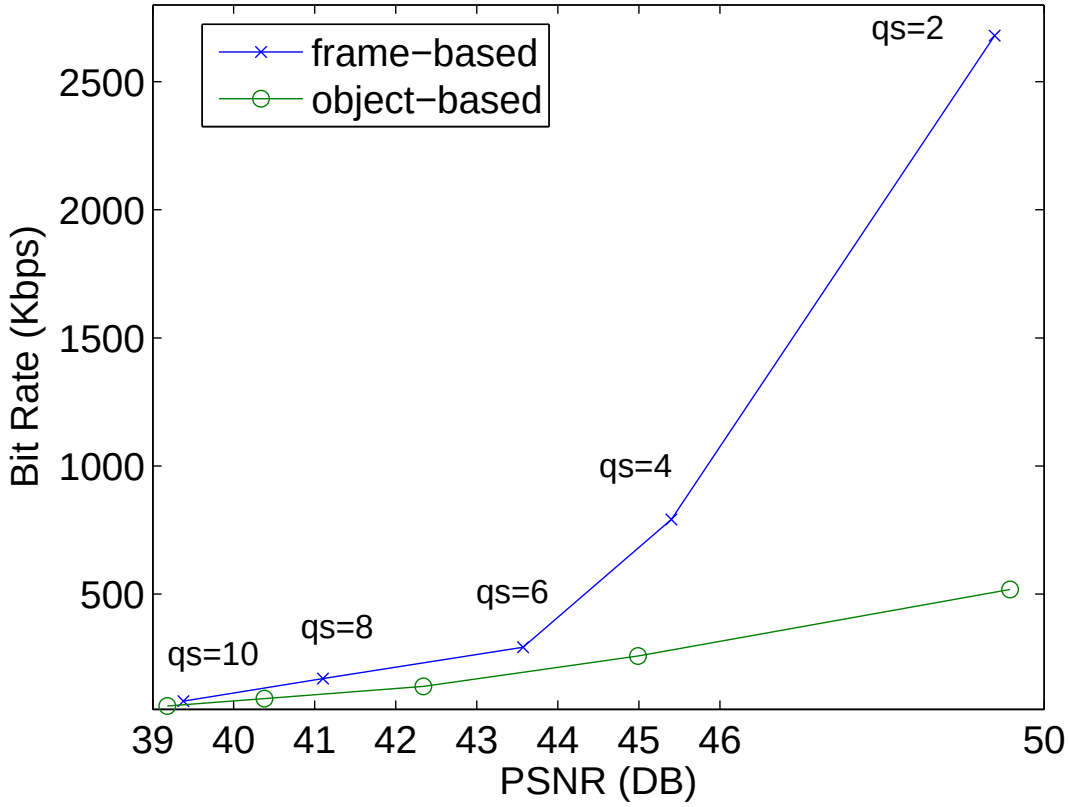

Figure 56: Coding results of the patient monitoring video sequence. The results of object-based and frame-based coding in constant quality mode are compared. The "qs" stands for quantization step. One can see that object-based coding outperforms frame-based coding.

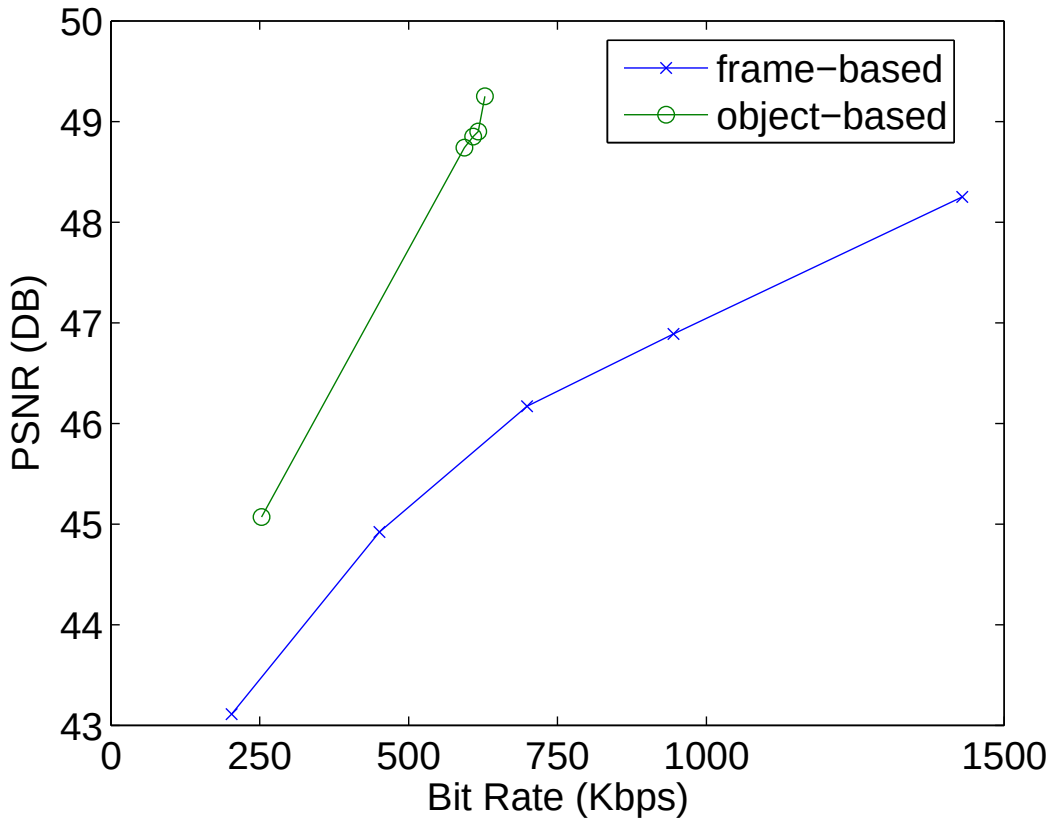

Figure 57: Coding results of the patient monitoring video sequence in constant bitrate mode. The object-based coding and frame-based coding results are compared. Great improvements were obtained via object-based coding.

4.5.4 Multi-camera patient monitoring video

Currently, most available epilepsy monitoring systems support VHS resolution video, e.g. 352×240 at 30 frames/second. With the improved coding efficiency, a substantial increase in video resolution is feasible. We present the investigation of a new video monitoring design, where three cameras are utilized. This system is highlighted in Fig. 58 with the three cameras mounted on three side-walls. DVD-resolution videos (e.g. 720×480 pixels) are collected from these cameras.

With this system, the entire view of the room can be observed. As a consequence, panning and tilting operations on the camera are not needed, since a favorable view of the patient is obtainable from one of the three angles. In addition, zooming operation can also be disregarded, because a reasonable clarity of the patient is usually achievable with the DVD resolution. Therefore, manual adjustment of the camera, a tedious and expensive operation, can be avoided.

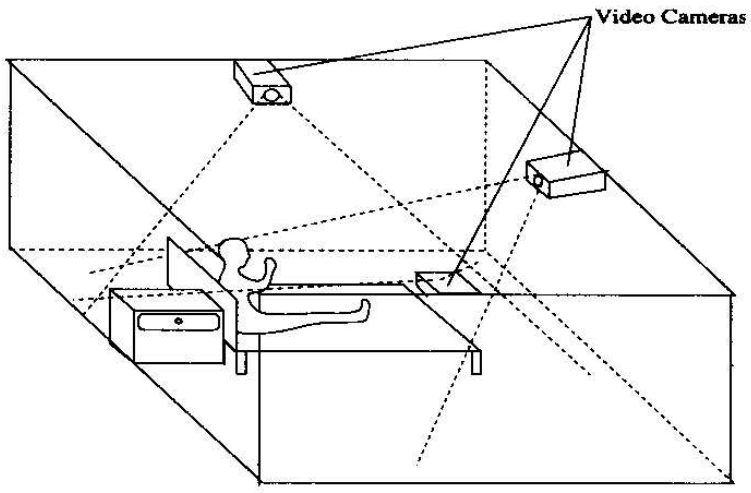

Figure 58: The three-camera system design with the cameras mounted on the side-walls. These cameras cover the entire view of the monitoring room. The remote operations on the cameras may not be in need.

Sample video frames from a three-camera system are shown in Fig. 59. Each video has 720×480 pixels in a frame and 30 frames per second. To encode video with such high definition, both the computation and the coding efficiency need to be considered. To reduce the computation time for the VOP construction, we carry out change detection on down scaled video frames (e.g. with a dimension of 180×120). The obtained CDMs are then postprocessed at the reduced scale level. The processed CDMs are resized to the original dimension to serve as the alpha planes.

(a)

(b)

Figure 59: Sample video frames taken from a three-camera system. High definition (720×480 at 30 fps) video is collected with this system. The top panels show the background scene from the three cameras. The bottom panels show the video frames with patients.

Several VOP samples are shown in Fig. 60, where (a) and (b) are the samples of VOP_2, and (c) and (d) are the samples of VOP_3. We see that in both VOP_2 and VOP_3, disturbances made appearance in the background area. Most of these disturbances were caused by the rapid flickering of the wall lamp in the recording room. These false detections were usually relatively small in size. Although they might be eliminated by size filtering, we kept them in the VOPs to reduce the risk of removing true foreground regions. Also, it can be seen that the shapes of the VOPs are blocky, which is due to the down-scaling operation performed in the VOP construction process.

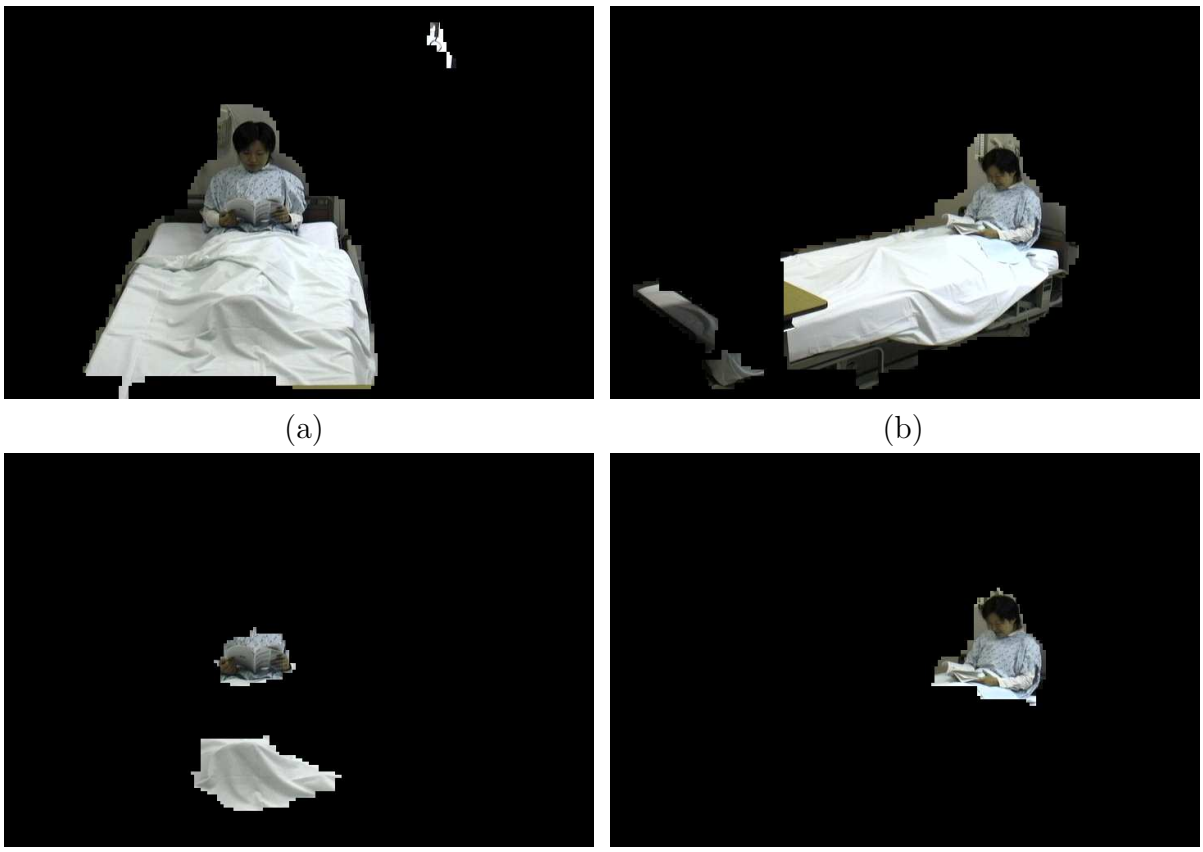

Figure 60: The VOPs of patient monitoring video: (a)(b) samples of VOP_2, where the patient and the bed were included, and (c)(d) the VOP_3 samples, which represent moving body parts.

The coding results are shown in Figs. 61 and 62. Extremely high bit rates were required to encode these high definition video via frame-based coding. For example, a bit rate of over 2 Mbps was necessary to encode the video from one camera for a 40 DB PSNR. It is nearly impossible to provide such a high bandwidth for a long distance transmission . However, with the object-based coding approach, less than 700 Kbps was needed to maintain the same quality within the region of interest. This bitrate is affordable for current Internet users with high speed transmission, e.g. DSL and cable.

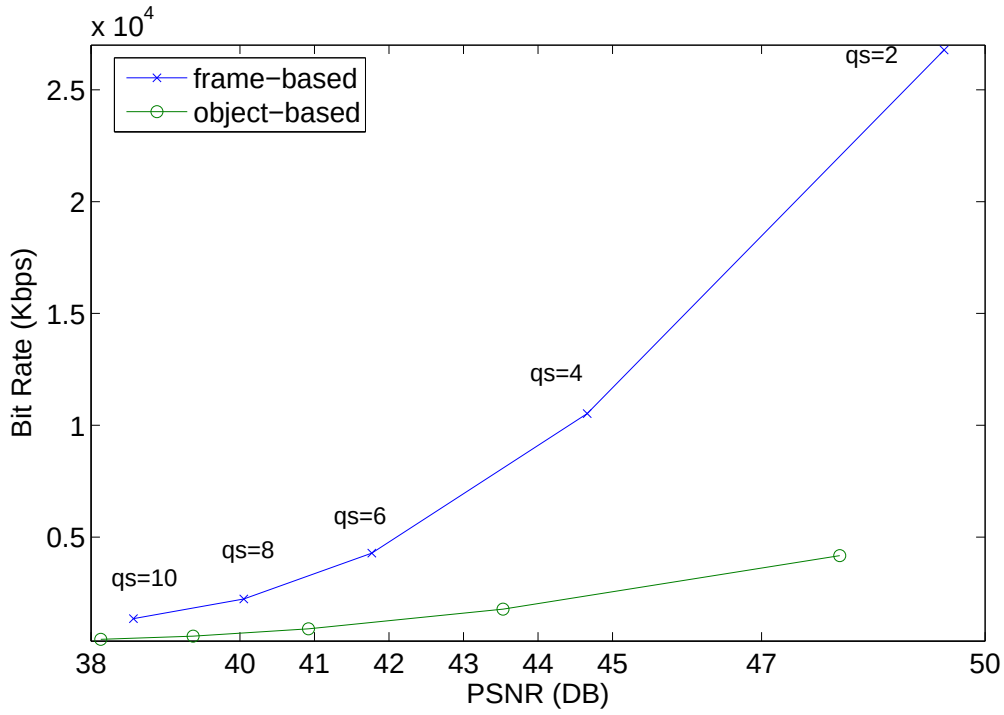

Figure 61: Coding results of the high definite patient monitoring video. The results of object-based and frame-based coding in constant quality mode are compared. The "qs" stands for quantization step. One can see that object-based coding outperforms frame-based coding significantly.

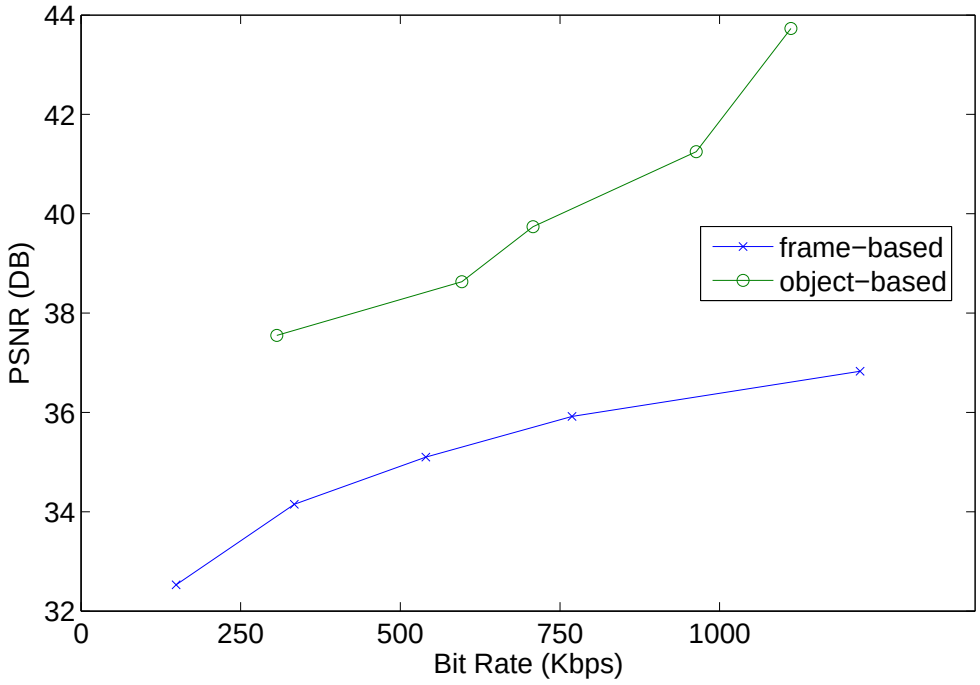

Figure 62: Coding results of the high definite patient monitoring video in constant bitrate mode. The object-based coding and frame-based coding results are compared. Great improvements were obtained via object-based coding.

4.5.5 Subjective evaluation

Subjective evaluation (DSCQS test) on the reconstructed video was performed on selected video clips, including both the MPEG reference video and patient monitoring video. These video clips were coded at bit rates between 500 Kbps and 1000 Kbps. In the test, each trial consisted of a pair of stimuli : the original sequence and the reconstructed. The two stimuli were rendered to the reviewers in alternating fashion with the order randomly chosen for each trial. The ordering was unknown to the reviewers. Each stimulus was rated by a quality scale, from "poor" to "execellent". Also, the reviewers were asked to indicate the one with better quality, in case that the two stimuli were rated the same scale.

In our test, 21 video clips were evaluated by 6 independent reviewers. Table 3 listed the summarized ratings on the quality of reconstructed video clips. The three columns "reconstructed", "equivalent" and "original" in Table 4 indicate the number of the reviewers that rated "the reconstructed clip looks better", "they are equivalent" and "the original clip looks better", respectively. One may see that most of the reconstructed clips were rated good quality, and there were few visual distortions between the reconstructed and the original sequences.

Table 3: Ratings on reconstructed video clips from six independent reviewers.

	excellent	very good	good	fair	poor	total
number of ratings	16	44	45	21	0	126
percentage(%)	12.7	34.9	35.7	16.7	0	100

Table 4: Comparison between the reconstructed and original sequence.

better sequence	reconstructed	equivalent	original	total
number of ratings	27	64	35	126
percentage(%)	21.4	50.8	27.8	100

4.6 DISCUSSION ON UPDATING VOP_1

In the duration of patient monitoring, the background contents (e.g. floor and wall decoration) do not change most of the time. The hypothesis of stationary background may be justified in this scenario. However, there are cases that the background scene is under change: 1) camera motion, i.e. panning, tilting and zooming, 2) repositioning of background content, such as adjustment of bed shape, and 3) appearance of new object, e.g. medical device placed in the background area. In all the cases, we need to update VOP_1 to adapt to the background changes.

We utilize a background registration approach to updating the background layer. This approach starts with an initial VOP_1 and updates it online. The initial VOP_1 may be

obtained by taking a snapshot of the recording environment before the patient makes appearance. The online updating is performed by the following process:

- The CDM resulted from covariance testing method indicates the moving regions contained in a group of consecutive frames. The "unchanged pixels" indicated by the CDM may belong to the background or the body parts that are motionless within the duration of the test frames. If a pixel is stationary for a long period, then there is a high probability that the pixel belongs to the background. Therefore, the idea to update VOP_1 can be carried out by checking the history of the "unchanged pixels". When a pixel stays stationary for a time period longer than a preselected threshold, then the pixel value in VOP_1 can be updated with the one in the current test frame.
- To record the history of the pixels, a background registration table is utilized. Each entry of the table records the duration (in number of frames) that a pixel stays continuously stationary. The initial value of a table entry is zero. Once a pixel is detected "moving", the corresponding entry in the registration table is reset to zero.
- When the entry value is accumulated to L, the specified threshold, the pixel is considered to be with background. At time point t_i', see Fig. 43, the background pixels are updated with the intensity values in the latest test frame. Considering that the patient monitoring video contains typically slow motion, we set L equal to the number of frames that span one minute.

Some experimental results of this background registration algorithm are shown in Fig. 63. In these experiments, the goal was to construct the background from the video frames. The initial background scene was set to blank (all zeros) for each of the experimental video. Covariance test change detection (CT-CD) method was carried on the video frames to identify the moving/stationary pixels. For the *hallway* sequence, we set the threshold $L = 30$, considering that this sequence contains fast moving contents. This setting meant that at 30 frames per second, any region that stayed stationary for 1 second would be assigned to the background area. We can see from the results that the background was well constructed and updated, e.g. the suit case placed on the deck was identified as part of the background. The patient monitoring sequence is different from the *hallway* sequence, as the patient usually

stays in bed and generates small movements. For this reason, we set $L = 1800$, a time span of 1 minute at 30 frames per second. From the results, we see that the recording environment except the bed, which was occupied by the patient, was correctly constructed.

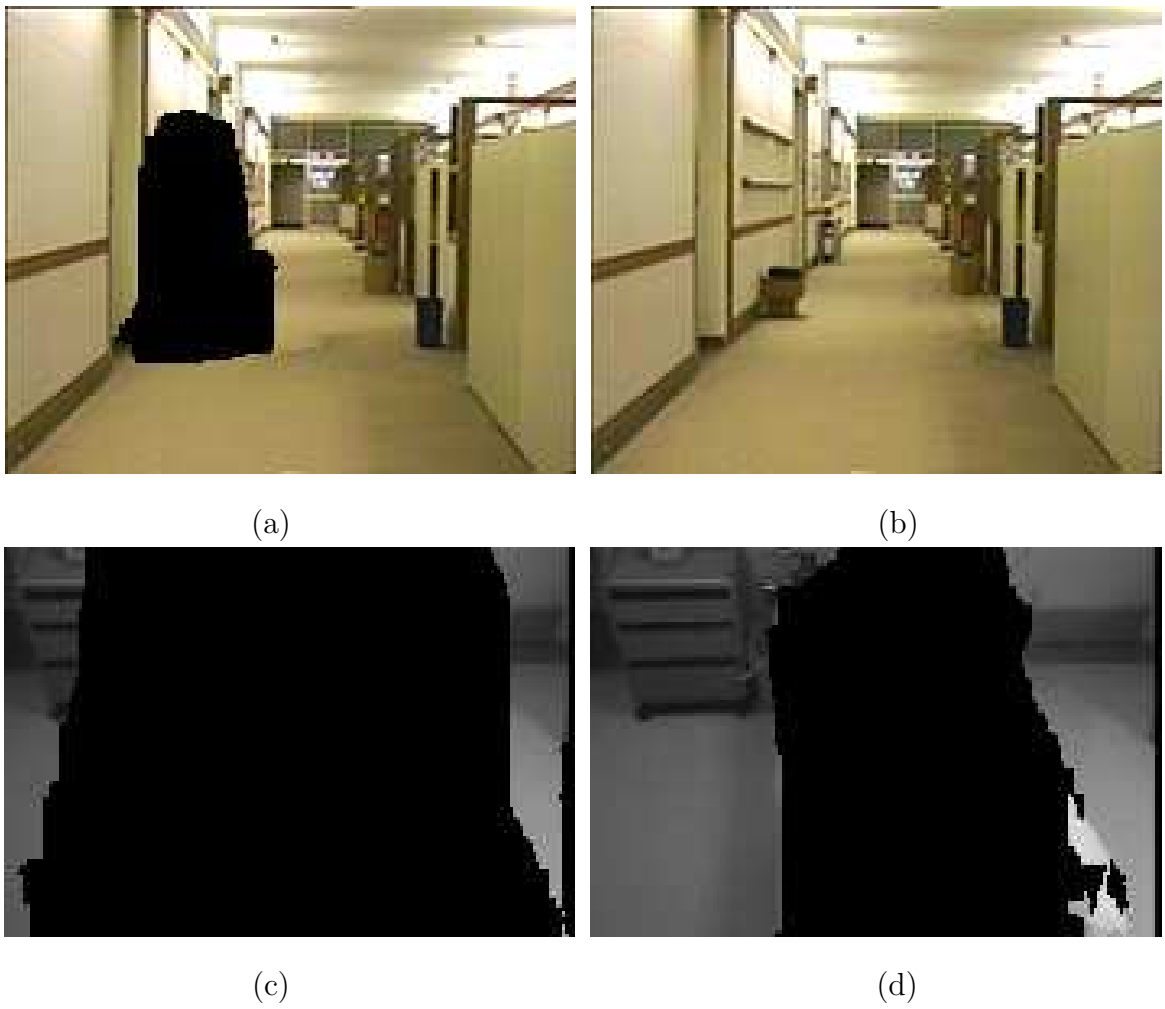

Figure 63: Experimental results of background registration algorithm.(a) and (b) The constructed background scenes at frame 75 and 300 of the *hallway* sequence, respectively. (c) and (d) The constructed background at frame 1950 and 2400 of patient monitoring video, respectively.

5.0 CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In this dissertation, an object-based approach has been provided to advance the video coding systems utilized for patient monitoring. We show that a scene can be represented by multiple video objects which enlighten content-driven coding applications. We apply this concept to encoding patient monitoring video. Each video frame is decomposed into three objects: background, short-time stationary foreground, and moving foreground. This decomposition reflects the features of patient monitoring video that the background contents are mostly stationary, and the motion activity is usually small and local.

Constructing the video objects is a critical step in the object-based coding system. We employ change detection as a key technique to accomplish this task. In order to address the weak points of the conventional methods in detecting small changes, we have presented two novel change detection approaches: 1) a MRF-MFT model which detects relevant changes between two images via an optimization process, and 2) a covariance test method which explores the temporal correlation contained in multiple video frames. Both approaches have shown great robustness in detecting small changes in image sequences.

The efficiency of coding the patient monitoring video can be greatly improved via the object-based approach. The underlining concept is to selectively code the video contents. In our application, only the moving foreground is coded at the full frame rate, while the other two objects are coded with much reduced temporal resolution. Statistical analysis is provided on the coding efficiency, where both texture coding and shape coding are investigated. The analytical results, as well as the experimental results on a variety of video sequences, show that at high coding fidelity, the object-based coding can outperform frame-based coding in

a wide margin. We have also examined a prototype of a multi-camera patient monitoring system in which each camera collects high definition digital video. Our results have showed that the substantial increase of video resolution can be successfully accomplished with the object-based approach.

5.2 FUTURE WORK

Some future work of this research is suggested as follows,

- Change detection exploiting color information should be investigated based upon the two methods presented. The presented models utilize only luminance to detect changes. The robustness can be further improved by taking color features into consideration. For example, in the MRF-MFT model, potential functions that formulate color difference between images can be designed to reflect constraints in the color space.
- Change detection at the presence of global motion is also worth of investigation. We have discussed updating the background scene when panning/tilting/zooming operations are performed on the camera. Another way that may be feasible to compensate the global motion is to utilize a background mosaic and apply affine motion model to register video frames with it. This background mosaic can be generated beforehand and utilized in a once-and-for-all manner. However, a critical problem that may be raised by this approach is the registration error. In order for a following change detection approach to function, this error has to be properly modeled.
- The constructed video objects may provide preliminary indexing functions for the multimedia patient record. The patient monitoring video may be summarized on the motion activities, where the size, position and the trajectory of the video objects may be analyzed to provide statistics of the patient activity. These statistics may be utilized as descriptors of the video such that retrieval of the multimedia content in a patient record may be facilitated.

BIBLIOGRAPHY

[1] Mohammed Ghanbari, *Video coding: an introduction to standard codecs,* London: Institution of Electrical Engineers, c1999.

[2] ISO/IEC 14 496-2: 2001, *Generic Coding of Audio-Visual Objects Part 2: Visual,* 2nd ed., 2001.

[3] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 7, No. 1, pp. 19-31, Sep. 1997.

[4] Ahan Shamin and John A. Robinson, "Object-based video coding by global-to-local motion segmentation," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 12, No. 12, pp. 1106-1116, Dec. 2002.

[5] Raj Talluri, et., "A robust, scalable, object-based video compression technique for very low bit-rate coding," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 7, No. 1, pp. 221-233, Feb. 1997.

[6] Nikolaos Doulamis, etc., "Low bit-rate coding of image sequences using adaptive regions of interest," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 8, No. 8, pp. 928-934, Dec. 1998.

[7] Kwok-Wai Wong, Kin-Man Lam, and Wan-Chi Siu, "An efficient low bit-rate video-coding algorithm focusing on moving regions," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 11, No. 10, pp. 1128-1134, Oct. 2001.

[8] Thomas Sikora, "The MPEG-7 Visual Standard for Content Description An Overview," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.11, No.6, pp. 696-702, June. 2001.

[9] H. Musmann, M. Hotter and J. Ostermann. "Object-oriented analysis-synthesis coding of moving images," *Signal Processing on Image Communication,* Vol. 1, No.2, Oct. 1989.

[10] V. M. Bove, Jr. "Multimedia based on object models: Some whys and hows," *IBM SYSTEMS JOURNAL,* Vol. 35, NOS 3&4, 1996.

[11] John Y. A. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. on Image Processing,* Vol. 3, No. 5, pp. 625-638, Sep. 1994.

[12] P. Salembier and F. Marques, "Region-based representations of image and video: Segmentation tools for multimedia services," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 9, No. 8, Dec. 1999.

[13] T. Meier and K. N. Ngan,"Segmentation and tracking of moving objects for content-based video coding", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.9, No.8, pp. 1190-1203, Dec 1999.

[14] M. Kim, J. G. Choi, D. Kim, etc., "A vop generation tool: automatic segmentation of moving objects in image sequences based on spatio-temporal information," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.9, No.8, pp. 1216-1226, Dec 1999.

[15] S. Chien, S. Ma, and L. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 12, No. 7, July 2002.

[16] S. Chien, Y. Huang and L. Chen, "Predictive watershed: a fast waterfhed algorithm for video segmentation," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 13, No. 5, pp. 453-461, Sep. 2003.

[17] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 8, No. 5, pp. 539-546, Sep. 1998.

[18] Y. Tsaig and A. Averbuch, "Automitic segmentation of moving objects in video sequences: a region labeling approach," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 12, No. 7, pp. 597-612, July. 2002.

[19] C. Kim and J. Hwang, "Fast and automatic video object segmentation and tracking for content-based applications," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 12, No. 2, pp. 122-129, Feb. 2002.

[20] C. Kim and J. Hwang, "Object-based video abstraction for video surveillance systems," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 12, No. 12, pp. 1128-1138, Dec. 2002.

[21] C. Gu and M. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 8, No. 5, pp. 572-584, Sep. 1998.

[22] S. Sun, D Haynor and Y. Kim, "Semiautomatic video object segmentation using vsnakes," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 13, No. 1, pp. 75-82, Jan. 2003.

[23] A. K. Jian, A. Ross and S. Prabhakar, "An introduction to biometric recognition", *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 14, No. 1, pp. 4-20, Jan. 2004.

[24] John Daugman, "Iris recognition," *American scientist,* Vol. 89, pp. 326-333, Jul.-Aug. 2001.

[25] John Daugman, "How iris recognition works," *IEEE Trans. on Circuits and Systems for Video Technology,* Vol. 14, No. 1, pp. 21-30, Jan. 2004.

[26] M. Bosc, F. Heitz, J. P. Armspach, I. Namer, D. Gounot, and L. Rumbach, "Automatic change detection in multimodal serial MRI: application to multiple sclerosis lesion evolution," *Neuroimage*, vol. 20, pp. 643-656, 2003.

[27] M. J. Dumskyj, S. J. Aldington, C. J. Dore, and E. M. Kohner, "The accurate assessment of changes in retinal vessel diameter using multiple frame electrocardiograph synchronized fundus photography," *Current Eye Research*, vol. 15, no. 6, pp. 632-652, June 1996.

[28] J.-P. Thirion and G. Calmon, "Deformation analysis to detect and quantify active lesions in three-dimensional medical image sequences," *IEEE Transactions on Medical Image Analysis*, vol. 18, no. 5, pp. 429-441, 1999.

[29] L. Lemieux, U. Wieshmann, N. Moran, D. Fish, and S. Shorvon, "The detection and significance of subtle changes in mixed-signal brain lesions by serial MRI scan matching and spatial normalization," *Medical Image Analysis*, vol. 2, no. 3, pp. 227-242, 1998.

[30] C.-Y. Fang, S.-W Chen, and C.-S. Fuh, "Automatic change detection of driving environments in a vision-based driver assistance system," *IEEE Trans. Neural Networks,* vol. 14, no. 3, pp. 646-657, May 2003.

[31] W. Y. Kan, J. V. Krogmeier, and P. C. Doerschuk, "Model-based vehicle tracking from image sequences with an application to road surveillance," *Opt. Eng.*, vol. 35, no. 6, pp. 1723-1729, 1996.

[32] M. J. Black, D. J. Fleet, and Y. Yacoob, "Robustly estimating changes in image appearance," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 8-31, 2000.

[33] Chris Stauffer and W. Eric L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. on PAMI,* Vol. 22, No. 8, pp. 747-757, Aug. 2000.

[34] Y. Yakimovsky, "Boundary and object detection in real world images," *JACM,* vol. 23, no. 4, pp. 598-619, Oct. 1976.

[35] Ming-Chieh Lee, et. al.,"A Layered Video Object Coding System Using Sprite and Affine Motion Model,"*IEEE Trans. on on Circuits and Systems for Video Technology*, Vol. 7, No. 1, pp. 130-145, Feb., 1997.

[36] A. Smolic, T. Sikora and J.-R Ohm, "Long-term global motion estimation and its application for sprite coding, content description, and segmentation," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 9, No. 8, pp. 1227-1242, Dec. 1999.

[37] Yan Lu, Wen Gao and Feng Wu, "Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques," *IEEE Trans. on on Circuits and Systems for Video Technology*, Vol. 13, No. 5, pp. 394-405, May 2003.

[38] M. Ghanbari, "Arithmetic coding with limited past history", *Electronics letters*, Vol. 27, No. 13, pp. 1157-1159, June 1991.

[39] H. Gish and J. Pierce "Asymptotically efficient quantizing," *IEEE Trans. on Information Theory,* Vol. 14, No.5, pp. 676-683, Sep. 1968.

[40] H. Hang and J. Chen, "Source model for transform video coder and its application — part 1: fundamental theory," *IEEE Trans. on on Circuits and Systems for Video Technology*, Vol. 7, No. 2, pp. 287 - 298, April 1997.

[41] "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. on on Circuits and Systems for Video Technology*, Vol. 11, No. 12, pp. 1226 - 1238, Dec. 2001.

[42] A. Katsaggelos, etc., "MPEG-4 and rate-distortion-based shape-coding techniques," *Proceedings of the IEEE,* Vol. 86, No. 6, pp. 1126-1154, June 1998.

[43] N. Brady, "MPEG-4 Standardized Methods for the Compression of Arbitrarily Shaped Video Objects", *IEEE Trans. on on Circuits and Systems for Video Technology,* Vol. 9, No. 8, pp. 1170-1189, Dec. 1999.

[44] N. Yamaguchi, T. Ida and T. Watanabe, "A binary shape coding method using modified MMR," *Proceedings of International Conference on Image Processing*, Vol. 1, pp. 26-29 Oct. 1997.

[45] N. Brady, F. Bossen, N. Murphy, " Context-based Arithmetic Encoding of 2D Shape Sequences", *Proceedings of ICIP*, pp. 29-32, 1997.

[46] A. J. Pinho "Adaptive Context-Based Arithmetic Coding of Arbitrary Contour Maps", *IEEE Signal Processing Letters,* Vol. 8, No. 1, pp. 4-6, Jan. 2001.

[47] M. Eden and M. Kocher, "On the performance of a contour coding algorithm in the context of image coding part1: contour segment coding," *Signal Processing*, Vol. 8, pp. 381 - 386, 1985.

[48] J. Koplowitz, On the performance of chain codes for quantization of line drawings, *IEEE Trans. Pattern Anal. Machine Intell.,* vol. PAMI-3, pp. 180-185, Mar. 1981.

[49] S. H. Lee, etc. "Binary Shape Coding Using Baseline-Based Method," *IEEE Trans. on on Circuits and Systems for Video Technology,* Vol. 9, No. 1, pp. 44-58, Feb. 1999.

[50] Haohong Wang, G. M. Schuster, A. K. Katsaggelos and T. N. Pappas, "An Efficient Rate-Distortion Optimal Shape Coding Approach Utilizing a Skeleton-Based Decomposition," *IEEE Trans. on Image Processing,* pp. 1181-1193, Vol. 12, No. 10, Oct. 2003.

[51] Y. Z. Hsu et al.,"New likelihood test methods for change detection in image sequences," *Comput. Vis. Graph. Image Process.*, vol. 26, pp. 73-106, 1984.

[52] K. Skifstad and R. Jain, "Illumination independent change detection for real world image sequences," *Comput. Vis. Graph. Image Process.*, Vol. 46, no. 3, pp. 387-399, 1989.

[53] T. Aach, A. Kaup, and R. Mester,"Statistical model-based change detection in moving video," *Signal Processing*, Vol. 31, pp. 165-180, 1993.

[54] T. Aach and A. Kaup,"Bayesian algorithms for adaptive change detection in image sequences using Markov random fields," *Signal Processing: Image Communication*, Vol. 7, pp. 147-160, 1995.

[55] Emrullah Durucan and Touradj Ebrahimi "Change Detection and Background Extraction by Linear Algebra," *Proceedings of the IEEE* vol.89, No.10, pp. 1368-1381, Oct. 2001.

[56] A. Neri, etc. "Automatic moving object and background separation," *Signal Processing* Vol. 66, pp. 219 - 232, 1998.

[57] B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM,* vol. 18, pp. 311-317, 1975.

[58] A. V. Oppenheim, R. W. Schafer, and T. G. Stockham Jr, "Nonlinear filtering of multiplied and convolved signals," *Proc. IEEE*, vol. 56, pp. 1264-1291, Aug. 1968.

[59] S. Kirkpatrick, C. D. Gellatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, pp. 671-680, 1983.

[60] J. Besag, "On the statistical analysis of dirty pictures (with discussions)," *J. Roy. Statist. Soc.,* ser. B, vol. 48, pp. 259-302, 1986.

[61] M. Hassner and J. Sklansky, "The use of Markov random fields as models of texture", *Comput. Graphics Image Processing*, vol. 12, pp.357-370, 1980.

[62] R. Chellappa and A. Jain, *Markov Random Fields Theory and Applications*, Boston : Academic Press, 1993.

[63] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721-741, Nov. 1984.

[64] S. Liu, C. Fu, and S. Chang, "Statistical change detection with moments under time-varying illumination," *IEEE Trans. Image Processing*, vol. 7, no. 9, pp. 1258-1268, September 1998.

[65] T. Kasetkasem and P. K. Varshney, "An Image Change Detection Algorithm Based on Markov Random Field Models", *IEEE Trans. Geoscience and Remote Sensing*, vol. 40, No. 8, pp. 1815-1823, Aug. 2002.

[66] L. Bruzone and D. F. Prieto, "An Adaptive Semiparametric and Context-Based Approach to Unsupervised Change Detection in Multitemporal Remote-Sensing Images", *IEEE Transactions on Image Processing*, Vol. 11, No. 4, pp. 452-466, April 2002.

[67] David Chandler, *Introduction to modern statistical mechanics*,New York : Oxford University Press, 1987.

[68] Jun Zhang, "The mean field theory in EM procedures for blind markov random field image restoration",*IEEE Transactions on Image Processing*, Vol.2, No.1, pp. 27-40, Jan. 1993.

[69] Jun Zhang and Gerald G. Hanauer, "The Application of Mean Field Theory to Image Motion Estimation",*IEEE Transactions on Image Processing*, Vol.4, No.1, pp. 19-33, Jan. 1995.

[70] Jie Wei and Ze-nian Li "An efficient two-pass MAP-MRF algorithm for motion estimation based on mean field theory",*IEEE Transactions on Image Processing*, Vol.9, No.6, pp. 960-972, Sep. 1999.

[71] Matthias De Geyter and Wilfried Philips, "A noise robust method for change detection," *Proc. IEEE ICIP'03*, Vol. 2, pp. 391-394, Sep. 2003.

[72] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *MIT technical report A.1.Memo N0.572*, April 1980.

[73] Li-Fen Chen, Hong-Yuan M. Liao and Ja-Chen Lin, "Wavelet-based optical flow estimation," *IEEE Transactions on Image Processing*, Vol. 12, No. 1, pp. 1-12, Jan. 2002.

[74] John W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. on PAMI,* Vol. PAMI-2, No. 6, pp. 554-562, Nov. 1980.

[75] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images - a review," *Proceedings of the IEEE,* Vol. 76, No. 8, pp. 917-935, Aug. 1988.

[76] O. Sukmarg adn K. R. Rao, "Fast object detection and segmentation in MPEG compressed domain," *Proc. IEEE TENCON,* Kuala Lumpur, Malaysia, Sept. 2000.

[77] R. Venkatesh Babu, K. R. Ramakrishnan and S. H. Srinivasan, "Video object segmentation: a compressed domain approach," *IEEE Trans. on circuits and systems for video technology,* Vol. 14, No. 4, pp. 462 - 473, April 2004.

[78] R. Jain, R. Kasturi and B. G. Schunck, *Machine Vision*, McGraw-Hill, New York, NY, 1995.

[79] A. Murat. Tekalp, *Digital video processing*, NJ: Prentice Hall PTR, 1995.

[80] Arun N. Netravali and Barry G. Haskell, *Digital pictures : representation and compression*, Plenum Press, New York, c1988.

[81] G. J. Conklin, etc., "Video coding for streaming media delivery on the internet," *IEEE. Trans. on circuit and systems for video technology,* Vol. 11, No. 3, pp. 269-281, Mar. 2001.

[82] Y. J. Zhang "A survey on evaluation methods for image segmentation", *Pattern Recognition*, Vol. 29, No. 8, pp. 1335-1346, 1996.

[83] Tan K.T., Ghanbari M. and Pearson D.E., "An objective measurement tool for MPEG video quality", *Signal Processing*, vol. 7, pp. 279-294, 1998.

[84] Th. Alpert and J.-P. Evain, "Subjective quality evaluation - the SSCQE and DSCQE methodologies," *EBU technical review*, pp. 12-20, Spring 1997.

[85] P. Correia and F. Pereira, "Objective evaluation of relative segmentation quality," in *Proc. Ieee Int. Conf. Image Processing,* Vol. 1, Sept. 2000, pp. 308 - 311.

[86] K. Mckoen, etc., "Evaluation of segmentation methods for surveillance applications," *EUSIPCO,* Tampere Finnland, Sep. 2000.

[87] C. Erdem, B. Sankur and A. Tekalp, "Performance measures for video object segmentation and tracking," *IEEE Trans. on Image Proc.,* Vol. 13, No. 7, July 2004.

[88] T. Alpert, etc. "Subjective evaluation of MPEG-4 video codec proposals: Methodological approach and test procedures," *Signal Processing: Image Communication,* Vol. 9, pp. 305-325, 1997.